Predicting Mood Disorder Risk
A Data Blending and Machine Learning Approach

MASTER’S THESIS

by
Alex Klein and Florian Reifschneider

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

Frankfurt Big Data Lab
Department of Computer Science
Goethe-University

Supervised by
Prof. Dott.-Ing. Roberto V. Zicari
Databases and Information Systems

November 21, 2016
Declaration of Authorship

In accordance with Masterordnung Informatik §10 (12)

We declare that we have authored this thesis independently, that we have not used other than the declared sources, and that we have explicitly marked all material which has been quoted either literally or by content from the used sources.

Frankfurt am Main on

Alex Klein

Florian Reifschneider
“Here is the tragedy: when you are the victim of depression, not only do you feel utterly helpless and abandoned by the world, you also know that very few people can understand, or even begin to believe, that life can be this painful. There is nothing I can think of that is quite as isolating as this.”

Giles Andreae
Abstract

The ability to predict the risk of developing a mood disorder for a given individual with sufficient accuracy could be of substantial help in diagnosing and treating mood disorders before they become a major health issue. This thesis is based on work that was conducted within the framework of the Geisinger Health Collider Project that was jointly held in cooperation by Geisinger Health Systems and UC Berkeley from fall 2015 to spring 2016. In this context, access was given to anonymized real-world clinical data on mood disorder patients, including their detailed patient history. Using a data blending and machine learning approach, this thesis seeks to develop and evaluate a prediction model centered around the patient history that is able to predict the individualized risk for mood disorder development in order to facilitate early diagnosis in undiagnosed individuals. As part of data blending, the clinical data provided by Geisinger Health Systems was combined with multi-disciplinary data acquired from various sources, such as the U.S. Census Bureau. The focus of the data blending strategy was to examine the relationship between an individual’s personal environment and their mood disorder risk. The actual prediction model was trained on the blended data using several machine learning algorithms and evaluated thoroughly in order to validate the hypothesis that the personal environment influences the mood disorder risk, as well as to proof that the patient history of an individual can be used to accurately predict their mood disorder risk.
Acknowledgements

Firstly, we would like to express our sincere gratitude to our advisor Prof. Dott.-Ing. Roberto V. Zicari for the continuous support of our master’s thesis. Without him, we would have never had the courage to tackle a problem as important and meaningful as the one that lies at the core of our thesis.

Further, we want to extend our deepest gratitude to Dr. Marco Pennacchiotti, who guided us through the jungle of machine learning with utter patience and competence. We feel lucky to have had an expert in the field of machine learning by our side throughout our work.

Our sincere thanks also go out to Geisinger Health Systems and the Sutardja Center at UC Berkeley for giving us a chance to participate in the Geisinger Health Collider. We would like to especially thank Dr. Nicholas Marko and the whole data science team at Geisinger, including Dr. Oleg Roderick, Dr. Joseph Klobusicky, David Sanchez, Debdipto Misra and Jason Brown.

We thank Dr. Anthony Bak for his assistance and valuable suggestions for the refinement part of our thesis.

We would also like to thank Dr. Brigitte Piniewski and Arash Nouri, who were also involved as experts in the Geisinger Health Collider. Without their passionate participation and input, we could have not finished our thesis.

We would also like to show our gratitude Prof. Dr. Andreas Reif and Dr. Sarah Kittel-Schneider for their valuable input.

Last but not the least, we would like to thank our families and close friends for supporting us spiritually throughout writing this thesis and our lives in general. Special thanks go out to Greta and Harry, for their tirelessness, frustration tolerance and continuous support.
Contents

Abstract vii
Acknowledgements ix

1 Introduction 1
1.1 Scope 2
1.2 Geisinger Health Collider 3
  1.2.1 Timeline 3
1.3 Problem Description 5
1.4 Overview 5

2 Strategy 7
2.1 Overview 8
2.2 Data Management 9
  2.2.1 Data Quality 9
2.3 Technical Approach 11
2.4 Data Blending 12
  2.4.1 Differences to Data Integration 13
  2.4.2 Data Quality Concerns 14
  2.4.3 Data Blending Diagrams 14
2.5 Initial Strategy 17
2.6 Domains of Data 20
  2.6.1 Data Sources 21
2.7 Predicting Mood Disorder Risk 23
  2.7.1 Evaluation 24

3 Implementation 27
3.1 Acquired Data 27
  3.1.1 Clinical Data by Geisinger Health Systems 28
  3.1.2 Population Data 32
  3.1.3 Economic Data 33
  3.1.4 Geographic Data 35
  3.1.5 Crime Data of Major US Cities 37
3.2 Data Preparation 39
  3.2.1 Dealing with Missing Data 39
  3.2.2 Census Community Data 40
  3.2.3 Estimation of Crimes in Pennsylvania 44
  3.2.4 Partitioning of Clinical Data 50
  3.2.5 Patient Demographics 51
  3.2.6 Calculation of Mood Disorder Prevalence 53
  3.2.7 Imputation of Mood Disorder Prevalence 59
  3.2.8 Patient History Matrix 64
## Appendix C  Python Libraries

<table>
<thead>
<tr>
<th>Library</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumPy</td>
<td>138</td>
</tr>
<tr>
<td>scikit-learn</td>
<td>139</td>
</tr>
<tr>
<td>RDF and OWL</td>
<td>139</td>
</tr>
<tr>
<td>Word2vec</td>
<td>141</td>
</tr>
</tbody>
</table>

## Appendix D  Machine Learning Background

### D.1  Prediction Models
- Linear Regression: 143
- Logistic Regression: 144
- Boosting: 145
- AdaBoost: 146
- Gradient Boosted Trees: 146

### D.2  Evaluation Methodology
- Cross Validation: 147
- Mean Square Error: 147
- Accuracy: 148
- R²: 148
- Log Loss: 148
- Receiver Operating Characteristic: 149

## Appendix E  Data Preparation

### E.1  Census Community Dataset
### E.2  Crime Prediction
- Crime ZIP Code Identification: 154
- Pig: 156

### E.3  Imputation of Mood Disorder Prevalence
### E.4  Patient Demographics
### E.5  Patient History
### E.6  Blended Dataset

## Appendix F  Data Analysis

### F.1  Imputation of Mood Disorder Prevalence
- Patient Coverage
  - Patient Coverage Pennsylvania: 167
  - Geisinger Facilities in Pennsylvania: 168
- Confidence: 169

### F.2  Crime Prediction
- Load Data: 172
- Training Model: 172

### F.3  Patient History

### F.4  Mood Disorder Risk Model
- Load Training and Test Data: 173
- Train Models: 173
- Evaluate Model: 175
- Load and Score: 175
- ROC: 175
- PR: 176
List of Figures

2.1 Adaption of Data Supply Chain build by Accenture for the Purpose of Data Ethics ................................................. 10
2.2 Overview of Strategy ............................................................................... 12
2.3 Conceptual difference between datasets commonly found in data integration (a) and data blending (b) ........................................................................................................... 13
2.4 Example Data Blending Diagram ............................................................ 15
2.5 Generalized Data Blending Approach ...................................................... 15
2.6 Generalized Data Blending Approach ...................................................... 16
2.7 Blending Diagram of All Data Domains into One Dataset ....................... 21

3.1 Data Blending Diagram of the Blending Strategy for Census Community Dataset ........................................................................................................................... 40
3.2 Distribution of the patient coverage over all 3,393 unique ZIP Codes found in the patient demographics dataset, plotted in both linear (a) and logarithmic scale (b) ........................................................................................................................................ 55
3.3 Heat map of patient coverage for all ZIP Codes in the state of Pennsylvania. The values for the heat map were normalized according to the power law scale, allowing for a more natural visualization ...... 56
3.4 Map of locations of all Geisinger hospitals and clinics ................................ 57
3.5 Distribution of the margin of error for the calculated mood disorder prevalence values ................................................................................................................................. 58
3.6 Mood disorder prevalence values and corresponding confidence intervals at the 95% confidence level for all ZIP Codes with a margin of error below 3% ................................................................................................................................... 59
3.7 Feature Correlation for Mood Disorder Prevalence Imputation ....... 62
3.8 Linear Scree Plot of the Singular Values of the Diagnosis-Patient Matrix ................................................................................................................................. 70
3.9 Logarithmic Scree Plot of the Singular Values of the Diagnosis-Patient Matrix................................................................................................................................. 70
3.10 Blending Diagram for the Final Dataset ................................................. 72
3.11 Evaluation Results for Mood Disorder Prevalence Features ............... 80
3.12 Evaluation Results for Crime Indicator Feature ................................... 82
3.13 Evaluation Results for Census Community Features .......................... 83
3.14 Evaluation Results for Patient Demographic Features features .......... 84
3.15 Evaluation Results for Patient History Features 90% Version Only .... 86
3.16 Evaluation Results for Patient History Features Scree Version Only .... 87
3.17 Evaluation Results for Gradient Boosted Trees Model on both ALL and ALL/SCREE configuration ......................................................................................................................... 89
3.18 Precision-Recall Curves for Gradient Boosted Trees Model on ALL and ALL/SCREE Configurations ......................................................................................................................... 90
3.19 F-Measure for Gradient Boosted Trees Model on ALL Configuration
List of Tables

2.1 Dimensions of Data Quality ........................................... 11
2.2 Selected Datasets for the Blending Process .......................... 22

3.1 Schema of the Demographic Dataset provided by Geisinger Health Systems ........................................ 28
3.2 Data Quality Analysis of Demographic Dataset provided by Geisinger Health Systems .............................. 29
3.3 Schema of the Problem List Dataset Provided by Geisinger Health Systems ........................................... 30
3.4 Data Quality Analysis of Problem List provided by Geisinger Health Systems ........................................ 31
3.5 Schema of the Population Dataset provided by the U.S Census Bureau ................................................ 32
3.6 Schema of the Economic Dataset provided the U.S. Census Bureau ....................................................... 34
3.7 Data Quality Analysis of the Economic Dataset provided by the U.S. Census Bureau .................................. 35
3.8 Schema of the Geographic Dataset provided by U.S. Census Bureau ....................................................... 36
3.9 Data Quality Analysis of Geographic Data provided by U.S Census Bureau ............................................ 37
3.10 Schema of Crime Dataset by the Police Department of Chicago ......................................................... 38
3.11 Overview of Data Manipulation of the Census Datasets ............... 41
3.12 Schema of blended Census Community Dataset ................. 43
3.13 Schema of Minimal Crime Datasets ................................... 45
3.14 Overview of Identified ZIP Codes for Crime Locations ................................. 48
3.15 Overview of Number of ZIP Codes in Crime Data and number of ZIP Codes after Joined with Census Community Features .................................................. 49
3.16 Evaluation Results of different Regression Models for the Crime Indicator Model .................................... 50
3.17 Schema of Patient Demographic Dataset ................................ 52
3.18 Results of Univariate Feature Selection for the Prediction of the Mood Disorder Prevalence ..................... 61
3.19 Evaluation Results of Mood Disorder Prevalence Imputation Model .................................................. 63
3.20 Number of Features of the Prepared Datasets ................................................. 71
3.21 Schema of Blended Dataset ................................................ 73
3.22 Evaluation Results for the Baseline .................................................. 78
3.23 Evaluation Results for Mood Disorder Prevalence Features ................................................................. 79
3.24 Evaluation Results for Crime Indicator Features ...................... 81
3.25 Evaluation Results for Census Community Features .................. 81
3.26 Evaluation Results of Patient Demographic Features .......... 85
3.27 Breakdown of Both Linear Regression Model and Gradient Boosted Trees Model on ALL and All/SCREE Configuration ........................................... 89
Chapter 1

Introduction

Mental health is a serious topic. The mind is the single most unique thing a person has. Our mind is what makes us who we are; it’s what drives us, what we feel, and what we think. But it’s what happens when things go awry that makes mental health so important.

Among the most common mental health problems is a group of illnesses aptly named mood disorders. The list of mood disorders includes sicknesses such as major depressive disorder, bipolar disorder and dysthymic disorder, with major depressive disorder being the most common among them. Mood disorders affect an alarmingly large number of people, in fact, approximately 9.5% of the U.S. population suffer from one or multiple mood disorders. This relates to 20.9 million affected individuals in the U.S. alone.

The symptoms of mood disorders are diverse and can occur in different levels of severity, however, they always include some form of mood disturbance, such as elevated or depressed mood. Other severe symptoms include anxiety, apathy and low energy in general, trouble sleeping and concentrating, and suicidal thoughts. Mood disorders are a major cause of suicides, with about one-half to two-thirds of suicides attributed to mood disorders. In total, mood disorders are responsible for over a half million of preventable deaths annually worldwide.

As a result to the direct symptoms, individuals affected by mood disorders also often suffer from functional impairment and reduced quality of life. Lost work productivity due to this, including missed time from work and unemployment, is a major problem for both the individual and society as a whole. The resulting economic burden on the society caused by mood disorders has been increasing steadily over the past years, from an estimated economic burden of 83.1 billion dollars in 2000 to 210.5 billion dollars in 2010. These enormous costs, especially the direct medical costs, are at least partly attributable to low detection rates of mood disorders despite their rather high prevalence. It is estimated that less than half of affected patients are correctly diagnosed in the primary care setting.
Chapter 1. Introduction

The ability to predict the risk of developing a mood disorder for a given individual with sufficient accuracy could be of substantial help in diagnosing and treating mood disorders before they become a major health issue for the individual. Not only would this improve the quality of life of those affected tremendously, but early diagnosis would also lower the economic burden on society.

More detailed information on mood disorders in general and the specific mood disorders talked about in this thesis can be found in Appendix B.

1.1 Scope

The goal of this thesis is to tackle the problem of predicting the individualized risk for developing a mood disorder in order to facilitate early diagnosis in undiagnosed individuals. This mood disorder risk can be understood as an indicator of how likely an individual is to develop a mood disorder and could be used in a clinical setting to help doctors and medical personnel with the diagnosis of actual mood disorder patients. Further, the mood disorder risk could be used to initiate preemptive care for high-risk individuals.

To this end, we chose to tackle the problem by using a combination of data blending and machine learning to develop a prediction model that is able to predict the mood disorder risk on an individual level with sufficient accuracy to be used in a real-world clinical setting. Data blending is an exploratory approach to data analysis that seeks to combine multi-disciplinary data in often non-obvious ways and includes the identification and acquisition of possible datasets, the preparation and transformation of the acquired data, and the actual joining of the individual datasets. Machine learning is used to train a prediction model on the blended data. The data blending strategy is centered around real-world clinical data that was acquired from Geisinger Health Systems in the scope of the Geisinger Health Collider project that is the subject of Section 1.2.

Using machine learning methods in a clinical setting is not a new approach, especially in the field of diagnostics [Kon01]. Prediction models that are able to predict the risk of developing certain disorders or diseases can be used to estimate the probability for an individual of either already suffering from or falling ill of a disease in the future [Hen+13]. For instance, machine learning models have been successfully used in the prediction and prognosis, as well as the detection and diagnosis of cancer [Mac+91, CW06]. Additionally, machine learning approaches have also been shown to be useful in the prognosis and diagnosis of cardiovascular diseases [Kuk+99, Men+12].
1.2 Geisinger Health Collider

This thesis is based on work that was conducted within the framework of the Geisinger Health Collider project that was jointly held in cooperation by Geisinger Health Systems and UC Berkeley from fall 2015 to spring 2016 [UC 15]. The Geisinger Health Collider project, called Collider for short throughout this thesis, was a competition focused on data blending and data analysis of clinical data provided by Geisinger Health Systems. The goal was to encourage the teams to supplement the provided data with additional, nonclinical data, to solve the technical problem of data blending and to demonstrate how effectively it can be to use and analyze multi-disciplinary data.

Teams were able to choose between one of three problems related to multi-disciplinary data analysis:

- Integrated data analysis for early warning of heart/lung failure
- Indirect data collection to support anti-obesity efforts in healthcare and society
- It’s not all in your head: multi-disciplinary data analysis of common psychological conditions

After choosing a problem, each team worked independently to identify novel data sources, to refine the hypothesis, and to formulate a tentative strategy for data blending and the subsequent analysis. At this stage, a premium was placed on creativity and the practical achievability of the project. Teams were also urged to use data that is tangible and publicly available. For each problem, Geisinger Health Systems provided a dataset with patients collected by the Geisinger Health facilities in Pennsylvania.

The Collider was addressed to graduate and Ph.D. students with a background in mathematics and computer science, but without a strong background in data science. The project also gave the participants the opportunity to learn how to blend data in a meaningful way and how to use common machine learning techniques while they were able to consult experts to discuss their problems.

As described in the motivation of the thesis, our team selected the topic of mood disorders.

1.2.1 Timeline

The Collider was divided into two phases, a first phase of strategic development followed by a second phase of the actual implementation of the strategy formulated in phase one.
Chapter 1. Introduction

The first phase spanned over a period of about four weeks and started with selecting one of the proposed topics. Phase one concluded with a submission by each team that consisted of a proposed strategy for the second phase.

After the first phase, the Geisinger Health Systems team reviewed every submission in detail and provided all teams with extensive feedback on the strategy and their prognosis for the feasibility of the proposal. The teams that were found to have submitted a feasible and promising strategy proposal by Geisinger were given the chance to proceed into the second phase. About ten teams took part in the competition initially, of which only five submissions were selected to proceed into the second phase after the review process of phase one, including a detailed review of each strategy by Geisinger.

The second phase, spanning a period of two months, started with getting access to the clinical data from Geisinger Health Systems. The primary component of phase two was to develop and evaluate the strategy proposed in the first phase. Teams had to refine their strategy based on the feedback or worked on problems that surfaced in the meantime.

Over the whole phase it was possible to communicate with the data scientists from Geisinger, with external experts in the medical area and in the technical area as well. The final submission for the second phase should include the following:

- Technical description of the strategy
- Data blending approach
- Machine learning approach
- Evaluation of the whole process
- Additional sources

It was important to describe the strategy development in detail and problems we had to deal with. After handing in the final submission, a final project presentation was set up by Geisinger Health Systems to show the effort over the last weeks to all the participants in the project and of course to the jury of Geisinger Health Systems that scores the data blending and machine learning approach.

As our team chose problem three, mood disorder risk, access was given to anonymized real-world clinical data on mood disorder patients, including their detailed patient history.

The following chapters will focus on the third topic proposal that is related to mood disorders, as this topic was chosen for the Collider by our team.
1.3 Problem Description

For every selectable topic in the Collider, Geisinger provided a brief introduction. In our case the description was about mood disorder:

*It’s not all in your head: multi-disciplinary data analysis of common psychological conditions*

Psychological mood disorders are often described as both social and medical phenomena. Recent studies in suicide prevention make connections between mood disorders and patterns in residential power use, logs of phone calls, and purchasing history, while older studies identify certain demographic groups as being more at risk. The problem of screening for and predicting the risk of mood disorders in the general population could have a major impact on population health. Can a combination of EMRs (containing clinical data) and other data sources improve upon current strategies for predicting the individualized risk for developing a mood disorder?

Based on this description our team started with doing research in the area of mood disorder, such as understanding what a mood disorder is, what could be a trigger for developing a mood disorder and how important it is to do research in this area. This kind of research was important to develop an objective including a hypothesis or research question to work on. For the final submission a clearly stated objective was elaborated, including possible publicly available datasets, the whole data blending process and the potential machine learning methods.

1.4 Overview

The main part of the thesis is divided into three major chapters, namely Strategy, Implementation and Refinement, that roughly align with the phases of the Collider.

Chapter 2 is concerned with the development of the data blending strategy that was created in the first phase of the Collider, including a short description of the adjustments made to the initial strategy.

Chapter 3 is focused on the actual implementation of the strategy, including acquisition and preparation of the needed datasets, the creation of the blended dataset, training the mood disorder risk model on the blended dataset, a detailed evaluation of the results, and a short summary of all results.

The last major chapter is Chapter 4 that highlights three approaches to refining our implementation, including a detailed discussion of the results found.
Finally, all the results and findings are summarized and possible future research is highlighted in Chapter 5.

The appendices of the thesis contain further evaluation results, medical background, technical background, fundamentals of data analysis, data preparation source code and data analysis source code.
Chapter 2

Strategy

As described in the introduction, predicting the risk for the development of mood disorders is a highly relevant and necessary field of study that has the potential of significantly helping a tremendous amount of people through early diagnosis and timely treatment of developing mental conditions. Inspired by the Collider, the presented work focuses on a data blending and machine learning approach to predict the individualized risk of developing a mood disorder or mood disorder risk for short.

The first step in the process of finding a solution to a given problem is coming up with a strategy to tackle this problem. Such a strategy should explain the chosen approach in detail and can later be used as a roadmap to implement the actual solution to the problem. Formulating the complete strategy at once works well for small problems, but for complex problems that involve a longer period of time working on them, the strategy often has to change in response to encountering unforeseen obstacles and new insights into the problem. This process of finding the right strategy and adapting it according to new insights is called strategy development.

Accordingly, this chapter will focus on the strategy that was developed as part of the presented work and the inherent strategy development process. This chapter roughly aligns with the first phase of the Collider, however, the strategy was tweaked over the course of the creation of this thesis. The following sections will showcase the different aspects of our approach, starting with a brief overview of the strategy in Section 2.1. As data blending and machine learning lie at the heart of our approach, the presented strategy is strongly focused on these two topics. As an introduction, Section 2.4 deals with the overall concept of data blending and the definition used in the context of this thesis. It also introduces a notation that can be used to describe data blending strategies, which is used throughout the thesis.

Due to the nature of the problem and the scope, in which it was tackled, the overall strategy had to be adapted several times over the course of the strategy development process. The strategy presented in this chapter includes all of these changes and differs from our initial strategy that was submitted after the first phase of the
Collider. However, this initial strategy and our reasoning for the adjustments are talked about in Section 2.5. This section also explains some of the design decisions found in the rest of the chapter. As some of the decisions that were made early on, but later were found to be not optimal, could not be changed anymore due to the amount of work these changes would have involved.

Section 2.6 deals with the concrete data blending strategy used in the rest of the thesis with a special focus on the domains of data that were chosen and where data from these domains can be acquired from.

Lastly, Section 2.7 focuses on the actual prediction of mood disorder risk. It is mainly concerned with how risk can be predicted, how the best prediction model for the task can be found and what methods already exist to measure the prediction quality.

### 2.1 Overview

It has been shown, that the personal environment of an individual can massively influence the individual’s mental health and as a result also the risk of developing a mood disorder [MO04]. The personal environment includes everything the individual is confronted with in their daily life, such as their social interactions, housing situation, work situation, education and family of origin. Factors relating to the individual’s personal environment that have been found to affect the mood disorder risk include the individual’s work environment and occupation [WP08], as well as employment status [Los+12]. Many of these are either directly or indirectly related to the socio-economic status of the individual.

As a result, the socio-economic status of an individual seems to have a considerable impact on the development of mood disorders [Doh+92]. It has been found that living in poverty is related with a high rate of emotional disturbance [HR07] and poor mental health [LM63]. It has been argued that social inequality, which is closely related to poverty and tied to the socio-economic status of the individual, causes psychosocial stress that can eventually lead to mental health problems, such as depression [Wil96].

All of this being the case, it seems feasible to focus on the personal environment when trying to predict the mood disorder risk. Unfortunately, information on an individual’s personal environment is rather hard to acquire, as this includes both highly sensitive data and a broad spectrum of possible data points. For instance, information on the highest level of education and personal relationships with family and friends of an individual could be of interest, when looking at the development of mood disorders.
2.2. Data Management

By taking part in the Collider, our team was provided access to data that is not accessible to researchers under normal circumstances. This access to real-world data, however, is invaluable when trying to solve real-world problems, as artificial datasets that are constructed for academic purposes tend to not reflect the challenges that need to be solved with real-world data, such as data cleaning and imputation, but also figuring out, which data is good and should be used and which data can be safely ignored for one reason or another. The payoff of using real-world data to solve real-world problems is mostly the quality of insights that can be extracted from the data. In the case of predicting the mood disorder risk and real-world clinical data, the quality of gathered insights should directly correlate with the prediction accuracy.

2.2 Data Management

When dealing with a huge number of datasets, a plan on how to manage the data has to be created. For all the acquired data in the research process it has to be ensured that the datasets are accurate, complete, authentic and reliable [C B06]. Data management becomes more important when acquiring sensitive data, as the data has to be stored in a trustful environment and results from the analysis performed on the sensitive data can only be published given the permission by the original provider of the sensitive data. Accenture elaborated a data supply chain in the area of digital trust [Acc16].

This data supply chain describes the different areas of data management starting with acquisition, storage, aggregation, analysis and use of data for selling or sharing. In the scope of this thesis, we mainly focus on the four topics of acquiring, storing, aggregating and analyzing data, highlighted by the dashed box in Figure 2.1.

Datasets were acquired from the Census of the United States and Geisinger Health Systems and stored in a trustful environment. These datasets were used for aggregation, integration and especially data blending. All the machine learning techniques, described in the presented work, are analyzing the manipulated data and different evaluation methods are used to examine if the analysis is good.

2.2.1 Data Quality

The foundation for data analysis is data of high quality. It is well known by experts in the field that “garbage in produces garbage out” [Kas+16] when it comes to training prediction models. But when talking about data quality it is important
Chapter 2. Strategy

FIGURE 2.1: Adaption of Data Supply Chain build by Accenture for the Purpose of Data Ethics [Acc16], licensed under Creative Commons Attribution 4.0 International License

Ingest data from sensors, systems, or humans, recording its provenance and consent for use wherever possible

Record data to a trusted location that is both secure and easily accessible for further manipulation

Combine disparate datasets to create a larger dataset that is greater than the sum of its parts

Examine and transform data with the purpose of extracting information and discovering new insights

Apply the insights gained from data analysis toward making decisions, affecting change, or delivering a product or service

Provide access to datasets or data insights to new sets of data manipulators or consumers

Remove data from servers to prevent release or use

FIGURE 2.1: Adaption of Data Supply Chain build by Accenture for the Purpose of Data Ethics [Acc16], licensed under Creative Commons Attribution 4.0 International License

to specify that it is not enough to assess whether a dataset is of good or bad quality. A dataset can be accurate, correct and free of errors, but not complete due to missing values. It is not easy to create a general data quality methodology for all the different areas of data quality, as the measurement of quality depends on the usage of data. Therefore we developed our own data quality methodology, shown in Table 2.1.

Over the last several decades a lot of research has been undertaken in the area of data quality [SYIT11]. Researchers described different perspectives on how to assess data quality. Data quality can be divided into multiple dimensions that are either task dependent or independent. In a task dependent metric, contextual knowledge is needed to assess the data, but the assessment process of a task independent metric can be applied to any dataset [PLW02]. The development of the dimension methodology shown in Table 2.1 was done in a task dependent way, as every


<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>How easily available is the data and are there any restrictions regarding access to the data?</td>
</tr>
<tr>
<td>Accuracy</td>
<td>How correct is the data and is it free of errors?</td>
</tr>
<tr>
<td>Completeness</td>
<td>Is the data missing values, does it include null values or default values?</td>
</tr>
<tr>
<td>Conformity</td>
<td>Do all data values conform to a consistent format?</td>
</tr>
<tr>
<td>Consistency</td>
<td>Does the data contradict itself or contain duplicate values</td>
</tr>
<tr>
<td>Credibility</td>
<td>To what extent can the data be trusted to be truthful and how reliable is the source of the data?</td>
</tr>
<tr>
<td>Objectivity</td>
<td>Is the data impartial and unbiased?</td>
</tr>
<tr>
<td>Timeliness</td>
<td>How current and up-to-date is the data?</td>
</tr>
<tr>
<td>Usability</td>
<td>Is the data easy to use, well structured and in an appropriate format?</td>
</tr>
</tbody>
</table>

Table 2.1: Dimensions of Data quality

dataset had to be assessed considering all dimensions.

It is possible to add more dimensions to a data quality methodology for better granularity, but all dimensions can be clustered into the eight separated groups of completeness, availability & accessibility, currency, accuracy, validity, usability & interpretability, reliability and credibility, and consistency [JSI13].

### 2.3 Technical Approach

As already mentioned before, our approach is focused on data blending and machine learning, with the goal of developing a predictor that can be used to predict the risk of developing a mood disorder for an individual. In broad terms the approach can be summarized as follows: To predict the mood disorder risk, a number of machine learning algorithms is used to train multiple prediction models on a blended set of data from different domains of data.

Our approach can be roughly divided into six separate phases or steps, which can be seen in Figure 2.2. The first two phases, problem assessment and data selection, are actually part of the strategy development process rather than the resulting strategy, yet we want to mention them here, as the strategy is primarily based on the result of those two phases.

In the first step, problem assessment, the overall problem was assessed and a possible solution was hypothesized, resulting in our main assumption that the personal
environment of an individual has a significant influence on the individual's mental state, influences the risk of developing a mood disorder.

The next two steps, data selection and data acquisition, are concerned with selecting the right kinds of data and getting access to the data. As data from one domain, namely clinical data, was initially provided by Geisinger Health Systems, the data selection step mainly consisted of thinking of new domains that could possibly be blended with the clinical data to gain further insights and improve the prediction quality. We identified four additional domains of data that both somehow relate to the clinical data and have datasets that are publicly available.

### 2.4 Data Blending

In the most general terms, data blending describes a relatively new and not well defined approach to data analysis, which is concerned with combining or blending, hence the name, heterogeneous and often disparate data from different domains in non-obvious ways to gather new or more meaningful insights. Being relatively new, a canonical and agreed-upon definition of data blending has yet to be defined. As a result, multiple and partially contradicting definitions exist. The following
2.4. Data Blending

thoughts on data blending relate to our understanding of data blending and might not align perfectly with one of the many definitions of data blending currently in circulation.

2.4.1 Differences to Data Integration

At a first glance, data blending seems to be closely related to the process of data integration, but the two concepts should not be confused, as they differ in both how the problem is approached and what their actual desired goal is. Data integration seeks to combine well-documented and mostly internal datasets by joining them over common attributes into a unified view of said datasets that can be used to solve a predefined data analytics task [Len02]. Data blending can be seen as a rather explorative approach to data analysis with the goal to find new opportunities to improve a predefined data analytics task by combining data that neither has to be well-structured nor directly related, while keeping the quality of said data as high as possible. Further, data blending is distinguished by combining datasets that belong to vastly different domains.

Figure 2.3 shows the conceptual difference between datasets typically found when dealing with data integration and data blending. It should be noted that joining datasets over common attributes is also commonly used as part of data blending, whereas combining non-overlapping datasets is usually not part of data integration, especially when those datasets stem from different domains.
One of the main challenges of data blending is to find a way to combine datasets that are not directly related and don’t share a direct overlap, but rather find a non-obvious common ground between the datasets that can be used to combine them, as can be seen in Figure 2.3. This common ground can be of many different forms and is unique to each data blending task. Based on this, it should be clear that data blending is also concerned with identifying such common grounds and finding ways to use them. As such, data blending is not a technical process, but rather involves a cognitive analysis of the problem at hand that relies heavily on domain knowledge and creative thinking to formulate a data blending strategy.

2.4.2 Data Quality Concerns

An important factor to keep in mind when dealing with data blending is data quality. As described in Section 2.2.1, data of good quality is crucial to the success of any data analysis task. When combining multiple independent datasets, data quality is an even more critical matter, as bad data can taint good data in terms of quality, if the necessary precautions are neglected or a wrong approach is chosen. The rule of thumb that input of bad data results in output of bad data also holds true for data blending. To give an example, joining a dataset that is missing a lot of values with a rather complete one will result in a considerably incomplete dataset.

But not only data of bad quality can lead to bad results. Another hurdle to look out for when blending data is the fact that the quality of source datasets does not directly correlate with the quality of the resulting dataset. For instance, joining two datasets of overall good quality by themselves can result in a dataset of rather bad quality in regards to consistency, if the two source datasets contain contradicting information. Thus, problems with data quality are easy to overlook when using data blending, if the issue is not handled with the utmost caution.

2.4.3 Data Blending Diagrams

Describing and visualizing a given data blending strategy is a challenge itself, as no canonical description language or notation exists that could be used to describe this new approach. In order to be able to express the data blending strategy presented in the scope of this thesis, we introduce a minimal notation that revolves around an abstract blending operator $+$ that represents the cognitive aspects of the data blending process. The proposed notation can be used to visualize a blending strategy using a data blending diagram. Data blending diagrams can be used to convey both top-level blending strategies between different domains of data and concrete blending approaches between distinct datasets, on a more detailed level.
2.4. Data Blending

In our notation, data is depicted as either belonging to certain domains of data, which are expressed by rounded rectangles, or consolidated into datasets, which are expressed by plain rectangles respectively. Connections between datasets, data domains and operators are denoted by solid lines. The $+$ operator takes $n$ inputs, denoted by data domains and datasets connected by lines with no arrows, and provides the blended dataset as an output, illustrated as a dataset connected by a line with an arrow facing the dataset. The $+$ operator always results in a single dataset, regardless of whether the input is data belonging to a certain data domain or dataset. This makes sense, as the goal of data blending is to create a single blended dataset. An exemplary data blending diagram can be seen in Figure 2.4.

Shown in this example is data from two data domains, namely climate data and social media data, that is blended into a resulting dataset using the $+$ operator.

Domains of data and datasets can also appear together in one data blending diagram on the condition that each dataset is contained by its data domain, as shown...
Dataset 1  Dataset 2  ...  Dataset n

+  

Blended Dataset

**Figure 2.6:** A generalized data blending approach between multiple datasets. The operator should be regarded as an abstract blending operator that is individual to each blending task.

in Figure 2.5. This kind of data blending diagram can be used to showcase the individual datasets while still being able to express the data blending between different domains of data on a conceptual level.

Blending operations, although not shown here, can be chained to convey a notion of order in which the datasets should be blended. This is of rather practical reason and is meant to give hints about the technical implementation, as the whole blending strategy could also just be modeled by a single blending operation that takes care of the whole blending process. Chaining is especially useful when one blending operation relies on the outcome of another blending operation, such as being able to join a third dataset on a field that is only available after having joined and aggregated two other datasets.

It should be noted, however, that the operator does not dictate the actual technical implementation, as it is only defined on a conceptual level. The actual implementation of a blending strategy will usually include a combination of operations that span the whole spectrum of technical possibilities, including simple joins over common attributes as well as post-aggregate joins.

Finally, a data blending diagram can omit the notion of data domains completely and only include data that belongs to specific datasets. This kind of data blending diagram is most useful for illustrating the details of a data blending strategy that has already been defined using a more top-level blending diagram. An example for a diagram of this kind can be seen in Figure 2.6. This diagram describes part of a data blending strategy that involves combining specific datasets. Again, the datasets are blended into one resulting dataset using the operator.
2.5 Initial Strategy

The initial strategy was developed within the scope of the first phase of the Collider in a given time frame of four weeks. As the teams did not get access to the clinical data for the first phase, no further datasets were provided and a strategy had to be developed from scratch.

The main goal of the strategy development in the first phase was to formulate a hypothesis or research question and find corresponding datasets that could be used to prove the hypothesis or answer the research question while being publicly accessible.

Geisinger Health Systems provided a corresponding data dictionary for every problem description, encompassing all attributes and ICD-9 codes that are part of the clinical dataset. With this data dictionary, it was possible to develop a prototypical blending strategy with the purpose of combining publicly accessible datasets with the clinical datasets. This prototypical strategy relies heavily on performing joins over common attributes or post-aggregate joins, i.e. aggregating the data over some fields before being able to join it.

The resulting strategy described a rudimentary approach to predicting a heightened individualized risk of developing a mood disorder by estimating the likelihood of having suffered from psychological trauma. This strategy built upon two main hypotheses, which were:

**Hypothesis I**

Psychological trauma in an individual’s personal history, such as sexual abuse and assault and physical abuse, especially in early stages of life, can lead to the development of a mood disorder or other mental health problem later in the individual’s life. If the likelihood that a person has suffered from psychological trauma in the past or will suffer from it in the future can somehow be estimated, we can predict a heightened risk of developing a mental health problem and especially a mood disorder for this individual.

**Hypothesis II**

Individuals that have suffered from psychological trauma in the past share certain features. This can be used to estimate the likelihood that a yet unknown individual has suffered or will suffer from psychological trauma.

Intuitively, the first hypothesis seems to be genuine and is in fact relatively easy to validate, as there has been extensive research in this area. In the case of childhood abuse and maltreatment, a positive correlation between psychological trauma and mood disorders has been shown. Additionally, sexual assault, especially in children and adolescents, has been found to play a significant role in developing major
depressive disorder (MDD). It has also been shown that certain mental health problems, such as depression, are significantly more prevalent in sexual assault victims in comparison to nonvictims.

The leading challenge was to validate the second hypothesis, as it has not been researched extensively. Although asking individuals in-depth questions about their personal history might already lead to some results in regards to their risk of mood disorder development, identifying abuse sufferers and assault victims is a rather relevant problem, as for instance only about 30% of affected individuals report sexual assault to police [Fin09]. Additionally, victims of sexual abuse might not be able to remember or may even actively suppress memories of their abuse.

To be able to challenge the second hypothesis, datasets of four different categories were selected:

- Crime Data
- Socio-Economic Data
- Medical History Data
- Child Abuse Data

Several police departments in the United States provide machine readable data on reported crimes in their subject area over a particular time (one year or more). Every crime report contains information about the date, crime type, city, state, and address (ZIP Code or intersection). This data can be used to estimate the sexual crime rate of a given ZIP Code in the United States.

The socio-economic data contains three datasets on ethnic makeup, economic structure, and the population density. The data is provided by the Census of the United States.

The clinical data provided by Geisinger Health Systems contains demographic and medical history data on treated patients in a Geisinger facility in Pennsylvania. As the data includes patients who suffered from a mood disorder and patients that are part of the control group, it is possible to estimate the mood disorder prevalence in a given ZIP Code.

The child abuse data contains two datasets on sexually / physically abused and nonabused children, including the children’s family environment, demographics, location of abuse, educational outcome, and mental health state of the affected children. This dataset could be used to estimate the likelihood of prior abuse in childhood and analyze the effect on the individual’s mental health state. The datasets are provided by the National Data Archive on Child Abuse and Neglect (NDA-CAN) and are subject to their rules of eligibility, which state that “only individuals
holding a faculty appointment or research position at an institution of higher education, a research organization, or a government agency” are eligible to attain access. It turned out that it was not possible for our team to use any data that concerned children in general. Therefore, the research strategy shifted in another direction. As the time frame of the Collider was very tight, it was not possible to develop a new strategy completely from scratch, however, adjustments to the strategy were possible. To avoid dealing with non-adult individuals, the focus of the strategy focused on women who were sexually abused in their previous life and were 18 years old or older at the time of the study.

The goal of the strategy was to blend the aforementioned intermediate datasets into one coherent dataset that can be used to train a classifier. Two important factors for the quality of the classification outcome are the selected features and the amount of training data. We have chosen the features to the best of our abilities, based on available data. The outcome of the blending process heavily relies on the selected features, as they determine the needed attributes.

As the intermediate datasets have no immediate overlap, they cannot simply be joined. To be able to blend them together, the individual entries in the intermediate datasets are divided into two distinct risk groups, that relate to a normal and a heightened risk of developing a mood disorder. As the second hypothesis states, this risk is derived from the direct correlation between the likelihood of having suffered from psychological trauma. The intermediate datasets are then joined on the risk group they belong to, which are either normal risk or heightened risk, to form a large dataset that is essentially a Cartesian product of all entries that belong to the same risk group. Each entry of the blended dataset consists of all selected features as well as the risk group to which it belongs. This blending process should generate sufficiently large training data for the classifier.

To classify individuals into normal and heightened risk, a classification model is needed which can be trained with the blended data. Such a classifier could be implemented as a multi-layer artificial neural network or deep neural network [AAA05][DHK13]. As we do have labeled training data in the form of our blended dataset, we can use this to train an ANN in a supervised manner using backpropagation with entries from the blended dataset being the input and the risk group being the expected output.

For the classification task at hand, an ANN with one hidden layer should be sufficient to classify reliably. Thus, the ANN’s architecture should be as follows: The input layer consists of \( n \) input neurons, where \( n \) is the number of selected features used to classify, which are fully connected to \( m < n \) nonlinear neurons in the hidden layer, which are in turn fully connected to a single output neuron in the output layer. Other supervised approaches to classification, such as Support Vector Machines, could also be used. However, in the course of the second phase,
the classification approach was exchanged for a regression task that predicts a risk expressed by a probability value instead of a simple risk group classification.

### 2.6 Domains of Data

In the scope of the Collider, Geisinger Health Systems provided the participating teams with real-world clinical data that could be used as a foundation for a data blending strategy. The goal of this data blending strategy, based on the provided clinical data, was to find a better answer to a defined research question or to find a better research question to answer in that process.

As mentioned in Section 2.1, research on mood disorder has revealed that the personal environment of an individual can massively influence the individual’s mental health. The main hypothesis in this thesis is that the personal environment of an individual is a key influence factor for the individual’s mental state and thus also affects the risk of developing a mood disorder. As the clinical data does not contain detailed information about the personal environment, except the sparse data of the demographic dataset, additional data is needed, preferably from other domains of data.

When developing a data blending strategy, an in-depth understanding of the provided data is required to be able to select additional datasets to use. It must be emphasized that data blending is an exploratory approach with the goal of finding new opportunities to improve a data analytics task, so any data that is even remotely related or has extractable information is conceivable.

In the initial strategy, the focus was placed upon data about child maltreatment and crime data. For the child maltreatment, two relevant datasets were selected that included information on reports of child maltreatment investigations by Child Protective Services and the socio-economic outcomes in mid to late adolescence of physically abused preadolescent children. Unfortunately, it turned out that it is not possible to get access to any datasets concerned with children, as such data is strictly regulated. Thus, the blending strategy had to be changed.

Instead of selecting information about the living situation of a maltreated child, the focus can be put on the neighborhood instead. Information about the prosperity of a region, the ethnic makeup or the population is needed to quantify the area. Therefore, dataset that are concerned with demographic, economic, geographic and crime data could be used. These datasets are useful to describe the surroundings of an individual and provides information on the prosperity, danger and potential damage in the corresponding neighborhood.
2.6. Domains of Data

To be able to predict danger and potential damage of an individual’s personal environment, the goal was to acquire detailed crime data of Pennsylvania, as all Geisinger facilities are located in Pennsylvania. Sadly, it was not possible to acquire detailed crime reports of the whole State of Pennsylvania or major cities in Pennsylvania over at least one year, another approach was needed. As it was possible to acquire demographic, economic and geographic data of all counties of the United States, detailed crime reports of major Cities of the United States were acquired with the idea to combine them with the stated datasets. This data should be used to train a prediction model, so the crime rate of all ZIP Codes in Pennsylvania can be predicted based on the demographic, economic and geographic data.

The demographic data extends the sparse information that the clinical data contains per patient such as, population or ethnic makeup. Economic data enriches the data with additional information, such as poverty rate or average household income. The geographic data adds the size of the ZIP Code area to the dataset, and the crime data describes the crime rate in a certain area.

With access to these datasets it should be possible to answer the research question how to predict mood disorder risk in a more precisely way than with only the clinical data, as there is much more information about the area a patient is living than without the additional data.

2.6.1 Data Sources

In the previous section, possible data domains were examined and their potential benefit to the clinical data described. The next challenge was to identify sources that provide these kinds of data on a ZIP Code level basis in a machine readable format.
Chapter 2. Strategy

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>Population data by ZIP Code</td>
<td>U.S. Census</td>
</tr>
<tr>
<td>Economy Data</td>
<td>Economy data by ZIP Code</td>
<td>U.S. Census</td>
</tr>
<tr>
<td>Density</td>
<td>Density data by ZIP Code</td>
<td>U.S. Census</td>
</tr>
<tr>
<td>Crime Data</td>
<td>Crime data by ZIP Code</td>
<td>Police Departments from major U.S. cities</td>
</tr>
<tr>
<td>Geisinger Demographics</td>
<td>Demographic ZIP Code level data of patients</td>
<td>Geisinger Health Systems</td>
</tr>
<tr>
<td>Geisinger Patient History</td>
<td>Patient history</td>
<td>Geisinger Health Systems</td>
</tr>
</tbody>
</table>

**Table 2.2: The selection of all used dataset in the data blending process**

For the population data, economic data and geographic data the API of Census of the United States[^1] was used, as it contains a powerful filter mechanism that can be used to only select the data needed for our purposes. The API is publicly available and provides machine readable datasets. Thereby, it was possible to search for demographic, economic and geographic data on a ZIP Code level basis.

The population dataset and economic dataset are part of the 2010-2014 American Community Survey. These 5-year estimates include results from the American Community Survey as well as from the Puerto Rico Community Survey. The data describes the time from January 1, 2010, until December 31, 2014. The 2010-2014 American Community Survey 5-year data results include estimates of demographic, social, housing and economic characteristics for people living in housing units and group quarters. In the 2010-2014 American Community Survey, Table DP03 titled "Selected Economic Characteristics" describes the economic data and Table B03002, titled "Hispanic or Latino Origin by Race" describes the population data.

The geographic data is provided by the 2010 Census gazetteer files for counties, county subdivisions (minor civil divisions / census county divisions), places, census tracts, 111th Congressional districts, American Indian Areas / Alaska Native Areas / Native Hawaiian Areas, school districts (elementary, secondary, and unified), state legislative districts (upper and lower), ZIP Code tabulation areas (ZCTAs), and urban areas for the 50 states, the District of Columbia, and Puerto Rico.

As no department in the United States provides publicly available data about all the reported crimes in the United States, many publicly available databases of major police departments had to be checked. Only detailed crime reports that include crime type, date, and an anonymous location, such as a ZIP Code or anonymous street address, were useful to create the crime dataset. If the crime scene or the

[^1]: http://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml
crime type was missing, the crime report had to be excluded. Unfortunately, this resulted in the exclusion of many datasets from police departments, not because of unstructured data, but rather due to incomplete crime reports or because the report was not provided in a machine readable format.

2.7 Predicting Mood Disorder Risk

The goal of this thesis, as stated in Chapter 1 and 1.4, is to train a prediction model on a blended dataset, created from datasets belonging to the described domains of data to predict the individualized risk of developing a mood disorder. This mood disorder risk is expressed as a probability value, where a low probability represents a low risk of developing a mood disorder and a high probability represents a high risk. Nonetheless, a moderate or high risk of developing a mood disorder for an individual is not an absolute indicator of developing a mood disorder, as the prediction model is trained on attributes describing the individual's environment and their medical history. Additional information, such as past occurrences of depression in the family or experienced significant trauma in their past, could improve the accuracy of the predicted risk as well.

Before training the described prediction model, it is important to know what the output of the model should be. If the output variable takes continuous values, a regression model should be used, but if the output takes class labels, a binary classification model should be used. As described in Section 2.5, the initial strategy called for a binary classification model, to classify an individual into a low-risk or high-risk group. Nevertheless, a binary classification approach is not enough to accurately represent the risk for an individual. Therefore, the classification task was replaced by a regression task that predicts a risk expressed by a probability value instead of a simple risk group classification.

As Geisinger Health Systems provided labeled clinical data, a supervised machine learning approach is used. Labeled data contains data points that are already classified or pre-clustered, whereas unlabeled data does not provide any information on which data point relates to which cluster. For unlabeled data, unsupervised machine learning approaches are used, with the goal of gathering the data points into certain groups.

For the prediction model, a particular form of regression called logistic regression is used. A logistic regression does not predict the binary value of the dependent variable, but rather predicts probabilities for each possible value, as shown in Section D.1.2.
In the strategy development process, it is not easy to select the best machine learning algorithm with the best quality in precision. For this reason, several algorithms are compared in the presented work, described in Section 2.1.

2.7.1 Evaluation

The process of building a complex machine learning model with good prediction quality requires a detailed understanding of the data and the machine learning tools utilized. Only with a well thought out approach, can the important aspects of a model be ensured and lead to a high-quality model. A critical part of this approach is the evaluation process of the model, which shows whether the stated goal was achieved or missed [RB99].

Before describing in detail how the models are evaluated, it is important to emphasize why it is crucial to assess the prediction quality of a model. A prediction model is created to predict the outcome based on input of never before seen data. In other words, the purpose of a prediction model is to not predict outcomes for samples found in the training data, but rather for input that was not part of the training set. It is evident that a prediction model that is trained and evaluated on the same data can achieve good prediction results, but this does not represent a real-world scenario and therefore cannot be used to estimate the generalization error. There is a multitude of machine learning algorithms that can be used to train a prediction model. The best model for given task can be found by evaluating and tweaking the model in order to identify the model with the best prediction quality.

For the presented work the following evaluation strategy is used:

1. Sample training and test set
2. Use cross-validation for model selection and optimization
3. Train model on the training set
4. Evaluate the model using the test set
5. Train the model on the whole dataset

The evaluation process starts by randomly splitting the prepared dataset into a training and a test set, e.g. 80 percent training data and 20 percent test data. Splitting the data before training is very important, as the evaluation results would be biased towards better results when using the same data for training as well as for testing.

Cross-validation is applied to choose the best machine learning model and optimize the hyper-parameters of said model. In the scope of this thesis, k-fold cross-validation is used. K-fold Cross-validation splits the training set into $k$ non-overlapping
subsets (called folds) and performs $k$ runs with $k-1$ folds as the training set and the remaining fold as the validation set. In every run, a different fold is used as the test set until every fold has been used, to predict the prediction quality once. A more detailed description of cross-validation can be found in Section D.2.1. With the help of cross-validation, the hyperparameters of the model can be tweaked to improve the prediction quality. When it is no longer possible to tweak the parameters while increasing the quality, the model is evaluated against the test data.

An alternative approach to this step could be the usage of an additional validation set that is created by splitting the dataset into three datasets or by using an additional dataset. With an unbiased additional dataset, it is possible to verify how good the model performs on real-world data. In the presented work a validation set is not part of the evaluation process, as additional clinical data could not be acquired. Instead, cross-validation is used to estimate the generalization error. The generalization error specifies how well the model will perform on out-of-sample data, i.e., data that was not used to train the model. Keeping an eye on the generalization error is important, as it can act as an indicator for overfitting. Keeping the generalization error small is a good way of ensuring that the model will perform well in a real-world scenario.

Step four of the evaluation process describes the model evaluation using the test set. This step is essential for an accurate evaluation, as so-called unseen data or out-of-sample data is used. In the process of model selection, optimization and training the best model, the test set is never used. After the best model is selected and trained, it is evaluated against the test set to evaluate the model accuracy on real-world data. As described above, if the model is evaluated on already seen data, the evaluation results are biased and not representative.

For the evaluation of a trained model, several metrics are useful to analyze the prediction quality. Therefore, the MSE, $R^2$, accuracy, ROC and precision and recall, described in Section D.2, are used in the presented work. These evaluation metrics can be used to compare the different models and to choose the one with the best prediction quality.
Chapter 3

Implementation

In the last chapter we have formulated a tentative strategy for our overall approach in order to develop a prediction model that can be used to predict the individualized risk of developing a mood disorder. This chapter focuses on the actual implementation of the prediction model, including the preliminary data preparation, the data blending process and the subsequent evaluation of the models trained.

The following sections will follow through the implementation process chronologically. We will start with the analysis of the data we will use in the implementation and a short description how the data was acquired in Section 3.1. Next, the data is cleaned and preprocessed to be used as training data for a prediction model, which is the focus of Section 3.2. Once the data is prepared, it can be joined into one coherent dataset that forms the base of our prediction model, which is described in Section 3.3.1. With the blended dataset ready to be used as training data, the actual prediction models are trained on the data in Section 3.3.2. Finally, Section 3.4 is concerned with the evaluation and analysis of the evaluation results of the trained prediction models.

3.1 Acquired Data

For the data blending process, several datasets from different domains of data were acquired with the goal of training a prediction model. Section 2.6 describes the selected data domains in detail and Section 2.6.1 states the used datasets and their corresponding sources.

Each dataset is described by its original structure and a data quality analysis, based on the data quality dimensions stated in Section 2.2.1. The data quality analysis focuses on the dimension of completeness, as datasets with missing values, such as missing values or default values, have to be preprocessed to deal with missing values. The approach that deals with missing values is described in Section 3.2.1.
Each dataset is assigned a unique namespace, which can be used to discriminate between the fields of the different datasets. For this purpose, each field of a given dataset is prefixed with the corresponding namespace.

### 3.1.1 Clinical Data by Geisinger Health Systems

The clinical data provided by Geisinger Health Systems contains two datasets on patient demographics and patient history. Both datasets contain a unique patient identifier that can be used to link the demographic and patient history of a patient.

The *patient demographics dataset* contains demographic data on patients that were treated at a Geisinger facility in Pennsylvania, including attributes, such as date of birth and gender. The *problem list dataset* contains the corresponding problem list for each patient, which is a list of diagnoses and treatments that the patient has received while under care by Geisinger Health Systems, including the date these diagnoses were given to the patient.

The patient demographics dataset contains 266,426 patients described by twelve different attributes.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIN_STUDY_ID</td>
<td>Study ID</td>
<td>A unique patient identifier</td>
</tr>
<tr>
<td>CLIN_DOB</td>
<td>Date of Birth</td>
<td>Patient’s date of birth</td>
</tr>
<tr>
<td>CLIN_DOD</td>
<td>Date of Death</td>
<td>Patient’s date of death</td>
</tr>
<tr>
<td>CLIN_GEN</td>
<td>Gender</td>
<td>Patient’s gender</td>
</tr>
<tr>
<td>CLIN_RA</td>
<td>Race</td>
<td>Patient’s Race</td>
</tr>
<tr>
<td>CLIN_MAR_STA</td>
<td>Marital Status</td>
<td>Patient’s current marital status</td>
</tr>
<tr>
<td>CLIN_REL</td>
<td>Religion</td>
<td>Patient’s religion</td>
</tr>
<tr>
<td>CLIN_ZIP</td>
<td>Patient ZIP Code</td>
<td>Patient’s current ZIP Code</td>
</tr>
<tr>
<td>CLIN_EMP_STA</td>
<td>Employment Status</td>
<td>Patient’s current employment status</td>
</tr>
<tr>
<td>CLIN_LANG</td>
<td>Language</td>
<td>Patient’s primary language</td>
</tr>
<tr>
<td>CLIN_INS</td>
<td>Insurance</td>
<td>Patient’s current insurance</td>
</tr>
<tr>
<td>CLIN_CP</td>
<td>Control Patient</td>
<td>Flag that determines whether patient is a control patient or not</td>
</tr>
</tbody>
</table>

| TABLE 3.1: Schema of the demographic dataset provided by Geisinger Health Systems |

The demographic data on the patients includes the unique patient identifier (CLIN_STUDYID), the patient’s date of birth (CLIN_DOB) and date of death (CLIN_DOD), gender (CLIN_GEN), race (CLIN_RAC), marital status (CLIN_MAR_STAT), religion (CLIN_REL), ZIP Code of residence (CLIN_ZIP), employment status (CLIN_EMP_STATUS), primary
3.1. Acquired Data

language (CLIN_LANG), current insurance (CLIN_INS) and the control patient status (CLIN_CP). The marital status, ZIP Code of residence, employment status and insurance do not describe the time of the initial diagnoses, but rather describe the current situation, as no historical data is available. The fields CLIN_GEN, CLIN_RAC, CLIN_MAR_STAT, CLIN_REL, CLIN_EMP_STATUS, CLIN_LANG, CLIN_INS are categorical. CLIN_GEN contains two valid values male and female. CLIN_RAC contains 15 different ethnicities, such as white, black or asian. CLIN_MAR_STAT contains five different states, including single, married and divorced. CLIN_REL can assume one of 68 different values, such as christian, roman catholic or no religious preference. CLIN_EMP_STATUS contains eight different types, including full time, part time and retired. CLIN_LANG can assume one of 66 different languages, for example English, Spanish and German. CLIN_INS contains more than 500 different health insurances including health insurances as well as medicare plans.

The data consists of two groups of patients, one group includes patients, who suffered from a mood disorder, while the other functions as the control group that includes patients that did not suffer from a mood disorder. The group to which a patient belongs is designated by the control patient status, in which a value of 0 denotes a patient that has suffered from a mood disorder and a value of 1 denotes a patient that is part of the control group. It is important to include patients from both groups, as it is not possible to run statistically significant analysis on the influence factors without a control group.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIN_STUDY_ID</td>
<td>Complete</td>
</tr>
<tr>
<td>CLIN_DB</td>
<td>Complete</td>
</tr>
<tr>
<td>CLIN_DD</td>
<td>3,653 deceased patients</td>
</tr>
<tr>
<td>CLIN_GEN</td>
<td>15 patients with gender unknown</td>
</tr>
<tr>
<td>CLIN_RAC</td>
<td>601 patients declined to provide and for 1,039 patients it was unable to obtain</td>
</tr>
<tr>
<td>CLIN_MAR_STAT</td>
<td>6,472 with unknown marital status</td>
</tr>
<tr>
<td>CLIN_REL</td>
<td>6 patients with no religious information, 18254 patients with information none and 2,471 with information not available</td>
</tr>
<tr>
<td>CLIN_ZIP</td>
<td>110 ZIP Codes for patients are missing</td>
</tr>
<tr>
<td>CLIN_EMP_STAT</td>
<td>8,227 patients no status was specified</td>
</tr>
<tr>
<td>CLIN_LANG</td>
<td>2 patients with no language information, 451 patients with unable to obtain</td>
</tr>
<tr>
<td>CLIN_INS</td>
<td>5,389 patients no information provided</td>
</tr>
<tr>
<td>CLIN_CP</td>
<td>Complete</td>
</tr>
</tbody>
</table>

TABLE 3.2: Data quality analysis of the demographic data provided by Geisinger Health Systems
The data quality analysis of the demographic data, described in Table 3.2, revealed that it is of rather good quality. According to the dimensions of accessibility and completeness, the dataset contains missing values or not useful data, as either patients did not provide further information or the information is not specified. The dataset is not publicly accessible, but access was granted to the teams that took part in the Collider, as part of a data sharing agreement.

Of the 266,426 records, each record contains a valid unique patient ID, a date of birth in a consistent format and the control patient status. The missing values for the date of death are not handled as an error, as the date is only added if the patient died. It is not specified if the patient died due to a diagnosed disease or due to another factor unrelated to patient history.

The gender attribute is not specified for 15 patients and and the ZIP Code is not defined for 110 patients.

The other attributes of race, marital status, religion, employed status, language, and insurance have many missing values, inconsistent values or values that are not useful.

The problem list dataset contains 2,322,729 records of diagnoses described by eight features.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIN_STUDY_ID</td>
<td>Study ID</td>
<td>A unique patient identifier</td>
</tr>
<tr>
<td>CLIN_DX_ID</td>
<td>DX ID</td>
<td>A unique ID for diagnosis</td>
</tr>
<tr>
<td>CLIN_DX NA</td>
<td>DX Name</td>
<td>Description of diagnosis</td>
</tr>
<tr>
<td>CLIN_ICD_9</td>
<td>ICD9</td>
<td>ICD9 code of diagnosis</td>
</tr>
<tr>
<td>CLIN_ICD_10</td>
<td>ICD10</td>
<td>ICD10 code of diagnosis</td>
</tr>
<tr>
<td>CLIN_NO_DAT</td>
<td>Noted Date</td>
<td>Date diagnosis was noted.</td>
</tr>
<tr>
<td>CLIN_RE_DAT</td>
<td>Resolved Date</td>
<td>Date diagnosis was resolved</td>
</tr>
<tr>
<td>CLIN_PRO_STA</td>
<td>Problem Status</td>
<td>Diagnosis status</td>
</tr>
</tbody>
</table>

Table 3.3: Schema of the problem List dataset provided by Geisinger Health Systems

The problem list on patients includes the unique patient identifier (CLIN_STUDY_ID), the diagnosis ID (CLIN_DX_ID), the diagnosis name (CLIN_DX NA), the ICD-9 code (CLIN_ICD_9), the ICD-10 code (CLIN_ICD_10), the noted date (CLIN_NO_DAT), the resolved date (CLIN_RE_DAT) and the status of the diagnosis (CLIN_PRO_STA).

The diagnosis ID and the diagnosis name are internally used in Geisinger facilities, whereas the ICD-9 code and the ICD-10 code are standardized disease codes. The noted date and the resolved date describe when the diagnoses were noted or resolved by a doctor. If the diagnoses are not yet resolved, the problem status is active and otherwise it is set to resolved.
The problem list contains 9,195 distinct ICD-9 codes and 13,700 distinct ICD-10 codes. As mentioned in Section B.1, the ICD-9 standard contains 13,000 distinct codes, whereas ICD-10 standard contains 68,000. Thus the data covers more than 70% of all ICD-9 codes and more than 20% of all ICD-10 codes.

The data quality analysis of the problem list dataset, described in Table 3.4, revealed that it is of rather poor quality. Regarding the dimensions of completeness, conformity and usability, the dataset contains a variety of missing values, such as missing ICD-9 codes and ICD-10 codes. Several records also contain more than one ICD-9 code. Additionally, a problem list does not exist for every patient in the provided dataset.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIN_STUDY_ID</td>
<td>Complete</td>
</tr>
<tr>
<td>CLIN_DX_ID</td>
<td>Complete</td>
</tr>
<tr>
<td>CLIN_DX_NA</td>
<td>Complete</td>
</tr>
<tr>
<td>CLIN_ICD_9</td>
<td>3,548 ICD9 codes missing</td>
</tr>
<tr>
<td>CLIN_ICD_10</td>
<td>143,065 ICD10 codes missing</td>
</tr>
<tr>
<td>CLIN_NO_DAT</td>
<td>5 dates missing</td>
</tr>
<tr>
<td>CLIN_RE_DAT</td>
<td>512,184 resolved dates are specified</td>
</tr>
<tr>
<td>CLIN_PRO_STA</td>
<td>Complete</td>
</tr>
</tbody>
</table>

Table 3.4: Data quality analysis of problem list provided by Geisinger Health Systems

Within the 2,322,729 records, each record contains a valid unique patient ID, valid diagnosis ID, diagnosis name and the problem status. Both, the ICD-9 and ICD-10 attributes have multiple missing values so that the records have to be dropped since a standardized diagnosis code is not specified. Another five records also have to be dropped, as they do not contain the noted date of the diagnosis and thus the diagnoses cannot be integrated into a chronological patient history. The resolved date is not evaluated in the process of the data quality analysis, since it is not possible to check if more than 512,184 diagnoses were already resolved.

As described above, a problem list does not exist for every patient, since the patient demographics dataset includes 266,426 unique patients, whereas the problem list dataset contains only a history for 206,004 individual patients.

Due to the fact that 47,463 records contain more than one ICD-9 code, a transformation into a consistent format is a must, as described in Section 3.2.8.

Our data blending approach is centered around the clinical data, as it is the only dataset that is provided by Geisinger Health Systems. All the additional acquired datasets are selected due to the fact that more information about the environment of patients is needed.
3.1.2 Population Data

The population dataset was acquired from the United States Census Bureau as part of the 2010-2014 American Community Survey[1]. The original name of the table from which this dataset was acquired is "Hispanic or Latino Origin by Race". The dataset contains detailed information of the ethnic makeup on a ZIP Code level basis, such as white, Hispanic, and various combinations of ethnicities. Statistical features such as the margin of error are also included in the dataset.

The dataset contains 33,120 rows and 76 attributes, but only 10 features are described in detail, as the other features are not used for the process of data blending, since they do not add meaningful insights to the personal environment of an individual.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP_ZIP</td>
<td>ZIP Code</td>
<td>The five digits ZIP Code</td>
</tr>
<tr>
<td>POP_TOTAL</td>
<td>Total Population</td>
<td>Population in the ZIP Code area</td>
</tr>
<tr>
<td>POP_HISP</td>
<td>Hispanic Population</td>
<td>Number of Hispanic residents in the ZIP Code area</td>
</tr>
<tr>
<td>POP_WHITE</td>
<td>White Population</td>
<td>Number of white residents in the ZIP Code area</td>
</tr>
<tr>
<td>POP_BLACK</td>
<td>Black Population</td>
<td>Number of black residents in the ZIP Code area</td>
</tr>
<tr>
<td>POP_AI</td>
<td>American Indian Population</td>
<td>Number of residents of American Indian ethnicity in the ZIP Code area</td>
</tr>
<tr>
<td>POP_ASIAN</td>
<td>Asian Population</td>
<td>Number of Asian residents in the ZIP Code area</td>
</tr>
<tr>
<td>POP_PI</td>
<td>Pacific Islander Popu-</td>
<td>Number of residents of Pacific Islander ethnicity in the ZIP Code area</td>
</tr>
<tr>
<td>POP_OTHER</td>
<td>Other Population</td>
<td>Number of residents of other ethnicities in the ZIP Code area</td>
</tr>
<tr>
<td>POP_MIXED</td>
<td>Mixed Population</td>
<td>Number of residents of mixed ethnicity in the ZIP Code area</td>
</tr>
</tbody>
</table>

The population dataset includes the ZIP Code (POP_ZIP), the total population (POP_TOTAL), the number of Hispanic residence (POP_HISP), the number of white residents (POP_WHITE), the number of black residents (POP_BLACK), the number of residents of American Indian ethnicity (POP_AI), the number of Asian residents (POP_ASIAN), the number of Pacific Islander ethnicity (POP_PI), the number of other ethnicities (POP_OTHER), and the number of mixed ethnicity (POP_MIXED).

[1]https://www.census.gov/programs-surveys/acs/
3.1. Acquired Data

(Pop\_Asian), the number of residence of Pacific Islander ethnicity (Pop\_PI), the number of residents of other ethnicities (Pop\_OTHER) and the number of residents of mixed ethnicity (Pop\_MIXED).

The data quality analysis of the demographic data revealed that it is of rather good quality. According to the data quality dimensions of completeness and the dimension of correctness, the dataset contains 145 rows where only zeros are specified for all ethnicities. These need to be replaced with average values, as described in Section 3.2.2.

As described in Section 2.6, the population data extends the sparse information that the clinical data contains per patient with the distribution of the ethnic makeup of the individual’s neighborhood to get better insights of the environment, in which the individual is living. It has been proposed that individuals who live in a neighborhood with a greater proportion of people from the same ethnic group are much less susceptible for depressive symptoms than individuals who live in a neighborhood with a lower proportion of the same ethnic group [Hal93] [HN00]. Conversely, it has also been discussed that the exact opposite is true, namely that a greater population of people from the same ethnicity might adversely affect mental health [Mai+10]. Regardless of which hypothesis, if either, is true, the ethnic makeup of the neighborhood an individual lives in is worth looking at in the context of the development of mood disorders.

3.1.3 Economic Data

The economic dataset was acquired from the United States Census Bureau as part of the 2010-2014 American Community Survey[^1]. The original name of the table from which this dataset was acquired is "Selected Economic Characteristics". It contains information on the average household income and its prosperity, the unemployment rate, and the occupation rate for several jobs. This data is important, as it provides better insight into the environment and prosperity of a ZIP Code. The economic dataset contains 33,120 rows and 551 attributes. Almost each piece of information is provided as both absolute value and percentage value for the corresponding ZIP code. Additionally, statistical attributes such as the margin of error are also part of the dataset.

As there are too many features to explain each in detail, only the features that are important for the blending process are described.

The economic dataset includes the ZIP Code (ECON\_ZIP), the percentage of unemployed people (ECON\_UNEMP), the percentage of people in occupation management.

[^1]: https://www.census.gov/programs-surveys/acs/
### Table 3.6: Schema of the economic dataset provided the U.S. Census Bureau

(ECON_MGT), the percentage of people in occupation service (ECON_SERV), the percentage of people in occupation sales (ECON_SALES), the percentage of people in occupation construction (ECON_CONST), the percentage of people in occupation transport (ECON_TRANS), the average commuting to work time (ECON_CTW), the percentage of people without insurance (ECON_INSURED), the percentage of people below poverty level (ECON_BELOW_POV), the median household income (ECON_INC_MED), and the average household income (ECON_INC_AVG).

The data quality analysis of the economic data revealed that it is of rather good quality. Regarding to the data quality dimension of completeness and correctness, every described feature in Table 3.6 contains missing values, except the ZIP Code feature.

2,828 records contain at least one feature without a valid value specified. These
3.1. Acquired Data

### Field ID Description

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECON_ZIP</td>
<td>Complete</td>
</tr>
<tr>
<td>ECON_UNEMP</td>
<td>512 records missing unemployment rate</td>
</tr>
<tr>
<td>ECON_OCC_MGT</td>
<td>546 records missing the occupation rate</td>
</tr>
<tr>
<td>ECON_OCC_SERV</td>
<td>546 records missing the occupation rate</td>
</tr>
<tr>
<td>ECON_OCC_SALES</td>
<td>546 records missing the occupation rate</td>
</tr>
<tr>
<td>ECON_OCC_CONST</td>
<td>546 records missing the occupation rate</td>
</tr>
<tr>
<td>ECON_OCC_TRANS</td>
<td>546 records missing the occupation rate</td>
</tr>
<tr>
<td>ECON_CTW</td>
<td>2,828 records missing commuting to work information</td>
</tr>
<tr>
<td>ECON_NO_INSUR</td>
<td>401 records missing the no insurance rate</td>
</tr>
<tr>
<td>ECON_BELOW_POV</td>
<td>509 records missing the below poverty rate</td>
</tr>
<tr>
<td>ECON_INC_MED</td>
<td>1,124 records missing the median household income value</td>
</tr>
<tr>
<td>ECON_INC_AVG</td>
<td>1,109 records missing the average household income value</td>
</tr>
</tbody>
</table>

| TABLE 3.7: Data quality analysis of the economic dataset provided by the U.S. Census Bureau |

need to be replaced with the corresponding median values, as described in Section 3.2.2.

Studies have shown that there is a positive connection between the admission rates of people who suffered from affective disorder and the fields of commuter balance, income inequality and unemployment rates [Los+12]. As the dataset contains the addressed fields of the study and various other fields, it might extend the information of the environment of an individual in a positive way, as described in Section 2.6.

3.1.4 Geographic Data

The geographic dataset is acquired from the United States Census Bureau 2010 Gazetteer Files on a ZIP Code level basis[^1]. It contains information on the population size and the land area size. It contains 33,120 records and nine features.

The dataset includes the ZIP Code (GEO_ZIP), the population size (GEO_POP), the housing unit count (GEO_HUC), the land area in square meter (GEO_LA), the water area in square meter (GEO_WA), the land area in square miles (GEO_LA_SM), the water area in square miles (GEO_WM_SM), the latitude (GEO_LAT) and the longitude (GEO_LON).

[^1]: https://www.census.gov/geo/maps-data/data/gazetteer2010.html
Table 3.8: Schema of the geographic dataset provided by U.S. Census Bureau

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO_ZIP</td>
<td>ZIP Code</td>
<td>The corresponding ZIP Code</td>
</tr>
<tr>
<td>GEO_POP</td>
<td>Population</td>
<td>Size of the population in the corresponding ZIP Code</td>
</tr>
<tr>
<td>GEO_HUC</td>
<td>Housing Unit Count</td>
<td>Housing unit count in the corresponding area</td>
</tr>
<tr>
<td>GEO_LA</td>
<td>Land Area (square meters)</td>
<td>Land area size of the corresponding area in square meters</td>
</tr>
<tr>
<td>GEO_WA</td>
<td>Water Area (square meters)</td>
<td>Water area size of the corresponding area in square meters</td>
</tr>
<tr>
<td>GEO_LA_SM</td>
<td>Land Area (square miles)</td>
<td>Land area size of the corresponding area in square miles</td>
</tr>
<tr>
<td>GEO_WA_SM</td>
<td>Water Area (square miles)</td>
<td>Water area size of the corresponding area in square miles</td>
</tr>
<tr>
<td>GEO_LAT</td>
<td>Latitude</td>
<td>Latitude of the corresponding area in square kilometers</td>
</tr>
<tr>
<td>GEO_LON</td>
<td>Longitude</td>
<td>Longitude of the corresponding area in square kilometers</td>
</tr>
</tbody>
</table>

The data quality analysis, described in Table 3.9, revealed that the geographic dataset is of rather good quality. According to the data quality dimensions of correctness and completeness, the dataset contains missing values and incorrect values that cannot be seen at first glance.

Assuming that the housing count number and the population size are correlated in a ZIP Code, it is not possible that the housing count number is zero, while the population size is greater than zero. Therefore, a housing count number greater than zero with a correlating population size of zero is attributable to the fact that the ZIP Code only contains hotels and motels as housing units – and therefore it is not considered as incomplete data.

Studies have revealed that the admission rates of people who suffered from affective disorders, especially with additional diseases like schizophrenia, is correlated with the population size and population density [Los+12]. However, the population density is not part of the dataset, but it be easily calculated with the population size and the land area.

As described in Section 2.5, information on the personal environment of an individual is needed. With the geographic data, it is possible to calculate the population density, meaning the amount of people living in a defined area, e.g. number of people per square mile.
3.1. Acquired Data

Field ID Description

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO_ZIP</td>
<td>Complete</td>
</tr>
<tr>
<td>GEO_POP</td>
<td>144 rows with a population size of zero</td>
</tr>
<tr>
<td>GEO_HUC</td>
<td>304 ZIP Codes with a population size greater than zero, but zero housing units</td>
</tr>
<tr>
<td>GEO_LA</td>
<td>Land area size of the corresponding area in square meters</td>
</tr>
<tr>
<td>GEO_WA</td>
<td>Water area size of the corresponding area in square meters</td>
</tr>
<tr>
<td>GEO_LA_SM</td>
<td>Land area size of the corresponding area in square miles</td>
</tr>
<tr>
<td>GEO_WA_SM</td>
<td>Water area size of the corresponding area in square miles</td>
</tr>
<tr>
<td>GEO_LAT</td>
<td>Complete</td>
</tr>
<tr>
<td>GEO_LON</td>
<td>Complete</td>
</tr>
</tbody>
</table>

TABLE 3.9: Data quality analysis of geographic data provided by U.S Census Bureau

3.1.5 Crime Data of Major US Cities

In the presented work, information about the sexual crime rate of sufficient ZIP Codes is needed to successfully create a crime dataset. To obtain a preferably high coverage of the United States, the cities of Chicago, Austin, Indianapolis, Philadelphia and Los Angeles, were chosen, as the local police departments of these cities provide detailed public data on all crimes reported to the police department over a period of at least one year. These public datasets slightly differ in their structure, but all of them contain the date of the crime, the crime type, the approximate location of the crime and several other fields, such as the responsible police department and the ID of the dispatched officer. Only the public crime data of the City of Austin contains information about the ZIP Code in which the crime was reported. The other datasets only provide an approximate location of the offense, i.e. street intersection or block.

Table 3.10 shows the schema of the crime data of the City of Chicago and how detailed a crime report is recorded. The data contains more than six million crime reports from 2001 to present with a delay of one week. Unfortunately, not all police departments in the United States provide their crime reports in the same way as the City of Chicago does. Therefore, a general data quality analysis of all datasets was not possible.

The crime dataset is of the overall worst quality compared to the other datasets. This stems from the fact that the single crime datasets differ in their structure. Some
### Table 3.10: Schema of crime dataset by the police department of Chicago

<table>
<thead>
<tr>
<th>Field ID</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIM_ID</td>
<td>ID</td>
<td>Unique identifier for the record</td>
</tr>
<tr>
<td>CRIM_CN</td>
<td>Case Number</td>
<td>Internal police department number</td>
</tr>
<tr>
<td>CRIM_DAT</td>
<td>Date</td>
<td>When the incident occurred</td>
</tr>
<tr>
<td>CRIM_BLK</td>
<td>Block</td>
<td>Partially redacted address</td>
</tr>
<tr>
<td>CRIM_PT</td>
<td>Primary Type</td>
<td>Primary description</td>
</tr>
<tr>
<td>CRIM_DES</td>
<td>Description</td>
<td>Secondary description</td>
</tr>
<tr>
<td>CRIM_LOC DES</td>
<td>Location</td>
<td>Location description</td>
</tr>
<tr>
<td>CRIM_ARR</td>
<td>Arrest</td>
<td>Indicates whether an arrest was made</td>
</tr>
<tr>
<td>CRIM_DOM</td>
<td>Domestic</td>
<td>Indicates whether an arrest was made</td>
</tr>
<tr>
<td>CRIM_BT</td>
<td>Beat</td>
<td>Indicates the beat where the incident occurred</td>
</tr>
<tr>
<td>CRIM_DIS</td>
<td>District</td>
<td>Indicates the police district where the incident occurred</td>
</tr>
<tr>
<td>CRIM_WA</td>
<td>Ward</td>
<td>The ward (City Council district) where the incident occurred</td>
</tr>
<tr>
<td>CRIM_CA</td>
<td>Community Area</td>
<td>The community area where the incident occurred</td>
</tr>
<tr>
<td>CRIM_FBI</td>
<td>FBI Code</td>
<td>Indicates the crime classification</td>
</tr>
<tr>
<td>CRIM_X</td>
<td>X Coordinate</td>
<td>The x coordinate of the location where the incident occurred</td>
</tr>
<tr>
<td>CRIM_Y</td>
<td>Y Coordinate</td>
<td>The y coordinate of the location where the incident occurred</td>
</tr>
<tr>
<td>CRIM_YEA</td>
<td>Year</td>
<td>Year the incident occurred</td>
</tr>
<tr>
<td>CRIM_UPD</td>
<td>Updated on</td>
<td>Date and time the record was last updated</td>
</tr>
<tr>
<td>CRIM_LAT</td>
<td>Latitude</td>
<td>The latitude of the location where the incident occurred</td>
</tr>
<tr>
<td>CRIM_LON</td>
<td>Longitude</td>
<td>The longitude of the location where the incident occurred</td>
</tr>
<tr>
<td>CRIM_LOC</td>
<td>Location</td>
<td>The location where the incident occurred (latitude and longitude)</td>
</tr>
</tbody>
</table>

of the crime datasets contain detailed information for each crime report, whereas others contain only sparse information with a lot of missing values. As a consequence, the usability of the crime data is bad and it has to be prepared with special attention before it can be used for the analysis.
3.2 Data Preparation

Before being able to do anything useful with newly acquired datasets, it is necessary to prepare the data for further use. This data preparation includes cleaning the data from invalid values, filling in or removing data points with missing values and combining data from different sources and data formats. Especially in regards to data blending, data preparation is responsible for transforming, aggregating and joining the different datasets according to the data blending strategy.

As described in Section 3.1, we acquired 10 distinct datasets in total that needed to go through the process of data preparation with the goal of creating a unified dataset that can be used to train a prediction model using a supervised learning algorithm.

3.2.1 Dealing with Missing Data

As can be seen from the list of acquired datasets and the respective evaluation of their data quality, one of the main challenges we had to deal with in this matter was incomplete, inaccurate and unavailable data.

There are several methods that can be used to deal with missing and inaccurate values. Two of the most widely used approaches are the following [HDA08]:

1. Ignore the samples that include missing or inaccurate values and only analyse the complete and accurate samples

2. Replace the inaccurate or missing values using some kind of imputation method

Ignoring samples that include missing values is a computationally cheap and conceptually easy way of dealing with missing data, as it only involves removing samples from the dataset. However, this is only a viable option if there is just a small percentage of affected samples. When missing values are spread throughout the dataset, ignoring the affected samples could mean losing a significant amount of data. This is especially hurtful if the number of samples is not particularly extensive in the first place.

In these cases, replacing inaccurate or missing values using an imputation method is more feasible, as the amount of data available for use is not decreased. Imputation can be conducted in several different ways. The most basic imputation method for missing values is to use a default value, such as the value that occurs the most, in place of the missing value. Another simple imputation technique involves calculating the mean or average value for the missing field and using this calculated value as the replacement for the missing value. It should be clear, that these imputation methods do not necessarily fill in the missing values with high-quality values in
Chapter 3. Implementation

Figure 3.1: Data blending diagram of the blending strategy for census community dataset

regards to the accuracy and correctness of the data, but they at least preserve some of the inherent truth of the data.

More powerful imputation methods involve imputing the values based on some kind of prediction model to generate more probable accurate values. Even more advanced imputation methods try to express the uncertainty that results from the imputation by means of confidence intervals and the like [Sch02].

For the census community data consisting of demographic, economic and geographic data acquired from the U.S. Census Bureau, we used simple imputation by the median value of the non-missing values. The statistically insignificant values of the mood disorder prevalence data is imputed by a combination of both the median value and a linear regression model trained on the significant values. The crime data is a completely synthetic dataset, in so far that all values for Pennsylvania are estimated using a regression model. For the clinical dataset, missing values are not imputed at all, records with missing values are simply ignored, as sufficient data is available.

3.2.2 Census Community Data

The combination of the Census community dataset is a part of the process of data blending. Therefore, the demographic, economic and geographic datasets are joined into one coherent dataset, named the census community dataset, which contains information on the environment of any ZIP Code that is part of the Census database.

The three datasets are joined on the ZIP Code feature that all datasets have in common. An inner join is used, so that only ZIP Codes that are part of all three datasets are joined into one record per ZIP Code. The resulting dataset contains 30,562 records, whereas 2,558 records were dropped since these ZIP Codes are not part of all datasets. Although in the United States approximately 43,000 five digits ZIP
### Field IDs

<table>
<thead>
<tr>
<th>Field IDs</th>
<th>Operation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP_HISP</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_WHITE</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_BLACK</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_AI</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_ASIAN</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_PI</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_OTHER</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>POP_MIXED</td>
<td>Divided by corresponding total of the demographic dataset</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_UNEMP</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_OCC_MGT</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_OCC_SERV</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_OCC_SALES</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_OCC_CONS</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_OCC_TRANS</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_INSURED</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>ECON_BELOWL_POV</td>
<td>Divided by 100</td>
<td>Value between 0 and 1</td>
</tr>
<tr>
<td>GEO_DENS</td>
<td>Total population divided by land area of the geographic dataset</td>
<td>Absolute value</td>
</tr>
</tbody>
</table>

*Table 3.11: Overview of data manipulation of the census datasets*
Codes exist, the datasets only include 33,120 records. The schema of the dataset is shown in Table 3.11.

Following, the resulting dataset has to be normalized. Normalization, is an important step, as a lot of features of the dataset are described by different units and scales, but a consistent format is needed for some machine learning algorithms. Therefore, the features `pop_hisp`, `pop_white`, `pop_black`, `pop_ai`, `pop_asian`, `pop_pi`, `pop_other`, `pop_mixed`, `econ_unemp`, `econ_occ_mgt`, `econ_occ_serv`, `econ_occ_sales`, `econ_occ_const`, `econ_occ_trans`, `econ_no_insur`, `econ_below_pov` and `geo_dens` had to be converted to a consistent format.

The ethnicity features are transformed into fractions, i.e. they are divided by the population size of the corresponding ZIP Code, resulting in a value between 0 and 1. All economic rates are divided by 100 to convert the percentage value into a value between 0 and 1.

As the density is not part of the geographic dataset, it is calculated by dividing the land area in square meters by 100,000 resulting in square kilometers. Finally, the result is divided by the population size of the corresponding ZIP Code.

The result is a dataset with the features `cc_pop_hisp`, `pcc_op_white`, `cc_pop_black`, `cc_pop_ai`, `cc_pop_asian`, `cc_pop_pi`, `cc_pop_other`, `cc_pop_mixed`, `cc_econ_unemp`, `cc_econ_occ_mgt`, `cc_econ_occ_serv`, `cc_econ_occ_sales`, `cc_econ_occ_const`, `cc_econ_occ_trans`, `cc_econ_insured` and `cc_econ_below_pov` that are in a correct, consistent format. However, the fields `geo_dens`, `econ_inc_med`, `econ_inc_avg` and `econ_ctw` are described by absolute values. Therefore, these five features were standardized by subtracting the feature’s average value from the original feature value divided by the standard deviation of the feature.

\[
x' = \frac{x - \bar{x}}{s} \tag{3.1}
\]

\(x\) describes the original feature value, \(\bar{x}\) the average of the feature and \(s\) the standard deviation of the feature. The resulting value of the computation is a value between 0 and 1.

The resulting standardized features are called `cc_geo_dens`, `cc_econ_inc_med`, `cc_econ_inc_avg` and `cc_econ_ctw`.

As the demographic dataset and the economic dataset contain missing values, as described in Section 3.1.2 and in Section 3.1.3, it is important to replace these missing values with representative values - records should not be dropped simply due to missing values in a record. Therefore, for each feature, the median was calculated and each missing value replaced by the corresponding median. Replacing the missing values with the median instead of the average is a better approach, as the means can be skewed by outliers.
### Table 3.12: Schema of blended census community dataset

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP_ZIP</td>
<td>The five digits ZIP Code</td>
</tr>
<tr>
<td>CC_TOTAL</td>
<td>Population in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_HISP</td>
<td>Fraction of Hispanic people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_WHITE</td>
<td>Fraction of white people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_BLACK</td>
<td>Fraction of black people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_AI</td>
<td>Fraction of American Indian people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_ASIAN</td>
<td>Fraction of Asian people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_PI</td>
<td>Fraction of Pacific Islander people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_OTHER</td>
<td>Fraction of other people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_POP_MIXED</td>
<td>Fraction of mixed people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_UNEMP</td>
<td>Fraction of unemployed people in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_OCC_MGT</td>
<td>Fraction of occupation management in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_OCC_SERV</td>
<td>Fraction of occupation service in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_OCC_SALES</td>
<td>Fraction of occupation sales in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_OCC_CONS</td>
<td>Fraction of occupation construction in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_OCC_TRANS</td>
<td>Fraction of occupation transport in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_CTW</td>
<td>Standardized absolute value in minutes in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_INSURED</td>
<td>Fraction of people with insurance in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_BELOW_POV</td>
<td>Fraction of people below poverty level in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_INC_MED</td>
<td>Standardized absolute value in dollars in the ZIP Code area</td>
</tr>
<tr>
<td>CC_ECON_INC_AVG</td>
<td>Standardized absolute value in dollars in the ZIP Code area</td>
</tr>
<tr>
<td>CC_GEO_DENS</td>
<td>Standardized density of the land area in the ZIP Code area</td>
</tr>
</tbody>
</table>
Thereby, in 2,343 records several missing values are replaced by the corresponding median. It should be noted that replacing missing values with the median does not distort the dataset, as it does not change the already calculated median of a feature. Following the process of standardization, the dataset contains only features that centered around 0.

The final census community dataset contains 21 features: nine from the population dataset, eleven from the economic dataset, one from the geographic dataset and the ZIP Code that is part of each dataset. All features and their description can be seen in Table 3.12.

### 3.2.3 Estimation of Crimes in Pennsylvania

All of the patients described in the clinical dataset provided by Geisinger Health Systems were treated in a Geisinger facility in Pennsylvania. As more than 97% of the patients live in Pennsylvania, our original idea was to acquire data on crimes to calculate the crime rate per ZIP Code. However, these detailed crime reports are not made publicly available by the State of Pennsylvania. However, detailed crime data is available for several major U.S. cities. To be able to use this crime data, we chose to construct a synthetic crime dataset for the ZIP Codes found in the clinical dataset from the crime data that is available. The approach we chose for this thesis is unconventional and in retrospective not advisable, as it introduces feature dependencies between the census community features and the constructed crime indicator feature. The main reason we chose to still include the approach in this thesis is that the crime indicator can later be used to demonstrate the principles of data blending.

Information about the crime rate of a sufficient number of ZIP Codes is needed to train an estimator successfully. To obtain a preferably high coverage of the United States, the cities Chicago, Austin, Indianapolis, Philadelphia and Los Angeles were chosen, as the local police departments of these cities provide detailed public data on all crimes reported to the police department over a period of at least one year. These public datasets slightly differ in their structure, but all of them contain the date of the crime, the crime type, the approximate location of the crime and several other fields, such as the responsible police department and the ID of the dispatched officer, as described in Section 3.1.5. The minimal set of features that are contained in every acquired crime dataset can be seen in Table 3.13.

The features CRIM_DAT, CRIM_TP, CRIM_ADD, CRIM_CIT and CRIM_ST contain enough information for every single crime report. Therefore, we are able to create our own crime dataset.
3.2. Data Preparation

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIM_DAT</td>
<td>Date of crime</td>
</tr>
<tr>
<td>CRIM_TP</td>
<td>Type of the crime</td>
</tr>
<tr>
<td>CRIM_ADD</td>
<td>Anonymized address of the reported crime or the ZIP Code</td>
</tr>
<tr>
<td>CRIM_CIT</td>
<td>The ZIP Code of the reported crime</td>
</tr>
<tr>
<td>CRIM_ST</td>
<td>State where the crime took place</td>
</tr>
</tbody>
</table>

Table 3.13: Schema of minimal crime datasets

The City of Austin’s public crime data is the only one, which contains information about the ZIP Code in which the crime was reported. The other datasets only provide an approximate location of the crime, (i.e. street intersection, block) for which a corresponding ZIP Code had to be identified. To obtain the corresponding ZIP Code for a given crime location, the APIs of the United States Census Bureau and the Google Geocoding API were used.

The Census API is the first choice to look for unknown ZIP Codes, as no daily request limit is set. If the Census API does not provide a valid ZIP Code for a given address, an additional request with the corresponding address is sent to the Google Geocoding API. The Google Geocoding API is used as a fallback and second source, as the API is not free of use. The free requests are limited to 2,500 per day. If the limit is reached the IP is blocked and additional requests are prohibited.

As mentioned in Section 2.5, the initial strategy is focused on reported sex crimes. Therefore, the crime datasets of the described cities were filtered by sexual crimes, resulting in 415 sex crimes in the City of Austin in 2014, 44,162 sex crimes for the City of Chicago from 2001 to 2015, 567 sex crimes in the City of Indianapolis in 2014, 2565 sex crimes for the City of Los Angeles in 2014 and 7,167 sex crimes for the City of Philadelphia from 2012 to 2014. This resulted in 54,461 crime reports in which the corresponding ZIP Code had to be identified, as the sex crimes in Austin, already contained the ZIP Code.

Since the geocoding API of the Census provides an endpoint for batch processing with a maximum of 1,000 addresses per request and an endpoint for single address requests, both endpoints were tested to determine the fastest endpoint for identifying the ZIP Codes. Therefore, two node.js scripts were developed, one for the batch processing endpoint and another that processes the addresses line by line and sends single requests.

The Census provides a detailed documentation on how to use both endpoints:

[https://geocoding.geo.census.gov/geocoder/Geocoding_Services_API.pdf](https://geocoding.geo.census.gov/geocoder/Geocoding_Services_API.pdf)
For the batch processing endpoint, the following parameters had to be specified to identify the corresponding ZIP Code to every single address.

**returntype** location

**benchmark** location

**addressFile** CSV file

The returntype is set to location, as only the geolocation is needed to identify the ZIP Code. Benchmark is a numerical ID or name that references the version of the tool. Benchmark 4 describes the Census 2010 Benchmark. The CSV file must be formatted to include the unique ID, street address, city, state and ZIP Code. If a component is missing from the dataset, it must still retain the delimited format with a missing value, as only the unique ID and the street address are required fields.

For the single record geocoding service endpoint, the following parameters had to be specified to identify the corresponding ZIP Code for the address.

**searchtype** address

**returntype** location

**benchmark** 4

**street** The street address

**city** The city

**state** The state

**format** default: json

The searchtype is set to address, as a detailed address is specified by street, city and state in the body request. The returntype and benchmark are equivalent to the batch processing endpoint.

For the Google Geocoding API, a third script was developed that is similar to the single request script for the U.S. Census Bureau. It was important to encapsulate both scripts, as the U.S. Census Bureau has no limit and the whole job does not need additional user input, whereas the Google Geocoding API has a per day limit and the limit can only be eluded by resetting the IP address by hand or paying for a business account.

For the process of identifying the ZIP Code, the crime reports were converted into a CSV file with the fields unique ID, street, city, and state, since the batch process endpoint expects a file with the described columns. Before all datasets were split

[https://geocoding.geo.census.gov/geocoder/returntype/addressbatch](https://geocoding.geo.census.gov/geocoder/returntype/addressbatch)

[https://geocoding.geo.census.gov/geocoder/returntype/searchtype?parameters](https://geocoding.geo.census.gov/geocoder/returntype/searchtype?parameters)
into multiple files containing 1,000 addresses, a test file was created to test both endpoints for quality and efficiency.

It was initially expected that the results would be the same for both endpoints, and that the batch process would be faster than single requests. It turned out that the batch processing performance is worse than the single requests. The badge processing endpoint identified 645 ZIP Codes of 1,000 and in the response, only the identified addresses were returned, whereas the single address endpoint identified 895 ZIP Codes for the same 1,000 addresses. Consequently, the single address endpoint was chosen for the ZIP Code identification of the 53,590 addresses of sex crimes.

As mentioned before, if the Census API response contains a valid ZIP Code, the ZIP Code is stored in the CSV file and the next line is processed, otherwise, the same request is sent to the Google Geocoding API and the ZIP Code in the response is stored in the CSV file. If the response does not contain a valid ZIP Code as well, the corresponding crime report is dropped, as the crime report cannot be allocated to the proper area.

Less than 2% of the data was dropped due to the fact that both APIs could not provide a valid ZIP Code.

In our initial implementation only sex crimes had been filtered from the crime datasets, as we were still only focusing on women and how sex crimes affect the mood disorder risk. Eventually, we decided to broaden our approach and include all patients and consequently also all crime types. As a result, the effort needed to analyze and prepare the crime data grew significantly, as a considerably larger number of ZIP Codes had to be looked up for the location of the crimes.

In the previous approach, the average crime rate in a ZIP Code was calculated over several years. However, selecting all types of crime for several years would result in too much data to process. Selecting all crime types for one year results in 726,701 records, whereas the sex crimes for several years contained 53,590 records.

It is possible to use the described system to identify the ZIP Codes for the crime reports. However, the system was not constructed for this amount of data. If all crime reports are processed by the Census Geocoding API, it would take more than one day. Although this high amount of requests is not prohibited by the Census, a faster and more stable approach was desired.

Therefore, we set up a PostgreSQL database with the PostGIS extension that adds support for geographic objects to the database. This allowed us to perform location queries via SQL.

\[\text{http://www.postgis.net/}\]
Chapter 3. Implementation

The U.S Census Bureau provides so called TIGER (Topologically Integrated Geographic Encoding and Referencing) products that are spatial extracts from the Census’s MAF/TIGER database. One of those products are the shapefiles, a geospatial data format for geographic information system environments that can easily be integrated into the PostgreSQL database. Only the shapefiles of the states of Illinois, Indiana, Pennsylvania and California are needed, as these states contain the selected cities of crime.

With this system, it was possible to identify 100 to 120 addresses per second on average, which is 10 times faster than using the single request Geocoding API of the U.S. Census Bureau or Google. This approach was used for more than 700,000 anonymized addresses of crime scenes.

<table>
<thead>
<tr>
<th>City</th>
<th>Reported</th>
<th>Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicago</td>
<td>262,670</td>
<td>262,667</td>
</tr>
<tr>
<td>Philadelphia</td>
<td>183,301</td>
<td>182,913</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>228,017</td>
<td>227,253</td>
</tr>
<tr>
<td>Indianapolis</td>
<td>52,713</td>
<td>52,649</td>
</tr>
</tbody>
</table>

**Table 3.14: Overview of identified ZIP Codes for crime locations**

The final results for all processed datasets contain 262,667 crime reports for the City of Chicago, 182,913 crime reports for the City of Philadelphia, 227,253 crime reports for the City of Los Angeles and 52,713 crime reports for the City of Indianapolis. 1,219 crime reports had to be dropped, as no ZIP Code could be identified.

For each ZIP Code of the respective city, the number of crimes per ZIP Code were calculated by grouping the crime reports by ZIP Code and counting the number of occurring crimes. Afterwards, the crime datasets were joined with the population dataset described in Section 3.1.2, as the demographic data contains the population count per ZIP Code. To calculate a normalized crime rate $r$, the number of crimes $n$ is divided by the population count $N$ and multiplied by 10,000, i.e.

$$ r = 10,000 \times \frac{n}{N}. $$

**Equation (3.2)**

The final result describes the number of crimes per 10,000 people in the corresponding ZIP Code.

Before it is possible to train a model on this data, the crime rate is standardized for each dataset individually, resulting in a crime indicator for each ZIP Code that is centered around 0. By doing this, the individual crime rates of the ZIP Codes of the

---

8 [https://www.census.gov/geo/maps-data/data/tiger.html](https://www.census.gov/geo/maps-data/data/tiger.html)
9 [https://www.census.gov/geo/maps-data/data/tiger-line.html](https://www.census.gov/geo/maps-data/data/tiger-line.html)
different cities become comparable. This is necessary, because the ZIP Code with
the least crimes in Chicago possibly sees more crimes in a year than most of the
ZIP Codes in Austin. Using the crime indicator instead of the crime rate solves this
problem, as the crime indicator is relative to the average crimes within the city.

After all crime report datasets were processed, missing ZIP Codes identified, and
the crime rate standardized into the crime indicator, they are finally joined with the
census community data and merged into one dataset.

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP Codes in Crime Data</th>
<th>ZIP Codes in Joined Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Chicago</td>
<td>65</td>
<td>64</td>
</tr>
<tr>
<td>Indianapolis</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>125</td>
<td>125</td>
</tr>
<tr>
<td>Philadelphia</td>
<td>47</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 3.15: Overview of number of ZIP Codes in crime data and num-
ber of ZIP Codes after joined with census community features

Due to the fact that the census community data does not contain all ZIP Codes that
are part of the crime dataset, one ZIP Code of each the City of Chicago and the City
of Philadelphia had to be dropped, as no census community data was available for
them. This resulted in crime indicators for 49 ZIP Codes in the City of Austin, 64
ZIP Codes in the City of Chicago, 34 ZIP Codes in the City of Indianapolis, 125 ZIP
Codes in the City of Los Angeles and 46 ZIP Codes in the City of Philadelphia or
318 ZIP Codes in total. The detailed number of ZIP Codes in the crime data and
the number of ZIP Codes in the joined dataset can be seen in Table 3.15.

Afterwards, it is possible train the model to estimate the crime indicator for every
ZIP Code that is part of the clinical dataset and also part of the census community
dataset. For the implementation of the estimator, several regression models were
tested and evaluated: a linear regression model, a ridge regression model, a gradi-
ent boosted trees model and an AdaBoost model. The different regression models
were separately trained, evaluated and the best performing model was chosen for
the estimation of the crime indicator.

Before the models are trained, the dataset is randomly sampled into 80% training
data and 20% test data. The resulting training set contains 254 records and the test
set 64 records.

The models are trained on the training data and five-fold cross validation is used
on the training data to tune the models’ hyperparameters.

The optimized models are then trained on the whole training set and evaluated
against the test set. For a detailed explanation of the evaluation methodology see Section D.2.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE (Test)</th>
<th>R² (Test)</th>
<th>MSE (5-fold CV)</th>
<th>R² (5-fold CV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression</td>
<td>0.4267</td>
<td>0.6281</td>
<td>0.4561 (+/- 0.1474)</td>
<td>0.4884 (+/- 0.2970)</td>
</tr>
<tr>
<td>Ridge Regression</td>
<td>0.4550</td>
<td>0.6035</td>
<td>0.5375 (+/- 0.2592)</td>
<td>0.4098 (+/- 0.3042)</td>
</tr>
<tr>
<td>Gradient Boosted Trees</td>
<td>0.3193</td>
<td>0.7217</td>
<td>0.4301 (+/- 0.2462)</td>
<td>0.5150 (+/- 0.3690)</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>0.4270</td>
<td>0.6278</td>
<td>0.4561 (+/- 0.1474)</td>
<td>0.4884 (+/- 0.2970)</td>
</tr>
</tbody>
</table>

Table 3.16: Evaluation results of different regression models for the crime indicator model

The evaluation results, that are illustrated in Table 3.16 show that the gradient boosted trees model performs considerably better than the other three tested models. With an R² of 0.7217, the gradient boosted trees model performs quite well on the test data and should be perform well enough to estimate the crime indicator for the ZIP Codes in the clinical dataset. However, the output of the model still introduces a good amount of uncertainty to our overall prediction task. This could lead to problems, when the crime indicator is used as a feature for the mood disorder risk model. The crime indicator feature is later evaluated in Section 3.4.4.

After the evaluation of the crime indicator estimation model, the gradient boosted trees model is used to estimate the crime indicator for all ZIP Codes found in the clinical dataset. The resulting data is later used as the crime indicator feature in the blended dataset.

The node.js script for the Zip Code identification and the Pig scripts for the data preparation for this section can be seen in Section E.2 of the appendix. Additionally, the Python scripts can be found in Section F.2 of the appendix.

3.2.4 Partitioning of Clinical Data

In Section 2.7.1 of the previous chapter, the steps of how to train and evaluate a strong prediction model are described in detail. In this section the focus is on the first step, the partition of the dataset into training set and test set. This step is important to evaluate the model against out-of-sample data at the end, since using data that is already part of the training set would bias the evaluation results.

The clinical data provided by Geisinger Health System is one of the main data sources that is used in the development of the imputation model of mood disorder prevalence, described in Section 3.2.6 and for the mood disorder risk model, described in Section 3.3. However, it is important that the models that include the
patient demographics dataset use a fixed number of patients for training purposes and the other patients only for evaluation purposes.

Therefore, the clinical data has to be partitioned into two groups of patients, one group used as training data and one group used as test data, with 80% of the patients assigned to the former and 20% assigned to the latter. For the patient demographics dataset, this results in 213,142 patients for the training set and 53,284 patients for the test set. As the patient demographics dataset and the problem list dataset of the clinical data provided by Geisinger Health Systems are linked by the unique patient ID, the split also divides the problem list dataset into 1,894,535 records for the training set and 475,702 records for the test set.

If an additional dataset is now joined with one of the clinical datasets, it is not joined with the original dataset, but rather with the training set of the respective dataset. This ensures that it is possible to train a prediction model on this data and evaluate the model against the corresponding test set. An additional split of the joined data is not needed. Through this approach, all models dealing with the clinical data used in this thesis are always evaluated on unseen patient data.

In the following sections, the clinical data is always used separately for training and testing. Even if it is not explicitly stated, transformations are never implemented with the whole clinical dataset for the reasons stated above.

### 3.2.5 Patient Demographics

The patient demographics dataset, described in Section 3.1.1, contains various personal information on each individual, but not all fields are in a useful format to train a prediction model, as several attributes such as the gender of an individual, the employment status and marital status are categorical attributes that cannot be directly used as a feature. For this reason, we need to extract the features needed from the data.

The fields of interest from the patients demographic dataset are the patient’s date of birth (CLIN_DOB), gender (CLIN_GEN), race (CLIN_RAC), marital status (CLIN_MAR_STAT), employment status (CLIN_EMP_STATUS), current insurance (CLIN_INS) and the control patient status (CLIN_CP). All the fields are shown in Table 3.1.1 in Section 3.1.

The date of birth is transformed into the more meaningful age of the patient which can be used as a feature. The age is calculated by subtracting the year of birth of the individual from the current year. The gender attribute only takes two values in the given dataset, which are male and female. This categorical field is transformed into a binary feature, where 0 represents male and 1 represents female. For 17 records, the gender is specified as unknown or missing all together. These missing values
<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD_STUDY_ID</td>
<td>A unique patient identifier</td>
</tr>
<tr>
<td>PD_ZIP</td>
<td>ZIP Code of residence of the patient</td>
</tr>
<tr>
<td>PD_AGE</td>
<td>Age of patient</td>
</tr>
<tr>
<td>PD_GENDER</td>
<td>Gender of patient, where 0 represents male patients and 1 represents female patients</td>
</tr>
<tr>
<td>PD_WHITE</td>
<td>Binary flag that specifies whether patient is white</td>
</tr>
<tr>
<td>PD_BLACK</td>
<td>Binary flag that specifies whether patient is black</td>
</tr>
<tr>
<td>PD_AsIAN</td>
<td>Binary flag that specifies whether patient is asian</td>
</tr>
<tr>
<td>PD_AI</td>
<td>Binary flag that specifies whether patient is of American Indian ethnicity</td>
</tr>
<tr>
<td>PD_PI</td>
<td>Binary flag that specifies whether patient is of Pacific Islander ethnicity</td>
</tr>
<tr>
<td>PD_MARRIED</td>
<td>Married status</td>
</tr>
<tr>
<td>PD_DIVORCED</td>
<td>Divorced status</td>
</tr>
<tr>
<td>PD_WIDOWED</td>
<td>Widowed status</td>
</tr>
<tr>
<td>PD_EMP_FULL</td>
<td>Full time employment status</td>
</tr>
<tr>
<td>PD_EMP_PART</td>
<td>Part time employment status</td>
</tr>
<tr>
<td>PD_EMP_STU</td>
<td>Employed student status</td>
</tr>
<tr>
<td>PD_EMP_SELF</td>
<td>Self-employed status</td>
</tr>
<tr>
<td>PD_EMP_RET</td>
<td>Retired status</td>
</tr>
<tr>
<td>PD_INS</td>
<td>Insured status</td>
</tr>
<tr>
<td>PD_MEDICARE</td>
<td>Medicare status</td>
</tr>
<tr>
<td>PD_MD</td>
<td>Mood disorder status</td>
</tr>
</tbody>
</table>

Table 3.17: Schema of the patient demographic dataset

are replaced by 0.5, as it is not possible to determine from the other data if the patient is male or female.

As stated before, the ethnicity field is also categorical and needs to be transformed. We chose to use one-hot encoding for the applicable values of the field. This results in five binary features, which are PD_WHITE, PD_BLACK, PD_AsIAN, PD_AI and PD_PI. Only one of the features can be 1 or all of them 0, meaning the patient is from another ethnicity.

The marital status has to be handled similarly. The categorical values are transformed into the features PD_MARRIED, PD_DIVORCED and PD_WIDOWED. For the transformation, we grouped the values of divorced and separated together in the PD_DIVORCED feature. If a patient is single, the marital status features are set to all zero.

As stated before, the extracted employment status is represented by the features PD_EMP_FULL, PD_EMP_PART, PD_EMP_STU, PD_EMP_SELF and PD_EMP_RET. The employment
3.2. Data Preparation

status contains values for both full time and part time students, but we decided to group both into one feature, as they don’t often occur.

Another important attribute about the individual’s personal life is the current insurance, since the dataset contains patients with a regular insurance, no insurance, and patients that make use of medicare. We extract two features from this attribute, which are \texttt{PD_INS} and \texttt{PD_MEDICARE}. To decide between regular insurances, no insurance and medicare, a naive approach was chosen that treats all insurances containing medicare in their name as a medicare supplemented plan, which leads to the feature \texttt{PD_MEDICARE} being 1. If the patient is not insured, both features are set to 0, otherwise the \texttt{PD_INS} is set to 1.

Finally, the dataset contains a field called control patient status. This status describes, if an individual has actually suffered from a mood disorder or is part of the control group. The status takes the value 0 for a diagnosed mood disorder patient and 1 if no mood disorder was diagnosed in the past. The mood disorder status \texttt{PD_MD} describes the negation of the control patient status. Consequently, the value 0 represents that the patient has not received a mood disorder diagnosis and the value 1 represents that the patient was diagnosed with a mood disorder.

As all of the aforementioned features are bound between the range of 0 and 1 except the age, only the feature \texttt{PD_AGE} is standardized. Additionally, the original dataset contains missing values and for several patients only some of the attributes are available. After the transformation, these values are replaced by the corresponding average of the feature.

The resulting set of features can be seen in Table 3.17 and the corresponding Pig Latin script used to clean and extract the features can be found in Listing E.12 in Section E.5 of the appendix.

3.2.6 Calculation of Mood Disorder Prevalence

The most interesting information that we extracted from the patient demographics dataset is the distribution of the mood disorder prevalence per ZIP Code. As stated in the overall strategy in Chapter 2, the mood disorder prevalence is an important indicator of the influence of the personal environment on the mood disorder risk.

A ZIP Code with a remarkably high mood disorder prevalence indicates the presence of endemic factors to blame for the heightened prevalence. Conversely, an extraordinarily low mood disorder prevalence in a given ZIP Code has to be at least partially explained by some characteristics of the ZIP Code. Data on the distribution of the mood disorder prevalence can be used to predict the mood disorder risk for an individual living in a given ZIP Code.
However, we were not able to find any comprehensive data on the distribution of mood disorder prevalence with a higher granularity than on the state level. As nearly all patients we received data on were contained in the state of Pennsylvania, data on the state level gives no further insights. For this reason, the patient demographics dataset provided by Geisinger Health Systems was used to calculate the mood disorder prevalence for every ZIP Code found in the dataset.

As the patient demographics dataset contains the patient’s ZIP Code of residence and information on whether the patient suffers from a mood disorder, the mood disorder prevalence for a given ZIP Code can be calculated by dividing the number of affected patients living in the ZIP Code by the total number of patients living in the ZIP Code. This can be expressed as

\[
P_{\text{ZIP}} = \frac{n_{\text{ZIP}}}{N_{\text{ZIP}}},
\]

where \(P_{\text{ZIP}}\) is the mood disorder prevalence for a given ZIP Code, \(n_{\text{ZIP}}\) denotes the number of affected patients in the ZIP Code, and \(N_{\text{ZIP}}\) is the total number of patients in the given ZIP Code.

It is important to note that the mood disorder prevalence calculated here only relates to the number of affected and unaffected patients. The resulting percentage, of course, does not necessarily equal the real mood disorder prevalence for a given ZIP Code, as all individuals used in the calculation are drawn from the population of patients that have been treated by Geisinger Health Systems, rather than the whole population living in the ZIP Code. In fact, the calculated percentages should be uniformly higher than the actual mood disorder prevalence, as healthy individuals that have not been treated are not factored in. However, the calculated mood disorder prevalence serves as a realistic estimate, albeit scaled by some unknown factor, as both affected and unaffected patients are sampled randomly from the whole patient population.

One problem with this approach, however, is the statistical significance of the calculated values. If only a few patients from a given ZIP Code are found in the data, the resulting value for the mood disorder prevalence is more or less meaningless, as it is not statistically significant and thus could be the result of mere coincidence. Unfortunately, this was the case for a majority of the ZIP Codes found in the patient demographics dataset.

To get an idea of how meaningful the calculated mood disorder prevalence would be, we initially looked at the patient coverage found in the provided data. The patient coverage is the percentage of patients living in a certain ZIP Code found in the dataset in relation to the actual population of the ZIP Code, i.e. 100 patients for a ZIP Code with a population of 10,000 corresponds to a patient coverage of 0.01 or
3.2. Data Preparation

![Graph showing distribution of patient coverage over ZIP Codes.](image)

(a) The patient coverage plotted using a linear scale for the patient coverage axis.

(b) The patient coverage plotted using a logarithmic scale for the patient coverage axis.

**Figure 3.2:** Distribution of the patient coverage over all 3,393 unique ZIP Codes found in the patient demographics dataset, plotted in both linear (a) and logarithmic scale (b).

1%. Figure 3.2 shows the distribution of the patient coverages per ZIP Code for all ZIP Codes found in the patient demographics dataset.

In the data at hand, the patient coverage follows a rough power-law distribution, as can be recognized by the almost linear appearance of the distribution in logarithmic scale. This means that for a majority of the included ZIP Codes, only a few samples exist in the dataset, whereas the majority of samples in the dataset belong to a small amount of ZIP Codes.

This can be explained naturally by the way the samples where acquired in the first place: Patients will generally visit a doctor’s office or health care facility in their approximate vicinity rather than a facility across the state or even out-of-state. This leads to a sampling bias in regards to the area the treated patients live in, as areas with Geisinger facilities will be much more prominently featured in the collected data. One exception to this are individuals that are currently away from home and are in immediate need of medical attention, as these types of patients will generally use the nearest health care facility to their current location. Accordingly, only 1,184
of the 3,393 ZIP Codes occurring in the patient demographics dataset are actually located in the state of Pennsylvania. In other words, about two thirds of the ZIP Codes found in the dataset are from out-of-state patients. The calculated mood disorder prevalence values for these out-of-state ZIP Codes are mostly useless for our purpose, as they are heavily under-sampled, i.e. only occurring once or twice in the dataset, and are therefore statistically insignificant.

The interesting ZIP Codes for our purpose are the 1,184 ZIP Codes found in the data that are located in the state of Pennsylvania. Figure 3.3 shows the distribution of patient coverages for all ZIP Codes in the state of Pennsylvania, including those not found in the data, in the form of a heat map. As the heat map shows, the patient coverage is especially high in ZIP Codes that are located in an area that stretches from the center of the state to the north-east. Accordingly, almost all of the western part of the state and a majority of the ZIP Codes in the south-east are more or less unrepresented in the data. This also shows that the patient coverage is mostly unrelated to the population and population density of the ZIP Codes as highly dense and metropolitan ZIP Codes, such as the ones found in Philadelphia, which is located in the south-east of the state, are heavily underrepresented.

This distribution is easily explained by looking at Figure 3.4, which shows all locations of Geisinger hospitals and clinics, where patients in the dataset were treated. The Geisinger locations align perfectly with the area that features the relatively high patient coverage. The ZIP Code with the highest patient coverage is ZIP Code 16853, which is a very small ZIP Code in terms of size and population located in the midst of the area covered by Geisinger facilities with a patient coverage of about...
93 %, i.e. 280 of the 300 residents of the ZIP Code are present in the patient demographics dataset. The next highest patient coverage lies at about 47%.

The analysis of the patient coverage distribution already shows that the mood disorder prevalence for a majority of ZIP Codes found in the patient demographics dataset cannot be calculated with a high amount of certainty. That being said, patient coverage does not necessarily align with statistical significance of the estimated values. A much more suited measure for the significance of the calculated values is the actual confidence in the calculated values expressed as a confidence interval or the margin of error.

In regards to estimated values, the confidence interval is a range of values, for which it is possible to say with a certain confidence that the actual value lies within it. A confidence interval always relates to a certain confidence level, which is expressed as the probability of the actual value falling into the confidence interval. Usual values for the confidence level are 90%, 95%, and 99%. The margin of error is defined as the half of the width of the confidence interval, which specifies the maximum deviation of the estimated value in regards to the actual value according to the confidence interval.

To determine which of the calculated mood disorder prevalence values are statistically significant, we chose to calculate confidence intervals at the 95% confidence level. The confidence intervals were calculated using the adjusted Wald method proposed by [AC98], which has been shown to be a good choice for binomial proportions, i.e. the proportion of people suffering from mood disorder [SL05].

The adjusted Wald method uses the Wilson point estimator rather than the maximum likelihood estimator we used before to calculate the center point of the confidence interval [Wil27]. The Wilson point estimator has been found to provide the most accurate estimation for proportions that lie beneath 50% [LS06]. It is calculated by
where \( \hat{p} \) is the point estimation, \( x \) is the number of positive outcomes and \( n \) is the number of samples, while \( z \) is the Z-score, which depends on the level of confidence wanted, in our case for the 95% confidence level \( z = 1.96 \).

Additionally, a finite population correction (FPC) factor was used to improve the quality of the confidence intervals, as the samples where gathered from ZIP Codes that partially exhibit very small population sizes [Iss18]. The FPC depends on both the population and the sample size and is calculated as

\[
FPC = \sqrt{\frac{N - n}{N - 1}}. 
\]

Resulting from this, the confidence intervals for the calculated mood disorder prevalence values can be calculate as

\[
\hat{p} \pm z \times \sqrt{\frac{\hat{p}(1 - \hat{p})}{n} \times \frac{N - n}{N - 1}},
\]

where \( \hat{p} \) is the resulting Wilson point estimation, which constitutes the midpoint of the confidence interval. The part after the plus-minus sign makes up the margin of error, corrected by the FPC.

The confidence intervals for the mood disorder prevalence were calculated for all ZIP Codes that occurred in the dataset. Figure 3.5 shows an excerpt of the distribution of the margin of error for these ZIP Codes. As can be seen, a statistical significant value for the mood disorder prevalence can only be determined for a
small fraction of the ZIP Codes that occur in the dataset. Only about 200 ZIP Codes show a margin of error below the 5% threshold, while even less ZIP Codes, 91 to be exact, fall under the 3% threshold.

Figure 3.6 shows the mood disorder prevalence and the corresponding confidence intervals at the 95% confidence level for all ZIP Codes that exhibit a margin of error below the 3% threshold.

With confidence intervals of this size, some of the calculated values cannot be used to accurately describe the mood disorder prevalence in the corresponding ZIP Code, much less be used to predict the mood disorder risk for an individual living in this ZIP Code. However, for most ZIP Codes there is at least some weak statistical evidence in the calculated values. Hence, it seems feasible to impute the mood disorder prevalence for only those ZIP Codes that are so heavily underrepresented in the data.

The Pig scripts for the data preparation for this section can be seen in Section E.3 of the appendix. Additionally, the Python scripts can be found in Section F.1 of the appendix.

### 3.2.7 Imputation of Mood Disorder Prevalence

The patient demographics dataset contains more than 3,000 distinct ZIP Codes and about 47% of those ZIP Codes belong to the state of Pennsylvania. On the other hand, about 73% of the ZIP Codes in the state of Pennsylvania are covered in the dataset. As described in the previous section, a statistically significant mood disorder prevalence value can only be calculated for part of the ZIP Codes occurring
in the clinical data, hence, some of mood disorder prevalence values have to be imputed.

We chose to use two separate imputation methods in parallel and evaluate which of the two methods performs better when used to predict the mood disorder risk. The first method is a simple imputation by average. The second is a hybrid method using both imputation by average and regression.

In the simple imputation by average, the mood disorder prevalence of ZIP Codes that exhibit a margin of error of over 15% are disregarded and replaced by the average, the computed mood disorder prevalence values for ZIP Codes with a margin of error below 15% are used as they are.

The second imputation method is a hybrid solution, combining both imputation by average and regression model. A regression model is trained on the most significant ZIP Codes using part of the census community features as predictor variables. It is important to note that this could introduces some feature dependencies, when the census community features and the mood disorder prevalence feature are used together in the blended dataset. However, as only a minor fraction of the ZIP Codes are actually imputed by the regression model and only part of the census community features are used in this approach, the possible feature dependency should not pose a problem to the mood disorder risk model. In fact, this pre-analysis of the data could take over some of the work of the mood disorder risk model, as certain correlations could already be included in the mood disorder prevalence.

As a first step, the optimal threshold for the margin of error, described in Section 3.2.6, has to be determined to select enough statistically significant data to train a strong model. However, if the chosen threshold is too high, additional uncertainty is introduced to the model, which will lead to worse prediction results and is thus not desired. The goal is to select as many samples with as little of overlap in the corresponding confidence intervals as possible, as this is essential to train an accurate estimator. It should be clear that training a model on data of high quality results in better model accuracy than training on data of lower quality.

As a compromise between volume and significance of training data, a threshold of 3% was chosen, which results in 91 statistically significant target values used to train the model. Additional features that can be used as predictor variables are needed to train the regression model. We chose to use the census community features as the independent variables for the regression model. For this reason, the selected records from the unimputed mood disorder prevalence data with a margin of error of 3% and below are joined with the census community dataset, described in Section 3.2.2, on the ZIP Code field. The resulting dataset contains the 21 census community features and the mood disorder prevalence of the corresponding
ZIP Code. As 91 samples are generally not enough to train a robust model on 21 features, a subset of the most important features have to be selected.

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>F-Measure</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC_POP_TOTAL</td>
<td>0.0053</td>
<td>0.9420</td>
</tr>
<tr>
<td>CC_POP_HISP</td>
<td>3.1794</td>
<td>0.0780</td>
</tr>
<tr>
<td>CC_POP_WHITE</td>
<td>1.6426</td>
<td>0.2033</td>
</tr>
<tr>
<td>CC_POP_BLACK</td>
<td>0.1407</td>
<td>0.7085</td>
</tr>
<tr>
<td>CC_POP_AI</td>
<td>0.5905</td>
<td>0.4442</td>
</tr>
<tr>
<td>CC_POP_ASIAN</td>
<td>3.0436</td>
<td>0.0845</td>
</tr>
<tr>
<td>CC_POP_PI</td>
<td>0.0575</td>
<td>0.8110</td>
</tr>
<tr>
<td>CC_POP_OTHER</td>
<td>0.9207</td>
<td>0.3399</td>
</tr>
<tr>
<td>CC_POP_MIXED</td>
<td>0.7835</td>
<td>0.3785</td>
</tr>
<tr>
<td>CC_ECON_UNEMP</td>
<td>0.1778</td>
<td>0.6743</td>
</tr>
<tr>
<td>CC_ECON_OCC_MGT</td>
<td>4.8723</td>
<td>0.0299</td>
</tr>
<tr>
<td>CC_ECON_OCC_SERV</td>
<td>3.8634</td>
<td>0.0525</td>
</tr>
<tr>
<td>CC_ECON_OCC_SALES</td>
<td>1.0202</td>
<td>0.3150</td>
</tr>
<tr>
<td>CC_ECON_OCC_CONST</td>
<td>1.4739</td>
<td>0.2279</td>
</tr>
<tr>
<td>CC_ECON_OCC_TRANS</td>
<td>4.8032</td>
<td>0.0310</td>
</tr>
<tr>
<td>CC_ECON_INSURED</td>
<td>0.0293</td>
<td>0.8646</td>
</tr>
<tr>
<td>CC_ECON_BELOW_POV</td>
<td>0.0023</td>
<td>0.9617</td>
</tr>
<tr>
<td>CC_ECON_DENS</td>
<td>3.3309</td>
<td>0.0713</td>
</tr>
<tr>
<td>CC_ECON_INC_MED</td>
<td>6.1660</td>
<td>0.0149</td>
</tr>
<tr>
<td>CC_ECON_INC_AVG</td>
<td>9.8199</td>
<td>0.0023</td>
</tr>
<tr>
<td>CC_ECON_CTW</td>
<td>0.0102</td>
<td>0.9198</td>
</tr>
</tbody>
</table>

Table 3.18: Results of the univariate feature selection of census community features for the prediction of the mood disorder prevalence

A generally accepted rule of thumb for selecting the right number of independent variables for any kind of prediction model suggests that ten outcome events per predictor variable are needed to achieve an accurate estimation \[\text{AS14, VM07}\]. Based on this rule, only a maximum of nine of the census community features should be selected for 91 significant samples to train the regression model.

The SelectKBest class from the sklearn.model_selection package is used to select the 9 best census community features. The goal is to choose the features with the highest F-score in regards to the target value, as the higher the F-score, the larger the impact on the estimation outcome should be. Therefore, univariate feature selection is used to estimate the linear dependency between the feature and the target value using an F-test. For a single feature \(x_i\), the F-score is calculated as
The results of each calculated F-score can be seen in the Table 3.18. The features with the highest F-scores, namely `cc_pop_hisp`, `cc_pop_white`, `cc_econ_occ_mgt`, `cc_econ_occ_trans`, `cc_econ_dens`, `cc_econ_inc_med`, `cc_econ_inc_avg` and `cc_econ_ctw`, are selected and used for training a regression model on these features with the mood disorder prevalence as target. The scatter plots of the selected features can be seen in Figure 3.7.

After the feature selection, the 91 significant ZIP Codes are randomly sampled into 80% of training data and 20% of test data, in order to be able to later evaluate the prediction results.

We are not able to use a regular logistic regression model, as the mood disorder prevalence is not a binary categorical feature, but rather a proportion between 0 and 1. To deal with this problem, a binomial generalized linear model is chosen that uses the logit transformation as a link function. Each calculated mood disorder prevalence \( p \) is transformed by the logit function

\[
\text{logit}(p) = \log\left(\frac{p}{1-p}\right).
\]

(3.8)

Subsequently, a regression model is trained on the training data and evaluated against the test samples to estimate the generalization error and finally, the model is trained on all the samples. The evaluation results are shown in Table 3.19. As can be seen from the \( R^2 \) score, the performance of the model leaves much to be desired.
3.2. Data Preparation

<table>
<thead>
<tr>
<th>Model</th>
<th>R²</th>
<th>Bier Score</th>
<th>F-Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logit OLS</td>
<td>0.247</td>
<td>0.0868</td>
<td>2.263</td>
</tr>
</tbody>
</table>

TABLE 3.19: Evaluation results of the mood disorder prevalence imputation model

Based on these results, it is questionable, if the model will give any improvement over the imputation by average. However, both imputation methods are evaluated separately in Section 3.4.3.

Once the model is evaluated, the model is used to predict the mood disorder prevalence values for all ZIP Codes occurring in the clinical data. Additionally, for each prediction a prediction interval is calculated using the `wls_prediction_std` method from the `statsmodels.sandbox.regression.predstd` package, which specifies the certainty of the model in the prediction. The imputation is now conducted as follows: computed mood disorder prevalence values are replaced by the results of the regression model, if the size of the calculated confidence interval of the computed value is larger than the prediction interval of the predicted value. However, prevalence values with confidence and prediction intervals that span more than 30% respectively are replaced by the average over the 91 significant mood disorder prevalence values, analogously to the first method.

The results of the two imputation methods are fundamentally similar and the two resulting features can be used interchangeably. Both approaches are evaluated in Section 3.4.3. We are aware that the second strategy is not conventional and might turn out to be of no benefit to the mood disorder risk model, however, analyzing the correlation between the census community features and the mood disorder prevalence has already provided some further insights into the development and prevalence of mood disorders.

We found that both the median and average household income and the fraction of individuals working in management positions are negatively correlated with the mood disorder prevalence in a given ZIP Code. This is interesting, as it suggests that the prosperity of the neighborhood has a significant influence on the development of mood disorders in this neighborhood. However, we were not able to show any significant correlation between the mood disorder prevalence and the fraction of individuals holding health insurance, the employment rate and the poverty rate of the ZIP Code. This is surprising, as these features should be positively correlated with the mood disorder prevalence according to the hypothesis above. This would suggest that the prosperity of a neighborhood does not affect the mood disorder prevalence. Which of the two or whether both are true to a certain extent should be investigated further as part of future research.
The Pig scripts for the data preparation for this section can be seen in Section F.1 of the appendix. Additionally, the Python scripts can be found in Section F.1 of the appendix.

### 3.2.8 Patient History Matrix

The second dataset that was provided as part of the clinical data we received from Geisinger Health Systems contains information on the medical history of the individuals described in the patient demographics dataset. As described in Section 2.1, this patient history data is essential for the prediction of the mood disorder risk, as it provides a profound look into the individual’s life. This includes evidence of certain behaviors, such as smoking and drug abuse, and indications of the individual’s personal environment, such as injuries that are often seen with certain professions and health problems that are related to certain lifestyles, i.e. obesity, diabetes and liver problems.

The patient history data also introduces a whole new dimension to the analysis that is missing from all other datasets, which is temporality. In contrast to the other datasets that only apply to a certain point in time, the patient history dataset includes chronological data that spans a whole period of time. This allows us to gain insight into the individual’s past, rather than only taking their current situation into consideration. For instance, the mood disorder prevalence and crime rate data can only be considered for the individual’s current place of residence, as no historical data of prior places of residence is included in the clinical dataset. As a result, an individual that has recently moved to a new place could be falsely flagged as a low-risk individual, although their personal environment before the recent change would have indicated a higher risk. This information is lost due to the nontemporal nature of the data. The patient history data can be used to compensate for these shortcomings.

As described in Section 3.1.1, the patient history data consists of detailed information on the medical history of more than 230,000 patients, with each record indicating a diagnosis given to a patient, including the exact date of the diagnosis and the type of diagnosis coded as both ICD-9 and ICD-10 codes. Consequently, it is possible to construct a detailed diagnosis history for each patient occurring in the dataset that is made up of a chronological succession of diagnoses.

In order to train a prediction model on this diagnosis history, it first must be transformed into features of some sort. Our approach of extracting features from the diagnosis history takes inspiration from the fields of natural language processing and information retrieval, especially the vector space model.
As proposed by [SWY75], the vector space model is an algebraic model that can be used to represent entities as vectors of identifiers. Each of these entity vectors describes a point in a common vector space, allowing for a comparison of entities by means of distance. In the original application, the entities of the vector space model are text documents and the identifiers correspond to the terms used in the document.

In its original application, the vector space is created by constructing a term-document matrix, where each column represents a document and each row represents a term occurring in one of the documents, with the values being term weights for the term in the respective document that can be calculated using numerous methods. Terms not occurring in the respective document receive a term weight of zero. It is important to note, that in the construction of the vector space, each document is treated as a bag-of-words. This means that each document is represented by a multiset, meaning a set that allows multiple instances of the same element, of terms occurring in the document. In this representation, the order, in which the terms occurred, is disregarded, as only the frequency of occurrences is needed to calculate the term weights.

The resulting document vectors can be compared using several distance measures for the purpose of document classification and similarity calculation. One of the most used distance measures used in this context is the cosine similarity, which is defined as the cosine of the angle $\theta$ between two vectors, that is

$$
\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|},
$$

where $A$ and $B$ are two vectors from the vector space. The cosine is very suited as a similarity measure, as it ranges from 0.0 for orthogonal vectors to 1.0 for vectors identical in direction [Sin01]. By using the cosine similarity, only the direction of the vectors is taken into account when calculating the similarity between two vectors, rather than their euclidean length. This is favorable, as the direction of a vector in the vector space model relates to the inherent meaning of the corresponding document, regardless of the document’s size [TP10].

The vector space model, however, is not only limited to documents and terms. As described above, the generic vector space model only deals in terms of entities and identifiers and thus can be used to solve a broad spectrum of problems. In our case, we chose to create a vector space from the patient history data, where the entities correspond to the patients and the identifiers represent the diagnoses the patients have received. Analogously, a diagnosis-patient matrix can be constructed from the patient history data, where every column represents a patient and every row represents a diagnosis that occurred in the patient history data.
A challenge in this approach is the question of how to encode the received diagnoses. Luckily, the patient history data already includes the diagnoses in an encoded form, namely as both ICD-9 and ICD-10 codes. As it stands, ICD-10 is the successor of ICD-9 and meant to be an extensive replacement, however, we chose to use the ICD-9 codes of the diagnoses, as 143,065 records, more than 6% of the data, did not contain the corresponding ICD-10 code, whereas only 3,548 records, less than 0.2% of the data, are missing the ICD-9 code. Additionally, the diagnoses are more spread out in ICD-10 opposed to ICD-9, as there are almost five times as many codes in ICD-10 as there are in ICD-9 that can be used to encode a given diagnosis. This results in a significantly more sparse diagnosis-patient matrix that has many more dimensions, which both takes more space to store and is more computationally expensive when being analysed.

As with the traditional vector space model approach, we also represent the patient history as an equivalent to a bag-of-words, with diagnosis codes instead of terms. We chose to use a binary diagnosis weight, where a weight of 1 is given to diagnoses that occur in the individual’s patient history and a weight of 0 is given to all other diagnoses that did not occur in the individual’s patient history.

This approach does admittedly disregard the chronological order of the diagnoses a patient has received in favor of a more simple way of using and analysing the data, however, the unordered set of occurred diagnoses should include ample information for the final prediction model to be of significant improvement to the prediction quality.

The diagnosis-patient matrix is constructed using the problem list data described in Section 3.1.1. The problem list data contains records that are not normalized, i.e. some records contain multiple ICD-9 codes, however, to construct the diagnosis-patient matrix, the diagnoses must exist in a normalized form. For this reason, the problem list dataset is first transformed into a normalized form by splitting records with multiple ICD-9 codes into multiple records with a single ICD-9 code using Pig. The Pig Latin script used to normalize the problem list data can be found in Listing E.13 in Section E.5 of the appendix.

To be able to use the patient history data to train a prediction model that can later be evaluated correctly, the problem list dataset has to be split into training and test set according to the technique described in Section 3.2.4. It is important to use the same split for the clinical data throughout the whole data preparation process, as the resulting prediction model that is trained on the final dataset must be evaluated correctly on out-of-sample data. If different splits on the data were to be used, the final evaluation would be biased towards better results, as the prediction model has seen part of the test data before. Consequently, we needed to construct two separate diagnosis-patient matrices, one from the training and one from the test set. The same split on the patients as in Section 3.2.4 used for this section.
3.2. Data Preparation

The first step in the construction of the diagnosis-patient matrices is to calculate the set of occurring diagnoses

\[ D = \{d_1, d_2, \ldots, d_j\} \]

that is used as the diagnosis vocabulary, where \( d_j \) is the \( j \)th unique diagnosis. The diagnosis vocabulary is calculated by scanning the training set of the problem list dataset consecutively and storing each uniquely encountered diagnosis code in the vocabulary. Constructing the diagnosis vocabulary first is important, as the entries in the vocabulary are used as the dimensions of the patient vectors. Diagnoses that only occur in the test set are of no interest, as they cannot be used to train the prediction model.

As mentioned before, each patient \( P_i \) is represented by a \( j \)-dimensional vector

\[ P_i = (p_{i1}, p_{i2}, \ldots, p_{ij}), \]

where \( p_{ij} \) represents the diagnosis weight of the \( j \)th diagnosis for the \( i \)th patient.

For each patient in the demographic dataset, a set of diagnoses is compiled from the training and test set of the problem list dataset separately, excluding ICD-9 codes 296, including all child codes, 300.4 and 311, which correspond to mood disorder diagnoses. These ICD-9 codes are excluded from the resulting patient diagnosis lists, as they only appear in the patient history for individuals who have suffered from a mood disorder and this information can already be deduced from the control patient status of the patient.

The patient vectors for both training and test set are constructed by scanning the patient diagnosis lists for training and test set, respectively. For each patient diagnosis list, the corresponding patient vector \( P_i \) is determined using the unique patient ID and for every diagnosis \( d_j \) occurring in the list, the diagnosis weight \( p_{ij} \) is set to 1. The diagnoses are looked up in the diagnosis vocabulary using the provided ICD-9 code.

The result of this process are two diagnosis-patient matrices, one for the training and one for the test set of the original problem list dataset. The diagnosis vocabulary generated from the training set of the problem list dataset contains 9,095 unique diagnoses. Further, the training matrix is about 180,000 columns wide, while the test matrix is about 46,000 columns wide, with both matrices containing about 9,000 rows, which relates to size of the vocabulary. It is important to note that the resulting matrices are very sparse, as the patients in the clinical datasets usually only have a small number of diagnoses in their patient history. So, to recapitulate, each column vector of the matrices now represents one patient taken from
the clinical data, each dimension signifying one of the diagnoses from the diagnosis vocabulary. These patient vectors can now be used as feature vectors to train a prediction model.

3.2.9 Dimensionality Reduction

For all intents and purposes, however, using feature vectors of this size and sparsity is not desirable, as training a robust prediction model on 9,095 sparse feature vectors is hard, because it is easy to fall victim to the curse of dimensionality [Rus97]. Points in high-dimensional space tend to be very far apart and highly prone to noise, which basically leads to every sample being an outlier of some sort. Fitting a machine learning model on these samples can easily lead to overfitting, as the model is not able to generalize enough. Additionally, training a model on a huge number of features is more computationally expensive and takes longer, which makes optimizing the model a very time consuming and inconvenient matter, as the model has to be retrained over and over again.

To solve these problems, the dimensionality of the matrix must be reduced, with the goal of creating a lower-dimensional yet more dense representations of the information contained in the original matrix. One way of doing this borrowed from LSA is singular value decomposition (SVD) [GR70]. In LSA, a truncated SVD is used to reduce the dimensionality of the term-document matrix by finding a low-rank approximation to the original matrix. By reducing the dimensionality, the sparsity and noise of the original matrix is also significantly reduced, while dimensions that are similar in meaning are combined into new semantic dimensions [LFL98].

Adapted to our problem, we can use truncated SVD to reduce the dimensionality of the diagnosis-patient matrix, while keeping most of the information contained in the matrix. The resulting dimensions of the patient vectors should also be semantically more expressive and more meaningful to a machine learning model.

The first step of our approach is to calculate a full SVD of the diagnosis-patient matrix. For this, our $m \times n$ diagnosis-patient matrix $A$ is decomposed into three matrices $U$, $\Sigma$ and $V$, so that

$$A = U\Sigma V^T, \quad (3.10)$$

where $\Sigma$ is a $m \times n$ matrix that contains the singular values of $A$ in descending order on its diagonal. $U$ is a $m \times m$ matrix where every row vector represents a diagnosis, while $V^T$ is a $n \times n$ matrix where every column vector represents a patient. Both $U$ and $V^T$ are unitary [GR70].
The idea behind truncated SVD is to calculate a low-rank approximation to $A$ with rank $k$. For this, only the $k$ largest singular values in $\Sigma$ are used, while the others are replaced with 0. As each singular value corresponds to a column in $U$ and a row in $V^T$, we eliminate the less important dimensions from both the patient vectors and the diagnosis vectors. This leaves us with the rank $k$ approximation

$$A_k = U_k \Sigma_k V_k^T,$$  \hspace{1cm} (3.11)

with $U_k$, $\Sigma_k$ and $V_k^T$ being the truncated versions of the original SVD. The resulting truncated patient vectors from $V_k^T$ are $k$-dimensional and can be used as feature vectors for a machine learning model.

The only important parameter for this method is the rank $k$, as it determines the number of dimensions of the resulting feature vectors. Naturally, choosing a large number for $k$ leads to feature vectors that are a better approximation to the original feature vectors, however, less reduction in terms of dimensionality. Therefore, the chosen parameter $k$ can have a grave impact on the performance of the machine learning model trained on the feature vectors.

For our model, we chose to try two different methods of determining a good value for $k$ that both reduces the dimensionality of the feature vectors significantly and results in them containing enough of the original information to be useful to a machine learning model.

The first method is concerned with the concept of retained energy in the low-rank approximation of the matrix. The energy of a matrix is defined as the sum of squared singular values. The retained energy of a reduced matrix can be calculated by dividing the energy of the reduced matrix by the energy of the full matrix \cite{GG13}, that is

$$E(A_k) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2}.$$  \hspace{1cm} (3.12)

A good rule of thumb is to select the rank $k$, so that 90% of the energy of the original matrix is retained in the low-rank approximation \cite{RU11}. Calculated on our diagnosis-patient matrix, this relates to a rank $k$ of 1,150 that satisfies the 90% of retained energy criterion.

The second method is called the scree test and is concerned with finding the optimal $k$ using a scree plot of the singular values. The scree plot visualizes the singular values in descending order, the resulting curve is examined by eye to find a sort of elbow in the curve, where the singular values stop decreasing rapidly and start to level off. The number of singular values to the left of this elbow point is a good candidate for the rank $k$, as it characterizes a fair trade-off between dimensionality
and meaningfulness. Figure 3.8 shows the scree plot on a linear scale. Due to the large amount of singular values, the scree plot looks quite clinched and the elbow is not easily inspected. Figure 3.9 shows the same plot using a logarithmic scale, which makes it significantly easier to examine.

Based on the visual inspection, we chose to use a rank $k$ of 200 in addition to the one determined using the 90% energy method. Both versions of the patient history features can now be used to train a machine learning model. The patient history features are assigned the PH namespace and called PH_F1 through PH_1150 for use in the blended dataset.

The Pig scripts for the data preparation for this section can be seen in Section 3.5 of the appendix. Additionally, the Python scripts can be found in Section F.3 in the appendix.
3.3 Predicting Mood Disorder Risk

The main goal of the thesis is to train a prediction model with the ability to predict the mood disorder risk for an individual. Using a data blending approach, multiple datasets stemming from different domains of data were acquired to enrich the clinical data provided by Geisinger Health Systems in order to get better insights into the individual’s personal environment and its influence on the development of mood disorders.

To this point, the acquired datasets were analyzed for data quality, as described in Section 3.1, and afterwards prepared for use as features for a prediction model, which is explained in detail in Section 3.2.

3.3.1 Blended Dataset

Before it is possible to train the mood disorder risk model, the final set of prepared datasets, namely the census community dataset, crime indicator dataset, mood disorder prevalence dataset, patient demographics dataset, and patient history dataset have to be combined into one coherent dataset that can then be used to train and evaluate the prediction model. Each record in this blended dataset represents a single individual and includes information about the individual’s life, their medical history, and their personal environment, as well as the mood disorder status of the individual that is calculated from the control patient status in the patient demographics dataset, as described in Section 3.1.1.

As a first step, the patient demographics dataset and the patient history dataset are joined on the unique patient identifier that is represented by the field `study_id` in both datasets. Afterwards, the result is joined with the mood disorder prevalence dataset, census community dataset and the crime indicator dataset on the patient’s ZIP Code that originates from the patient demographics dataset. A blending diagram describing the creation of the blended dataset can be seen in Figure 3.10.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Features</th>
<th>Field/s Used for Join</th>
</tr>
</thead>
<tbody>
<tr>
<td>Census Community Dataset</td>
<td>21</td>
<td>ZIP Code</td>
</tr>
<tr>
<td>Crime Indicator Dataset</td>
<td>1</td>
<td>ZIP Code</td>
</tr>
<tr>
<td>Mood Disorder Prevalence Dataset</td>
<td>2</td>
<td>ZIP Code</td>
</tr>
<tr>
<td>Patient Demographics Dataset</td>
<td>18</td>
<td>ZIP Code, Study ID</td>
</tr>
<tr>
<td>Patient History Dataset</td>
<td>1150</td>
<td>Study ID</td>
</tr>
</tbody>
</table>

Table 3.20: Number of features of the prepared datasets
The blended dataset includes the 21 demographic features from the census community dataset, the crime indicator feature from the crime indicator dataset, the two differently imputed mood disorder prevalence features from the mood disorder prevalence dataset, the 17 patient demographic features and the mood disorder status from the patient demographics dataset, and the 1,150 features from the patient history dataset. Additionally, we include the unique patient ID and the patient’s ZIP Code as found in the patient demographic dataset in order to check the resulting dataset for correctness. A compact breakdown of the features every dataset contributed to the blended dataset including the fields used to join the datasets can be seen in Table 3.20. In total, the blended dataset consists of 1,195 features, a detailed breakdown of the dataset’s schema is shown in Table 3.21. The first feature in the dataset is the mood disorder status of the patient and represents the target for our prediction model. The following two features are the patient’s unique ID and their ZIP Code, which are included for the purpose of being able to cross-reference each row with the original datasets. All remaining features are used as the predictor variables for the prediction model.

In accordance to the evaluation methodology and the resulting split on the patient
3.3. Predicting Mood Disorder Risk

The blended dataset is created for both training and test patients independently to ensure that the trained model can be evaluated on a set of unseen data. The resulting training set contains 187,505 records and the corresponding test set consists of 46,898 records.

Table 3.21: Schema of blended dataset

<table>
<thead>
<tr>
<th>Feature Number</th>
<th>Feature ID</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PD_MD_STATUS</td>
<td>Patient Demographics Dataset</td>
</tr>
<tr>
<td>2</td>
<td>STUDY_ID</td>
<td>Patient Demographics Dataset</td>
</tr>
<tr>
<td>3</td>
<td>ZIP_CODE</td>
<td>Patient Demographics Dataset</td>
</tr>
<tr>
<td>4</td>
<td>MDP_AVG</td>
<td>Mood Disorder Prevalence Dataset</td>
</tr>
<tr>
<td>5</td>
<td>MDP_MODEL</td>
<td>Mood Disorder Prevalence Dataset</td>
</tr>
<tr>
<td>6</td>
<td>CR_CI</td>
<td>Crime Indicator Dataset</td>
</tr>
<tr>
<td>7</td>
<td>CC_POP_TOTAL</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>8</td>
<td>CC_POP_HISPANIC</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>9</td>
<td>CC_POP_WHITE</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>10</td>
<td>CC_POP_BLACK</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>11</td>
<td>CC_POP_AI</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>12</td>
<td>CC_POP ASIAN</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>13</td>
<td>CC_POP_PI</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>14</td>
<td>CC_POP OTHER</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>15</td>
<td>CC_POP_MIXED</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>16</td>
<td>CC_ECON_UNEMP</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>17</td>
<td>CC_ECON_OCC_MGT</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>18</td>
<td>CC_ECON_OCC_SERV</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>19</td>
<td>CC_ECON_OCC_SALES</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>20</td>
<td>CC_ECON_OCC_CONST</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>21</td>
<td>CC_ECON_OCC_TRANS</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>22</td>
<td>CC_ECON_CTW</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>23</td>
<td>CC_ECON_INSURED</td>
<td>Census Community Dataset</td>
</tr>
<tr>
<td>24</td>
<td>CC_ECON BELOW_POV</td>
<td>Census Community Dataset</td>
</tr>
</tbody>
</table>

Continued on next page

10 These features are only included for debug purposes and are not used as features for the prediction model.
### Table 3.21 – continued from previous page

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>CC_ECON_INC_MED</td>
</tr>
<tr>
<td>26</td>
<td>CC_ECON_INC_AVG</td>
</tr>
<tr>
<td>27</td>
<td>CC_GEO_DENS</td>
</tr>
<tr>
<td>28</td>
<td>PD_AGE</td>
</tr>
<tr>
<td>29</td>
<td>PD_GENDER</td>
</tr>
<tr>
<td>30</td>
<td>PD_WHITE</td>
</tr>
<tr>
<td>31</td>
<td>PD_BLACK</td>
</tr>
<tr>
<td>32</td>
<td>PD_ASIAN</td>
</tr>
<tr>
<td>33</td>
<td>PD_AI</td>
</tr>
<tr>
<td>34</td>
<td>PD_PI</td>
</tr>
<tr>
<td>35</td>
<td>PD_MARRIED</td>
</tr>
<tr>
<td>36</td>
<td>PD_DIVORCED</td>
</tr>
<tr>
<td>37</td>
<td>PD_WIDOWED</td>
</tr>
<tr>
<td>38</td>
<td>PD_EMP_FULL</td>
</tr>
<tr>
<td>39</td>
<td>PD_EMP_PART</td>
</tr>
<tr>
<td>40</td>
<td>PD_EMP_STU</td>
</tr>
<tr>
<td>41</td>
<td>PD_EMP_SELF</td>
</tr>
<tr>
<td>42</td>
<td>PD_EMP_RET</td>
</tr>
<tr>
<td>43</td>
<td>PD_EMP_INS</td>
</tr>
<tr>
<td>44</td>
<td>PD_MEDICARE</td>
</tr>
<tr>
<td>45</td>
<td>PH_F1</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>1195</td>
<td>PH_F1150</td>
</tr>
</tbody>
</table>

The Pig scripts for join of all acquired and prepared dataset into one coherent blended dataset can be seen in Section E.6 of the appendix.

#### 3.3.2 Prediction Model

Everything we have talked about up until now boils down to the construction of a model that is able to predict the individualized risk of developing a mood disorder. In the most general terms, a prediction model able to solve this problem uses a set of sample individuals, for whom it is known whether they have developed a
mood disorder, to predict the mood disorder risk for individuals it has not seen before. In this context, there are only two observable states: either the individual has developed a mood disorder or they haven’t. That being the case, the prediction task can also be viewed as a binomial classification task, where the class labels relate to the two observable states and the corresponding groups of individuals.

In terms of this classification task, the mood disorder risk now can be expressed as the probability of belonging to the group of individuals that have developed a mood disorder. Using this fact, the mood disorder risk model can be implemented using any classification model that can predict probabilities for its predicted class assignments.

For the implementation of the mood disorder risk model, we chose to use two separate machine learning models in parallel, namely a logistic regression model (LR) and gradient boosted trees model (GBT). The logistic regression model is used as a simple baseline model that is computationally cheap to give a rough understanding of how well the chosen features are suited for the prediction task. In contrast, the gradient boosted trees model is used as the state-of-the-art model that would be applied to the real-world task, as it yields better performance in part due to its non-linear nature. The two models are implemented in Python using the scikit-learn library.

For both models, logistic regression and gradient boosted trees, the training set of the features from the blended dataset described in Section 3.3.1 are used to train the models. The mood disorder status is used as the target that the models are trained on.

The logistic regression model is implemented using the LogisticRegression class provided by the sklearn.linear_model package. The model is trained using L2 regularization and liblinear as the solver backing the model [Fan+08]. The code that is used to train the logistic regression model can be seen in Section ?? of the appendix.

The GradientBoostingClassifier class from the sklearn.ensemble package is used to implement the gradient boosted trees model. This type of model offers a great number of hyperparameters that can be tweaked to increase the performance of the model for a specific classification task. It is important to not use the test set for this hyperparameter optimization to prevent overfitting the model to the test data. Rather, techniques like cross-validation should be used to optimize the hyperparameters. As the optimal set of hyperparameters heavily depends on the exact set of data used to train the model and finding these optimal hyperparameters is a tedious and computationally expensive task, we opted to use the model with the default hyperparameters as they are provided by scikit-learn. These hyperparameters include the number of estimators used, which is set to 100, and a learning rate of 0.1. Additionally, the maximum depth of the estimators is set to 3. The code that
is used to train the gradient boosted trees model can be seen in Section ?? of the appendix.

Both models are first evaluated on the training data using 5-fold cross validation. The prediction accuracy at a threshold of 50%, the log loss and the ROC AUC are calculated using the `cross_val_score` method from the `sklearn.model_selection` package. The cross validation results give a first insight into how well the models will perform on out-of-sample data and can reveal signs of overfitting and other serious problems the model might have early on.

After the first evaluation round, the models are trained on the whole training set using the `fit` method of the corresponding model instance. Finally, the `predict_proba` method of the models is used to predict the mood disorder risk for the test set and the results are evaluated on the mood disorder status of the test patients. Once the models are trained and evaluated, they can be used to predict the mood disorder risk for unseen individuals.

### 3.4 Evaluation

The prediction model introduced in this thesis has many real-world applications. However, after the model is trained, it has to be evaluated in order to know how good it will perform in these real-world applications.

In order to not only see how good the overall model performs but also be able to see if the data blending approach described in this thesis improves the model, we chose to train the model in multiple configurations that include different combinations and versions of feature sets. We use these different model configurations to find the best possible version of the mood disorder risk model and examine how the data from each source contributes to the overall model’s performance.

Each model configuration is trained and evaluated as outlined in the previous section. The metrics used to evaluate the model include the prediction accuracy at the 50% threshold, the Brier score, the log loss, the ROC curve, including the corresponding AUC, the precision-recall curve and for the most interesting model configurations also the F-measure curve including the F-measure at the optimal threshold. A detailed explanation of the used evaluation metrics can be found in Appendix D in Section D.2.

#### 3.4.1 Model Configurations

The model configurations evaluated in the following sections can be grouped into three major categories:
3.4. Evaluation

1. Model configurations that only encompass features from a single source, including different versions of the same feature sets

2. Model configurations including the complete blended dataset with different versions of the patient history features

3. Model configurations including all but one feature source

The first group of model configurations is concerned with evaluating the individual feature sets and how well the models perform when only a single set of features is available. It is also meant to identify the best version of each feature set to use for the final model. The model configurations that are evaluated as part of this group are:

**Two Versions of Mood Disorder Prevalence Features (MDP)**
In these two configurations, the models are trained on the two differently imputed mood disorder prevalence features.

**Crime Indicator Feature (CI)**
In this configuration, the model is trained on the crime indicator feature only.

**Census Community Features (CC)**
In this configuration, the model is trained on the census community features only.

**Patient Demographics Features (PD)**
In this configuration, the model is trained on the patient demographics features only.

**Patient History Features (PH)**
In this configuration, the model is trained on the patient history features only.

The next group of model configurations evaluated include the complete blended dataset with the best performing feature versions as determined using the model configurations in the first group. The two model configurations that fall into this group are:

**All Features using Patient History with 90% Retained Energy (ALL)**
In this configuration, the model is trained on all 1,195 features of the blended dataset using the patient history features that contain 90% of the original energy of the diagnosis-patient matrix.

**All Features using Patient History Reduced by Scree Test (ALL/SCREE)**
In this configuration, the model is trained on the blended dataset including only 200 patient history features according to the scree test.

The third and final group of model configurations is concerned with explaining why the final model performs as it does. For this purpose, all model configurations
in this group are based of the ALL model configuration, while the features of one particular feature set are left out for each model configuration.

**Leave out Patient History (ALL-PH)**
In this configuration, the model is trained on all features except the patient history features.

**Leave out Patient Demographics (ALL-PD)**
In this configuration, the model is trained on all features except the patient demographics features.

**Leave out Mood Disorder Prevalence (ALL-MD)**
In this configuration, the model is trained on all features except the mood disorder prevalence feature.

**Leave out Census Community (ALL-CC)**
In this configuration, the model is trained on all features except the patient demographics features.

**Leave out Crime Indicator (ALL-CI)**
In this configuration, the model is trained on all features except the crime indicator feature.

### 3.4.2 Evaluation Baseline

When trying to evaluate different machine learning models, it is useful to define a baseline, to which the individual models can be compared. Baselines are usually naive in their approach and can be anything from guessing random outputs to a simple regression model. Using a baseline allows us to see if the evaluated models actually perform better than this naive approach.

The baseline used in the following sections is rather simple. It uses the class prior probabilities calculated on the training set as a constant prediction output. As 52,700 of the 187,505 patients found in the training set belong to the group of patients suffering from a mood disorder, this leaves the baseline with a constant prediction of a probability of 28.11% for the class of affected patients.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.5000</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
</tbody>
</table>

**Table 3.22: Evaluation results for the baseline**

At the 50% threshold, the baseline yields an accuracy of 0.7162 on the test patients, as all patients are predicted to not suffer from a mood disorder and 71.62% of the test patients in fact don’t suffer from a mood disorder. The ROC AUC is naturally...
0.5, as any threshold below 28.11% leads to both the true positive rate and the false positive rate being 0, while any threshold above this leads to both the true positive rate and the false positive rate being 1. A detailed breakdown of the baseline’s performance can be seen in Table 3.22.

### 3.4.3 Mood Disorder Prevalence (MDP)

<table>
<thead>
<tr>
<th>Model</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imputation by Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.5728</td>
<td>0.2008</td>
<td>0.5902</td>
<td>0.7157</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.5755</td>
<td>0.2005</td>
<td>0.5899</td>
<td>0.7157</td>
</tr>
<tr>
<td>Imputation by Regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.5721</td>
<td>0.2008</td>
<td>0.5903</td>
<td>0.7159</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.5743</td>
<td>0.2006</td>
<td>0.5900</td>
<td>0.7160</td>
</tr>
</tbody>
</table>

**Table 3.23:** Evaluation results for the mood disorder prevalence features

In this section, both the mood disorder prevalence feature imputed by a regression and the mood disorder prevalence feature imputed by the average, both described in Section 3.2.7, are trained separately and evaluated to see which imputation strategy ultimately performs better. The better performing version of the mood disorder prevalence feature is used for the final model.

The results for both feature versions show that the gradient boosted trees model performs slightly better than the logistic regression model regardless of the imputation method used, although the results for all four models is quite close. For instance, the ROC AUC of the gradient boosted trees model trained on the dataset imputed by the average is slightly larger at 0.576 than the ROC AUC of the logistic regression model trained on the same dataset at 0.573. The Brier score, log loss and the accuracy are almost identical for all models.

Overall, the models trained on the feature imputed by the average perform slightly better than the models trained on the feature imputed by the regression. The concrete evaluation results for all four models trained can be seen in Table 3.23.

As the evaluation results of both feature versions are nearly identical, the following further discussion of the results will only focus on the slightly better performing feature imputed by the average.

The ROC curve and the precision-recall curve of the logistic regression model trained on the feature imputed by the average are shown in Figure 3.11a and Figure 3.11b. For the gradient boosted trees model trained on the feature imputed by the average, both plots can be seen in Figure 3.11c and Figure 3.11d. The ROC curves show
that both models perform better than random guessing, as the ROC AUC is larger than 0.5. The precision-recall curves highlight that the chosen threshold has only a minimal effect on the precision and recall values of the models.

As can be seen from the evaluation results, the both models trained on the mood disorder prevalence feature imputed by the average yield some improvement over the baseline. Especially the gradient boosted trees model shows that including the mood disorder prevalence feature in the final model could lead to better results.

### 3.4.4 Crime Indicator Only (CI)

For the crime indicator evaluation, both models are trained on the crime indicator dataset, described in Section 3.2.3.
3.4. Evaluation

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.5085</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.5665</td>
<td>0.2013</td>
<td>0.5916</td>
<td>0.7162</td>
</tr>
</tbody>
</table>

Table 3.24: Evaluation results for the crime indicator features

The results of both models show that the gradient boosted trees model performs better than the logistic regression model, as expected. The ROC AUC of the gradient boosted trees model is larger at 0.567 than the ROC AUC of the logistic regression model at 0.509. The Brier score and the log loss are both lower for the gradient boosted trees model than for the logistic regression model, meaning the gradient boosted trees model performs better. Although the gradient boosted trees model has a larger ROC AUC and lower Brier Score and log loss, the accuracy of both models is identical at 71.62% of correctly classified individuals. The concrete evaluation results for both models can be seen in Table 3.24.

The ROC curve and the precision-recall curve of the logistic regression model are shown in Figure 3.12a and Figure 3.12c. For the gradient boosted trees model, both plots can be seen in Figure 3.12b and Figure 3.12d.

The ROC curve of the logistic regression model shows that the logistic regression is almost identical to random guessing, as the ROC curve is almost linear. Additionally, the precision-recall curves highlight that the chosen threshold has only a minimal effect on the precision and recall values of the models.

However, the analysis of the crime indicator features shows that gradient boosted trees model provides meaningful insights on the crime indicator feature, whereas the logistic regression does not. Only the gradient boosted trees model shows that including the crime indicator feature in the final model could lead to better results.

3.4.5 Census Community Only (CC)

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.5365</td>
<td>0.2026</td>
<td>0.5948</td>
<td>0.7162</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.5783</td>
<td>0.2002</td>
<td>0.5891</td>
<td>0.7161</td>
</tr>
</tbody>
</table>

Table 3.25: Evaluation results for the census community features

For the evaluation of the census community dataset, both models are trained on the census community features, described in Section 3.2.2.

As expected, based on the previous evaluations, the results of both models show that the gradient boosted trees model performs better than the logistic regression.
model. The ROC AUC of the gradient boosted trees model is larger at 0.578 than the ROC AUC of the logistic regression model at 0.537. The Brier score and the log loss are both lower for the gradient boosted trees model than for the logistic regression model, meaning the gradient boosted trees model performs better. Although the gradient boosted trees model has a larger ROC AUC and lower Brier Score and log loss, the accuracy of both models is almost identical at 71.6% of correctly classified individuals. This is comparable with the previous evaluation of the crime indicator feature, in which the accuracy is identical, but the Brier score and log loss are not. The concrete evaluation results for both models can be seen in Table 3.25.

The ROC curve and the precision-recall curve of the logistic regression model are shown in Figure 3.13a and Figure 3.13c. For the gradient boosted trees model, both plots can be seen in Figure 3.13b and Figure 3.13d.

Both ROC curves of the models show that the models perform better than random guessing. At a first glance, the ROC of the logistic regression looks almost linear,
3.4. Evaluation

but the ROC AUC at 0.54 shows that it cannot be linear. Additionally, the precision-recall curves highlight that the chosen threshold has only a minimal effect on the precision and recall values of the models.

As can be seen from the evaluation results, the both models trained on the census community yield some improvement to the baseline. Especially the gradient boosted trees model shows that including the census community feature in the final model could lead to better results.

3.4.6 Patient Demographics Only (PD)

In this section, the patient demographic dataset extracted from the clinical dataset, as described in Section 3.2.5, is evaluated. Therefore, both models are trained on the patient demographic features.

The results of both models show that the gradient boosted trees model performs better than the logistic regression model, but both models perform considerably
(a) ROC curve of the logistic regression model

(b) ROC curve of the gradient boosted trees model

(c) Precision-recall curve of the logistic regression trees model

(d) Precision-recall curve of the gradient boosted trees model

(e) Precision-recall curve of the logistic regression trees model

(f) F-Measure curve of the gradient boosted trees model

**Figure 3.14:** Evaluation results for the patient demographic features
3.4. Evaluation

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.6712</td>
<td>0.1893</td>
<td>0.5605</td>
<td>0.7162</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.7313</td>
<td>0.01756</td>
<td>0.5181</td>
<td>0.7349</td>
</tr>
</tbody>
</table>

Table 3.26: Evaluation results of the patient demographic features. Results are shown for the logistic regression model and the gradient boosted trees model.

better than the already evaluated models. The ROC AUC of the gradient boosted trees model is larger at 0.731 than the ROC AUC of the logistic regression model at 0.671. The Brier score, log loss and the accuracy are much better for gradient boosted trees than for logistic regression. The concrete evaluation results for both models trained can be seen in Table 3.26.

The ROC curve and the precision-recall curve of the logistic regression model are shown in Figure 3.14a and Figure 3.14c. The same curves for the gradient boosted tree model are shown in Figure 3.14b and Figure 3.14d.

The ROC curves show that both models perform better than random guessing, as the ROC AUC is larger than 0.5.

It is the first evaluation, in which the precision-recall curves highlight that choosing the right threshold has an significant effect on the precision and recall values of the models. The F-measure curves for the logistic regression model and the gradient boosted trees models are shown in Figure 3.14e and Figure 3.14f. From the F-measure curve of the logistic regression model it can be seen that the best threshold lies at 25.5%, resulting in a recall of 0.7118 and a precision of 0.3927. For the gradient boosted trees model the best threshold lies at 28.7%, resulting in a recall of 0.7496 and a precision of 0.4154.

The analysis shows that both models trained on the patient demographic features yield a significant improvement to the baseline. Particularly the gradient boosted trees model shows that including the patient demographic features in the final model should most definitely lead to better results.

3.4.7 Patient History Only (PH)

In Section 3.2.9 the dimensionality reduction of the diagnosis-patient matrix using SVD is described, in which two strategies are illustrated to reduce dimensionality without losing too much information, as the original matrix is too large to train a prediction model on. The first strategy is to retain 90% of the energy of the matrix resulting in 1,150 features and the second is to perform a scree test on the decomposed matrix, in which we chose to use the first 200 features.
Chapter 3. Implementation

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.8037</td>
<td>0.1891</td>
<td>0.5628</td>
<td>0.7176</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.8036</td>
<td>0.1540</td>
<td>0.4722</td>
<td>0.7775</td>
</tr>
</tbody>
</table>

(a) Breakdown of evaluation results

(b) ROC curve of the gradient boosted trees model

c) ROC curve of the gradient boosted trees model

(d) Precision-recall curve of the logistic regression model

e) Precision-recall curve of the gradient boosted trees model

(f) F-Measure curve of the gradient boosted trees model

(g) Precision-recall curve of the gradient boosted trees model

Figure 3.15: Evaluation results for the models trained on the patient history features 90% version only
3.4. Evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.469282</td>
<td>0.203125</td>
<td>0.596199</td>
<td>0.716214</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.706808</td>
<td>0.182808</td>
<td>0.545692</td>
<td>0.728901</td>
</tr>
</tbody>
</table>

(a) Breakdown of evaluation results

(b) ROC curve of the gradient boosted trees model

(c) ROC curve of the gradient boosted trees model

(d) Precision-recall curve of the gradient boosted trees model

(e) Precision-recall curve of the gradient boosted trees model

(f) F-Measure curve of the gradient boosted trees model

(g) Precision-recall curve of the gradient boosted trees model

Figure 3.16: Evaluation results for the models trained on the patient history features scree version only
Both versions of the patient history features are trained on separately and evaluated to see which dimensionality reduction strategy ultimately performs better. As enough training data is available, it should not be a problem to train a prediction model on more than 1,000 features. It is to be expected that the patient history features retaining 90% of the energy will perform significantly better, however, we would like to determine how much better. It is also obvious that training a regression model on 1,150 features is considerably more computationally expensive than training the same model on 200 features. For this reason, we evaluate both versions of the features to see if the performance of the model trained on the patient history features retaining 90% of the original energy is worth the extra work in comparison to the version constructed using the scree test.

The results of the models trained on both versions of the patient history features show that linear regression model and the gradient boosted trees model trained on the version that retains 90% of the original energy perform both significantly better than both models of the scree version. The ROC AUC for both models trained on the 90% energy version are almost identical, with the ROC AUC of the gradient boosted trees model at 0.8036 and the ROC AUC of the logistic regression model at 0.8037. This can be seen in Table 3.15a. The ROC curve of both models can be seen in Figure 3.15b and Figure 3.15c.

This is the first time in the evaluation process that the logistic regression model has a larger ROC AUC than the gradient boosted trees model. However, the Brier score, log loss and the accuracy are significantly better for the gradient boosted trees model, which is consistent with our other results.

The precision-recall curve and the F-measure curve of the 90% energy version on the logistic regression model can be seen in Figure 3.15d and Figure 3.15f. For the gradient boosting both curves can be seen in Figure 3.15e and Figure 3.15g. For both versions on both regression models, it can be seen that it is possible to use the maximum f-measure value to optimize both precision and recall. Especially, the threshold of the gradient boosted trees model on the 90% energy version lies at 28.7%, resulting in a recall of 0.7070 and a precision of 0.5234.

The evaluation results of the models trained on the scree version, shown in Figure 3.16, show that the models perform worse in comparison to the models trained on the 90% energy version. The gradient boosted trees model has a ROC AUC at 0.7068, whereas the logistic regression model’s ROC AUC is at 0.4693. In other words every random guessing approach would score better than our logistic regression model, which means that it also performs worse than our baseline. The corresponding threshold of the gradient boosted trees model on the scree version lies at 24.8%, resulting in a recall of 0.7730 and a precision of 0.3916. However, the gradient boosted trees model trained on the scree version is accurate enough to justify using it in combination with the other feature sets in the following section.
3.4. Evaluation

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression (90 %)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>Gradient Boosting (90 %)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
<tr>
<td>Scree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression (Scree)</td>
<td>0.6818</td>
<td>0.1870</td>
<td>0.5547</td>
<td>0.7207</td>
</tr>
<tr>
<td>Gradient Boosting (Scree)</td>
<td>0.7738</td>
<td>0.1650</td>
<td>0.4932</td>
<td>0.7530</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.5000</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
</tbody>
</table>

**TABLE 3.27:** Breakdown of both the linear regression model and the gradient boosted trees model on ALL and All/SCREE configuration

![ROC Curve](image1.png)  ![ROC Curve](image2.png)

(a) ROC curve of the gradient boosted trees model on the ALL/SCREE configuration  
(b) ROC curve of the gradient boosted trees model on the ALL configuration

**FIGURE 3.17:** Evaluation results for gradient boosted trees model on both ALL and ALL/SCREE configuration

### 3.4.8 All Features (ALL)

The ALL model configuration represents the whole blended dataset that we have constructed throughout this chapter. The models trained on this model configuration are principally what this thesis is about, the construction of a prediction model for mood disorder risk of an individual. The best performing model of the models evaluated in this section is a good candidate to be used as the actual mood disorder risk model that can be applied to the real-world task of predicting the mood disorder risk of an individual in a clinical context.

Two versions of the blended dataset are used for training the regression model. First one is the blended dataset with the patient history retaining 90% of the energy of the patient history matrix and the second one contains instead the scree version of the patient history. Both a logistic regression model and a gradient boosted trees model are trained on both dataset separately. Based on the previous evaluation
The results show that both gradient boosted trees models for the two different datasets perform significantly better than the linear regression model. The gradient boosted trees model for the 90% energy version performs best with a ROC AUC value at 0.83, whereas the ROC AUC of the second version is 0.77. Both ROC curves can be seen in Figure 3.18a and Figure 3.18b.

For both regression models the maximum threshold can be chosen to optimize the precision and recall of the corresponding model. Especially, for the gradient boosted trees model trained on the full dataset containing the 90% energy version the threshold of 29.3% optimizes the precision up to 0.5431 and the recall to 0.7357. The threshold for the corresponding model is shown in Figure 3.19.

The evaluation of the models trained on the whole blended dataset shows that
models trained on all 1,195 features of the blended dataset, including the 1,145 patient history features that retain 90% of the energy of the original diagnosis-patient matrix, perform significantly better than the ones trained on the blended dataset with the smaller number of 200 patient history features that were determined using the scree test. The overall best performing model in this regard is unsurprisingly the gradient boosted trees model trained on all 1,195 features.

In the following analysis of which sets of features contribute to the overall performance, in regards to predicting the mood disorder risk, only the models trained on all 1,195 features are considered.

### 3.4.9 Importance of Features

Leaving out certain feature sets enables us to evaluate the importance of said features to the final model. The goal is to inspect the difference between the models trained on all features and the ones where the particular set of features is left out, to see if these features contribute to the overall performance and if so, how much. As described in Section 3.4.1, we defined five different model configurations that relate to the five feature sets that form the blended dataset.

For each model configuration both the logistic regression and gradient boosted trees model are trained on four of the five feature sets in order to reveal how much the fifth feature set influences the prediction performance. Table 3.28 shows a breakdown of the evaluation results for the five different model configurations. Visualizations of the ROC curve, the precision-recall curve and the F-measure curve for all five model configurations can be found in Section A.3 of Appendix A.

As expected, when leaving out the patient history features from the models, the prediction performance is significantly reduced. Across all evaluation metrics it can be seen that the models trained without the patient history features are performing significantly worse. Especially the ROC AUC for both the logistic regression model and the gradient boosted trees model takes a significant hit at 0.6814 and 0.7407 respectively. On a positive note, the models trained without the patient history features still perform better than the baseline.

This is mostly due to the patient demographics features, as can be seen when leaving them out. The models trained on the ALL - PD model configuration perform astoundingly bad. The logistic regression model trained on the ALL - PD model configuration performs almost as bad as the same model trained on the ALL - PH model configuration across all evaluation metrics, only the ROC AUC is slightly better at 0.6988 compared to 0.6814 with the ALL - PH configuration. The gradient boosted trees model does perform better when trained on the ALL - PD model configuration in comparison to the same model trained on the ALL - PH configuration,
but is also significantly worse than the same model trained on the ALL configuration. This shows that the patient demographics features do in fact contribute to the prediction performance quite a lot, especially when used to train a gradient boosted trees model.

We assumed from the evaluation of the models trained on the mood disorder prevalence feature, presented in Section 3.4.3, that the mood disorder prevalence of the ZIP Code the individual lives in does in fact give some insight into the individual’s mood disorder risk. Leaving out the mood disorder prevalence feature from the final model can show us, how much the mood disorder prevalence contributes to the overall performance of the model.

The evaluation of the ALL - MDP model configuration shows that both the logistic regression and the gradient boosted trees model in this configuration still perform very closely to the ALL model configuration. However, removing the mood disorder prevalence feature from the model results in a slight decrease across all evaluation metrics. This shows, that the mood disorder prevalence, if only slightly, increases the prediction performance when predicting the mood disorder risk for unseen individuals.

To our surprise, two of the evaluated model configurations actually perform slightly better then the ALL configuration we are using as a benchmark when being used to train the gradient boosted trees model. The two model configurations in question

<table>
<thead>
<tr>
<th>Model</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL - PH (LR)</td>
<td>0.6814</td>
<td>0.1871</td>
<td>0.5549</td>
<td>0.7208</td>
</tr>
<tr>
<td>ALL - MDP (LR)</td>
<td>0.7239</td>
<td>0.1787</td>
<td>0.5347</td>
<td>0.7315</td>
</tr>
<tr>
<td>ALL - CI (LR)</td>
<td>0.7264</td>
<td>0.1777</td>
<td>0.5314</td>
<td>0.7350</td>
</tr>
<tr>
<td>ALL - CC (LR)</td>
<td>0.7261</td>
<td>0.1776</td>
<td>0.5316</td>
<td>0.7346</td>
</tr>
<tr>
<td>ALL - PD (LR)</td>
<td>0.6988</td>
<td>0.1872</td>
<td>0.5578</td>
<td>0.7200</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>ALL - PH (GB)</td>
<td>0.7407</td>
<td>0.1734</td>
<td>0.5128</td>
<td>0.7387</td>
</tr>
<tr>
<td>ALL - MDP (GB)</td>
<td>0.8267</td>
<td>0.1468</td>
<td>0.4487</td>
<td>0.7847</td>
</tr>
<tr>
<td>ALL - CI (GB)</td>
<td>0.8285</td>
<td>0.1461</td>
<td>0.4467</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL - CC (GB)</td>
<td>0.8289</td>
<td>0.1459</td>
<td>0.4463</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL - PD (GB)</td>
<td>0.8062</td>
<td>0.1533</td>
<td>0.4702</td>
<td>0.7775</td>
</tr>
<tr>
<td>ALL (GB)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.5000</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
</tbody>
</table>

Table 3.28: Results of the evaluation of the importance of the different sets of features used in the blended dataset. The model names relate to the abbreviations of the model configurations introduced in Section 3.4.1.
3.4. Evaluation

are ALL - CI and ALL - CC. When being used to train the logistic regression model, the two model configurations perform ever so slightly worse than our benchmark, however, when used to train the gradient boosted trees model, they actually perform better than the ALL model configuration.

This is very interesting, as it means that the final model could possibly perform better in a real-world scenario if both the crime indicator feature and the census community features are left out of the model. On the other hand, it could also mean that the crime indicator and the census community features are somehow contradictory and cancel out each other’s positive effect on the model. To confirm which of the two is true, we chose to also evaluate a sixth model configuration which excludes both the crime indicator and the census community features (ALL - CC - CI). If the resulting models still perform better than the ALL model configuration, this would mean that the crime indicator and the census community features actually impair the model’s ability to predict the mood disorder risk. On the other hand, if the model performs worse without the two feature sets, this would confirm the latter hypothesis.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL - CI - CC (LR)</td>
<td>0.7257</td>
<td>0.1777</td>
<td>0.5318</td>
<td>0.7342</td>
</tr>
<tr>
<td>ALL - CI - CC (GB)</td>
<td>0.8291</td>
<td>0.1459</td>
<td>0.4462</td>
<td>0.7869</td>
</tr>
<tr>
<td>ALL - CI (LR)</td>
<td>0.7264</td>
<td>0.1775</td>
<td>0.5314</td>
<td>0.7350</td>
</tr>
<tr>
<td>ALL - CI (GB)</td>
<td>0.8285</td>
<td>0.1461</td>
<td>0.4467</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL - CC (LR)</td>
<td>0.7261</td>
<td>0.1776</td>
<td>0.5316</td>
<td>0.7346</td>
</tr>
<tr>
<td>ALL - CC (GB)</td>
<td>0.8289</td>
<td>0.1459</td>
<td>0.4463</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>ALL (GB)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
</tbody>
</table>

Table 3.29: Evaluation results for the models trained on all features except the crime indicator and the census community features

The first two rows of Table 3.29 show the results for the models trained on the three remaining feature sets excluding the crime indicator and the census community features. The results, however, are quite a bit contradicting. As before, the ROC AUC of the logistic regression model trained on the ALL - CI - CC configuration is also slightly smaller than the one of the same model trained on the ALL configuration. This decrease in performance is also reflected in the other metrics, but overall the loss of performance is minuscule. On the other hand, the gradient boosted trees model trained on the same model configuration performs better than the same model trained on the ALL configuration. In fact, the gradient boosted trees model trained on the ALL - CI - CC configuration is the overall best performing model we have evaluated so far with an ROC AUC of 0.8291.
These results possibly mean two things: The crime indicator and census community features do add a tiny bit of value to the training data, but this gain only manifests when they are used to train the simpler logistic regression model. The more sophisticated gradient boosted trees model is not able to pull any further insights from the features, further, they actually seem to impair the model. This is bad news for our blending strategy, as the crime indicator and the census community features are a major part of our blending approach.

3.4.10 Summary of Results

In the evaluation process, we have used various model configurations to see which perform best and which feature sets have the highest impact on the mood disorder risk prediction. Table 3.30 shows a detailed breakdown of all model configuration that were evaluated in this chapter.

The most important feature set of the blended dataset is the patient history. For the patient history features, two versions were evaluated, namely a version that retains 90% of the energy of the original diagnosis-patient matrix with 1,150 features and a version that was created using a scree test with only 200 features. The evaluation results show that the 90% energy version performs significantly better than the scree version, which was expected, as the 90% energy version encompasses significantly more features. However, the additional work that has to be put into training the 90% energy version is well worth the improvement in performance compared to the scree version.

To our surprise, the second best performing feature set are the patient demographic features. Although the feature set contains shallow information about an individual by itself, the patient demographics provide valuable insights in regards to the mood disorder risk. Especially noteworthy is the yield of performance of the patient demographic features with respect to the small amount of features needed to express the contained information.

However, the mood disorder prevalence feature, the census community features and the crime indicator feature do not have a tremendous impact on the overall model. Using the gradient boosted trees model, they do give some insights in regards to the mood disorder risk, nonetheless. Of the three mentioned feature sets, the mood disorder prevalence feature adds the most value to the overall model.

The evaluation results of the models trained on all feature sets of the blended dataset show that the model configuration that includes the patient history that retains 90% of the energy performs better than the one including the scree version of the patient history features. However, the scree version profits more of the additional feature set than the 90% energy version.
### 3.4. Evaluation

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP (LR)</td>
<td>0.5728</td>
<td>0.2008</td>
<td>0.5902</td>
<td>0.7157</td>
</tr>
<tr>
<td>CI (LR)</td>
<td>0.5085</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
<tr>
<td>PH (LR)</td>
<td><strong>0.8037</strong></td>
<td>0.1891</td>
<td>0.5628</td>
<td>0.7176</td>
</tr>
<tr>
<td>PD (LR)</td>
<td>0.6712</td>
<td>0.1893</td>
<td>0.5605</td>
<td>0.7162</td>
</tr>
<tr>
<td>CC (LR)</td>
<td>0.5365</td>
<td>0.2026</td>
<td>0.5948</td>
<td>0.7162</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td><strong>0.1774</strong></td>
<td><strong>0.5314</strong></td>
<td><strong>0.7353</strong></td>
</tr>
<tr>
<td>ALL/SCREE (LR)</td>
<td>0.6815</td>
<td>0.1870</td>
<td>0.5549</td>
<td>0.7208</td>
</tr>
<tr>
<td>ALL - PH (LR)</td>
<td>0.6814</td>
<td>0.1871</td>
<td>0.5549</td>
<td>0.7208</td>
</tr>
<tr>
<td>ALL - PD (LR)</td>
<td>0.6988</td>
<td>0.1872</td>
<td>0.5578</td>
<td>0.7200</td>
</tr>
<tr>
<td>ALL - MDP (LR)</td>
<td>0.7239</td>
<td>0.1787</td>
<td>0.5347</td>
<td>0.7315</td>
</tr>
<tr>
<td>ALL - CC (LR)</td>
<td>0.7261</td>
<td>0.1776</td>
<td>0.5316</td>
<td>0.7346</td>
</tr>
<tr>
<td>ALL - CI (LR)</td>
<td>0.7264</td>
<td>0.1775</td>
<td><strong>0.5314</strong></td>
<td><strong>0.7350</strong></td>
</tr>
<tr>
<td>ALL - CI - CC (LR)</td>
<td>0.7257</td>
<td>0.1777</td>
<td>0.5318</td>
<td>0.7342</td>
</tr>
<tr>
<td>MDP (GBT)</td>
<td>0.5755</td>
<td>0.2005</td>
<td>0.5899</td>
<td>0.7157</td>
</tr>
<tr>
<td>CI (GBT)</td>
<td>0.5665</td>
<td>0.2013</td>
<td>0.5916</td>
<td>0.7162</td>
</tr>
<tr>
<td>PH (GBT)</td>
<td>0.8036</td>
<td>0.1540</td>
<td>0.4722</td>
<td>0.7775</td>
</tr>
<tr>
<td>PD (GBT)</td>
<td>0.7313</td>
<td>0.1756</td>
<td>0.5181</td>
<td>0.7349</td>
</tr>
<tr>
<td>CC (GBT)</td>
<td>0.5783</td>
<td>0.2002</td>
<td>0.5891</td>
<td>0.7161</td>
</tr>
<tr>
<td>ALL (GBT)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td><strong>0.7870</strong></td>
</tr>
<tr>
<td>ALL/SCREE (GBT)</td>
<td>0.7682</td>
<td>0.1665</td>
<td>0.4964</td>
<td>0.7494</td>
</tr>
<tr>
<td>ALL - PH (GBT)</td>
<td>0.7407</td>
<td>0.1734</td>
<td>0.5128</td>
<td>0.7387</td>
</tr>
<tr>
<td>ALL - PD (GBT)</td>
<td>0.8062</td>
<td>0.1533</td>
<td>0.4702</td>
<td>0.7775</td>
</tr>
<tr>
<td>ALL - MDP (GBT)</td>
<td>0.8267</td>
<td>0.1468</td>
<td>0.4487</td>
<td>0.7847</td>
</tr>
<tr>
<td>ALL - CC (GBT)</td>
<td>0.8289</td>
<td><strong>0.1459</strong></td>
<td>0.4463</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL - CI (GBT)</td>
<td>0.8285</td>
<td>0.1461</td>
<td>0.4467</td>
<td>0.7866</td>
</tr>
<tr>
<td>ALL - CI - CC (GB)</td>
<td><strong>0.8291</strong></td>
<td><strong>0.1459</strong></td>
<td><strong>0.4462</strong></td>
<td><strong>0.7869</strong></td>
</tr>
<tr>
<td>Baseline</td>
<td>0.5000</td>
<td>0.2033</td>
<td>0.5965</td>
<td>0.7162</td>
</tr>
</tbody>
</table>

*Table 3.30: Comparison of all evaluation results on all model configurations*
Altogether the gradient boosted trees model trained on all the features performs best. The ROC AUC of the model lies at 0.8284. The best threshold for optimizing the classification performance lies at 29.30%. At this point a f-measure of 0.625 is measurable that results in a precision of 0.7357 and a recall of 0.5431.

Additionally, the importance of each feature set was analysed for the overall performance of the final model by leaving each of them out one at a time and evaluate each of these model configurations. Our initial assumption that the patient history and the patient demographics have the greatest impact on the prediction accuracy of the final model was validated. To our surprise, however, we could improve the overall performance by leave out both the census community feature set and the crime indicator feature.

The census community features and the crime indicator feature seem to reduce the performance of the model slightly. As both datasets lie at the core of our data blending strategy, this leads to the question of whether the blending strategy is not good enough, the used datasets are of poor quality, the data is just not correlated enough or data blending in general is not the right approach for our purpose.
Chapter 4

Refinement

The previous chapter followed through the implementation of our strategy and concluded with the evaluation of the models that were trained on the different configurations of our blended dataset. As the evaluation results in Section 3.4 show, additional data on the personal environment of an individual has little to no influence on the prediction quality, when the patient history is present. Leaving out the census community features and the crime indicator feature that were used as features for the personal environment of the patients did not reduce the prediction performance significantly and the model trained on the model configuration that left out both feature sets even outperformed the model trained on the whole blended dataset.

This raises the question, if the data blending approach used in this thesis is flawed or data blending in general is not applicable to the problem of mood disorder risk prediction. Answering this question, however, would exceed the scope of this thesis, albeit it seems reasonable to address this question in the future.

At the same time, the patient history features and the patient demographic features that were extracted from the clinical data provided by Geisinger Health Systems were found to be almost wholly responsible for the prediction performance of the models trained on the blended data. Based on this, it seems feasible to focus on improving the analysis of the clinical data, especially the patient history, in order to improve the prediction performance.

The following section will follow through the refinement process chronologically. Starting with the advances analysis of the patient history in Section 4.1 including three different approaches. Finally, Section 4.2 is concerned with the final discussion of the main results of all the findings of the whole thesis.
4.1 Advanced Analysis of Patient History

As described in Section 3.2.8, we constructed a diagnosis-patient matrix from the patients’ medical histories, in which each patient is represented by a vector that contains every diagnosis the patient has received in the form of an ICD-9 code, given it is available in the patient’s medical history.

To recapitulate, the row vectors of the diagnosis-patient matrix, each representing one patient, could be used as feature vectors for a prediction model. However, these patient vectors potentially have a vast amount of dimensions, one for each diagnosis that occurred in the patient population, training on these high-dimensional vectors is both inefficient and prone to a phenomenon called the curse of dimensionality. Therefore, to be able to use the patient vectors, the dimensionality of diagnosis-patient matrix needs to be reduced. One approach to reduce the dimensionality of the patient matrix is to apply a truncated SVD to the matrix, as shown in Section 3.2.9.

4.1.1 Tf-idf Weighting

As stated before, the diagnosis-patient matrix is conceptually analogous to the term-document matrix, as it is commonly found in the context of the vector space model. Another highly related concept might help with improving the prediction results is weighting the matrix using tf-idf.

Tf-idf, which stands for *term frequency–inverse document frequency*, is a weighting scheme intended to reflect the importance or information content of the words in a document [Ram03]. At the core of tf-idf lies the assumption that the more often a word occurs in a document, the more important it is for the meaning of a document, while the more documents a word occurs in, the less important it is for the meaning of all documents.

Translated to the diagnosis-patient matrix, this would mean that diagnoses that were received by a patient multiple times are more expressive for their medical history, while diagnoses that were given to a considerable amount of patients are less expressive than the ones that only a few patients have received; a hypothesis that seems worth exploring.

The tf-idf weight for a given term $t$ in document $d$ is calculated by multiplying the *term frequency* denoted by $tf_{d,t}$ with the *inverse document frequency* denoted by $idf_t$, thus

$$ tf-idf_{t,f} = tf_{d,t} \times idf_t. \tag{4.1} $$
Table 4.1: Evaluation results of tf-idf weighted patient history features in place of the patient history features in the blended dataset (ALL/T-FIDF)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL/TFIDF (LR)</td>
<td>0.7390</td>
<td>0.1744</td>
<td>0.5240</td>
<td>0.7406</td>
</tr>
<tr>
<td>ALL/TFIDF (GBT)</td>
<td>0.8343</td>
<td>0.1461</td>
<td>0.4464</td>
<td>0.7878</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>ALL (GBT)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
</tbody>
</table>

The inverse document frequency is defined as the negative logarithm of the quotient of the document frequency denoted by \( df \) and the total number of documents denoted by \( N \), which yields

\[
idf_t = - \log \left( \frac{df_t}{N} \right),
\]

(4.2)

where the document frequency \( df \) is the number of documents the term occurs in.

The diagnosis-patient matrix is now weighted using tf-idf by calculating the tf-idf weight for each diagnosis of each patient. Thereafter, the truncated SVD is again applied to the matrix. To be able to compare the models trained on the tf-idf weighted patient history features, we chose to use a rank \( k \) of 1,150, the same rank that was determined for the 90% of retained energy on the original diagnosis-patient matrix in Section 3.2.8. The resulting feature vectors for the patients are identical in form to the feature vectors resulting after applying truncated SVD without the tf-idf weighting and as such can be used as a direct replacement in the blended dataset. To see, how well the new features perform when used to train a model, we define a new model configuration based on the ALL configuration that switches out the patient history features for the new tf-idf weighted patient history features. We call this additional model configuration ALL/TFIDF.

Table 4.1 shows the evaluation results of the two models trained on the new ALL/T-FIDF model configuration. The models trained on the ALL/TFIDF both perform better than the ones trained on the ALL configuration. The gradient boosted trees model trained on the ALL/TFIDF configuration performs better than any other model that has been evaluated thus far, with an ROC AUC of 0.8343 compared to an ROC AUC of 0.8284 for the same model trained on the ALL configuration. The ALL/TFIDF configuration also performs better than almost all other model configurations when used to train the logistic regression model, the only exception being the logistic regression model trained on the patient history features only.

The ROC curves of the two models can be seen in Figure 4.1. Compared to the
ROC curves of the models trained on the ALL model configuration in Section 3.4.8 no significant difference can be seen. The same applies to the precision-recall curves of the two models that is shown in Figure 4.2.

Especially the gradient boosted trees model trained on the ALL/TFIDF configuration is interesting, as it is the best performing model yet, when it comes to predicting the mood disorder risk. The model reaches its best F-measure when a threshold of 38.6% is used for the classification task. At this threshold, the F-measure lies at 0.632 compared to 0.625 for the same model trained on the ALL configuration.

This shows that applying tf-idf weighting to the diagnosis-patient matrix is a valid and useful addition to our approach. The little extra work that needs to be put into the data preparation is well worth the improved prediction performance.
4.1. Advanced Analysis of Patient History

4.1.2 Dimensionality Reduction: Ontology

Up to this point, we have only followed one approach in regards to the reduction of the dimensionality of the diagnosis-patient matrix, namely applying truncated SVD. However, this is not the only valid approach, in fact there is significant room for improvement. In the following sections we present two other approaches to the dimensionality reduction with the goal of improving our overall performance.

The dimensions of the diagnosis-patient matrix relate to the ICD-9 codes of the diagnoses that the patients in the clinical dataset received. However, these ICD-9 codes do not just plainly exist side by side, but rather live in a hierarchical structure. That being the case, it seems feasible to use this external knowledge when trying to reduce the dimensionality of the diagnosis-patient matrix.

To be able to use the external knowledge, it needs to be available in a machine readable format. One machine readable knowledge source is the Unified Medical Language System (UMLS), which includes a digital ontology with more than 900,000 concepts and 12 million relations between these concepts [Bod04]. Embedded into the UMLS is the ICD-9-CM ontology that holds information on the whole ICD-9 vocabulary including all hierarchical relations between ICD-9 codes. Sadly, the UMLS is not publicly available and can only be accessed by means of proprietary tools, which, although available and distributed along with the UMLS data, are not easily integrated into the Python-based pipeline used in this thesis.

Fortunately, there is a version of the UMLS ICD-9-CM ontology available that has been converted to RDF containing the complete ICD-9 hierarchy. This version uses OWL to describe the concepts in form of ICD-9 codes and aggregated ICD-9

\footnote{The ICD-9-CM ontology in RDF can be found at \url{https://bioportal.bioontology.org/ontologies/ICD9CM}. Information on the original UMLS ontology can be found at \url{https://www.nlm.nih.gov/research/umls/sourcedocu.../index.html}.}
code groups with hierarchical relations between them. OWL being a W3C standard, many open source tools exist to read and interact with OWL data, including extensive tools for use with Python. Listing 4.1 shows an excerpt directly taken from the ICD-9-CM ontology serialized in the Turtle format. As can be seen, the ontology contains objects for each ICD-9 code that live together in one coherent hierarchy.

Figure 4.4 shows an excerpt from the ICD-9-CM ontology that illustrates the hierarchical structure that ICD-9 codes live in. In the ICD-9-CM ontology, every ICD-9 code is subsumed by a more generalized ICD-9 code or an aggregated category of ICD-9 codes, such as the category of neurotic disorders, personality disorders, and other nonpsychotic mental disorders that spans the range of ICD-9 codes 300 through 316.99. All ICD-9 codes live under one of the top-level categories diseases and injuries (001–999.99), procedures (00–99.99), supplementary classification of external causes of injury and poisoning (E00–E999.9) and supplementary classification of factors influencing health status and contact with health services (V01–V91.99).
In order to reduce the dimensionality of the diagnosis-patient matrix, multiple ICD-9 codes have to be combined to form new dimensions while including as much of the original information as possible. The quality of the resulting matrix heavily depends on choosing the right dimensions to be combined together. Luckily, the ICD-9-CM ontology already includes aggregated ranges of ICD-9 codes that we can use as part of the ICD-9 hierarchy, namely the categories described above.

For the ICD-9 codes living under the top-level category of diseases and injuries, we selected not the direct children categories as dimensions, as these cover a broad range of different diagnoses and amount to only 17 categories, but rather the children of these categories, which are more specific and amount to 142 concepts in total including the categories of psychoses (290–299.99), intellectual disabilities (317–319.99) and neurotic disorders, personality disorders, and other nonpsychotic mental disorders (300–316.99) that can be seen in Figure 4.4. For the two supplementary classification top-level categories, we chose to use the direct children categories, as they
are not nearly as deep and broad as the diseases and injuries category. The child categories of the supplementary classification of external causes of injury and poisoning category amount to 17 categories, while the child categories of the supplementary classification of factors influencing health status and contact with health services category amount to 21 categories. For the top-level category of procedures, no dimensions where chosen, as these don’t occur in the diagnosis-patient matrix. In total, this amounts to 180 selected concepts from the ontology that can be used as the new dimensions of the reduced diagnosis-patient matrix. Listing 4.2 shows the SPARQL query that was used to query the RDF version of the ICD-9CM ontology for the selected concepts.

However, identifying the new dimensions is only part of the solution. The existing values of the diagnosis-patient matrix also need to be transformed into the values for the new dimensions. One way of calculating the value for the combined dimension is simply summing the values of the dimensions that where combined. Another approach could be using the average over the values. As diagnoses are additive by nature, meaning that an individuals health is negatively correlated to the cumulative sum of diagnoses they received, i.e. being sick of two illnesses is worse than being sick of only one of them, we chose to sum the values for the new combined dimensions.

As described in Section 3.2.8, the diagnosis-patient matrix is constructed for both
4.1. Advanced Analysis of Patient History

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL/Ontology (LR)</td>
<td>0.8063</td>
<td>0.1535</td>
<td>0.4709</td>
<td>0.7781</td>
</tr>
<tr>
<td>ALL/Ontology (GBT)</td>
<td>0.8326</td>
<td>0.1442</td>
<td>0.4410</td>
<td>0.7880</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>ALL (GBT)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
</tbody>
</table>

Table 4.2: Evaluation results of the ontology-based patient history features in place of the patient history features in the blended dataset (ALL/Ontology)

A training and a test set, containing 80% and 20% of the patients in the clinical dataset respectively. When reducing the dimensionality using SVD, the SVD was performed on the training set only, as the test set needs to be treated as out of sample data to be able to estimate the prediction quality of the resulting prediction models. When using an external knowledge source to reduce the dimensionality of the matrix exclusively, it is irrelevant whether the training and test set are reduced individually or as one big matrix, as samples from the test set do not influence the samples of the training set in the resulting reduced matrix.

The result of the dimensionality reduction using the ICD-9-CM ontology is a reduced diagnosis-patient matrix, in which the column vectors represent the patients and each row represents a group of diagnoses. Each patient vector has 180 dimensions that represent the 180 diagnosis categories described above. Analogously, the resulting patient vectors can again be used as feature vectors for our prediction model. For the purpose of evaluation, we refer to this new model configuration, in which the 1,150 original patient history features are replaced by the 180 ontology-based patient history features, as ALL/Ontology.

Table 4.2 contains a breakdown of the evaluation results of the ALL/Ontology configuration. The improvement in performance over the ALL model configuration is astounding, especially when used to train the logistic regression model. The ROC AUC of the logistic regression model trained on the ALL/Ontology lies at 0.8063, which is well beyond the ROC AUC of 0.7265 that the same model trained on the ALL configuration scored. Surprisingly, it still scores slightly worse in the ROC AUC metric than the logistic regression model trained on the PH model configuration. However, it beats all other logistic regression models across the remaining metrics. In fact, its performance is on par with the gradient boosted trees model trained on the ALL model configuration, which can also be seen in the ROC curve shown in Figure 4.5a. This is an extremely good result for a logistic regression model and underlines the validity of the ontology approach, especially considering that only 180 features are needed to express the patient history.

The gradient boosted trees model trained on the ALL/Ontology configuration also
outperforms the same model trained on the ALL configuration, with an ROC AUC of 0.8326 compared to the ROC AUC of 0.8284 of the latter model. However, it does not beat the gradient boosted trees model trained on the ALL/TFIDF configuration, which has a slight edge over the ALL/Ontology model in the ROC AUC score. In regards to both the Brier score and the log loss, however, the model trained on the ALL/Ontology configuration scores slightly better. Overall, the performance of the gradient boosted trees model trained on the ALL/Ontology is very much in line with the same model trained on the ALL/TFIDF configuration. This can also be seen in the ROC curve in Figure 4.5b and the precision-recall curve in Figure 4.2b, which are nearly identical to corresponding curves for ALL/TFIDF. The best F-measures of both models is also identical, which can be seen in Figure 4.3 of the F-measure curve of the gradient boosted trees model trained on ALL/Ontology.
uses 180 patient history features in contrast to the 1,150 features used with the AL-L/TFIDF configuration. Not only does this use less storage space, the model also converges significantly faster. This basically means that by using the ICD-9-CM ontology as an external knowledge source to reduce the diagnosis-patient matrix it is possible to compress the information contained in the 1,150 tf-idf weighted patient history features of the ALL configuration into a mere 180 ontology-based patient history features.

### 4.1.3 Dimensionality Reduction: Word2vec

The second approach to reducing the dimensionality of the diagnosis-patient matrix we want to evaluate in this chapter uses a combination of word2vec and k-means clustering to create clusters of similar diagnoses that act as the new dimensions of the reduced diagnosis-patient matrix.

Word2vec is a procedure for creating word embeddings from an unlabeled text corpus by creating a shallow, two-layer neuronal network to capture the linguistic context of each word that was proposed by [Mik+13]. A more detailed description of word2vec can be found in Section C.8 of the appendix C.

In its essence, word2vec creates a vector space of word vectors, one for each word occurring in the corpus. The vector space is created in such a way that the word vectors of words that appear in similar linguistic contexts end up being close to one another in the vector space. This allows us to create a sense of semantic distance between words, using measures such as the euclidean distance between points in the vector space or the cosine distance between the word vectors. Using this sense of distance, it is possible to cluster words by proximity using a clustering algorithm, such as k-means clustering. This enables us to form semantic word clusters, who’s centroid vectors represent the common meaning of the words contained in the clusters. These properties of word2vec are interesting, as we can use them to reduce the dimensionality of the diagnosis-patient matrix.

In this approach, the chronological sequence of ICD-9 codes of the diagnoses a single patient received in the problem list dataset can be treated as the words of a sentence that describes the patient history of the corresponding individual. To construct these patient history sentences, the list of diagnoses of each patient in the problem list dataset is sorted chronologically by the date of the diagnosis and a space delimited string of ICD-9 codes is formed from the list. After that, word2vec is trained on the corpus of patient history sentences and word embeddings for our diagnosis pseudo-words are created. The result is a vector space of diagnosis vectors, each representing an ICD-9 code that occurred in the clinical dataset.
These diagnosis vectors allow us to compare diagnoses and calculate the semantic similarity between them. For instance, the list of most similar diagnoses to the ICD-9 code 311, which is the code for major depressive disorder, includes diagnoses such as dysthymic disorder (300.4) and adjustment disorder with depressed mood (309.0), as well as several versions of major depressive affective disorders (296.X). This is astounding, as all diagnoses on this list belong to the group of mood disorder diagnoses, even though word2vec has never been told what a mood disorder is. This validates our assumption that word2vec is able to infer meaning from the unlabeled patient history data, when it is represented as a collection of patient history sentences.

Using k-means clustering, the resulting diagnosis vectors can now be clustered into groups of semantically similar diagnoses. The outcome of this clustering process, however, heavily relies on the parameter \( k \) chosen for the k-means clustering, which specifies, how many clusters should be created. One interesting metric to find a good parameter \( k \) is the explained variance of the resulting clustering, which is defined as the between-cluster variance divided by the total variance. Figure 4.7 shows the distribution of explained variances as a function of the number of clusters used.

As with the rank \( k \) for the truncated SVD before, there are several ways to determine a good \( k \) for the k-means clustering. The two methods used in this section are somewhat related to the ones used with truncated SVD in Section 3.2.9. The first method selects the minimum parameter \( k \) that scores 90% of the variance explained, which relates to the criterion of 90% of retained energy in the reduced diagnosis-patient matrix used for the truncated SVD. In the second method, the curve is examined by eye to find some sort of elbow in the curve, where increasing the number of clusters does not give significantly better results, as the increase in the variance explained starts to level off.
According to the first method, a $k$ of 700 is selected for the 90\% of variance explained criterion, while we chose to use a $k$ of 100 using the second method. The two resulting clusterings, with 700 and 100 clusters respectively, are then used to reduce the dimensionality of the diagnosis-patient matrix.

To reduce the diagnosis-patient matrix, a look-up table is created that stores the corresponding cluster for each ICD-9 code. Afterwards, a new diagnosis-patient matrix is constructed with the diagnosis clusters being the new rows instead of the individual diagnoses. The value of each cell of the new matrix is calculated from the old diagnosis-patient matrix using the look-up table: For each diagnosis a patient has received, the corresponding cluster is looked up and the old value from the diagnosis-patient matrix is added to the current value of the matrix cell. This process, applied to both the 700 clusters and the 100 clusters version, results in two new diagnosis-patient matrices with 187,505 columns and 700 and 100 rows respectively. As with the original diagnosis-patient matrix, the resulting column vectors can be used as feature vectors for the patient history. For the evaluation of the newly reduced matrix, we define two new mode configurations ALL/W2V100 and ALL/W2V700, which include the 100 and 700 word2vec-reduced patient history features instead of the original patient history features respectively.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL/W2V100 (LR)</td>
<td>0.8177</td>
<td>0.1492</td>
<td>0.4599</td>
<td>0.7834</td>
</tr>
<tr>
<td>ALL/W2V700 (LR)</td>
<td>0.8301</td>
<td>0.1442</td>
<td>0.4477</td>
<td>0.7935</td>
</tr>
<tr>
<td>ALL/W2V100 (GB)</td>
<td>0.8350</td>
<td>0.1435</td>
<td>0.4396</td>
<td>0.7901</td>
</tr>
<tr>
<td>ALL/W2V700 (GB)</td>
<td>0.8336</td>
<td>0.1455</td>
<td>0.4476</td>
<td>0.7870</td>
</tr>
<tr>
<td>ALL (LR)</td>
<td>0.7265</td>
<td>0.1774</td>
<td>0.5314</td>
<td>0.7353</td>
</tr>
<tr>
<td>ALL (GBT)</td>
<td>0.8284</td>
<td>0.1462</td>
<td>0.4468</td>
<td>0.7870</td>
</tr>
</tbody>
</table>

**Table 4.3:** Evaluation results of the word2vec-reduced patient history features in place of the patient history features in the blended dataset for both 100 and 700 clusters (ALL/W2V100 and ALL/W2V700)

The evaluation results for both model configurations can be seen in Table 4.3. Both model configurations perform exceptionally well when used to train both the logistic regression model and the gradient boosted trees model. The logistic regression model trained on both configurations outperforms any other configuration we have tested. With an ROC AUC of 0.8177 for the ALL/W2V100 configuration and an ROC AUC of 0.8350 for the ALL/W2V700 compared to an ROC AUC of 0.8036 for the logistic regression model trained on the 1,150 patient history features only, the word2vec-reduced patient history features improve the prediction performance of the logistic regression model significantly. The detailed results of the logistic regression model trained on the two model configurations can be found in Section A.9 in Appendix A.
The gradient boosted trees model also sees improvements from the word2vec-reduced patient history features. Surprisingly, the 100 features version of the word2vec-reduced patient history performs better than the 700 features version when used to train the gradient boosted trees model. This could be an indicator that we chose to cluster the diagnoses into too many clusters with the 700 cluster version. It stands to reason that the results could be improved even more by finding a middle ground between the 100 and 700 clusters version. However, finding the best number of clusters to use is beyond the scope of this thesis and should be looked into as part of future research.

Overall the gradient boosted trees model may not gain performance by the same margin as the logistic regression model when trained on the word2vec-reduced patient history features, but enough for the gradient boosted trees model trained on the 100 word2vec-reduced features to take the top spot among evaluated models. With an ROC AUC of 0.8350, it outperforms all other models, including the models trained on the ALL/Tfidf and the ALL/Ontology model configurations.
The ROC curve of the gradient boosted trees model trained on both configurations can be seen in Figure 4.8. As expected, the ROC curve of the model trained on the ALL/W2V100 model configuration is slightly more bulged, which explains the slightly larger ROC AUC. The precision-recall curves, shown in Figure 4.9, display almost no difference between the ALL/W2V100 and ALL/W2V700 versions.

![F-Measure curve for 100 clusters (GB)](image1)

![F-Measure curve for 700 clusters (GB)](image2)

**Figure 4.10:** F-measure curves for the gradient boosted trees model trained on both the 100 clusters and the 700 clusters

There is, in fact, a slight difference between the model configurations when it comes to the F-measure. The two F-measure plots corresponding to the ALL/W2V100 and ALL/W2V700 model configuration, depicted in Figure 4.10, show that the model trained on the 700 features scores slightly higher than the 100 features version in regards to the F-measure at the optimal threshold. However, with an F-measure of 0.637 at the 29.1% threshold compared to an F-measure of 0.633 at the 29.5% threshold, the model trained on the ALL/W2V700 configuration should only slightly outperform the model trained on the ALL/W2V100 configuration in a real-world scenario. This is an impressive result, especially considering that only 100 patient history features are used to train the model with the ALL/W2V100 model configuration, the smallest number of patient history features among all model configurations.

Ultimately, the evaluation shows that word2vec is very good at extracting meaning from unlabeled data. What surprised us most was the accurateness, with which word2vec is able to determine semantic similarity between diagnoses, especially considering that it has no understanding of what a diagnosis is and what any of the ICD-9 codes mean. The performance of the evaluated models shows that word2vec is able to group diagnoses in a way that inherently makes sense, seemingly on par or even better than we were able to do with the help of domain-specific knowledge in form of the ICD-9-CM ontology. This is highly interesting, as the approaches differ on a fundamental basis: one using logic and reason to project meaning onto the data, while the other seeks to extract meaning from the inherent patterns contained within it.
4.2 Discussion of Main Results

In this thesis, several hypotheses and assumptions regarding the factors that cause or are correlated with the development of mood disorders were tested and evaluated.

The main result of our work lies in the confirmation that the medical history of a patient is heavily correlated with their risk of developing a mood disorder. We have shown that a detailed patient history alone can be used to predict the mood disorder risk with reasonable accuracy.

Additionally, personal information on the patients that can be obtained from their medical records, including their age and gender, ethnicity, marital and employment status, and whether they are currently holding health insurance, has been found to be also strongly correlated with the development of mood disorders. This could be partly explained by the fact that different demographic groups are varyingly susceptible to mood disorders in general. However, the marital and employment status, as well as whether they are insured, are also good indicators for the patient’s socioeconomic status, which in turn seems to have a significant impact on the development of mood disorders. The exact effects of the individual parts of the personal information on the development of mood disorders, however, need to be investigated to a greater extent as part of further research.

The main hypothesis that we wanted to test using our blending approach was that the personal environment of an individual has a strong influence on their risk of developing a mood disorder. We used the patient’s neighborhood, in the form of the ZIP Code they live in, as a way to quantify their personal environment. We could show that the mood disorder risk of patients is slightly correlated with the economic and demographic characteristics of the ZIP Code they live in. We also found a weak correlation between the estimated crime statistics in the ZIP Code of residence and the mood disorder risk, however, the findings could well be a result of the way the crime statistics were estimated and thus require further investigation.

One explanation for the relatively weak correlation between the neighborhood data and the mood disorder risk could be that we only took the current place of residence into account, as no information on the patient’s past places of residence were available. Current changes to the personal environment, like moving to another neighborhood or city, are hard to consider in this regard, as the change in environment could have not had enough time to influence the mood disorder risk, whether it be for the better or the worse. Additionally, the neighborhood an individual lives
4.2. Discussion of Main Results

in does not tell the whole story about their personal environment. Important factors, such as relationships with friends and family, cannot be inferred from the place of residence.

Based on our results we can neither disregard the original hypothesis that the personal environment is a factor for the development of mood disorders as invalid nor validate it beyond reasonable doubt. However, we have found enough evidence to justify further research into this topic.

As part of this thesis, we also evaluated the correlation between various economic and demographic features and the mood disorder prevalence in a given ZIP Code. We found that several of the tested features are quite strongly correlated with the mood disorder prevalence. Among them are both the median and average household income and the fraction of individuals working in management positions, which were all found to be negatively correlated with the mood disorder prevalence. However, we were not able to show any significant correlation for the fraction of individuals holding health insurance, the employment rate and the poverty rate of the ZIP Code with the mood disorder prevalence.

In summary, we believe that our work is a contribution to the field of mood disorder diagnosis and prediction that has the potential to further the acceptance of machine learning based methods as auxiliary diagnostic tools. Additionally, we have found evidence that strongly suggests that further research into automated risk prediction based on the patient’s medical history is needed, as we believe that there is great potential for improvements. Further, we believe that our work can also be of use in the study of the causes and risk factors of mood disorders.
Chapter 5

Conclusions and Future Work

Throughout this thesis we worked towards the goal of using data blending and machine learning techniques in order to predict the individualized risk for the development of mood disorders. Our data blending approach was centered around real-world clinical data on roughly a quarter million patients that we were able to obtain from Geisinger Health Systems.

In addition to the clinical data, we acquired additional datasets from multiple domains of data to form a blended dataset used to conduct multi-disciplinary data analysis. This additional data included economic, demographic, and geographic data acquired from the U.S. Census Bureau, as well as crime data from several major U.S. cities. As the clinical data included the ZIP Codes of residence of the patients, the additional data was acquired on a ZIP Code level, so it could be cross-referenced with patients in the clinical data.

The purpose of this additional data was to enrich the clinical data with information on the personal environment of the individual in form of the neighborhood they live in and use this additional information to identify potential mood disorder risk factors stemming from the personal environment. Further, we wanted to examine the connection between these factors and the actual mood disorder risk of the patients.

After all datasets were acquired, the data was cleaned, imputed, and transformed in preparation for being used as features in the blended dataset. The data acquired from the U.S. Census was combined and transformed into the census community features. The clinical data was used to calculate mood disorder prevalence for the ZIP Codes that occurred in the clinical data. Additionally, the data on crimes in major U.S. cities was used to estimate the crimes for the ZIP Codes found in the clinical data. A major part of the work was concerned with making the medical history data provided by Geisinger Health Systems usable by the prediction model. For this, we created a diagnosis-patient matrix and afterwards reduced it using truncated SVD to use the column vectors as feature vectors.
For the evaluation, we trained both a logistic regression model and a gradient boosted trees model on several different configurations of the blended dataset to examine the correlation between the different feature sets and the mood disorder risk. The results showed that the patient history features and the patient demographics features were strongly correlated with the mood disorder risk, however, the additional data in form of the census community features and the crime indicator feature was found to have only a slight impact on the mood disorder risk.

Afterwards, we focused on more advanced analyses of the patient history, including one method of weighting the diagnosis-patient matrix differently and two new approaches to the dimensionality reduction of the diagnosis-patient matrix. Weighting the diagnosis-patient matrix using tf-idf resulted in improved performance. Using an external knowledge source in the form of the ICD-9-CM ontology to reduce the dimensionality resulted in similarly increased performance as with the tf-idf weighted matrix, however, the number of needed features for the patient history was reduced to 180 features. The best performing model was achieved by using word2vec to cluster diagnoses by their semantic similarity and using these clusters to reduce the diagnosis-patient matrix. Not only resulted this in the overall best performing model, we could also reduce the number of features needed to represent the patient history to 100.

The main takeaway from this is that the analysis of the patient history using machine learning methods can have a significant impact not only on the diagnosis and prevention of mood disorders, but on the field of diagnostics as a whole. We believe that a great deal of research and effort has to be put into even more advanced analyses of the patient history to achieve even greater improvements in the prediction performance.

Throughout this thesis, we have followed one approach to the problem of predicting the individualized risk for a group of related disorders, however, it is also conceivable to apply the presented approach to a different set of diseases. In fact, the approach could be applied to a wide array of illnesses, such as cardiovascular diseases and metabolic disorders. On the other hand, additional approaches to predicting the individualized risk of developing certain diseases are also conceivable and should be taken into consideration for future research.

One interesting approach that could be looked at in the future is using recurrent neural networks to predict the most likely next diagnosis from a given patient history. In the past, statistic methods like Markov models have been used to achieve something similar, however, recent improvements in neural networks might lead to significant improvements over the older statistical models \cite{SB93, Sri15}. Being able to predict the most likely next diagnosis for a given patient could naturally be applied to predicting the risk for all possible diseases.
Admittedly, we were not able to show significant improvements in the prediction performance using our blending strategy, however, we firmly believe that blending of clinical and public data can lead to more robust prediction models and a better understanding of the data at hand. Using data blending as an exploratory tool to examine correlations and connections between the clinical data and our additional datasets helped us gain great insight into the field of mood disorders and the factors that are related to the development of this group of illnesses. Future research should further evaluate, which possible additional data sources could be used to extend the blending approach presented in this thesis.

In the future, prediction models that are able to calculate the patient’s risk for certain diseases and illnesses will be used to an even greater extent to help doctors and other medical personnel with the diagnosis of unidentified ailments in patients. Using a patient’s predicted risks for certain diseases, screening and testing could be guided towards the most likely diagnoses for the patient. This will lead to faster diagnostics in general and a lower chance for misdiagnosis, which in turn helps the patient get the correct treatment faster and has the potential for an overall decrease in medical costs.

Further, these risk predictions could be used to identify high-risk individuals that are likely to fall ill with a certain disease in the future. The preemptive diagnosis of serious illnesses could help with the mitigation of symptoms and even prevention of the affliction in the first place. Preventing illnesses before they can manifest is a desirable goal, as staying healthy is generally easier, faster and more cost-effective than recovering from sickness.

Both preemptive diagnosis and diagnostics enhanced by machine learning methods in general have the potential to significantly improve the quality of care and consequently patient outcomes. Ultimately, further research in this area has the opportunity to save uncountable lives and improve the quality of life of even more.

However, being able to predict the individualized risk for certain diseases also has some moral and ethical implications. As promising as machine learning guided diagnostic sounds, the same methods and approaches could be used in ways that negatively impact the patients. For instance, insurance companies could use high-quality risk predictions to discriminate against potentially unprofitable applicants. Individuals that are predicted to be at high risk of developing a serious illness in the near future would have to pay horrendous insurance premiums, if they could find insurance at all. Even if this discrimination based on predicted risk could be prevented by regulatory means, insurances could still use the prediction results to adjust their coverage of certain preemptive treatments and medication that appear unjustifiable based on the risk predictions.
When used by employers, the aforementioned prediction models could be used to assess the health of their employees and to a greater extend also job applicants. Employees that are found to be at higher risk for certain diseases could be either supported by their employer through countermeasures and preemptive care opportunities or terminated from their job in order to prevent the costs of future lost work productivity, depending on the moral and ethical alignment of the company. Health predictions could also play a huge factor in the employment process: job applicants that are predicted to be at high risk of falling ill could face major problems getting hired, as companies try to mitigate their risk of hiring employees, who’s long-term benefit to the company could fall short due to health problems.

A general ethical and partly philosophical issue that arises from using machine learning systems in a clinical context is accountability. If a prediction model can automatically diagnose patients and does so with better accuracy than a medical doctor, it is disputable, whether the practitioner following the guidance of the prediction model can and should be held responsible for misdiagnoses and medical mistreatment. Issues like this could heavily limit the use of automated prediction systems in real-world clinical applications.

All that said, we ultimately believe that the potential benefits of more accurate prediction models used to assist the diagnosis and prognosis of diseases outweigh these concerns manyfold. However, we agree that the legal and ethical concerns need to be addressed in order for these machine learning methods to gain universal acceptance.
Appendix A

Evaluation

In Section 3.4 of Chapter 3 the whole evaluation process of the various model configurations is described. Additional results that are not part of the stated section can be seen in the following sections.

A.1 MDP

(a) ROC curve of the logistic regression model on the MDP with model imputation configuration

(b) ROC curve of the logistic regression model on the MDP with average imputation configuration

FIGURE A.1: ROC curves for logistic regression model on both the MDP with model imputation and MDP with average imputation configuration
Appendix A. Evaluation

(A) Precision-recall curve of the logistic regression model trained on the MDP with model imputation configuration

(B) Precision-recall curve of the logistic regression model on the MDP with average imputation configuration

Figure A.2: Precision-recall curves for logistic regression model on both the MDP with model imputation and MDP with average imputation configuration

A.2 ALL and ALL/SCREE

(A) ROC curve of the linear regression model on the full ALL/SCREE configuration

(B) ROC curve of the linear regression model on the full ALL configuration

Figure A.3: Precision-recall curves for logistic regression model on both the ALL/SCREE and ALL configuration
A.3 Importance of Features

(a) Precision-recall curve of the logistic regression model trained on the full ALL/SCREE configuration

(b) Precision-recall curve of the logistic regression model trained on the full ALL configuration

**Figure A.4: Precision-recall curves for logistic regression model on both the ALL/SCREE and ALL configuration**

(a) F-measure curve of the logistic regression model trained on the full ALL/SCREE configuration

(b) F-measure curve of the logistic regression model trained on the full ALL configuration

**Figure A.5: F-measure curves for logistic regression model on both the ALL/SCREE and ALL configuration**

A.3 Importance of Features

A.4 ALL - PH

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.6814</td>
<td>0.1871</td>
<td>0.5549</td>
<td>0.7208</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.7407</td>
<td>0.1734</td>
<td>0.5128</td>
<td>0.7387</td>
</tr>
</tbody>
</table>

**Table A.1: Breakdown of Evaluation Results of ALL-PH Configuration**
Appendix A. Evaluation

Figure A.6: ROC curves for the logistic regression model and the gradient boosted trees model on the ALL-PH configuration

(a) ROC curve of the logistic regression model on the ALL-PH configuration

(b) ROC curve of the gradient boosted trees model on the ALL-PH configuration

Figure A.7: Precision-recall curves for the logistic regression model and the gradient boosted trees model on the ALL-PH configuration

(a) Precision-recall curve of the logistic regression model trained on the ALL-PH configuration

(b) Precision-recall curve of the gradient boosted trees model trained on the ALL-PH configuration

Figure A.8: F-measure curves for the logistic regression model and the gradient boosted trees model on the ALL-PH configuration

(a) F-measure curve of the logistic regression model trained on the ALL-PH configuration

(b) F-measure curve of the gradient boosted trees model trained on the ALL-PH configuration
A.5  ALL - MDP

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% energy</td>
<td>Logistic Regression</td>
<td>0.7239</td>
<td>0.1787</td>
<td>0.5347</td>
</tr>
<tr>
<td></td>
<td>Gradient Boosting</td>
<td>0.8267</td>
<td>0.1468</td>
<td>0.4487</td>
</tr>
</tbody>
</table>

**Table A.2:** Breakdown of Evaluation Results of ALL-MDP Configuration

(a) ROC curve of the logistic regression model on the ALL-MDP configuration

(b) ROC curve of the gradient boosted trees model on the ALL-MDP configuration

**Figure A.9:** ROC curves for the logistic regression model and the gradient boosted trees model on the ALL-MDP configuration

(a) Precision-recall curve of the logistic regression model trained on the ALL-MDP configuration

(b) Precision-recall curve of the gradient boosted trees model trained on the ALL-MDP configuration

**Figure A.10:** Precision-recall curves for the logistic regression model and the gradient boosted trees model on the ALL-MDP configuration
Appendix A. Evaluation

A.6 ALL - CI

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.7264</td>
<td>0.1775</td>
<td>0.5314</td>
<td>0.7350</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.8285</td>
<td>0.1461</td>
<td>0.4467</td>
<td>0.7866</td>
</tr>
</tbody>
</table>

Table A.3: Breakdown of Evaluation Results of ALL-CI Configuration

(a) ROC curve of the logistic regression model on the ALL-CI configuration
(b) ROC curve of the gradient boosted trees model on the ALL-CI configuration

Figure A.12: ROC curves for the logistic regression model and the gradient boosted trees model on the ALL-CI configuration
A.7 ALL - PD

![Precision-recall curve for logistic regression model on ALL-CI configuration](image1)

![Precision-recall curve for gradient boosted trees model on ALL-CI configuration](image2)

(a) Precision-recall curve of the logistic regression model on the ALL-CI configuration
(b) Precision-recall curve of the gradient boosted trees model on the ALL-CI configuration

**Figure A.13:** Precision-recall curves for the logistic regression model and the gradient boosted trees model on the ALL-CI configuration

![F-Measure curve for logistic regression model trained on ALL-CI configuration](image3)

![F-Measure curve for gradient boosted trees model trained on ALL-CI configuration](image4)

(a) Precision-recall curve of the logistic regression model trained on the ALL-CI configuration
(b) Precision-recall curve of the gradient boosted trees model trained on the ALL-CI configuration

**Figure A.14:** F-Measure curves for the logistic regression model and the gradient boosted trees model on the ALL-CI configuration

### A.7 ALL - PD

<table>
<thead>
<tr>
<th>Variations</th>
<th>ROC AUC</th>
<th>Brier Score</th>
<th>Log loss</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.6988</td>
<td>0.1872</td>
<td>0.5578</td>
<td>0.7200</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.8062</td>
<td>0.1533</td>
<td>0.4702</td>
<td>0.7775</td>
</tr>
</tbody>
</table>

**Table A.4:** Breakdown of Evaluation Results of ALL-PD Configuration
Appendix A. Evaluation

FIGURE A.15: ROC curves for the logistic regression model and the gradient boosted trees model on the ALL-PD configuration

(a) ROC curve of the logistic regression model on the ALL-PD configuration
(b) ROC curve of the gradient boosted trees model on the ALL-PD configuration

FIGURE A.16: Precision-recall curves for the logistic regression model and the gradient boosted trees model on the ALL-PD configuration

(a) Precision-recall curve of the logistic regression model trained on the ALL-PD configuration
(b) Precision-recall curve of the gradient boosted trees model trained on the ALL-PD configuration

FIGURE A.17: F-measure curves for the logistic regression model and the gradient boosted trees model on the ALL-PD configuration

(a) Precision-recall curve of the linear regression model on the ALL-PD configuration
(b) Precision-recall curve of the gradient boosted trees model on the ALL-PD configuration
A.8  ALL - CC

Variations  ROC AUC  Brier Score  Log loss  Accuracy

90% energy
Logistic Regression  0.7261  0.1776  0.5316  0.7346
Gradient Boosting  0.8289  0.1459  0.4463  0.7866

Table A.5: Breakdown of Evaluation Results of ALL-CC Configuration

(a) ROC curve of the gradient boosted trees model on the ALL-CC configuration
(b) ROC curve of the gradient boosted trees model on the ALL-CC configuration

Figure A.18: ROC curves for the logistic regression model and the gradient boosted trees model on the ALL-CC configuration

(a) Precision-recall curve of the gradient boosted trees model on the ALL-CC configuration
(b) Precision-recall curve of the gradient boosted trees model trained on the ALL-CC configuration

Figure A.19: Precision-Recall curves for the logistic regression model and the gradient boosted trees model on the ALL-CC configuration
Appendix A. Evaluation

(a) F-measure curve of the logistic regression model on the ALL-CC configuration

(b) F-measure curve of the gradient boosted trees model on the ALL-CC configuration

FIGURE A.20: F-measure curves for the logistic regression model and the gradient boosted trees model on the ALL-CC configuration

A.9 ALL/W2V100 and ALL/W2V700

(a) ROC curve of the logistic regression model on the ALL/W2V100 configuration

(b) ROC curve of the logistic regression model on the ALL/W2V700 configuration

FIGURE A.21: ROC curves for logistic regression model on both the ALL/W2V100 and ALL/W2V700 configuration
(a) Precision-recall curve of the logistic regression model on the ALL/W2V100 configuration

(b) Precision-recall curve of the logistic regression model on the ALL/W2V700 configuration

**Figure A.22:** Precision-recall curves for logistic regression model on both the ALL/W2V100 and ALL/W2V700 configuration

(a) F-Measure curve of the gradient boosted trees model on the ALL/W2V100 configuration

(b) F-Measure curve of the gradient boosted trees model on the ALL/W2V700 configuration

**Figure A.23:** F-measure curves for logistic regression model on both the ALL/W2V100 and ALL/W2V700 configuration
A.10 ALL/TFIDF

Figure A.24: F-measure for linear regression model on the ALL/TFIDF configuration

A.11 ALL/Ontology

Figure A.25: F-measure for linear regression model on the ALL/Ontology configuration
Appendix B

Medical Background

In the scope of this thesis, we got access to clinical data provided by Geisinger Health Systems. This data includes patients who have suffered from a mood disorder in the past. The main goal of this thesis is to train a prediction model that can be used to predict the mood disorder risk of an individual. To grasp the topics addressed in this thesis, it is important to have a fundamental understanding of the fields of mood disorders and clinical studies in general. This chapter seeks to give a brief overview of the knowledge needed in this regard.

Mood disorders are mental disorders characterized by periods of depression, sometimes alternating with periods of elevated mood [Far16]. Disorders, such as major depressive disorder, bipolar disorder and dysthymic disorder are all encompassed under the category of mood disorders. To get a feeling for how important the treatment of mood disorders is, the following facts describe the percentage of people in the U.S. who suffer from mood disorders every day:

- 2.6% of all adults live with a bipolar disorder [Nat16c]
- 6.7% of all adults had at least one major depressive episode in 2015 [Nat16f]
- 18.1% of all adults have experienced an anxiety disorder such as posttraumatic stress disorder, obsessive-compulsive disorder and specific phobias [Nat16b]

This short breakdown underlines the importance of research in the field of mood disorders, as millions are suffering from the consequences of these horrible diseases.
Appendix B. Medical Background

B.1 International Classification of Diseases (ICD)

To differentiate diseases and their severity the International Classification of Diseases (ICD) is used as the standard diagnostic tool for “epidemiology, health management and clinical purposes” [Wor92]. The ICD includes analyses, such as general health situation of populations and countries and detailed information on possible symptoms [Wor92]. All known diseases are described using unique ICD codes, which are revised periodically to keep them up to date. The medical data used in this thesis includes ICD codes from the ICD-9-CM standard, called ICD-9 codes for short throughout this thesis, where the number 9 describes the revision of the standard and the CM suffix, which stands for clinical modification, denotes an optimized version of the standard by the U.S. National Center for Health Statistics [Nat16a]. ICD is used in more than 100 countries and has been translated into 46 languages. The current version of the standard is revision 10, while the next version, revision 11, is already underway and will be released in 2018 [Wor92].

B.2 Major Depressive Disorder

Major depressive disorder, which is expressed by ICD-9 code 311, is a psychological disorder that involves a distinct change in mood, expressed as sadness or irritability, and includes psychophysiological changes, such as “disturbances in sleep, appetite, or sexual desire; constipation; loss of the ability to experience pleasure in work or with friends; crying; suicidal thoughts; and slowing of speech and action” [RG08]. The treatment of major depressive disorder is increasing, which is encouraging, but inappropriate treatment is a problem, as major depressive disorder is a common disorder, which is widely distributed in the population [Kes+03].

Among mental disorders, major depressive disorder is the most common one in the U.S with more than 5.7% of affected people. [Nat16f]

B.3 Episodic Mood Disorders

The class of episodic mood disorders, which is expressed by ICD-9 code 296, contains disorders, including several types of episodic bipolar and manic disorders. Both bipolar and manic disorders are denoted by different ICD-9 codes. ICD-9 code 296.0 corresponds to bipolar disorder of type one with a single manic episode, whereas 296.1 corresponds to manic disorder with recurrent episodes [Alk15].

A bipolar disorder is a complex disorder caused by both genetic and environmental factors, in which the fundamental disturbance in mood is ranging from “extreme
elation, or mania, to severe depression usually accompanied by disturbances in thinking and behavior” [CJ99] [AHS12].

In 2005, about 2.6% of the U.S. population was affected by a bipolar disorder [Nat16c]. Further, more than 6% of the patients affected are known to commit suicide within a period of twenty years after their diagnosis [AHS12].

B.4 Dysthymic Disorder

Dysthymic disorder, which is expressed by ICD-9 code 300.4, is characterized by a depressed mood for a period of two or more years (more days with depressed mood) without clear criteria for a major depression. Dysthymic disorder differs from major depressive disorder, as the symptoms for a dysthymic disorder are “poor appetite or overeating, insomnia or hypersomnia, low energy or fatigue, low self-esteem, poor concentration or difficulty making decisions, and hopelessness” [Dun98].

In 2005, about 1.7% of the U.S. population was affected by a dysthymic disorder [Nat16e].

B.5 Treatments

Mood disorders can be treated successfully with antidepressant and mood stabilizing medications, psychotherapy, family therapy or other therapies [Nat16d]. Further, a combined treatment including medication, individual psychotherapy and family therapy is recommend to treat complex chronic mood disorders [MK96]. Although advances are recognizable in the area of psycho-pharmacological medication, the treatment of major depression and other mood issues challenges the mental health providers [QA13]. However, conventional treatment methods often do not suffice for many patients, which leads them to call upon other treatment options, such as complementary and alternative medicine. For example, these include homeopathy, acupuncture and herbal treatments [QA13].

Bright light therapy, which can lead to significant reduction of depression symptoms [PM03], and electroconvulsive therapy (ECT) are also possible treatment for mood issues. The ECT therapy is an electroshock therapy for patients with depressive episodes, especially in case of psychotic symptoms or potential suicide. However, this treatment has side affects, such as short-term memory loss and nausea [Web16].
B.5.1 Current Research

Mood disorders continues to be a topic of current research, as there is still a high amount of uncertainty about how mood disorders form and how they can be efficiently treated.

Several Studies have revealed that there is a relationship between HIV infection and risk of depressive disorder. The frequency of HIV-positive subjects is two times higher than in HIV-negative subjects. However, the depression rates do not appear to be related to the sexual orientation or disease stage of infected individuals [CR01].

Additionally, a systematic psychiatric evaluation revealed that a high amount of people who suffered from multiple sclerosis, had to deal with depressions in the past [Jof+87]. This leads to the conclusion that people who suffered from a major disease in the past, have a higher risk to get mood disturbance in the future.

Further, a good amount of research is spent on the effects of neurochemistry on the development of mood disorders, such as the correlation between the hippocampal complex and mood disorders [Wos+14] or endocrine disturbances in mood disorder patients [BA99].
Appendix C

Software Tools and Frameworks

The following sections highlight the software tools and frameworks used in the development of the mood disorder risk model. The tools used in this thesis were chosen based on our experience with various programming languages and data processing frameworks in addition to best practices and generally excepted standard tooling.

C.1 Hadoop

Hadoop is an open source framework for distributed computing that is maintained by the Apache Software Foundation. Hadoop provides a distributed file system called HDFS and the ability to run distributed calculations using the MapReduce paradigm \cite{Whi12}. The main purpose of HDFS is to store large amounts of data distributed across a large number of individual machines (nodes) and have them readily available for processing \cite{Shv+10}. In order to process complex tasks with

![Diagram of MapReduce process]

\textbf{Figure C.1:} Example of how a simple word count would be calculated using MapReduce
very large datasets MapReduce is used to transform key/value pairs into intermediate key/value pairs (map function). The values of those intermediate pairs are later merged by the associated intermediate key (reduce function) [DG08].

A very famous example to describe MapReduce in an easy way is the word count example. The process of word count with map reduce is described in Figure C.1.

Additionally, services like Pig [C.2] or Hive (a distributed data warehouse) are built on top of the Hadoop ecosystem for a higher-level abstraction. For this thesis the Cloudera Virtual Machine in version 5.5 including a running Hadoop cluster is used[1]. The virtual machine is used to run pig scripts for the transformation and preparation of data.

C.2 Pig

Pig is a scripting platform for processing and analyzing large datasets. With Pig, complex MapReduce transformations can be written using the scripting language Pig Latin, to process the different Extract-Transform-Load (ETL) tasks [EZ09]. An ETL task consists of the three stages, which are extract, transform and load. Data is extracted from one or multiple data sources and transformed into a suitable format or preprocessed for the purpose of data analysis. Afterwards, the data is loaded into a structured format like a database or a CSV file [GR09]. Pig transforms the Pig Latin into multiple Map Reduce jobs based on Java [Ols+08].

It should be noted how simple it is to write a Pig script for a complex operation such as a joining two independent datasets.

```
populationData = LOAD './populationData.csv' USING PigStorage('','') AS
                   (ZIPCODE:chararray, TOTAL:float)

averageIncomeData = LOAD './averageIncome.csv' USING PigStorage('','')
                   AS (ZIPCODE:chararray, AVERAGE_INCOME:float);

joinedData = JOIN populationData by $0, averageIncomeData by $0;

STORE joinedData INTO './joinedIncomeAndAverage.csv' USING
                   PigStorage('','');
```

Listing C.1: Example of a join of two CSV files in Pig. Both CSV files are loaded and parsed using PigStorage, joined on the ZIPCODE field, and stored in the output CSV file.

[1] https://www.cloudera.com/documentation/enterprise/5-5-x/topics/cloudera_quickstart_vm.html
C.3. Node.js

The Listing C.1 shows how join two datasets and store the result. In this example a dataset containing containing the ZIP code and the population count is joined with dataset containing the ZIP Code and the average income. As both dataset have the ZIP Code feature in common, it is possible to join the datasets on the ZIP Code. Afterwards the result is stored in HDFS.

Even though the script at hand looks simple, the calculation needed to solve the task can be rather complex.

In the thesis Pig is used for most of the data preparation tasks, described in Section 3.2, such as data manipulation, transformation and normalization. It is also used for the analysis of data quality, described in Section 2.2.1.

C.3 Node.js

As described in Section C.2, the majority of ETL tasks, especially those dealing with great volumes of data and comparatively simple transformations, are best tackled by big data tools such as Pig. For smaller tasks and tasks that are less compute-intensive or require more complex operations, a general-purpose programming language can be used. For the work at hand, we chose to use JavaScript combined with Node.js, a JavaScript runtime environment that is built on top of the V8 JavaScript engine used by Google Chrome [TV10].

Node.js makes use of an event-driven, non-blocking I/O model, which means that asynchronous I/O operations do not block the whole execution of the process, but are only dealt with once the corresponding callback is called [Nod16]. Another useful property of Node.js is its concept of readable and writable streams. These streams allow for asynchronous stream processing, where data flows from a readable stream, such as an input file or the standard input (stdin), through a number of pipes to one or multiple writable streams, such as an output file or the standard output (stdout). This is especially useful when dealing with amounts of data that don’t fit into the available memory of the process, as input files can be gradually read and processed instead of reading the whole file into memory and processing it in bulk [Tai13].

Another benefit is the ability to rely on Node.js’ package manager and ecosystem, called npm, which “is the largest ecosystem of open source libraries in the world” according to Node.js’ official website [Nod16]. Serving as both the package manager and the online package registry, npm encompasses more than a quarter of a million reusable software libraries [npm16], including packages such as csv[2] that can be used to read, transform and write CSV files using Node.js streams. Listing C.2 shows the usage of the csv package in combination with Node.js streams.

https://www.npmjs.com/package/csv
var fs = require('fs');
var path = require('path');
var csv = require('csv');

var parser = csv.parse();

var file = path.join(__dirname, process.argv[2]);

fs.createReadStream(file)
  .pipe(parser)
  .pipe(csv.transform(function(record){
    if(record[3] == "Female")
      return [record[0], record[record.length-1]];
  })))
  .pipe(csv.stringify())
  .pipe(process.stdout);

LISTING C.2: Example of stream processing a CSV file in Node.js. The CSV file is parsed on a record-by-record basis using the csv.parse() streaming parser, the records are transformed using csv.transform() and finally written to the stdout stream after being turned into CSV records again by csv.stringify().

Node.js is mainly used for the identification of anonymized addresses of crime scenes using the Census Geocoding API and the Google Geocoding API, as described in Section 3.2.3.

C.4 Python

For the implementation of the single prediction models described in Section D.1 the programming language Python is used. Python is a powerful programming language that contains efficient high-level data structures and an effective approach to object-oriented programming [Ros95]. With its simple syntax and dynamic typing, Python is often used for scripting and rapid prototyping on many platforms. There are also a lot of third party modules especially in the area of data science, such as scikit-learn (C.5) and NumPy (C.6) that makes Python so powerful.

C.5 NumPy

Complex mathematical operations like the multiplication of matrices or a singular value decomposition on matrix (SVD) are needed in the presented work. Therefore
the Python library NumPy is used, that provides the already addressed methods and other high level mathematical functions. The ndarray data structure for n-dimensional arrays is the core functionality of NumPy. The ndarray is in contrast to Python’s built-in list data structure, a homogeneously typed array (all elements must be of the same type) [Oli06].

### C.6 scikit-learn

Scikit-learn is a powerful machine learning library for Python that provides state-of-the-art implementations of many well known machine learning algorithms for supervised and unsupervised problems. The library focuses on the easy integration of machine learning concepts in a general-purpose high-level language. As the library is distributed under the BSD license, it can be used for academic and commercial work [Ped+11].

```python
import numpy as np
from sklearn.cross_validation import train_test_split

test_split = 0.2

loaded = np.loadtxt(open('.../all_standardized_crime_pruned.csv', 'rb'), delimiter=',').astype(np.float32)

X = loaded[:, 4:]
y = loaded[:, 1]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_split, random_state=9873453)
```

Listing C.3: Plotting Cross-Validated Predictions

The example in Listing C.3 shows very plainly that the usage of scikit-learn is very simple. In this example a dataset is loaded and splitted into 80% training data and 20% test data.

### C.7 RDF and OWL

The Resource Description Framework (RDF) is a directed, labeled graph data format for representing information in the Web. It was originally designed by the W3C as a standard for describing meta data, but now RDF is a core feature of the semantic web. In the RDF model every statement consists of a subject, a predicate
and an object, whereas a resource (subject) can be described with another resource or a literal (object) and with a third resource (predicate) resulting in a triple [BG04].

```xml
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:si="http://www.w3schools.com/rdf/">
  <rdf:Description rdf:about="http://www.w3schools.com">
    <si:title>W3Schools</si:title>
    <si:author>Jan Egil Refsnes</si:author>
  </rdf:Description>
</rdf:RDF>
```

**Listing C.4: RDF Example**

To express queries across diverse data sources, the query language SPARQL (SPARQL Protocol And RDF Query Language) can be used. “SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs” [PS08].

```sparql
PREFIX abc: <http://example.com/exampleOntology#>

SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
  abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
  abc:isInContinent abc:Africa .
}
```

**Listing C.5: SPARQL Example**

Based on RDF the Web Ontology Language (OWL), a semantic markup language, was designed for creating and sharing ontologies. OWL is more powerful than RDF, as it possesses more features for describing the relation between different resources [Bec+04].

In the thesis RDF and OWL are used for shrinking the dimensional spaces of ICD-9 codes, a RDF ontology is used to put the single codes into subordinated groups. This process is described in detail in Section 4.1.2.
Another approach for calculating correlations between diseases is to consider every disease of the patient history as a word and create a huge vector space.

Word2vec is a procedure for creating word embeddings by creating a shallow, two-dimensional neuronal network to reconstruct the context of words. Word2vec can be used with the continuous bag of words model (CBOW) or the skip-gram model, where CBOW is faster, but achieves less accuracy, and skip-gram performs slower while leading to more accurate results [Mik+13]. The input of word2vec is a bag of words that is transformed into vector space, where every row describes a unique word of the bag. This vector space can be used to perform a k-means clustering to calculate clusters with words that have a similar context [GL14].

Word2vec is used in Section 4.1.3 to reduce the dimensionality of the patient history to calculate new clusters.
Appendix D

Machine Learning Background

The following chapter describes the fundamental background of the machine learning methodologies used in this thesis, including all prediction models used and also the evaluation methodology.

D.1 Prediction Models

One of the big challenges arising from the task of predicting the individualized risk for the development of mood disorders is using the right predictive models to calculate a prediction based on input data. Predictive models are mathematical models that use statistics to predict outcomes, such as predictions for future or otherwise unknown events, and are a major part of the field of machine learning [Gei93]. In this thesis, the predictive models, among other things, are used to predict risk, expressed as a probability of the occurrence of an undesirable event. A base example for a prediction model is a sales forecast for products based on the sales statistics of the last quarter.

D.1.1 Linear Regression

As stated in [Jam+13, p. 59], linear regression, a variation of regression analysis used to model a linear relationship between one or multiple independent variables and one dependent variable, is a simple example for a predictive model using a supervised learning approach. When dealing with only a single independent variable, also known as the predictor variable, the model is called simple linear regression. Simple linear regression seeks to model a linear relationship between the single independent variable $X$ and the dependent variable $Y$ that can be expresses as

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

(D.1)
where $\beta_0$ is equivalent to the intercept and $\beta_1$ to the slope in terms of the linear model. The $\epsilon$ represents the random error term that reflects the model’s inaccuracy. Together $\beta_0$ and $\beta_1$ are considered to be the model’s coefficients that are estimated during the training of the model. Using the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$, a prediction $\hat{y}$ given a specific $X = x$ can be calculated as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$  \hspace{1cm} (D.2)

In most situations it is desirable to not only predict based on a single predictor variable, but to use multiple independent variables for the prediction. **Multiple linear regression**, as the name implies, extends the simple linear regression model by giving each predictor variable $X_1, X_2, \cdots, X_n$ its own coefficients $\beta_1, \beta_2, \cdots, \beta_n$. The model now be expressed as

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon.$$  \hspace{1cm} (D.3)

The corresponding prediction term is now

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_n x_n.$$  \hspace{1cm} (D.4)

As mentioned before, the coefficients of the model are not known and, therefore, need to be estimated in order to fit the model to the training data. To achieve a fit with the least possible error, the coefficients are estimates by minimizing the residual sum of squares

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$  \hspace{1cm} (D.5)

where $y_i$ is an actual target value and $\hat{y}_i$ is the corresponding predicted value of the model, given the same input.

**D.1.2 Logistic Regression**

**Logistic regression** is a special form of regression model where the dependent variable is categorical, which means that the variable only has a finite set of valid values. In the special case of only two valid values for the dependent variable, the model is called binary logistic regression. In contrast to linear regression, logistic regression does not predict the concrete value of the dependent variable, but rather predicts probabilities for each possible value. As such, logistic regression can be
used to solve classification problems, by choosing the value or label with the highest probability.

![Figure D.1: The Use of a logistic regression for diabetes classification](image)

A simple example for a logistic regression is a blood test to classify if a person suffered from diabetes. Two hours after the meal, the person measures its blood glucose. In Listing D.1 the x axis describes the analyzed blood glucose value and the y axis describes if the person has diabetes or not.

The logistic regression model is used in all model configurations and for the mood disorder prevalence imputation model, as described in Section 3.3.2 and in Section 3.2.7.

### D.1.3 Boosting

Boosting stems from the theoretical discussion, of whether it is possible to aggregate numerous weak learners into a strong learner by combining them, as it easier to train weak learners than to train strong learners. In the context of binary classification, a learner is defined as weak, if it is slightly better than random guessing, whereas a strong learner is defined as approximately impeccable. The conclusion of this discussion was that it is possible to transform a weak learn into strong learner using and iterative procedure [May+14].

In boosting algorithms, such as AdaBoost or Gradient Boosted Trees the models do not manipulate the base-learner. Adaboost reweights the observations of the used training data, while gradient boosted trees works on the residuals of the target values. In each step of the iteration the algorithm finds a new solution.

Boosting builds upon the previous iteration step to concentrate on difficult classification problems. The final step of boosting aggregates all results of the base-learner and creates a more precise model [May+14].
D.1.4 AdaBoost

AdaBoost, short for Adaptive Boosting, was the first boosting algorithm and is researched very well with a widely use in applications of numerous fields [Sch13]. It applies the described boosting strategy and automatically adapts the parameters to the data based on the performance of the current iteration. The final model is the aggregation of all learner results [May+14].

As boosting concentrates on difficult classification problems, the issue of overfitting is often discussed. Overfitting in the area of machine learning is the side effect when a predictor is very closely fitted to the problems of the training data. Overfitting of a prediction model results in a very good prediction quality on seen data, but with poor quality on new data. To avoid overfitting, it is possible to stop the iteration. Stopping the iteration to late, may lead to overfitting, whereas stopping it to early result in a poor prediction model [May+14].

D.1.5 Gradient Boosted Trees

Gradient boosted trees is a machine learning algorithm for classification and regression tasks, which uses decision trees as weak learners [Fri01]. A decision tree is a non paramedic unsupervised machine learning algorithm that extracts its decision rules from the data attributes to build a model for predicting the target value [Mit97].

Gradient boosting starts with an imperfect model, such as the model that outputs the mean of the target value. For each iteration step a new model is created by adding an additional estimator in the form of a decision tree to the imperfect model of the previous iteration step. The additional estimator is fitted on the residual of the target value in regards to the output of the imperfect model of the last iteration step. The new model is created by adding the output of the imperfect model of the previous iteration step and the output of the newly trained estimator [Fri02; NK13].

The gradient boosted trees model is used in all model configurations and as described in Section 3.3.2.

D.2 Evaluation Methodology

The evaluation process in the area of machine learning is very important, as it is easy to train a weak prediction model, but complex to train a strong prediction
D.2. Evaluation Methodology

model with good prediction quality. In the following section the evaluation methodology used in the presented work is described.

D.2.1 Cross Validation

For the process of hyper parameter optimization, cross validation (CV) can be used to resample the existing training data into training and validation sets [Zhe15].

The k-fold variation of cross validation splits a dataset into \( k \) groups, called folds, with an equal size. One of the \( k \) folds is used as a validation set, whereas the remaining \( k - 1 \) folds are used for training the model. The procedure is performed \( k \) times and in every round a different fold is treated as the validation set. By doing this, it is possible to estimate the generalization error, as the validation set in each iteration can be seen as out-of-sample data. The validation results are determined by calculating the average over the \( k \) iterations [Jam+13]. This also allows for the calculation of confidence intervals of the evaluation results. Calculating the MSE using k-fold cross validation can be expressed as

\[
CV(k) = \frac{1}{k} \sum_{i=1}^{n} MSE_i. \tag{D.6}
\]

The leave-one-out variation of cross validation is a special form of the cross validation, where the \( k \) folds are represented by the number of data rows the dataset contains. This variant is valuable when the training dataset is too small to hold out data for the validation objective [Zhe15].

D.2.2 Mean Square Error

When a statistical model is trained on a dataset, the Mean Square Error (MSE) is used to evaluate how good the prediction of the trained model matches the original data. Measuring the MSE is important to know if the prediction is close to the original value or far off. The MSE is given by

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2, \tag{D.7}
\]

where \( n \) is the number of records, \( y_i \) is the original value in the \( i \)th operation, \( \hat{f}(x_i) \) is the predicted value that \( \hat{f} \) returns in the \( i \)th operation [Jam+13].
Appendix D. Machine Learning Background

D.2.3 Accuracy

To measure the quality of a predictor, the accuracy is a good indicator for how many right classifications were made. All the single classification results, 1 for right and 0 for false, are summed up and finally divided by the number of classifications:

\[
ACC = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i),
\]

where \(\hat{y}_i\) is the predicted class label for the \(i\)th classification \(\hat{f}\) and \(I(y_i \neq \hat{y}_i)\) is an indicator variable that equals one if \(y_i \neq \hat{y}_i\) and zero if \(I(y_i = \hat{y}_i)\). If 0 then the \(i\)th observation was classified correctly by the classification method; otherwise it was misclassified [Jam+13].

D.2.4 \(R^2\)

The coefficient of determination, known as \(R^2\), indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. The variability of a dataset set can be measured by calculating the square sum \(SS_{tot}\), where \(y_i\) describes the original value and \(\bar{y}\) the mean

\[
SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2
\]

and the square sum \(SS_{res}\), where \(y_i\) describes the original value and \(\hat{y}\) the predicted value.

\[
SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y})^2
\]

The general description of the \(R^2\) for a linear regression model with one independent variable is:

\[
R^2 = 1 - \frac{SS_{tot}}{SS_{res}}
\]

D.2.5 Log Loss

As the output of a predictor can be either 0 or 1 or a probability, log loss can be used to measure the accuracy of a classifier.
D.2. Evaluation Methodology

\[ ACC = \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{ij} \log p_{ij}), \]  

(D.12)

D.2.6 Receiver Operating Characteristic

Another way to look at a predictors quality is the *receiver operating characteristic* (ROC) represented by a curve that describes the proportion of decision made by a binary classifier. A binary classifier always has an outcome, positive or negative, that specifies whether the input record is predicted to be part of the positive or negative group. If an input record is predicted correctly into the positive group, we are talking about a true positive, and if it is predicted falsely into the negative group, it is called a false positive [Jam+13; Bra97].

The relative frequency distribution for a given threshold is determined by the true positive rate and the false positive rate. The resulting values are added into a diagram with false positive rate horizontal and true positive rate vertical. The idea is to select as many different thresholds as possible to draw a curve through all the points in the diagram. The most upper left points in the diagram the optimum [FBH05]. The area under curve (AUC) is calculated using the composite trapezoidal rule that approximates the region under a given function as a number of individual trapezoids, for which the area can be calculated easily [DDC88].
Appendix E

Data Preparation

In the scope of this thesis several datasets were analyzed and prepared. Further, the important code snippets of the described tasks, can be seen in the following sections.

E.1 Census Community Dataset

```scala
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;
register /home/cloudera/pig/lib/datafu-pig-incubating-1.3.0.jar

define VAR datafu.pig.stats.VAR();

densityData = LOAD '[[..]/density.csv' USING PigStorage(',',:) AS
  (ZIP:chararray, DENSITY:float);

ethnicityData = LOAD '[[..]/ethnicity.csv' USING PigStorage(',',:) AS
  (ZIP:chararray, TOTAL:float, HISPANIC:float, WHITE:float, BLACK:float,
   AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float,
   MIXED:float);

economicData = LOAD '[[..]/economy.csv' USING PigStorage(',',:) AS
  (ZIP:chararray, EMPLOYMENT_STATUS:float, COMMUTING_TO_WORK:float,
   OCC_MANGEMENT:float, OCC_SERVICE:float, OCC_SALES:float,
   OCC_CONSTRUCTION:float, OCC_TRANSPORTATION:float, INCOME_MEDIAN:float,
   INCOME_AVERAGE:float, HEALTH_INSURANCE_YES:float,
   HEALTH_INSURANCE_NO:float, BELOW_POVERTY:float);

joinedAllData = JOIN densityData BY $0, ethnicityData BY $0, economicData BY
  $0;

result = FOREACH joinedAllData GENERATE ethnicityData::ZIP as ZIP,
  TOTAL,
  (HISPANIC/TOTAL) as HISPANIC,
```
Appendix E. Data Preparation

```java
(WHITE/TOTAL) as WHITE, (BLACK/TOTAL) as BLACK, (AMERICAN_INDIAN/TOTAL) as AMERICAN_INDIAN,
(ASIAN/TOTAL) as ASIAN, (PACIFIC_ISLANDER/TOTAL) as PACIFIC_ISLANDER,
(OTHER/TOTAL) as OTHER,
(MIXED/TOTAL) as MIXED, EMPLOYMENT_STATUS/100 as EMPLOYMENT_STATUS,
OCC_MANGEMENT / 100 as OCC_MANGEMENT,
OCC_SERVICE / 100 as OCC_SERVICE, OCC_SALES/100 as OCC_SALES,
OCC_CONSTRUCTION/100 as OCC_CONSTRUCTION,
OCC_TRANSPORTATION/100 as OCC_TRANSPORTATION, HEALTH_INSURANCE_YES/100 as HEALTH_INSURANCE_YES,
BELOW_POVERTY/100 as BELOW_POVERTY, density as DENSITY, INCOME_MEDIAN,
INCOME_AVERAGE, COMMUTING_TO_WORK;
```

averageAndStandardDeviation = FOREACH (GROUP result ALL) GENERATE

```sql
SORT(VAR(result.TOTAL)) as STANDARD_DEVIATION_TOTAL,
SORT(VAR(result.DENSITY)) as STANDARD_DEVIATION_DENSITY,
SORT(VAR(result.INCOME_MEDIAN)) as STANDARD_DEVIATION_INCOME_MEDIAN,
SORT(VAR(result.INCOME_AVERAGE)) as STANDARD_DEVIATION_INCOME_AVERAGE,
SORT(VAR(result.COMMUTING_TO_WORK)) as STANDARD_DEVIATION_COMMUTING_TO_WORK,
AVG(result.TOTAL) as AVERAGE_TOTAL, AVG(result.DENSITY) as AVERAGE_DENSITY,
AVG(result.INCOME_MEDIAN) as AVERAGE_INCOME_MEDIAN,
AVG(result.INCOME_AVERAGE) as AVERAGE_INCOME_AVERAGE,
AVG(result.COMMUTING_TO_WORK) as AVERAGE_COMMUTING_TO_WORK;
```

normalizedData = FOREACH result GENERATE ZIP, (TOTAL -

```sql
(double)averageAndStandardDeviation.AVERAGE_TOTAL) /
(double)averageAndStandardDeviation.
```

STANDARD_DEVIATION_TOTAL) as NORMALIZED_TOTAL, HISPANIC, WHITE, BLACK,
AMERICAN_INDIAN, ASIAN, PACIFIC_ISLANDER, OTHER, MIXED,
EMPLOYMENT_STATUS, OCC_MANGEMENT, OCC_SERVICE, OCC_SALES, OCC_CONSTRUCTION,
OCC_TRANSPORTATION, HEALTH_INSURANCE_YES,
BELOW_POVERTY, (TOTAL -

```sql
(double)averageAndStandardDeviation.AVERAGE_DENSITY) /
(double)averageAndStandardDeviation.
```

STANDARD_DEVIATION_DENSITY) as NORMALIZED_DENSITY,
(INCOME_MEDIAN - (double)averageAndStandardDeviation.AVERAGE_INCOME_MEDIAN) /
(double)averageAndStandardDeviation.

STANDARD_DEVIATION_INCOME_MEDIAN) as NORMALIZED_INCOME_MEDIAN,
(INCOME_AVERAGE - (double)averageAndStandardDeviation.AVERAGE_INCOME_AVERAGE) /
(double)averageAndStandardDeviation.

STANDARD_DEVIATION_INCOME_AVERAGE) as NORMALIZED_INCOME_AVERAGE,
(COMMUTING_TO_WORK - (double)averageAndStandardDeviation.AVERAGE_COMMUTING_TO_WORK) /
(double)averageAndStandardDeviation.

STANDARD_DEVIATION_COMMUTING_TO_WORK) as NORMALIZED_COMMUTING_TO_WORK;

STORE normalizedData INTO '[:]/census_all' USING PigStorage(',,,-schema');
LISTING E.1: Pig Latin script that joins all acquired and prepared census datasets into the census community dataset

```pig
medianValuesData = LOAD '[...]/all_median_values.csv' USING PigStorage(',,')
→ AS (HISPANIC:float, WHITE:float, BLACK:float, AMERICAN_INDIAN:float,
→ ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float, MIXED:float,
→ EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float, OCC_SERVICE:float,
→ OCC_SALES:float, OCC_CONSTRUCTION:float, OCC_TRANSPORTATION:float,
→ HEALTH_INSURANCE_YES:float, BELOW_POVERTY:float, DENSITY:float,
→ INCOME_MEDIAN:float, INCOME_AVG:float, COMMUTING_TO_WORK:float);

normalizedDemographicData = LOAD '[...]/census_all.csv' USING
→ PigStorage(',,') AS (ZIP:chararray, TOTAL:float, HISPANIC:float,
→ WHITE:float, BLACK:float, AMERICAN_INDIAN:float, ASIAN:float,
→ PACIFIC_ISLANDER:float, OTHER:float, MIXED:float,
→ EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float, OCC_SERVICE:float,
→ OCC_SALES:float, OCC_CONSTRUCTION:float, OCC_TRANSPORTATION:float,
→ HEALTH_INSURANCE_YES:float, BELOW_POVERTY:float, DENSITY:float,
→ INCOME_MEDIAN:float, INCOME_AVG:float, COMMUTING_TO_WORK:float);

result = FOREACH normalizedDemographicData GENERATE
→ ZIP,
→ TOTAL,
→ ((HISPANIC IS NULL) ? medianValuesData.HISPANIC : HISPANIC) as HISPANIC,
→ ((WHITE IS NULL) ? medianValuesData.WHITE : WHITE) as WHITE,
→ ((BLACK IS NULL) ? medianValuesData.BLACK : BLACK) as BLACK,
→ ((AMERICAN_INDIAN IS NULL) ? medianValuesData.AMERICAN_INDIAN : 
→ → AMERICAN_INDIAN) as AMERICAN_INDIAN,
→ ((ASIAN IS NULL) ? medianValuesData.ASIAN : ASIAN) as ASIAN,
→ ((PACIFIC_ISLANDER IS NULL) ? medianValuesData.PACIFIC_ISLANDER : 
→ → PACIFIC_ISLANDER) as PACIFIC_ISLANDER,
→ ((OTHER IS NULL) ? medianValuesData.OTHER : OTHER) as OTHER,
→ ((MIXED IS NULL) ? medianValuesData.MIXED : MIXED) as MIXED,
→ ((EMPLOYMENT_STATUS IS NULL) ? medianValuesData.EMPLOYMENT_STATUS : 
→ → EMPLOYMENT_STATUS) as EMPLOYMENT_STATUS,
→ ((OCC_MANGEMENT IS NULL) ? medianValuesData.OCC_MANGEMENT : OCC_MANGEMENT) 
→ → as OCC_MANGEMENT,
→ ((OCC_SERVICE IS NULL) ? medianValuesData.OCC_SERVICE : OCC_SERVICE) as 
→ → OCC_SERVICE,
→ ((OCC_SALES IS NULL) ? medianValuesData.OCC_SALES : OCC_SALES) as OCC_SALES,
→ ((OCC_CONSTRUCTION IS NULL) ? medianValuesData.OCC_CONSTRUCTION : 
→ → OCC_CONSTRUCTION) as OCC_CONSTRUCTION,
```
Appendix E. Data Preparation

```java
((OCC_TRANSPORTATION IS NULL) ? medianValuesData.OCC_TRANSPORTATION : → OCC_TRANSPORTATION) as OCC_TRANSPORTATION,
((HEALTH_INSURANCE_YES IS NULL) ? medianValuesData.HEALTH_INSURANCE_YES : → HEALTH_INSURANCE_YES) as HEALTH_INSURANCE_YES,
((BELOW_POVERTY IS NULL) ? medianValuesData.BELOW_POVERTY : BELOW_POVERTY) → as BELOW_POVERTY,
((DENSITY IS NULL) ? medianValuesData.DENSITY : DENSITY) as DENSITY,
((INCOME_MEDIAN IS NULL) ? medianValuesData.INCOME_MEDIAN : INCOME_MEDIAN) → as INCOME_MEDIAN,
((INCOME_AVG IS NULL) ? medianValuesData.INCOME_AVG : INCOME_MEDIAN) as → INCOME_AVG,
((COMMUTING_TO_WORK IS NULL) ? medianValuesData.COMMUTING_TO_WORK : → COMMUTING_TO_WORK) as COMMUTING_TO_WORK;
STORE result INTO '[[...]/census_all_median replacement' USING
→ PigStorage('', '', '-schema');
```

Listing E.2: Pig Latin script that replaces all null values in the census
community dataset with the corresponding median.

E.2 Crime Prediction

E.2.1 Crime ZIP Code Identification

```java
var csv = require('csv');
var fs = require('fs');
var bhttp = require('bhttp');
var querystring = require('querystring');
var RateLimiter = require('limiter').RateLimiter;

var parser = csv.parse();
var stringifierOutput = csv.stringify();
var stringifierError = csv.stringify();
var readStream = fs.createReadStream('input.csv');
var writeStreamOutput = fs.createWriteStream('output.csv');
var writeStreamError = fs.createWriteStream('error.csv');
var limiter = new RateLimiter(4, 500);

var counter = 0;
var failed = 0;

function geocode(id, street, city, state) {
    var parameters = {
        "street": street,
        "city": city,
        "state": state,
```
"benchmark": "4",
"format": "json"
);

console.log("get", id);

bhttp.get("http://geocoding.geo.census.gov/geocoder/locations/address?" + queryString.stringify(parameters)).then(function(response) {
    counter++;
    var output = [id, parameters.street, parameters.city,
                  parameters.state];
//console.log(response);
    if(response.body.result.addressMatches.length > 0) {
        console.log(counter +": "+ failed);
//console.log(response.body.result.addressMatches[0].
            matchedAddress.split(',').[3].trim());
        output.push(response.body.result.addressMatches[0].
            matchedAddress.split(',').[3].trim());
        stringifierOutput.write(output);
    }
    else{
        failed++;
        console.log(counter, id, "not found");
        output.push("");
        stringifierError.write(output);
    }
}).catch(function(err) {
    limiter.removeTokens(1, function() {
        console.log("rerun");
        geocode(id, street, city, state);
    });
});

parser.on('readable', function(){
    while(data = parser.read()){
        var id = data[0];
        var street = data[1];
        var city = data[2];
        var state = data[3];

        limiter.removeTokens(1, function() {
            geocode(id, street, city, state);
        });
    }
});
Listing E.3: Node.js script that identifies anonymized ZIP Codes using the Census Geocoding API

E.2.2 Pig

crimeData = LOAD '[...]/crime_with_zip.csv' USING PigStorage(';',') AS
-> (DATE:chararray, BLOCK:chararray, CITY:chararray, STATE:chararray,
-> ZIP:chararray);

groupByYear = GROUP crimeData by ZIP;

result = FOREACH groupByYear GENERATE group, COUNT($1) as AVERAGE;

STORE result INTO '[...]/crime_avg_per_zip' USING PigStorage(';',',','-schema');

Listing E.4: Pig Latin script that calculates the average crime rate per year of each ZIP Code that is part of the loaded dataset

register /home/cloudera/pig/lib/piggybank-0.12.0.jar;

register /home/cloudera/pig/lib/datafu-pig-incubating-1.3.0.jar

define VAR datafu.pig.stats.VAR();

demographicData = LOAD '[...]/all_normalized.csv' USING PigStorage(';',') AS
-> (ZIP:chararray, TOTAL:float, HISPANIC:float, WHITE:float, BLACK:float,
-> AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float,
-> MIXED:float, EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float,
-> OCC_SERVICE:float, OCC_SALES:float, OCC_CONSTRUCTION:float,
-> OCC_TRANSPORTATION:float, HEALTH_INSURANCE_YES:float,
-> BELOW_POVERTY:float, DENSITY:float, INCOME_MEDIAN:float,
-> INCOME_AVG:float, COMMUTING_TO_WORK:float);

crimeData = LOAD '[...]/crime_avg_zip.csv' USING PigStorage(';',') AS
-> (ZIPCODE:chararray, CRIMES:float);

joinedData = JOIN crimeData by $0, demographicData by $0;

averageAndStandardDeviation = FOREACH (GROUP joinedData ALL) GENERATE
-> SQRT(VAR(joinedData.CRIMES)) as standardDeviationCrimes,
-> AVG(joinedData.CRIMES) as avgCrimes;
E.3 Imputation of Mood Disorder Prevalence

result = FOREACH joinedData GENERATE crimeData::ZIPCODE, ((CRIMES -
  \(\rightarrow\) (double)averageAndStandardDeviation.aggCrimes)
  \(\rightarrow\) (double)averageAndStandardDeviation.standardDeviation.aggCrimes) as
  CRIMES as crimes, averageAndStandardDeviation.aggCrimes as AVERAGE, TOTAL,
  \(\rightarrow\) demographicData::HISPANIC as HISPANIC,
  demographicData::WHITE as WHITE, demographicData::BLACK as BLACK,
  \(\rightarrow\) demographicData::AMERICAN_INDIAN as AMERICAN_INDIAN,
  demographicData::ASIAN as ASIAN, demographicData::PACIFIC_ISLANDER as
  PACIFIC_ISLANDER, demographicData::OTHER as OTHER,
  demographicData::MIXED as MIXED, demographicData::EMPLOYMENT_STATUS as
  EMPLOYMENT_STATUS, demographicData::OCC_MANGEMENT as OCC_MANGEMENT,
  demographicData::OCC_SERVICE as OCC_SERVICE, demographicData::OCC_SALES as
  OCC_SALES, demographicData::OCC_CONSTRUCTION as OCC_CONSTRUCTION,
  demographicData::OCC_TRANSPORTATION as OCC_TRANSPORTATION,
  \(\rightarrow\) demographicData::HEALTH_INSURANCE_YES as HEALTH_INSURANCE_YES,
  demographicData::BELOW_PORTEY as BELOW_PORTEY, demographicData::DENSITY,
  \(\rightarrow\) demographicData::INCOME_MEDIAN, demographicData::INCOME_AVG,
  demographicData::COMMUTING_TO_WORK;

STORE result INTO ' [...]/crime_standardized_zip_with_features' USING
  PigStorage('',''-'schema');

Listing E.5: Pig Latin script that joins the average crime per ZIP Code
  with the census community dataset

E.3 Imputation of Mood Disorder Prevalence

-- Alex 01.09.2016

demographicData = LOAD ' [...]/demographics_training.csv' USING PigStorage('','') AS
  (STUDYID:chararray, BIRTH_DATE:M:chararray,
   \(\rightarrow\) DEATH_DATE:M:chararray, GENDER:chararray, RACE:chararray,
   \(\rightarrow\) MARITAL_STATUS:chararray, RELIGION:chararray, PATIENT_ZIPCODE:chararray,
   \(\rightarrow\) EMPLOYED_STATUS:chararray, LANGUAGE:chararray, INSURANCE:chararray,
   \(\rightarrow\) CONTROL_PATIENT:int);

ethnicityData = LOAD ' [...]/ethnicity.csv' USING PigStorage('','') AS
  (ZIPCODE:chararray, TOTAL:float, HISPANIC:float, WHITE:float,
   \(\rightarrow\) BLACK:float, AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float,
   \(\rightarrow\) OTHER:float, MIXED:float);

cleanedZips = FOREACH demographicData GENERATE SUBSTRING(PATIENT_ZIPCODE, 0,
  \(\rightarrow\) 5) as ZIP, CONTROL_PATIENT as CP;

groupByZIP = GROUP cleanedZips BY (ZIP);
patientsPerZIP = FOREACH groupByZIP GENERATE group as ZIP,
   COUNT(cleanedZips) as PATIENTS, COUNT(cleanedZips) - SUM(cleanedZips.CP)
   as MOOD_DISORDER_PATIENTS;

patientsPerZIPWithEthnicity = JOIN patientsPerZIP BY ZIP, ethnicityData by
   ZIPCODE;

coveragePerZIP = FOREACH patientsPerZIPWithEthnicity GENERATE
   patientsPerZIP::ZIP as ZIP, (patientsPerZIP::PATIENTS /
   ethnicityData::TOTAL) as COVERAGE,
   patientsPerZIP::patients as SAMPLES, patientsPerZIP::MOOD_DISORDER_PATIENTS
   / (float)patientsPerZIP::PATIENTS as PREVALENCE_MLE,
   (patientsPerZIP::MOOD_DISORDER_PATIENTS + 1.926) / (patientsPerZIP::PATIENTS
   + 3.8416) as PREVALENCE_WP, ethnicityData::TOTAL as POPULATION;

x = FOREACH coveragePerZIP GENERATE ZIP, PREVALENCE_WP, COVERAGE,
   PREVALENCE_WP, 1.96 * SQRT(((PREVALENCE_WP * (1-PREVALENCE_WP)) / (SAMPLES + 3.8416)) *
   ((POPULATION - SAMPLES) / (POPULATION - 1))) as MARGIN_OF_ERROR,
   POPULATION, SAMPLES;

result = ORDER x by MARGIN_OF_ERROR;

STORE result INTO '[[...]/patient_coverage_and_confidence_per_zip_adjusted
   _wald_ordered_by_moe' USING PigStorage(',','=schema');

Listing E.6: Pig Latin script that calculates the patient coverage coverage
and confidence on a ZIP Code level basis using the adjusted wald method

REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

demographicData = LOAD '[[...]/all_normalized.csv' USING PigStorage(',') AS
   (ZIP:chararray, TOTAL:float, HISPANIC:float, WHITE:float, BLACK:float,
   AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float,
   MIXED:float, EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float,
   OCC_SERVICE:float, OCC_SALES:float, OCC_CONSTRUCTION:float,
   OCC_TRANSPORTATION:float, HEALTH_INSURANCE_NO:float,
   BELOW_POVERTY:float, DENSITY:float, INCOME_MEDIAN:float,
   INCOME_AVG:float, COMMUTING_TO_WORK:float);

coverage = LOAD '[[...]/patient_coverage_and_confidence_zip_adjusted_
   wald_moe_3_ordered.csv' USING PigStorage(',');
E.3. Imputation of Mood Disorder Prevalence

Listing E.7: Pig Latin script that join the significant samples with the the census community dataset

```
    AS (ZIP:chararray, PREVALENCE_MLE:float, COVERAGE:float, PREVALENCE_WP:float,
        MARGIN_OF_ERROR:float, POPULATION:float, SAMPLES:float);

    joined = JOIN coverage BY ZIP, demographicData BY ZIP;

    result = FOREACH joined GENERATE demographicData::ZIP,
        coverage::PREVALENCE_MLE, coverage::MARGIN_OF_ERROR, TOTAL,
        demographicData::HISPANIC, demographicData::WHITE,
        demographicData::BLACK, demographicData::AMERICAN_INDIAN,
        demographicData::ASIAN, demographicData::PACIFIC_ISLANDER,
        demographicData::OTHER, demographicData::MIXED,
        demographicData::EMPLOYMENT_STATUS, demographicData::OCC_MANGEMENT,
        demographicData::OCC_SERVICE, demographicData::OCC_SALES,
        demographicData::OCC_CONSTRUCTION, demographicData::OCC_TRANSPORTATION,
        demographicData::HEALTH_INSURANCE_NO, demographicData::BELOW_POVERTY,
        demographicData::DENSITY, demographicData::INCOME_MEDIAN,
        demographicData::INCOME_AVERAGE, demographicData::COMMUTING_TO_WORK;

    STORE result INTO '[[...]/mood_disorder_prevalence_confidence_30_features'
        USING PigStorage('','','-schema');
```

```
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

demographicData = LOAD '[[...]/all_normalized.csv' USING PigStorage('','') AS
    (ZIP:chararray, TOTAL:float, HISPANIC:float, WHITE:float, BLACK:float,
     AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float,
     MIXED:float, EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float,
     OCC_SERVICE:float, OCC_SALES:float, OCC_CONSTRUCTION:float,
     OCC_TRANSPORTATION:float, HEALTH_INSURANCE_NO:float,
     BELOW_POVERTY:float, DENSITY:float, INCOME_MEDIAN:float,
     INCOME_AVERAGE:float, COMMUTING_TO_WORK:float);

coverage = LOAD '[[...]/patient_coverage_and
    _confidence_zip_adjusted_wald_moe_3_ordered.csv' USING PigStorage('','')
    AS (ZIP:chararray, PREVALENCE_MLE:float, COVERAGE:float,
     PREVALENCE_WP:float, MARGIN_OF_ERROR:float, POPULATION:float,
     SAMPLES:float);

    joined = JOIN coverage BY ZIP, demographicData BY ZIP;

    result = FOREACH joined GENERATE coverage::ZIP, coverage::PREVALENCE_WP,
### Appendix E. Data Preparation

Listing E.8: Pig Latin script that joins the significant samples with the nine features with the highest F-measures

```pig
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;


confidenceData = LOAD '[...]/mood_disorder_prevalence_confidence_over_30.csv' USING PigStorage(';',') AS (ZIP:chararray, PREVALENCE_MLE:float, MARGIN_OF_ERROR:float);

joined = JOIN confidenceData BY ZIP, demographicData BY ZIP;

result = FOREACH joined GENERATE confidenceData::ZIP,
confidenceData::PREVALENCE_WP,
demographicData::HISPANIC, demographicData::WHITE, demographicData::ASIAN,
OCC_MANGEMENT, OCC_SERVICE, OCC_TRANSPORTATION,
demographicData::DENSITY, demographicData::INCOME_MEDIAN,
→ demographicData::INCOME_AVG;

STORE result INTO '[...]/mood_disorder_prevalence_confidence_over_30_with_nine_features' USING PigStorage(';',')'-schema');
```

Listing E.9: Pig Latin script that joins the non significant samples with the census community data

```pig
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

demographicData::HISPANIC, demographicData::WHITE, demographicData::ASIAN,
OCC_MANGEMENT, OCC_SERVICE, OCC_TRANSPORTATION,
demographicData::DENSITY, demographicData::INCOME_MEDIAN,
→ demographicData::INCOME_AVERAGE;

STORE result INTO '[...]/mood_disorder_prevalence_confidence_30
→ _with_nine_features' USING PigStorage(';',')'-schema');
```
E.3. Imputation of Mood Disorder Prevalence

```scala
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

modelData = LOAD '[...]/zip_with_predictions_confidence_intervals.csv' USING PigStorage(';', ) AS (ZIP:chararray, PREVALENCE:float, SE:float,
    LOWER_CONF:float, UPPER_CONF:float);

data = LOAD '[...]/patient_coverage_confidence_per_zip_adjusted_wald_ordered_by_prevalence.csv' USING PigStorage(';', ) AS (ZIP:chararray,
    PREVALENCE_MLE:float, COVERAGE:float, PREVALENCE_WP:float,
    MARGIN_OF_ERROR:float, POPULATION:float, SAMPLES:float);

joined = JOIN modelData BY ZIP LEFT OUTER, data BY ZIP;

cleaned = FOREACH joined GENERATE modelData::ZIP as ZIP, PREVALENCE as
    MODEL_PREVALENCE, LOWER_CONF as MODEL_LOWER, UPPER_CONF as MODEL_UPPER,
    PREVALENCE_WP as CALCULATED_PREVALENCE,
    ((PREVALENCE_WP IS NOT NULL) ? (PREVALENCE_WP - MARGIN_OF_ERROR) : -10000) as
    CALCULATED_LOWER,
    ((PREVALENCE_WP IS NOT NULL) ? (PREVALENCE_WP + MARGIN_OF_ERROR) : 10000) as
    CALCULATED_UPPER;

replacements = FOREACH cleaned GENERATE ZIP as ZIP,
    (((ABS(MODEL_LOWER - MODEL_UPPER)) < (ABS(CALCULATED_LOWER -
    CALCULATED_UPPER))) ? MODEL_PREVALENCE : CALCULATED_PREVALENCE) as FINAL_PREVALENCE,
    (((ABS(MODEL_LOWER - MODEL_UPPER)) < (ABS(CALCULATED_LOWER -
    CALCULATED_UPPER))) ? MODEL_LOWER : CALCULATED_LOWER) as FINAL_LOWER,
    (((ABS(MODEL_LOWER - MODEL_UPPER)) < (ABS(CALCULATED_LOWER -
    CALCULATED_UPPER))) ? MODEL_UPPER : CALCULATED_UPPER) as FINAL_UPPER,
    (((ABS(MODEL_LOWER - MODEL_UPPER)) < (ABS(CALCULATED_LOWER -
    CALCULATED_UPPER))) ? 1 : 0) as REPLACEMENT,
    MARGIN_OF_ERROR;

result = FOREACH replacements GENERATE ZIP,
    (((ABS(FINAL_LOWER - FINAL_UPPER)) < 0.3) ? FINAL_PREVALENCE :
    0.2454462645) as FINAL_PREVALENCE,
    (((ABS(FINAL_LOWER - FINAL_UPPER)) < 0.3) ? FINAL_LOWER : 0.2454462645) as
    FINAL_LOWER,
    (((ABS(FINAL_LOWER - FINAL_UPPER)) < 0.3) ? FINAL_UPPER : 0.2454462645) as
    FINAL_UPPER,
    (((ABS(FINAL_LOWER - FINAL_UPPER)) < 0.3) ? REPLACEMENT : 2) as REPLACEMENT,
    MARGIN_OF_ERROR;

final = FOREACH result GENERATE ZIP, FINAL_PREVALENCE, FINAL_LOWER,
    FINAL_UPPER, MARGINOFERROR, REPLACEMENT;
```
finalResult = ORDER final BY ZIP ASC;

STORE finalResult INTO '[[...]/model_and_calculation_replacement' USING PigStorage('','','-schema');

Listing E.10: Pig Latin script that replaces the non significant samples with the model imputation or the average.

REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

modelData = LOAD '[[...]/zip_with_predictions_confidence_intervals.csv' USING PigStorage('','') AS (ZIP:chararray);

data = LOAD '[[...]/patient_coverage_confidence_per_zip_adjusted_wald_ordered_by_prevalence.csv' USING PigStorage('','') AS (ZIP:chararray,
  PREVALANCE:float, COVERAGE:float, PREVALANCE_WP:float,
  MARGIN_OF_ERROR:float, population:float, samples:float);

joined = JOIN modelData BY ZIP LEFT OUTER, data BY ZIP;

cleaned = FOREACH joined GENERATE modelData::ZIP as ZIP, PREVALANCE_WP as 
  CALCULATED_PREVALENCE,
  ((PREVALANCE_WP IS NOT NULL)?(PREVALANCE_WP - MARGIN_OF_ERROR) : -10000) as 
  CALCULATED_LOWER,
  ((PREVALANCE_WP IS NOT NULL)?(PREVALANCE_WP + MARGIN_OF_ERROR) : 10000) as 
  CALCULATED_UPPER, MARGIN_OF_ERROR;

result = FOREACH cleaned GENERATE ZIP as ZIP,
  (((ABS(CALCULATED_LOWER - CALCULATED_UPPER)) < 0.3) ? 
    CALCULATED_PREVALENCE : 0.2454462645) as FINAL_PREVALENCE,
  (((ABS(CALCULATED_LOWER - CALCULATED_UPPER)) < 0.3) ? CALCULATED_LOWER : 
    0.2454462645) as FINAL_LOWER,
  (((ABS(CALCULATED_LOWER - CALCULATED_UPPER)) < 0.3) ? CALCULATED_UPPER : 
    0.2454462645) as FINAL_UPPER,
  MARGIN_OF_ERROR;

final = FOREACH result GENERATE ZIP, FINAL_PREVALENCE, FINAL_LOWER,
  FINAL_UPPER, MARGIN_OF_ERROR;

finalResult = ORDER final BY ZIP ASC;

STORE finalResult INTO '[[...]/model_and_calculation_replacement' USING PigStorage('','','-schema');
E.4. Patient Demographics

LISTING E.11: Pig Latin script that replaces the non significant samples the average

E.4 Patient Demographics

LISTING E.12: Pig Latin script that cleans the demographic datset and transforms various categorical features into new features

E.5 Patient History

```java
REGISTRATION /home/cloudera/pig/lib/piggybank-0.12.0.jar;

problemListData = LOAD '/user/master/data/geisinger/problem_list.csv' USING
  org.apache.pig.piggybank.storage.CSVExcelStorage() AS
  (STUDYID:chararray, DX_ID:chararray, DX_NAME:chararray,
   CURRENT_ICD9_LIST:chararray, CURRENT_ICD10_LIST:chararray,
   NOTED_DATE_M:chararray, RESOLVED_DATE_M:chararray,
   PROBLEM_STATUS:chararray);

demographicData = LOAD '/user/master/data/geisinger/demographics.csv' USING
  PigStorage(',' ) AS (STUDYID:chararray, BIRTH_DATE_M:chararray,
  DEATH_DATE_M:chararray, GENDER:chararray, RACE:chararray,
  MARITAL_STATUS:chararray, RELIGION:chararray, PATIENT_ZIPCODE:chararray,
  EMPLOYED_STATUS:chararray, LANGUAGE:chararray, INSURANCE:chararray,
  CONTROL_PATIENT:int);

joinedData = JOIN demographicData BY STUDYID, problemListData BY STUDYID;

diagnosesWithStudyIdICD9DATE = FOREACH joinedData GENERATE
  problemListData::STUDYID as STUDY_ID,
  FLATTEN(TOKENIZE(problemListData::CURRENT_ICD9_LIST, ',')) as ICD9,
  problemListData::NOTED_DATE_M as NOTED_DATE;

orderedList = ORDER diagnosesWithStudyIdICD9DATE BY STUDY_ID;

STORE orderedList INTO '[][transformed_ordered_problem_list_with
  _studyId_icd9_date' USING PigStorage(',' ,'-schema');

LISTING E.13: Pig Latin script that transforms the records in the patient history containing more than one ICD-9 code into individual records
**Appendix E. Data Preparation**

```java
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

problemListData = LOAD '[[..]]/transformed_ordered_problem_list_study_id_icd9_date.csv' USING 
org.apache.pig.piggybank.storage.CSVExcelStorage() AS 
(STUDYID:chararray, ICD9:chararray, NOTED_DATE:chararray);

demographicsTrainingData = LOAD '[[..]]/demographics_training.csv' USING 
PigStorage('(','') AS (STUDYID:chararray, BIRTH_DATE_M:chararray, 
DEATH_DATE_M:chararray, GENDER:chararray, RACE:chararray, 
MARITAL_STATUS:chararray, RELIGION:chararray, PATIENT_ZIPCODE:chararray, 
EMPLOYED_STATUS:chararray, LANGUAGE:chararray, INSURANCE:chararray, 
CONTROL_PATIENT:int);

joined = JOIN demographicsTrainingData BY STUDYID, problemListData BY 
STUDYID;

result = FOREACH joined GENERATE demographicsTrainingData::STUDYID, ICD9, 
NOTED_DATE;

STORE result INTO '[[..]]/ordered_problem_list_training' USING 
PigStorage('(','','-schema');

LISTING E.14: Pig Latin script that calculates the training set of the ordered problem list

```
E.6 Blended Dataset

```sql
REGISTER /home/cloudera/pig/lib/piggybank-0.12.0.jar;

--check

demographicsTrainingData = LOAD
  '[[...]/demographics_extended_cleaned_training.csv' USING PigStorage(',,')
  AS (STUDYID:chararray, ZIP:chararray, age:float, GENDER:float,
    WHITE:float, BLACK:float, ASIAN:float, AMERICAN_INDIAN:float,
    PACIFIC_ISLANDER:float, MARRIED:float, DIVORCED:float, WIDOWED:float,
    EMP_FULL:float, EMP_PART:float, EMP_STUD:float, EMP_SELF:float,
    EMP_RETIRED:float, INSURANCE:float, MEDICARE:float, CP:int);

patientHistoryData = LOAD '[[...]/patient_history_training_90percent.csv'
  USING PigStorage(',,') AS
    (STUDYID:chararray, f1:float,[...], f1150:float);

--check
demographicData = LOAD '[[...]/all_normalized.csv' USING PigStorage(',,') AS
  (ZIP:chararray, TOTAL:float, HISPANIC:float, WHITE:float, BLACK:float,
   AMERICAN_INDIAN:float, ASIAN:float, PACIFIC_ISLANDER:float, OTHER:float,
   MIXED:float, EMPLOYMENT_STATUS:float, OCC_MANGEMENT:float,
   OCC_SERVICE:float, OCC_SALES:float, OCC_CONSTRUCTION:float,
   OCC_TRANSPORTATION:float, HEALTH_INSURANCE_YES:float,
   BELOW_POVERTY:float, DENSITY:float, INCOME_MEDIAN:float,
   INCOME_AVG:float, COMMUTING_TO_WORK:float);

--check
prevalenceAVGData = LOAD
  '/user/master/data/mood_disorder_prevalence/prevalence_avg_data.csv'
  USING PigStorage(',,') AS (ZIP:chararray, PREVALENCE:float);

--check
prevalenceModelData = LOAD
  '/user/master/data/mood_disorder_prevalence/prevalence_model_data.csv'
  USING PigStorage(',,') AS (ZIP:chararray, PREVALENCE:float);

--check
crimeIndicatorData = LOAD
  '/user/master/data/crimeclincial_zips_crime_indicator.csv' USING
  PigStorage(',,') AS (ZIP:chararray, CRIME_INDICATOR:float);

clinicalDemographicsAndMatrix = JOIN patientHistoryData BY STUDYID,
    demographicsTrainingData BY STUDYID;

joined = JOIN demographicData BY $0, prevalenceAVGData BY $0,
    prevalenceModelData BY $0, crimeIndicatorData BY $0,
    clinicalDemographicsAndMatrix BY demographicsTrainingData::ZIP;
```
result = FOREACH joined GENERATE

demographicsTrainingData::CP,
demographicsTrainingData::STUDYID,
demographicData::ZIP,
-- FEATURES
prevalenceAVGData::PREVALENCE,
prevalenceModelData::PREVALENCE,
crimeIndicatorData::CRIME_INDICATOR,

demographicData::TOTAL, demographicData::HISPANIC, demographicData::WHITE,
demographicData::BLACK, demographicData::AMERICAN_INDIAN,
  → demographicData::ASIAN,
demographicData::PACIFIC_ISLANDER, demographicData::OTHER,
  → demographicData::MIXED,
demographicData::EMPLOYMENT_STATUS, demographicData::OCC_MANGEMENT,
  → demographicData::OCC_SERVICE,
demographicData::OCC_SALES, demographicData::OCC_CONSTRUCTION,
  → demographicData::OCC_TRANSPORTATION,
demographicData::HEALTH_INSURANCE_YES, demographicData::BELOW_POVERTY,
  → demographicData::DENSITY,
demographicData::INCOME_MEDIAN, demographicData::INCOME_AVG,
  → demographicData::COMMUTING_TO_WORK,

demographicsTrainingData::age, demographicsTrainingData::GENDER,
  → demographicsTrainingData::WHITE,
demographicsTrainingData::BLACK, demographicsTrainingData::ASIAN,
  → demographicsTrainingData::AMERICAN_INDIAN,
demographicsTrainingData::PACIFIC_ISLANDER,
  → demographicsTrainingData::MARRIED, demographicsTrainingData::DIVORCED,
demographicsTrainingData::WIDOWED, demographicsTrainingData::EMP_FULL,
  → demographicsTrainingData::EMP_PART,
demographicsTrainingData::EMP_STUD, demographicsTrainingData::EMP_SELF,
  → demographicsTrainingData::EMP_RETIRED,
demographicsTrainingData::INSURANCE, demographicsTrainingData::MEDICARE,

patientHistoryData::F1,,[...],patientHistoryData::F1150;

STORE result INTO ' [...]/matrix_all_training_90_percent' USING
  → PigStorage('','');

Listing E.16: Pig Latin script that joins all acquired and prepared datasets into one coherent dataset.
Appendix F

Data Analysis

In the scope of this thesis several machine learning were trained and evaluated. Further, various datasets were prepared and analyzed. The important code snippets of the described tasks, can be seen in the following sections.

F.1 Imputation of Mood Disorder Prevalence

F.1.1 Patient Coverage

F.1.1.1 Patient Coverage Pennsylvania

```python
import matplotlib as mpl
from matplotlib.patches import Polygon
from matplotlib.collections import PatchCollection
import matplotlib.colors as colors

# Create figure and two axes: one to hold the map and one to hold
# the colorbar
plt.figure(figsize=(15, 7.5), dpi=120)
map_axis = plt.axes([0.0, 0.0, 1, 1])
cb_axis = plt.axes([0.9, 0.1, 0.02, 0.8])

# Define colormap to color the ZIP codes.
# You can try changing this to cm.Blues or any other colormap
# to get a different effect
cmap = plt.get_cmap("plasma")

# Create the map axis
plt.axes(map_axis)
plt.axis([-81, -73.5, 39.5, 42.4])
plt.gca().set_axis_off()

# Read in ZIP code boundaries for Pennsylvania
d = read_ascii_boundary('..Data/zt42_d00')

patches = []
coverages = []
```
for polygon_id in d:
    polygon_data = np.array(d[polygon_id][\'polygon\'])
    zipcode = d[polygon_id][\'name\']

    if zipcode in zip_coverage_map:
        coverage = zip_coverage_map[zipcode]
    else:
        coverage = 0
    coverages.append(coverage)
    # Draw the ZIP code
    patch = Polygon(polygon_data, True)
    patches.append(patch)

coverages = np.array(coverages)
coverages_nonzero = coverages[np.nonzero(coverages)]
smallest_coverage = sorted(coverages_nonzero.flat)[1]
coverages = np.maximum(coverages, smallest_coverage)

p = PatchCollection(patches, cmap=cmap, edgecolor=(1, 1, 1, 1), linewidth=.5,
                     norm=colors.PowerNorm(0.25))
p.set_array(coverages)
plt.gca().add_collection(p)

cb = mpl.colorbar.ColorbarBase(cb_axis, cmap=cmap,
                                norm=colors.PowerNorm(0.25))
plt.savefig("[...]/Map_Distribution_Coverage_Pennsylvania.svg")

LISTING F.1: Python script that draws the patient coverage in the State of Pennsylvania

F.1.1.2 Geisinger Facilities in Pennsylvania

import matplotlib as mpl
from matplotlib.patches import Polygon
from matplotlib.collections import PatchCollection
import matplotlib.colors as colors

# Create figure and two axes: one to hold the map and one to hold
# the colorbar
plt.figure(figsize=(15, 7.5), dpi=128)

# Define colormap to color the ZIP codes.
# You can try changing this to cm.Blues or any other colormap
F.1. Imputation of Mood Disorder Prevalence

### Listing F.2: Python script that draws the map with all the Geisinger Health facilities in Pennsylvania

```python
# to get a different effect
cmap = plt.get_cmap("plasma")

# Create the map axis
plt.axis([-81, -73.5, 39.5, 42.4])
plt.gca().set_axis_off()

# Read in ZIP code boundaries for Pennsylvania
d = read_ascii_boundary('../Data/zt42_d00')

patches = []
for polygon_id in d:
    polygon_data = np.array(d[polygon_id]["polygon"])

    # Draw the ZIP code
    patch = Polygon(polygon_data, True)
    patches.append(patch)

p = PatchCollection(patches, cmap=cmap, edgecolor=(0, 0, 0, 0.5),
                    facecolor=(1,1,1,1), linewidth=.5)
plt.gca().add_collection(p)

y, x = zip(*geisinger_locations)

print x
print y

plt.scatter(x, y, s=40, c="red", edgecolor=(0.3,0,0))
plt.savefig("[...]/Map_Geisinger_Locations_ Pennsylvania.svg")
```

### F.1.1.3 Confidence

```python
import numpy as np
from sklearn.cross_validation import train_test_split

mood_disorder_prevalence_confidence_3 =
    np.loadtxt(open("[...]\patient_coverage_and_
                confidence_zip_adjusted_wald_3_ordered_by_prevalence.csv","rb"),
                delimiter=",").astype(np.float32)

import matplotlib
import numpy as np
```
import matplotlib.pyplot as plt
from matplotlib2tikz import save as tikz_save

prevalence = mood_disorder_prevalence_confidence_3[:, 3:4]
margin_of_error = mood_disorder_prevalence_confidence_3[:, 4:5]
x = np.arange(prevalence.shape[0])

plt.figure(figsize=(14, 4))
plt.axis((-2, 92, 0.07, 0.42))
plt.xlabel('ZIP Codes')
plt.ylabel('Mood Disorder Prevalence')

plt.plot(x, prevalence)
plt.errorbar(x, prevalence, yerr=margin_of_error, fmt='-o', markersize=4,
            label="Mood Disorder Prevalence\ with Confidence Intervals")
plt.legend(loc='lower right')
tikz_save("[...]/Plot_Mood_Disorder_Prevalence_Margin_of_\ Error_3Percent.tex", figureheight='8cm', figurewidth='18cm')
plt.show()

Listing F.3: Python script that calculates the confidence intervals for a margin of error of three or below based on the patient coverage

F.1.2 Loading Data

import numpy as np
from sklearn.cross_validation import train_test_split

test_split = 0.2

loaded = np.loadtxt(open('mood_disorder_prevalence_\ confidence_30\_with\_nine\_features.csv','rb'),
delimiter=',').astype(np.float32)

X = loaded[:,2:]
y = loaded[:,1:2]

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=test_split, random_state=534654)

def logit(x):
    return np.log(x/(1-x))
def logistic(x):
    return np.exp(x)/(np.exp(x)+1)

Listing F.4: Python script than loads a dataset and transforms the data using the logit function

F.1.3 Train Model

ols_endog_all = logit(y)

ols_exog_all = sm.add_constant(X, prepend=False)

ols_all = sm.OLS(ols_endog_all, ols_exog_all)
res_ols_all = ols_all.fit()
print(res_ols_all.summary())

ols_endog_all = logit(y)
ols_exog_all = sm.add_constant(X, prepend=False)

ols_all = sm.OLS(ols_endog_all, ols_exog_all)
res_ols_all = ols_all.fit()
print(res_ols_all.summary())

Listing F.5: Python script that trains a regression model

F.1.4 Imputation

all_zips_with_features =
    np.loadtxt(open(' [...]/zip_with_nine_features.csv',"rb"),
               delimiter=",").astype(np.float32)

X_pred = all_zips_with_features[:,1:]
X_pred_const = sm.add_constant(X_pred, prepend=False)

y_pred_all = logistic(res_ols.predict(X_pred_const))
prstd_all, iv_l_all, iv_u_all = wls_prediction_std(res_ols, X_pred_const)
results = np.column_stack((np.ravel(all_zips_with_features[:, 0]),
                           y_pred_all, prstd_all, logistic(iv_l_all), logistic(iv_u_all)))

np.savetxt(open(' [...]/zip_with_predictions_confidence_intervals.csv','wb'), results)

Listing F.6: Python script that imputes the mood disorder prevalence
F.2 Crime Prediction

F.2.1 Load Data

```python
import numpy as np
from sklearn.cross_validation import train_test_split

test_split = 0.2

loaded = np.loadtxt(open('./all_standardizedcrime_pruned.csv', 'rb'),
                   delimiter=',').astype(np.float32)

X = loaded[:, 4:]
y = loaded[:, 1]

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                     test_size=test_split, random_state=9873453)
```

Listing F.7: Python script that loads the crime dataset

F.2.2 Training Model

```python
from sklearn.linear_model import LinearRegression
from sklearn import cross_validation

reg = LinearRegression()

scores_mse = cross_validation.cross_val_score(reg, X_train, y_train, cv=5,
                                            scoring='mean_squared_error')
scores_r2 = cross_validation.cross_val_score(reg, X_train, y_train, cv=5,
                                           scoring='r2')

print("Mean square error: %0.4f (+/- %0.4f") % (scores_mse.mean(),
                                              scores_mse.std() * 2))
print("R^2: %0.4f (+/- %0.4f") % (scores_r2.mean(), scores_r2.std() * 2))
```

Listing F.8: Python script that trains a linear regression model on the crime dataset

F.3 Patient History

```python
import cPickle
import scipy.sparse.linalg
import matplotlib.pyplot as plt

with open('./matrix_training.pkl', 'rb') as f:
    data = cPickle.load(f)
    f.close()
```
F.4. Mood Disorder Risk Model

F.4.1 Load Training and Test Data

```python
import numpy as np

data = np.loadtxt(' [...]/final_all_90percent_training.csv',
                   delimiter='\,').astype(np.float32)
data[:, 0] = 1 - data[:, 0]
X = data[:, 3:]
y = data[:, 0]

data_test = np.loadtxt(' [...]/final_all_90percent_test.csv',
                        delimiter='\,').astype(np.float32)
data_test[:, 0] = 1 - data_test[:, 0]
X_test = data_test[:, 3:]
y_test = data_test[:, 0]
```

Listing F.10: Python scripts that loads the training and test data for the mood disorder risk model

F.4.2 Train Models

```python
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import cross_validation

gb = GradientBoostingClassifier()

gb_score_log_loss = cross_validation.cross_val_score(gb, X, y, cv=5,
                                                      scoring='log_loss', n_jobs=5)
gb_scores_roc_auc = cross_validation.cross_val_score(gb, X, y, cv=5,
                                                      scoring='roc_auc', n_jobs=5)
```
```python
from sklearn.linear_model import LogisticRegression
from sklearn import cross_validation

lr = LogisticRegression()

lr_scores_log_loss = cross_validation.cross_val_score(lr, X, y, cv=5,
                                                  scoring='log_loss', n_jobs=5)
lr_scores_roc_auc = cross_validation.cross_val_score(lr, X, y, cv=5,
                                                  scoring='roc_auc', n_jobs=5)
lr_scores_accuracy = cross_validation.cross_val_score(lr, X, y, cv=5,
                                                  scoring='accuracy', n_jobs=5)
lr.fit(X, y)

lr_predictions_proba = lr.predict_proba(X_test)
lr_predictions = lr.predict(X_test)

from sklearn.externals import joblib
joblib.dump(lr, '[...]/all_90percent_lr.pkl.gz')

np.savetxt('[...]/all_90percent_lr_predictions_proba.csv',
           lr_predictions_proba, delimiter=',')
np.savetxt('[...]/all_90percent_lr_predictions.csv', lr_predictions,
           delimiter=',')
```

Listing F.12: Python scripts that trains a logistic regression model
F.4. Mood Disorder Risk Model

F.4.3 Evaluate Model

F.4.4 Load and Score

```python
1 data_test = np.loadtxt(’[...]final_cp_test.csv’,
                       delimiter=’,’).astype(np.float32)
y_test = 1 - data_test

2 predictions_proba =
                       np.loadtxt(’[...]all_90percent_lr_predictions_proba.csv’,
                       delimiter=’,’).astype(np.float32)
predictions = np.loadtxt(’[...]all_90percent_lr_predictions.csv’,
                       delimiter=’,’).astype(np.float32)

3 print(”Accuracy: %0.6f” % metrics.accuracy_score(y_test, predictions))
4 print(”Brier Score: %0.6f” % metrics.brier_score_loss(y_test,
                                                      predictions_proba[:, 1]))
5 print(”Log loss: %0.6f” % metrics.log_loss(y_test, predictions_proba[:, 1]))
6 print(”ROC AUC: %0.6f” % metrics.roc_auc_score(y_test, predictions_proba[:, 1]))
```

Listing F.13: Python script that loads the training model and evaluates the model against the test set

F.4.5 ROC

```python
1 %matplotlib inline
2
3 import matplotlib
4 import matplotlib.pyplot as plt
5
6 matplotlib.rcParams.update({’font.size’: 13})
7
8 fpr, tpr, thresholds = metrics.roc_curve(y_test, predictions_proba[:, 1])
9
10 plt.figure(figsize=(5, 3.5))
11 plt.plot(fpr, tpr, label=’ROC curve (area = %0.2f)’ %
           metrics.roc_auc_score(y_test, predictions_proba[:, 1]))
12 plt.plot([0, 1], [0, 1], ’--’, color=’0.8’)  
13 plt.xlim([0.0, 1.0])
14 plt.ylim([0.0, 1.05])
15 plt.xlabel(’False Positive Rate’)  
16 plt.ylabel(’True Positive Rate’)  
17
18 chosen_thresholds_x = []
19 chosen_thresholds_y = []
20
21 for i, threshold in enumerate(thresholds):
```
if i % 1700 == 700 and i < 16400:
    chosen_thresholds_x.append(fpr[i])
    chosen_thresholds_y.append(tpr[i])
    plt.annotate('%.2f'.format(threshold), xy=(fpr[i], tpr[i]),
                 xytext=(30, -10), ha='right',
                 textcoords='offset points')

plt.plot(chosen_thresholds_x, chosen_thresholds_y, 'o', mfc='none',
          mec='blue', markersize=5, label='Thresholds')
plt.legend(loc="lower right", num points=1, fontsize=13)
plt.savefig("[...]/all_90percent_lr_roc.pdf", bbox_inches='tight',
            format='pdf')
plt.show()

Listing F.14: Python script that plots a ROC curve

F.4.6 PR

```python
%matplotlib inline

import matplotlib
import matplotlib.pyplot as plt

matplotlib.rcParams.update({'font.size': 13})

precision, recall, thresholds = metrics.precision_recall_curve(y_test,
                                                                predictions_proba[:, 1])

plt.figure(figsize=(5, 3.5))
plt.plot(recall, precision, label='Precision-Recall Curve')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Recall')
plt.ylabel('Precision')

chosen_thresholds_x = []
chosen_thresholds_y = []

markers = [10000, 20000, 30000, 37000, 42500, 46000]

for i, threshold in enumerate(thresholds):
    if i in markers:
        chosen_thresholds_x.append(recall[i])
        chosen_thresholds_y.append(precision[i])
    if i == 10000:
        textpos = (10, -17)
```
F.4. Mood Disorder Risk Model

```python
else:
    textpos = (25, 6)
    plt.annotate('{:2f}'.format(threshold), xy=(recall[i], precision[i]),
                 xytext=textpos, ha='right',
                 textcoords='offset points')

plt.plot(chosen_thresholds_x, chosen_thresholds_y, 'o', mfc='none',
          mec='blue', markersize=5, label='Thresholds')
plt.legend(loc='upper right', numpoints=1, fontsize=13)

plt.savefig('...', all_90percent_lr_prc.pdf', bbox_inches='tight',
            format='pdf')
plt.show()
```

LISTING F.15: Python script that plots the precision-recall curve

### F.4.7 F-Test

```python
f_measure = (2 * (precision*recall)/(precision+recall))[::-1]

max_index = np.argmax(f_measure)

max_f_measure = f_measure[max_index]
max_threshold = thresholds[max_index]

plt.figure(figsize=(5,3.5))
plt.plot(thresholds, f_measure, label='F-Measure Curve')
plt.plot([0, max_threshold, max_threshold], [max_f_measure, max_f_measure, 0], '--', color='0.6')
plt.plot(max_threshold, max_f_measure, 'o', mfc='none', mec='blue',
         label='Maximum')
plt.annotate('{:2f}'.format(max_f_measure), xy=(0, max_f_measure),
             xytext=(15, 25), ha='left', textcoords='offset points',
             arrowprops=dict(arrowstyle='->'))
plt.show()
```
LISTING F.16: Python script that plots the f-measure curve
Bibliography


[AC98] Alan Agresti and Brent A Coull. “Approximate is better than "exact" for interval estimation of binomial proportions”. In: The American Statistician 52.2 (1998), pp. 119–126. ISSN: 00031305. DOI: [10.2307/2685469](http://www.jstor.org/stable/2685469?origin=crossref&seq=1#%5C%7Bpage%5C%7Dscan%5C%7Dtab%5C%7Dcontents)


[LS06] James R. Lewis and Jeff Sauro. “When 100% really isn’t 100%: improving the accuracy of small-sample estimates of completion rates”. In: Journal of Usability Studies 1.3 (2006), pp. 136–150. ISSN: 1931-3357.


[MO04] Vijaya Murali and Femi Oyebode. “Poverty, social inequality and mental health”. In: Advances in Psychiatric Treatment 10.3 (2004), pp. 216–224. ISSN: 1355-5146. DOI: 10.1192/apt.10.3.216. URL: http://apt.rcpsych.org/content/10/3/216.abstract


187
pp. 613–620. ISSN: 00010782. DOI: 10.1145/361219.361220. URL: http://portal.acm.org/citation.cfm?doid=361219.361220


