Application of Supervised Machine Learning to Predict the Mortality Risk in Elderly Using Biomarkers

Priyanka Sonkar
Technological University Dublin

Follow this and additional works at: https://arrow.tudublin.ie/scschcomdis

Part of the Computer Engineering Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie, vera.kilshaw@tudublin.ie.
Application of supervised machine learning to predict the mortality risk in elderly using biomarkers

Priyanka Sonkar

A dissertation submitted in partial fulfillment of the requirements of Dublin Institute of Technology for the degree of M.Sc. in Computing (Data Analytics)

August 2017
Declaration

I certify that this dissertation which I now submit for examination for the award of MSc. in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institutes guidelines for ethics in research.

Signed: [Signature]

Date: 31 August 2017
Abstract

The idea of long-term survival amongst older individuals has been a major medical and social concern. A wide range of biomarkers have been identified to prospectively predict disability, morbidity, and mortality outcomes in older adult populations. The machine learning techniques applied with clinically relevant biomarkers provide new ways of understanding diseases and solutions to tackle challenges to the health of the aging population.

This paper describes two supervised machine learning techniques, Logistic Regression (LR) and Support Vector Machine (SVM) which are used in the prediction of the mortality in elderly people. LR is one of the traditionally used predictive modeling methods in clinical research where the probability of occurrence of two classes is a dichotomous criterion whereas, SVM is an emerging classification supervised learning technique based on building models using maximum-margin hyperplane. An attempt has been made to measure the classifier accuracy of each model and the performance of both the models is compared on a set of biomarker features of old patients. The experimental result shows that the SVM model outperformed the LR model in the prediction of survivorship among old individuals with statistically significant results (p<0.01).

Keywords: Mortality prediction, Biomarkers, Supervised Machine Learning, Logistic Regression, Support Vector Machine
Acknowledgements

I would like to express my sincere gratitude towards my supervisor **Dr. Luca Longo** for his expert advice, constructive suggestions and enthusiastic encouragement and recommendations during the course of this research project.

I would also like to acknowledge **Dublin Institute of Technology academic staff (DIT)** especially my professors at School of Computing for their knowledge, help, guidance and kindness.

I would like to say thanks to my father **J.L. Sonkar**, my mother **Rama Devi**, my mother-in-law **Promila Kher** and my sisters **Archana & Alka** for their love, care, support and encouragement during my thesis.

Finally, I wish to say special thanks to my loving husband **Vamshi Kher**, who always believed in me and encouraged me to pursue masters in my area of interest. Your support and motivation gave me the strength to enhance my abilities and strive for academic and professional success.
Contents

Declaration I
Abstract II
Acknowledgments III
Contents IV
List of Figures VII
List of Tables IX
List of Acronyms X

1 Introduction 1
   1.1 Background ....................................................... 1
   1.2 Research project .................................................. 3
   1.3 Research objectives .............................................. 3
   1.4 Research methodologies ......................................... 4
   1.5 Scope and limitations ........................................... 4
   1.6 Document outline ................................................ 5

2 Review of existing literature 7
   2.1 Mortality risk prediction ......................................... 8
       2.1.1 Mortality ....................................................... 8
       2.1.2 Importance of mortality prediction ......................... 9
2.1.3 Mortality in elderly ........................................... 10
2.1.4 Mortality prediction techniques .............................. 11
2.1.5 Features to predict mortality ................................ 11
2.2 Bio-markers ......................................................... 12
  2.2.1 Theory ......................................................... 12
  2.2.2 Biomarkers application ...................................... 13
  2.2.3 Biomarkers based techniques ............................... 15
2.3 Machine learning ................................................ 16
  2.3.1 Unsupervised learning ...................................... 17
  2.3.2 Supervised learning ......................................... 19
2.4 Application of machine learning in mortality prediction using biomarkers ........................................... 21
2.5 Summary of literature, limitations and gaps .................... 23

3 Experiment design and methodology .............................. 25
  3.1 Business understanding ........................................ 27
  3.2 Data Understanding ............................................. 27
  3.3 Data preparation .................................................. 29
    3.3.1 SMOTE Algorithm: Balanced dataset .................... 29
    3.3.2 Normalization and Standardization: Z-score ............ 31
    3.3.3 One-Hot encoding: Categorical variables ............... 32
    3.3.4 Random Forest Classifier: Feature importance ......... 32
  3.4 Modeling .......................................................... 33
  3.5 Evaluation ........................................................ 35
  3.6 Strength and limitations ....................................... 38
  3.7 Summary of design ............................................. 39

4 Implementation and results ........................................ 40
  4.1 Business Understanding ........................................ 40
  4.2 Data understanding ............................................. 41
  4.3 Data preparation ................................................ 46
    4.3.1 SMOTE Algorithm: Balanced dataset .................... 46
List of Figures

1.1 Document Outline ........................................... 5

2.1 Literature review layout .................................... 7
2.2 Age trajectory of mortality .................................. 9
2.3 Validation of biomarkers .................................... 13
2.4 Causal pathway from initiation to occurrence of disease ........ 14
2.5 Causal pathway from initiation to occurrence of disease ........ 16
2.6 K-means clustering for heart disease patients ................. 18
2.7 Accuracy v/s Precision ...................................... 19
2.8 Schematic representation of the optimum hyperplane by a SVM .... 20

3.1 CRISP-DM model ........................................... 26
3.2 High level design of the research experiment ..................... 26
3.3 Sampling for Imbalanced data ................................ 30
3.4 SMOTE algorithm KNN approach ............................. 31
3.5 Standard normal distribution (Bell curve) ....................... 32
3.6 Support Vector Machine .................................... 34
3.7 Support Vector Machine kernels ............................... 34
3.8 Stratified 5-fold cross validation technique ...................... 35
3.9 ROC-AUC curve ........................................... 37

4.1 Statistical Analysis of data ................................. 41
4.2 Distribution plot of numeric features with target ............... 42
4.3 Distribution plot of numeric features with target ............... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Distribution plot of categorical features with target</td>
<td>44</td>
</tr>
<tr>
<td>4.5</td>
<td>Heatmap matrix of variables with Target</td>
<td>45</td>
</tr>
<tr>
<td>4.6</td>
<td>Positive-Negative correlation with Target</td>
<td>46</td>
</tr>
<tr>
<td>4.7</td>
<td>ROC curve: Logistic Regression</td>
<td>49</td>
</tr>
<tr>
<td>4.8</td>
<td>ROC curve: Support Vector Machine - Linear</td>
<td>51</td>
</tr>
<tr>
<td>4.9</td>
<td>ROC curve: Support Vector Machine - Radial</td>
<td>51</td>
</tr>
<tr>
<td>4.10</td>
<td>ROC curve: Support Vector Machine - Polynomial</td>
<td>52</td>
</tr>
<tr>
<td>4.11</td>
<td>ROC curve: Logistic Regression</td>
<td>53</td>
</tr>
<tr>
<td>4.12</td>
<td>ROC curve: Support Vector Machine - Linear</td>
<td>54</td>
</tr>
<tr>
<td>4.13</td>
<td>ROC curve: Support Vector Machine - Radial</td>
<td>54</td>
</tr>
<tr>
<td>4.14</td>
<td>ROC curve: Support Vector Machine - Polynomial</td>
<td>55</td>
</tr>
<tr>
<td>5.1</td>
<td>Model comparison using box-whisker plot: Unbalanced dataset</td>
<td>58</td>
</tr>
<tr>
<td>5.2</td>
<td>Model comparison using box-whisker plot: Balanced dataset</td>
<td>59</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Biomarkers information ........................................ 28
3.2 Confusion Matrix .................................................. 38
4.1 Missing Value Analysis ............................................. 42
4.2 SMOTE oversampling .............................................. 47
4.3 One-hot Encoding ................................................... 48
4.4 SVM kernels and tuning parameters ............................. 50
4.5 SVM kernels and tuning parameters ............................. 53
5.1 AUROC on imbalanced and balanced target (mortality risk) ... 59
5.2 Statistical significance of experimental results .................. 60
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>Area Under the Curve</td>
</tr>
<tr>
<td>CRISP-DM</td>
<td>Cross Industry Standard Process for Data Mining</td>
</tr>
<tr>
<td>CV</td>
<td>Cross Validation</td>
</tr>
<tr>
<td>EMR</td>
<td>Electronic Medical Records</td>
</tr>
<tr>
<td>FDA</td>
<td>Food and Drug Administration</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
<tr>
<td>GRSH</td>
<td>General Self-Rated Health</td>
</tr>
<tr>
<td>HBV</td>
<td>Hepatitis B Virus</td>
</tr>
<tr>
<td>IADL</td>
<td>Instrumental Activities of Daily Living</td>
</tr>
<tr>
<td>ICU</td>
<td>Intensive Care Unit</td>
</tr>
<tr>
<td>KNN</td>
<td>K Nearest Neighbour</td>
</tr>
<tr>
<td>LOO</td>
<td>Leave One Out</td>
</tr>
<tr>
<td>LR</td>
<td>Logistic Regression</td>
</tr>
<tr>
<td>MSIA</td>
<td>Mass Spectrometric Immunoassay</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>ROM</td>
<td>Risk of Mortality</td>
</tr>
<tr>
<td>SNP</td>
<td>Single Nucleotide Polymorphism</td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>SMOTE</td>
<td>Synthetic Minority Over-sampling Technique</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

In high-income countries, much of the health care spending are devoted to diseases where breakthroughs in diagnosis and treatment are elusive. Chronic diseases such as cardiovascular diseases, cancer, diabetes and chronic respiratory problems are serious challenges to the health of aging populations and health care budgets. The survey conducted by World Health Organisation in 2005\footnote{https://www.oecd.org/health/biotech/49023036.pdf} estimated that 60% of the global mortality is due to chronic disease out of which 80% of this mortality occurred in low and middle-income countries (Tunstall-Pedoe 2006).

Health challenges associated with aging is a major medical and social concern. Risk prediction functions have been used to generate risk scores for age-related macular degeneration, cardiovascular disease, Alzheimer, cancer, stroke or multi-organ failure leading to decreasing life expectancy (Rosero-Bixby & Dow 2012). Epidemiological observational studies have been conducted to quantify prevalence and risk factors for disease (Dipnall et al. 2016).

The biomarkers have been in use for centuries and play a crucial role in improving understanding of disease mechanisms and response of patient towards a particular therapy. Biomarkers are the genes or measurable indicators of a biological state used to diagnose disease state, stage of illness, risk, or treatment response (Dipnall et al.)
The clinical use of biomarkers in drug development and medical diagnosis has recently accelerated the improvement in global health equity.

Although, biomarkers provide a powerful and dynamic approach in clinical studies to understanding the spectrum of chronic disease or future death, it is necessary to identify potential biomarkers out of thousands of biomarkers of a patient. Diagnostic tests are required for the translation of a biomarker into a clinically relevant biomarker, which includes generation and analysis of significant amount of data and knowledge. The generation of information is often logistically difficult, expensive and time-consuming. To overcome the hurdles associated with the generation of large amount of necessary data for the development of biomarkers, analysis techniques have been introduced.

The application of data mining techniques incorporating machine learning algorithms are rising across many disciplines over last two decades (Dipnall et al., 2016). In recent years, data mining techniques are used in the field of health care to analyze the massive amount of data to improve understanding of the disease and provide information on the presence of disease or susceptibility to disease, in an individual, or monitor patient response to therapeutic interventions. Statistical classification techniques and Supervised Machine Learning algorithms such as Support Vector Machine, Neural Networks, Logistic Regression and Decision trees are commonly used to build predictive models for medical diagnosis (Maroco et al., 2011).

Traditionally, these predictive models had been implemented with organ-specific functional indicators (Rosero-Bixby & Dow, 2012) such as handgrip strength, walking speed, pulmonary peak flow which are associated with the mortality risk in a patient. The recognition of features that serve as early indicators in aging prediction is required to predict long-term survival in elderly (Pritom, Munshi, Sabab, & Shihab, 2016). To increase the prediction accuracy of the models, new variables such as biomarkers of the patient are used to build the models. Biomarkers applied with machine learning have the potential to transform the current health care model, from a reactive approach to one which is more proactive.
1.2 Research project

The aim of the project is to compare the performance of two supervised machine learning techniques, Logistic Regression and Support Vector Machine for the classification of a group of elderly patients having mortality risk or not. The sample data set contains information about the biomarkers of the patient which is collected during the time span of three years by a doctor while monitoring her patients. The biomarkers of the patient are the independent feature variables which will be used to predict the target variable i.e. risk of mortality. Mortality risk predictive models will be built and later the comparison will be done based on the performance of the classifiers that have been trained on the same set of biomarkers data. Average accuracy of the classifiers will be used for evaluating the classifiers empirically. The research question answered from this project can be stated as:

“Can accuracy of models build with Support Vector Machine (SVM) for the prediction of mortality risk in elderly using biomarkers of the patients outperforms models build with Logistic Regression (LR) ?”

The results of the study would act as a proof of concept for the use of Support Vector Machine algorithms upon regression algorithms in the medical diagnosis.

1.3 Research objectives

The research objective of the project is to determine whether predictive models build with Support Vector Machine can yield greater classification accuracy than Logistic Regression model. In order to achieve the goal of the study, few experiments are performed as mentioned below:

- Explore the existing knowledge base on the application of machine learning in mortality risk prediction using biomarkers and perform a comprehensive analysis.

- Select a sample of data set having information about the biomarkers of the patients.

- Perform cleaning, feature scaling and feature engineering on the dataset.
CHAPTER 1. INTRODUCTION


- Compare models and select best prediction model evaluated based on the average classification accuracy in predicting the mortality risk.

1.4 Research methodologies

The study is focused on the comparison of two well-known machine learning algorithms in predicting survivorship among older individuals and hence comes under secondary research. The experiment will be performed on the existing dataset and no new data is collected for this study. As part of the Secondary research, literature related to similar research conducted on supervised machine learning techniques, mortality risk prediction, feature engineering for variable selection and biomarkers are studied to complete a comprehensive literature review.

The research methodologies used are Quantitative (Epidemiological) and empirical research. The experiment will be conducted to produce concrete results which will be further taken into consideration to address the hypothesis. The performance of the classifiers will be evaluated by comparing the average classification accuracy of each classifier. On the basis of the result, we will conclude which model turned out to be a better predictor and hence Inductive.

1.5 Scope and limitations

The study will focus on the prediction of the mortality risk by implementing machine learning techniques on the biomarkers of the patients. The dataset used for the experiment contains 37 biomarkers to study the impact of potential biomarkers on the chances of survival of an individual. Two supervised machine learning techniques, Logistic Regression and Support Vector Machine will be used to build the models using biomarkers as independent features to predict the mortality risk (Verplancke et al., 2008). The models will be trained and then compared to discuss how they vary in
both their prediction performance and accuracy.

A limitation of our study is the relatively small sample size, only 93 records of patients are available for carrying out the experiment. Also, few of the predictor feature variables are not normalized. This can have a knock-on effect on the learning capability of the model as this may induce biasing in the results hence needed to be handled.

Furthermore, trends in the dependent variable (mortality risk) is again biased. There are approximately 20% of the records containing information with mortality risk as true, whereas 80% of the records have mortality risk as false. Data upscaling technique, SMOTE sampling will be used for data preprocessing while performing the experiment to get rid of small-sized biased data. Also, the research study will target only the elderly population and since mortality varies largely between age groups, our results cannot be automatically extrapolated to younger populations.

1.6 Document outline

The layout of the report will be outlined as shown in figure 1.1 with six chapters in total, each divided into sections:
• Chapter 2 (Literature review) gives an overview of the current state-of-the-art relating to mortality prediction, biomarkers and supervised machine learning, primarily Logistic Regression and Support Vector Machine. A systematic literature review has been conducted to gain an insight into the related work done in the prediction of the mortality risk in old age persons using supervised machine learning algorithms. The gaps in the existing research will be highlighted to propose the research question for this project.

• Chapter 3 (Experimentation design and methodology) will elaborate the design of the experiment having an explanation of each experiment to be performed and the motivation behind each step. A complete understanding of the data to be used will be outlined. Various techniques proposed to be implemented as part of the research will be listed and described in this chapter.

• Chapter 4 (Implementation and results) details the practical implementation of the experiments and describes the results obtained at each step.

• Chapter 5 (Discussion on findings) will provide the detailed analysis of the experimental results and based on the results, a decision will be made to either accept or reject the hypothesis. Lastly, this chapter will also elaborate the strength and weakness of the research.

• Chapter 6 (Conclusion) will summarize the findings of the research undertaken during the dissertation including problem definition, critical analysis of the design, experiments conducted, evaluation of the results and scope for the further research.
Chapter 2

Review of existing literature

This chapter provides a detailed review of the relevant literature to gain an insight into the related work done in the prediction of mortality risk in elderly using supervised machine learning and biomarkers. The review is broadly classified into three sections: mortality prediction, biomarkers and machine learning as shown in figure 2.1.

Figure 2.1: Literature review layout
The state-of-art of the mortality risk prediction is presented in section 2.1 including the general theory about mortality, importance to predict mortality, various mortality risk prediction techniques and the features used to predict the mortality risk. The next section 2.2 provides details about biomarkers, explaining what are biomarkers, the application of biomarkers the and mortality risk prediction techniques based on biomarkers. The information and application of machine learning techniques in mortality risk prediction is given in section 2.3 which is further divided into unsupervised and supervised algorithms.

Last section 2.4 is the reflection of analysis of existing research around mortality risk prediction using machine learning algorithms with biomarkers. This analysis will help in finding the limitations and gaps in the existing research and motivation for the proposed research question.

2.1 Mortality risk prediction

2.1.1 Mortality

The risk of mortality (ROM) estimates the likelihood of death of a patient and provides a medical classification of patient mortality. Human mortality and the physiological processes of aging are fundamentally linked together and used in the process used to determine biological aging in human populations. The linkage between age and mortality is termed as the age dependence of mortality and is represented by Gompertz function (Manton 1999).

The increased scientific knowledge has increased the average length of life or life expectancy in the developed or developing countries leading to weakening of this link-age (Manton 1999). Figure 2.2 shows the age and mortality relation. The quadratic function is plotted for four ages (50, 59, 68 and 95 years) to analyze the impact of age on risk factor. Below figure 2.2 illustrates that the risk factor-age interaction is weak at age 50 and is maximum at age 95, infers that as age increases, mortality risk factor also increases.
CHAPTER 2. REVIEW OF EXISTING LITERATURE

2.1.2 Importance of mortality prediction

Several studies have been conducted to assess the general health of a person based on the survey rating provided by individuals as a response to the question how would you rate your health? excellent, very good, good, fair or poor. The score is useful in finding a rough estimate of the individuals who are not in a healthy condition and are seeking for medical assistance. As, per the experiment performed by DeSalvo, Bloser, Reynolds, He & Muntner in 2006, it is found that there is a statistically significant relationship between general self-rated health (GSRH) and high risk of mortality. Individual with poor GSRH had a 2-fold higher mortality risk as compared to the person with self-rated health as excellent (DeSalvo, Bloser, Reynolds, He, & Muntner, 2006).
Health planners and policy makers are trying to find out a feasible method to identify the most vulnerable person with highest health requirements. GSRH can be used to improve the health care given to a patient through the identification of groups who are more susceptible to mortality risk. The collection of such data may help in offering a beneficial tool in health and care planning sector.

### 2.1.3 Mortality in elderly

The recognition of factors that contribute to healthy aging helps in the prediction of long-term survivorship among older individuals. A study conducted by Swindell et al. (2010) to identify predictors of long-term survival in older women. Individuals exhibiting healthy aging patterns maintain a high quality of life with very few daily living impairments whereas an unhealthy aging trajectory is associated with diminished quality of life and increased mortality rate. The efficiency and quality of care can be improved by identifying at-risk individuals by predicting future health outcomes.

Prognostic information about the life expectancy of older people is important in clinical decision making and providing pro-active care to older people with limited life expectancy. In 2016, Houwelingen et al. conducted a study on 85-year-old inhabitants and collected samples over a time period of five years for mortality prediction. The collected sample of routine laboratory measurements was compared with gait speed and Instrumental Activities of Daily Living (IADL). It is concluded that the mortality risk predicted using laboratory tests are as accurate as models based on gait speed or IADL disability (van Houwelingen et al., 2013). In elderly aged > 85 years, gait speed and IADL are predictors of survival where slow gait speed is associated with high risk of mortality. Also, the study supports that the poor ability in IADL was associated with a significantly higher 2-year mortality risk (Taekema, Gussekloo, Westendorp, de Craen, & Maier, 2012).
2.1.4 Mortality prediction techniques

Prediction of future health states can be significant in the medical domain as it can contribute to early detection of a disease, effective treatment and prevention. The health-related information of any individual is stored in Electronic Medical Records (EMRs) and can be used to generate accurate predictions for the occurrence of health issues in an individual and increasing risk of mortality. The two main approaches used for the prediction of mortality risk (Hoogendoorn, el Hassouni, Mok, Ghassemi, & Szolovits, 2016): (1) build a mortality predictive model based on the temporal features extracted from EMRs, and (2) defining a patient similarity matrix and predicting risk using the outcome of the similar patient.

2.1.5 Features to predict mortality

In order to extract the information contained in EMR data variety of approaches have been developed so as to fully exploit the wealth of information. The focus is on the extraction of features from EMRs and further to use these features for the generation of predictive models. As per Hoogendoorn, el Hassouni, Mok, Ghassemi & Szolovits these features can be implemented in commonly used classification approaches such as logistic regression or on defining patient similarity using instance-based learning such as K-nearest neighbor (KNN) approach. To select the most promising features correlation coefficient is calculated to find out the most correlated attributes. An iterative process is used where features with the highest correlation with the target (mortality risk) are selected (Hoogendoorn et al., 2016). A number of studies have been conducted to study the features which are responsible for the high mortality risk in patients. Previous studies have shown that there is a causal relationship between the social relationships and the mortality rate (Holt-Lunstad, Smith, & Layton, 2010). Bixby and Dow explored few other most powerful mortality predictors: hand-grip strength, pulmonary peak flow, walking speed, blood markers, blood pressure and cholesterol level (Rosero-Bixby & Dow, 2012).
2.2 Bio-markers

2.2.1 Theory

The National Institutes of Health Biomarkers Definitions Working Group in 1998 defined biomarker as “a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacological responses to a therapeutic intervention” (Strimbu & Tavel, 2010). Biomarkers come under the umbrella of medical signs which stands for the indicators or symptoms which help in identifying the health of a patient. Biomarkers include almost everything, the complex laboratory tests such as tissues or blood cells to the simplest characteristics such as pulse rate or blood pressure. Biomarkers are also considered as surrogates for individual clinical end-points (Strimbu & Tavel, 2010), which implies that a biomarker is capable of accurately predicting the clinical outcome i.e. physical or mental traits of a person.

The evaluation of biomarkers is necessary to identify it as surrogate endpoints which is done based on its relevance and validity. The relevance of a biomarker refers to its ability to provide clinically relevant information to health-care providers or to the public whereas validity refers the effectiveness of a biomarker as a surrogate endpoint. As per the studies conducted by Strimbu and Travel, the U.S Food and Drug Administration (FDA) has approved the use of biomarkers as surrogate endpoints in the treatment development process.

Groopman and Kensler developed a validation model to study development of hepatocellular carcinoma on exposure to aflatoxin B1 in humans (Bonassi, Neri, & Puntoni, 2001). This approach was designed for chemical-specific biomarkers, and is applicable to small scale only. Further, to provide a more general approach for the validation of biomarkers as early predictor of disease on humans, three steps process is designed as shown in figure 2.3.
CHAPTER 2. REVIEW OF EXISTING LITERATURE

The first step is biomarker development which includes the evaluation of sensitivity, sensibility, reliability and accuracy of the assay. The second step evaluates the variability of a biomarker in the human population. The last step is to assess the causal relationship between a disease and a biomarker. The validation process concentrates on those biomarkers that are directly responsible in the prediction of a disease or mortality. The closer a biomarker to the causal pathway, more precisely it will help in the disease prediction (Bonassi et al., 2001).

2.2.2 Biomarkers application

The major challenge of modern epidemiological research is to investigate the cause of a disease or death due to exposure to a potentially harmful substance (Bonassi et al., 2001). The knowledge of the harmful compound causing health issues can help in interrupting or eradicating a disease on time. In 1990, research has been conducted on biomarkers and molecules of individuals that lead to illness or disease are measured. The most promising feature of a biomarker is the ability to predict a disease and monitor public health. It plays a significant role in clinical research and medicines by providing insights into pathogenic mechanisms. The events that are on the direct
CHAPTER 2. REVIEW OF EXISTING LITERATURE

pathways from the initiation of the occurrence of a disease is shown in below figure 2.4. As suggested by Bonassi & Au, longitudinal studies are often required to study and evaluate the impact of biomarkers on clinical endpoints such as recurrence of cancer or survival (Bonassi & Au, 2002). As a substitute of long-term studies surrogate endpoints are created which are measured in short period of time.

Figure 2.4: Causal pathway from initiation to occurrence of disease

(Bonassi & Au, 2002)

Biomarkers such as CD4 cell counts and HIV viral loads are proved beneficial for anti-AIDS treatments. Researchers have done an analysis to discover the root cause behind liver cancer and it is found that infection with hepatitis B virus (HBV) and exposure to a food-borne mutagen, aflatoxin (AFB1), are the major risk factors for liver cancer. Another example is to study the relationship between cigarette smoking and lung carcinogenesis. Almost 50% of the tumors having P53 gene and P53 mutation initiates lung cancer. Further clinical experiments showed that benzo[a]pyrene causes the mutation of P53 gene causing lung cancer (Bonassi & Au, 2002). Although biomarker helps in determining the potential treatments that are worth the efforts and resources, there is always a risk of false negative and false positives (Strimbu & Tavel, 2010). In terms of biomarkers, false negatives indicate the biological process
that led to improved clinical outcomes and is not captured by the biomarkers whereas, false positives indicate that biomarkers when used as surrogate outcomes do not predict true clinical outcomes. Therefore, clinical endpoint analysis is important with biomarker analysis to avoid over reliance only on biomarkers.

2.2.3 Biomarkers based techniques

Over the years, epidemiological researchers have fused biomarkers with a number of techniques to recognize key biomarkers associated with a disease, leading cause of mortality among the patients. In 2011, Buddi, Taylor, Borges, & Nelson explored a laboratory based technique of Mass Spectrometric Immunoassay (MSIA) and discovered that MSIA is used to analyze potential biomarkers. MSIA is based on immunoassay technique which is used in biochemistry to detect an analyte in a solution. The first step of the process of MSIA is to attach antibodies to the MSIA tip. These antibodies are incubated with the antigen and antibody-antigen complex is washed repetitively to get rid of any non-specific antibody. Later, the bound antigen molecules are removed and antigens are extracted onto a mass spectrometer probe. This solution is dried and then molecular variants present in the sample are identified (Buddi, Taylor, Borges, & Nelson, 2011). The detailed process of MSIA is shown in figure 2.5. These clinically relevant biomarkers have a tendency to make distinctions with higher accuracy if used with machine learning techniques such as support vector machine classifiers.

Machine learning boosted regression algorithm and logistic regression can also be used to identify potential biomarkers (Song, Mitnitski, Cox, & Rockwood, 2004). Supervised machine learning techniques are now widely used to investigate the prediction of mortality using biomarkers along with non-linear techniques such as Support vector machines, artificial neural networks etc. Supervised and unsupervised machine learning techniques are further elaborated in following section 2.3.
2.3 Machine learning

Machine learning is a method of building analytical models by first performing data analysis to find hidden insights and using algorithms that iteratively learn from the historical data and helps in predicting unseen data. Machine learning offers advantages over statistical methods used for predictions i.e. easing the process of knowledge acquisition from a system or reducing the time consumption \citep{pietersma2003}.

Day-to-day activities are comprised of machine learning techniques including, fraud detection, credit scoring, real time ads, pricing models, email spam filtering, text based sentimental analysis, medical diagnosis, pattern and image recognition, web search results and so on.

The biggest challenge in the field of health and sciences is to extract useful information from a large volume of data. Machine learning is now emerging as a solution
CHAPTER 2. REVIEW OF EXISTING LITERATURE

for the problem. Machine learning modelling methods such as supervised classification, clustering, probabilistic graphical methods for knowledge discovery, deterministic and stochastic heuristics for optimization are used in bioinformatics (Larranaga et al., 2006). According to the research conducted by Kononenko, the quality of the result depends on the selection of the classifier and concluded that combination of classifiers are more reliable in diagnostic systems problems instead of single classifier (Kononenko, 2001). Also, the classification performance is highly impacted by data preprocessing and tuning of algorithms (Pietersma et al., 2003).

The following sub-sections will give an overview of different classifiers which are broadly categorized as unsupervised and supervised machine learning techniques and their application in the prediction of the mortality risk among elderly.

2.3.1 Unsupervised learning

Unsupervised machine learning is used to draw inferences from unlabelled datasets. Cluster analysis is the most common unsupervised learning method to find grouping in data. The clusters are created based on the similarity measure such as Euclidean or probabilistic distance. The information in the resultant clusters are sorted in a way such that there are high intra-cluster similarity and low inter-cluster similarities (Gupta, Thakral, & Sharma, 2016).

Unsupervised algorithms are applied in data mining for pattern mining, sequence analysis and genetic clustering in bioinformatics, web mining, voice mining, medical imaging, and object recognition.

Nithya, Duraiswamy & Gomathy described few commonly used clustering techniques in their paper (2013). Hierarchical clustering has cluster nodes arranged in the form of a dendrogram containing child cluster, sibling clusters and parent node. Hierarchical clustering is further divided into agglomerative and divisive clustering. Inter-cluster similarity coefficients are calculated in agglomerative clustering and two similar clusters are merged into a new cluster. A divisive clustering has one cluster of all the data points at the beginning and further this cluster is split and recursive splitting is applied until each pattern has its own singleton cluster (Nithya, Duraiswamy, 2013).
K-means clustering is another most favorable unsupervised clustering technique due to its elucidation simplicity, easy execution and flexibility to sparse information. According to Gupta, Thakral & Sharma normalization techniques may be used to improve the accuracy of k-means clustering, by clustering the remaining non-clustered points. K-means clustering is used to group persons into high and low risk of having heart disease by creating clusters of high blood pressure and cholesterol level as shown in figure 2.6.

The major drawback of k-means clustering is that it requires a number of clusters in advance and doesn’t fit to identify non-convex shaped clusters.

Furthermore, existing researches in genetics revealed that to understand the biological processes, gene expression levels are measured at different development phases of different organisms, clinical conditions or body tissues. Traditional approaches to genomic research is based on the examination and collection of data from a single gene which is now extended to monitoring hundreds and thousands of genes under one experiment. Cluster analysis is used to analyze the tremendous amount of data obtained from micro-array studies (Tasoulis, Plagianakos, & Vrahatis, 2004). To improve the quality of clustering it is necessary to identify genes which have significant contribution.
Wang, Songtag & Wang in 2014 presented a disease progression modelling technique based on unsupervised learning. The proposed approach is flexible to accommodate new sources of data and further work can be done to modify existing model to learn a mixture of disease trajectories (Wang, Sontag, & Wang, 2014).

Similarly, supervised learning is another widely used machine learning technique in predictive modelling which is discussed in 2.3.2.

2.3.2 Supervised learning

Supervised classification technique is the most frequently carried out by Intelligent Systems and is applicable on the instances having labeled data. The supervised learning is focused not just only on constructing model by capturing input features but also to provide robust prediction of the output i.e. accuracy and precision (Palmer & Chakravarty, 2014). Both accuracy and precision sound like the same thing but there is a major difference between both as accuracy refers to the lack of bias in predictions whereas precision refers to the lack of variance as shown in figure 2.7.

The equation of a model that can be used to learn from existing data and to predict the outcome of new future data can be written as,
\[ Y = f(X, \theta) + \epsilon \] (2.1)

where, \( f(X, \theta) \) represents a function that maps input X to input Y (Palmer & Chakravarty, 2014).

Binomial Logistic Regression (LR) is the traditional standard predictive modeling method used in clinical research and studies. LR models are basically used in building predictive models where the probability of occurrence of one of the two classes is a dichotomous criterion. Logit transformation gives linear combination of predictor and is given by below equation,

\[
\ln\left(\frac{\hat{\pi}_i}{1 - \hat{\pi}_i}\right) = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_p X_{pi}
\]

Figure 2.8: Schematic representation of the optimum hyperplane by a SVM (Maroco et al., 2011)

New artificial intelligence methods are emerging for classification purposes in medical diagnosis. Support Vector Machines (SVMs) are the newest supervised machine learning technique which revolves around the notion of a ‘margin’ i.e. a hyperplane as shown in figure 2.8 which separates the two data classes. Maroco et al. provided
insights that the generalization error can be reduced by maximizing the margin in
order to create the largest possible distance between the separating hyperplane and
the instances on either side of the hyperplane (Maroco et al., 2011).

Maroco et al. proposed a new hypothesis in the prediction of Dementia and cogni-
tive impairment and concluded that an instance is classified into success group if the
estimated probability is greater than 0.5 otherwise it is classified into failure group.
The experimental results showed that newer statistical classification models such as
neural networks, support vector machine and random forests results into improved
accuracy, sensitivity and specificity (Maroco et al., 2011).

Similar researches have been done to compare the accuracy of the machine learning
models in the prediction of mortality in critically ill patients. In 2008, Verplancke
compared support vector machine and logistic regression models after trained these
models using 252 patient records and validated it using rest 100 records. The area
under the curve is used for validating the performance of the models and also statistical
significance of the results is measured. The accuracy of both the classifiers comes out
to be comparable but SVM has the possibility to improve patient care in near future by
facilitating data modeling in the Intensive Care Unit (ICU) (Verplancke et al., 2008).

2.4 Application of machine learning in mortality
prediction using biomarkers

Gathering insights from the above sections, this section will provide information on the
machine learning applications collaborated with biomarkers in mortality risk prediction
in old individuals.

Machine learning algorithms are used to analyze medical datasets from the very
beginning. The implementation of data mining techniques in medical field has im-
mense potential for exploring hidden patterns in the data and utilize these patterns
for clinical diagnosis. Medical diagnosis is the most complex task as it needs to be
precise. This section will focus on the contribution of machine learning techniques in
the area of research in medicine and epidemiology. Rose used prediction methods to
generate risk scores and risk prediction functions for timely detection of disease and to provide effective treatments to the patients based on the predicted risk of a disease or death (2013). Machine learning techniques are widely used to generate risk scores for heart disease, breast cancer, strokes and age-related physical performance. Prediction practices have initially relied on Parametric regression methods but now newer machine learning methods such as random forests and neural networks are implemented for prediction. Researchers are working on the application of different techniques and investigating that which individual algorithm should be implemented on different datasets to derive better performance (Rose, 2013).

The increasing number of covariates including biological, clinical and genomic data, demands flexible algorithms that may capture potential predictor features. Therefore, dimensionality reduction or feature selection techniques has become an apparent need in the bioinformatics applications. Both feature selection and dimensionality reduction aim towards reducing the number of features and can be applied to both supervised and unsupervised learning. The most important objectives of feature selection identified by Saeys, Inza & Larrañaga are (a) improve model performance by avoiding overfitting in the model, (b) to provide faster and cost-effective predictive model (c) to gain deeper insights of data generation and selection process (2007). However, the only overhead is that feature selection techniques increase the complexity of the model by introducing an additional layer in the modeling (Saeys, Inza, & Larrañaga, 2007).

Hamid, Omar & Mabrouk in 2016 implemented correlation-based feature selection and chi-squared feature selection machine learning methods to find the most effective Single Nucleotide Polymorphisms (SNPs). These markers are useful in identifying geriatrics diseases (El Hamid, Omar, & Mabrouk, 2016) and achieved almost 76.70% accuracy using significant SNPs features. Similarly, Ding, Li & Wang proposed combination of just-in-time learning (JITL) and extreme learning machine (ELM) in order to improve mortality prediction of ICU patients. JITL-ELM showed better performance in terms of area under the curve than neural network, logistic regression model and traditional score models (Ding, Li, & Wang, 2016).
2.5 Summary of literature, limitations and gaps

A review is presented in this chapter relating to the application of machine learning to identify the clinically relevant predictors for the prediction of geriatric chronic diseases leading to the mortality risk. Initially, literature related to the mortality risk are studied to gain knowledge about the risk of mortality in elderly, the importance of mortality predictions and the features for mortality prediction followed by biomarker’s literature focusing on biomarkers application and techniques. In machine learning section emphasis is given on two supervised machine learning techniques, Logistic Regression (LR) and Support Vector Machine (SVM) which are applied in the area of medical sciences.

Non-linear techniques such as support vector machine with polynomial kernels and radial basis functions (RBF) need to be investigated in future by researchers. These relationships are desirable to be discovered but a prior experiment is needed to interpret the importance of nonlinear techniques on a particular dataset (Song et al., 2004). Also, Delen, Walker & Kadam suggested that better promising methods such as support vector machines and rough sets can be used over decision trees, artificial neural networks and logistic regression in order to improve the prediction accuracy (Delen, Walker, & Kadam, 2005). In their further research, they proposed to implement a hybrid intelligent system. In a hybrid system, data mining results would be augmented with expert opinions and captured into an expert system. Liu emphasized on the comparison of different models to obtain better results (Liu, 2007).

According to the simulations performed by Liu, variable selection and survival function estimation of highly correlated features can prove beneficial for medical diagnosis (Liu, 2007). Hamid, Omar and Mabrouk worked on the identification of genetic biomarkers associated with Alzheimer’s disease. Different gene selection methods to discover new biomarkers using machine learning algorithms can be further explored as Phase II of this work (El Hamid et al., 2016). Fuzzy rule extraction method is used by Pal in 2007 to find a small set of most important features to build the model out of thousand of available features. It is found that different set of algorithms identify
different sets of important features hence investigations should be undertaken to set a threshold for the modulator values \cite{Pal2007}. The identification of the most effective biomarkers and accuracy of classifier on the number of features is suggested by Buddi, Taylor, Borges & Nelson for future research \cite{Buddi2011}. Biomarkers such as hypertension, hypercholesterolemia and stress can be explored and used for prediction \cite{Gruenewald2006, Roserobixby2012}.

The gaps and limitations are addressed by proposed research question which is given as,

"Can accuracy of models build with Support Vector Machine (SVM) for the prediction of mortality risk in elderly using biomarkers of the patients outperforms models build with Logistic Regression (LR)?"

Following sections will elaborate the design, methodology and experiments performed to answer the research question.
Chapter 3

Experiment design and methodology

This chapter will elaborate the plan and design of the experiments proposed to answer the research question. The standard approach of the CRISP-DM process (Chapman et al., 2000) will be implemented which is a structured data mining project planning methodology. The experiment will be carried out in five phases i.e. business understanding, data understanding, data preparation, modeling and evaluation. All the experimental steps will be performed in Python language using scikit-learn library. Scikit-learn (Scikit-learn.org, 2017) is an open machine learning library featuring various classification, regression and clustering algorithms.

The main aim of the research is to build a mortality risk predictive model based on the historical data of biomarkers of elderly patients which has been collected over a time span of three years. The chapter is divided into sub-sections adhere to CRISP-DM framework as shown in figure 3.1. Subsequent subsections will provide detailed information of the experiment.
Figure 3.1: CRISP-DM model

(Chapman et al., 2000)

The high-level experiment of the research is illustrated in below figure 3.2

Figure 3.2: High level design of the research experiment
3.1 Business understanding

The key focus of the study is to compare two supervised machine learning algorithms for the prediction of mortality risk. In order to achieve the primary goal of the research two models will be trained on the data containing biomarker features of the patients. Later the performance of both the models will be evaluated using average classification accuracy of the models. The following hypothesis will be taken into consideration to address the research question:

“H0 The accuracy of the model built with Support Vector Machine outperforms Logistic Regression for prediction of the mortality in elderly using bio-markers of the patient, with p-value <0.01.”

3.2 Data Understanding

The dataset contains information about the mortality risk factors in community dwelling elderly which had been collected from a European hospital during a time span of three years. This dataset consists records of 93 patients and information about 37 different biomarker features of each individual as shown in table 3.1. The age-group of the patients are divided into six categories i.e. <60, 60-65, 66-70, 71-75, 76-80, >80.

The mortality risk of the patient is the dependent variable and may take values either ‘0’ or ‘1’. The predictor variables are both continuous and categorical in nature. Table 3.1 contains the summary of the dataset, where attributes are the biomarkers and type represents the data type of each variable i.e. ‘N’ for numeric values and ‘C’ for categorical values.

Data investigation will be performed using Python code and include following steps:

1. Statistical analysis: Basic statistics such as distribution, average, max, min, standard deviation, normalization, mode, the skewness of variables will be computed for analysis.

2. Missing value analysis: Count and percentage count of the missing values of target and predictor variables will be calculated.
### CHAPTER 3. EXPERIMENT DESIGN AND METHODOLOGY

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>N</td>
<td>&lt; 60, 60 – 65, 65 – 70, 71 – 75, 76 – 80, &gt;80</td>
</tr>
<tr>
<td>sex</td>
<td>C</td>
<td>Female, Male</td>
</tr>
<tr>
<td>hyper</td>
<td>C</td>
<td>Diagnosis of Hypertension</td>
</tr>
<tr>
<td>DM</td>
<td>C</td>
<td>Diagnosis of Diabetes mellitus</td>
</tr>
<tr>
<td>HbA1c</td>
<td>N</td>
<td>Glycosilated Haemoglobin (%) - a marker of an average blood glucose in a three-month period</td>
</tr>
<tr>
<td>chol</td>
<td>N</td>
<td>Total cholesterol</td>
</tr>
<tr>
<td>CVD</td>
<td>C</td>
<td>Cardiovascular disease</td>
</tr>
<tr>
<td>BMI</td>
<td>N</td>
<td>Body mass index (a measure of the body weight)</td>
</tr>
<tr>
<td>w/h</td>
<td>N</td>
<td>Waist to hip ratio</td>
</tr>
<tr>
<td>skinf</td>
<td>N</td>
<td>Triceps skinfold thickness</td>
</tr>
<tr>
<td>COPD</td>
<td>C</td>
<td>Chronic Obstructive Pulmonary Disease</td>
</tr>
<tr>
<td>Allerd</td>
<td>C</td>
<td>Allergic disease (rhinitis/asthma)</td>
</tr>
<tr>
<td>Draller</td>
<td>C</td>
<td>Allergy to drugs</td>
</tr>
<tr>
<td>Analg</td>
<td>C</td>
<td>Long-term use of analgesics/nonsteroidal anti-inflammatory drugs</td>
</tr>
<tr>
<td>Neo</td>
<td>C</td>
<td>No, malignant disease in a stable phase, skin malignancy</td>
</tr>
<tr>
<td>Psy</td>
<td>C</td>
<td>Anxiety/depression, Parkinsons disease, cognitive impairment</td>
</tr>
<tr>
<td>MMS</td>
<td>N</td>
<td>Neuropsychologic test for screening on cognitive impairment Mini Mental Score</td>
</tr>
<tr>
<td>CMV</td>
<td>N</td>
<td>Cytomegalovirus infection</td>
</tr>
<tr>
<td>EBV</td>
<td>N</td>
<td>Epstein-Barr virus infection</td>
</tr>
<tr>
<td>HPA</td>
<td>N</td>
<td>Helicobacter pylori infection</td>
</tr>
<tr>
<td>LE</td>
<td>N</td>
<td>White blood cell (WBC) count</td>
</tr>
<tr>
<td>NEU</td>
<td>N</td>
<td>Neutrophils % in WBC differential</td>
</tr>
<tr>
<td>LY</td>
<td>N</td>
<td>Lymphocytes % in WBC differential</td>
</tr>
<tr>
<td>CRP</td>
<td>N</td>
<td>C-reactive protein</td>
</tr>
<tr>
<td>E</td>
<td>N</td>
<td>Red Blood Cell (RBC) count</td>
</tr>
<tr>
<td>HB</td>
<td>N</td>
<td>Hemoglobin</td>
</tr>
<tr>
<td>HTC</td>
<td>N</td>
<td>Hematocrite (Erythrocyte volume blood fraction)</td>
</tr>
<tr>
<td>Clear</td>
<td>N</td>
<td>Creatinine clearance - an indicator of chronic renal impairment</td>
</tr>
<tr>
<td>RF</td>
<td>N</td>
<td>Rheumatoid factor - the auto-antibody, increased in patients with rheumatoid arthritis</td>
</tr>
<tr>
<td>VITB12</td>
<td>N</td>
<td>Vitamin B12</td>
</tr>
<tr>
<td>FOLNA</td>
<td>N</td>
<td>Folic acid</td>
</tr>
<tr>
<td>CORTIS</td>
<td>N</td>
<td>Serum cortisol in the morning</td>
</tr>
<tr>
<td>PRL</td>
<td>N</td>
<td>Prolactin in the morning - the anterior pituitary gland hormone</td>
</tr>
<tr>
<td>TSH</td>
<td>N</td>
<td>Thyroid-stimulating hormone - the anterior pituitary gland hormone</td>
</tr>
<tr>
<td>FT3</td>
<td>N</td>
<td>Free triiodothyronine - the thyroid gland hormone</td>
</tr>
<tr>
<td>FT4</td>
<td>N</td>
<td>Free thyroxine the thyroid gland hormone</td>
</tr>
<tr>
<td>ANA</td>
<td>N</td>
<td>Anti-nuclear antibody - the auto-antibody - a diagnostic marker in rheumatic autoimmune diseases</td>
</tr>
<tr>
<td>IGE</td>
<td>N</td>
<td>Immunoglobulin E - a class of antibody included in the allergic reactions</td>
</tr>
</tbody>
</table>

Table 3.1: Biomarkers information
3. Outlier analysis: Outlier analysis will be performed to find out the values lying out of range e.g. age group of patients.

4. Exploratory data analysis: Frequency distribution plots of numeric and categorical variables with respect to the target variable.

5. Heatmap correlation matrix: Heatmap matrix will be produced to analyze the correlation between the target variable (mortality risk) with the predictor variables (biomarkers). Also, a separate plot will be generated to understand if the variables are positively or negatively correlated with the target variable.

The next section provides the detailed overview of the steps carried out for data preparation and feature engineering used for the research work.

### 3.3 Data preparation

The data preparation phase covers all the activities involved in the transformation and cleaning of the data to make it fit to use in the modeling phase (Learn, 2017). The missing values, noise and outliers present in the data identified during data understanding phase will be removed in data pre-processing. The identified missing values will be imputed using the mean value of the variable.

#### 3.3.1 SMOTE Algorithm: Balanced dataset

The main concern is that the data is highly imbalanced and small in size. There are approx. 20% records which have mortality risk as ‘1’ and rest 80% of records have mortality risk as ‘0’. If the experiment is proceeding further to modeling phase without balancing the data then the model will be trained with biasing and cost of misclassifying minority class could be very high. Sampling techniques: under-sampling or over-sampling as shown in figure 3.3, should be implemented to get rid of imbalanced data set.
Since the data set is quite small in size, each instance is highly important and can’t risk to loosing any information. Hence, under-sampling technique is discarded and over-sampling technique will be used for the experiment.

Synthetic Minority Over sampling Technique (SMOTE) algorithm will be used for creating synthetic samples. SMOTE applies KNN approach where it selects K nearest neighbors, joins them and creates the synthetic samples in the space as shown in figure 3.4. The algorithm takes the feature vectors and its nearest neighbors and computes the distance between these vectors. The difference is multiplied by a random number between (0, 1) and it is added back to feature.
3.3.2 Normalization and Standardization: Z-score

Learning algorithms are known to provide reckless predictions on unscaled or unstandardized features. To ensure that all the feature values are on the same scale, normalization or standardization is a mandatory step to be carried out before proceeding to model building. Standardization is a preprocessing step to standardize values of features from different dynamic range into a specific range.

Standard score or commonly known as z-score converts scale of all the parameters having zero mean and unit variance and can be given as,

$$z_i = \frac{x_i - \bar{x}}{S}$$  \hspace{1cm} (3.1)

where,

\(\bar{x}\) is the mean of the sample

\(S\) is the standard deviation of the sample

A z-score can be placed on a normal distribution curve as shown in figure 3.5. Z-scores range from -3 standard deviations to +3 standard deviations.
3.3.3 One-Hot encoding: Categorical variables

The biomarker data set has 10 categorical variables out of 37 feature variables. In order to build LR and SVM model, categorical variables need to be converted into numeric variables as regression analysis and vector machine only work on numeric variables. To overcome this problem, dummy variables will be constructed out of categorical variables using pandas library in python.

3.3.4 Random Forest Classifier: Feature importance

Lastly, to find out the importance of each variable in predicting the mortality risk, Random Forest algorithm will be applied \cite{Svetnik2004}. Random Forest is an ensemble modeling technique which is based on iteratively removing low-ranking variables and assessing the learning performance by cross-validation.

A score will be generated for each variable which provides the importance of the variable in the model. Highest the score of the variable, highest the importance of that particular variable. Variables with the lowest score may be considered as least important and can be eliminated. Data pre-processing techniques help in the extraction of more useful information and build a model with higher accuracy and performance.
3.4 Modeling

The main aim of this study is to investigate the use of a Support Vector Machine (SVM) based classification model for determining the mortality risk by comparing it with Logistic Regression (LR) based classification model. Therefore, two supervised machine learning algorithms, Logistic Regression and Support Vector Machine will be implemented to build the models.

Logistic regression is the most commonly used prognostic modeling method whereas, support vector machine is relatively new classification method which has been developed by Vapnik et al. in 1990s as a result of the collaboration between the statistical and the machine learning. The target variable is categorical with binary values (0,1), hence logistic regression is fit to use for the prediction of the mortality risk amongst other regression techniques. The LR method uses a weighted least squares algorithm, i.e. the prediction is based on the construction of a regression line as the best fit through the data points by minimizing a weighted sum of the squared distances to the fitted regression line (Verplancke et al., 2008).

Logistic regression is based on logistic function $\sigma(t)$ which is defined as follows:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$  \hspace{2cm} (3.2)

SVM is also a supervised machine learning technique similar to LR which is used for both regression and classification problems. SVM, in contrast, tries to model the input variables by finding a boundary for the classification of target variable which is called hyper-plane. The data points nearest to the hyper-plane are the support vectors and are considered as the critical elements of the data set because the removal of the points will lead to the alteration of the dividing hyper-plane. SVM can be used for both regression and classification purposes. If no separation is possible within a high number of input variables, the SVM algorithm finds a separation boundary for the classification as shown in figure 3.6. The separation boundary is generated by transforming the input variables by increasing the dimensionality of the variable space.
Three different SVM models will be built as part of the experiment which consists of SVM Linear kernel model, SVM radial kernel model and SVM polynomial kernel model each tuned with different values of tuning parameter ‘C’ and ‘γ’. SVM model uses a kernel function to separate classes which can’t be separated using line or plane. Therefore, a non-linear region is required by the classifier to separate such classes. This is also known as kernel trick, transformation of the data into higher dimensional feature space in order to separate it linearly as shown in figure 3.7.

Figure 3.6: Support Vector Machine

(Verplancke et al., 2008)

Figure 3.7: Support Vector Machine kernels
3.5 Evaluation

This experiment uses two steps to evaluate the performance of the predictive models. Firstly, a Stratified K-fold cross validation technique will be used for the validation of the model. In stratified k-fold cross validation, the folds are selected in such a way that each class labels are equally distributed in each fold. The target variable is binary and hence the experiment comes under dichotomous classification, this means that each fold contains roughly the same proportions of the two types of class labels.

The data set will be divided into k subsets where k =10, each time one of the k subsets will be used as the test set and the k-1 subsets will be used as a training set. In this way, every data point will be part of the test set exactly once and gets to be in training set k-1 times. The average results from the k folds will be taken and single estimation will be produced. The only disadvantage of using k-fold cross validation is that the algorithm takes time for training. k=10 is the standard value which is ideally used in the experiments. The training and test split in 5-fold cross validation is shown in below figure 3.8.

![Figure 3.8: Stratified 5-fold cross validation technique](Slideshare.net, 2017)

The performance of the models will be compared based on the accuracy obtained
in the prediction of mortality risk. Accuracy of the model will be computed in each fold and in the end there would be 10 accuracies per model. Accuracy of the model is given as,

\[ \text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \]  \hspace{0.3cm} (3.3)

where,

TP (True Positive) i.e. positive instances that are classified as positive,
FP (False Positive) i.e. negative instances that are classified as positive,
FN (False Negative) i.e. positive instances that are classified as negative,
TN (True Negative) i.e. negative instances that are classified as negative

The evaluation parameters such as, average classification accuracy, receiver operation curve (ROC) (Verplancke et al., 2008) and area under the curve (AUC) (Wu, Roy, & Stewart, 2010) will be obtained. ROC-AUC plot of each model will be generated to visualize the mean accuracy of each model. ROC curve is based on two metrics, True Positive Rate (TPR) and False Positive Rate (FPR).

True positive rate (TPR), also known as sensitivity, hit rate or recall, is defined as

\[ TPR = \frac{TP}{TP + FN} \]  \hspace{0.3cm} (3.4)

Intuitively this metric corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points will be missed.

False positive rate (FPR) or fall-out is defined as

\[ FPR = \frac{FP}{FP + TN} \]  \hspace{0.3cm} (3.5)

FPR can also be generated from specificity as

\[ FPR = 1 - \text{Specificity} \]  \hspace{0.3cm} (3.6)

where specificity is defined as

\[ FPR = \frac{TN}{TN + FP} \]  \hspace{0.3cm} (3.7)
This metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, higher the FPR, the more negative data points will be missclassified.

In order to combine FPR and TPR into one single metric i.e. to generate AUC, two former metrics with different threshold is calculated and then plotted on a single graph with FPR values on x-axis and TPR values on the y-axis. The resulting curve is called AUROC as shown in figure 3.9.

A confusion matrix is another evaluation metric which is used to describe the performance of a classifier by calculating evaluation parameters and is shown in table 3.2. The values of true positive rate and false positive rate are generated using confusion matrix.
The comparison of both the models will be helpful in finding out the performance difference between the models in terms of classification accuracy. The final step is to determine the statistical significance of each experiment and to accept or reject the stated hypothesis. ‘Wilcoxon Signed-Rank Test’ will be used to test the statistical significance with p-value set as 0.01.

### 3.6 Strength and limitations

This section outlines the strength and limitations of the research design. Firstly, two different families of models (Regression and SVM) are adopted to perform an experiment which is believed to be effective in order to gain insight of performance of distinct models on the mortality risk prediction. Regression is known to provide better interpretability and information about the significance of the predictive feature based on the coefficients assigned to each feature whereas, SVM has higher predictive power. Also, different SVM models with linear, radial and polynomial kernels will be trained on the data set with an optimal level of tuning.

The models will be trained using stratified 10-fold cross validation technique and should gives more accurate predictions. The model will be trained and evaluated 10 times and hence the average of all the accuracies obtained in 10 iterations will be taken to evaluate the classifier accuracy.

The major limitation of the study is the relatively small size of dataset, only 93 records of patients are available against 37 predictor variables. In general, for binary values there should be $2^n$ possible cases for ’n’ features. Higher the number of predictor features, higher the number of samples used to train the model. In real life, model can’t be trained for all possible number of combinations but it is good to use maximum
possibilities so that model can learn the exact behavior of target variable vs predictor variables.

Secondly, trends in the dataset are biased, records containing information with mortality risk as true are \((1/4)^{th}\) of the records with zero mortality risk. Synthetic data samples are created using data upscaling technique. This might produce difference in the results obtained using real world sample data and synthetic data.

Lastly, extensive tuning of SVMs models is not proposed while training the model due to time constraints and insufficient computational powers.

\section*{3.7 Summary of design}

This chapter is dedicated to providing the breakdown of the experiment to be carried out for the thesis. The strength and limitations of the experiments are also proposed in this chapter.

The chapter starts with a brief description of the dataset including the variable types and data source. One of the main concern are the issues present in the raw data. Hence, data pre-processing machine learning techniques are briefed which will be used for cleaning and normalizing the data to make it fit for modeling. These pre-processing techniques include SMOTE algorithm to balance dataset, Z-score for data standardization and One-hot encoding to generate dummy variables.

Further, four different models will be trained, LR, SVM with linear kernel, SVM with radial kernel and SVM with polynomial kernels. In the end, evaluation of all the models will be done and best model will be selected based on the average classification accuracy of each model.

The following chapter details the practical implementation of the proposed design and experiments.
Chapter 4

Implementation and results

This chapter details the results of the experiments performed. The layout of the chapter is a mirror of the previous chapter 'Design and Implementation' so that comparison can be made between the proposed design of each phase and the actual results obtained for that phase.

4.1 Business Understanding

The primary goal of this section as outlined in the previous chapter is reproduced here. This section will ensure that the targets are met according to the phases.

1. Exploratory analysis of input dataset.

2. Missing value and outlier analysis followed by data cleaning.

3. Impute missing values and generate dummy variables.

4. Train models on feature sets.

5. Score each model and analyze the results.

6. Compare the models in order to answer the research question.

7. Evaluate the statistical significance of the results.
4.2 Data understanding

The descriptive statistics of data is shown in Figure 4.1. The target variable is binary and has value either '1' i.e. mortality risk is true or '0' i.e. no mortality risk. The table provides information about the mean, standard deviation, maximum value, minimum value and distribution (quartile range) of each numeric variable.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Variables</th>
<th>Count</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Minimum</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>age</td>
<td>93</td>
<td>67.645</td>
<td>7.961</td>
<td>47</td>
<td>63</td>
<td>68</td>
<td>73</td>
<td>89</td>
</tr>
<tr>
<td>2</td>
<td>HbA1c</td>
<td>93</td>
<td>5.006</td>
<td>4.728</td>
<td>2.89</td>
<td>3.82</td>
<td>4.14</td>
<td>4.55</td>
<td>48.3</td>
</tr>
<tr>
<td>3</td>
<td>Chol</td>
<td>93</td>
<td>6.187</td>
<td>1.384</td>
<td>3</td>
<td>5.2</td>
<td>6.2</td>
<td>7.1</td>
<td>8.9</td>
</tr>
<tr>
<td>4</td>
<td>BMI</td>
<td>93</td>
<td>29.044</td>
<td>4.205</td>
<td>20.24</td>
<td>26.33</td>
<td>29.38</td>
<td>31.65</td>
<td>43.1</td>
</tr>
<tr>
<td>5</td>
<td>w/h</td>
<td>93</td>
<td>0.951</td>
<td>0.066</td>
<td>0.76</td>
<td>0.91</td>
<td>0.95</td>
<td>1.01</td>
<td>1.11</td>
</tr>
<tr>
<td>6</td>
<td>skinf</td>
<td>93</td>
<td>33.366</td>
<td>7.378</td>
<td>16</td>
<td>28</td>
<td>33</td>
<td>39</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>MMS</td>
<td>93</td>
<td>25.129</td>
<td>3.490</td>
<td>14</td>
<td>23</td>
<td>25</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>CMV</td>
<td>93</td>
<td>6.229</td>
<td>3.602</td>
<td>0.06</td>
<td>3.4</td>
<td>5.7</td>
<td>8.1</td>
<td>17.8</td>
</tr>
<tr>
<td>9</td>
<td>EBV</td>
<td>93</td>
<td>135.589</td>
<td>50.239</td>
<td>15.8</td>
<td>121.2</td>
<td>170</td>
<td>170</td>
<td>170</td>
</tr>
<tr>
<td>10</td>
<td>HPA</td>
<td>93</td>
<td>32.612</td>
<td>51.386</td>
<td>2.3</td>
<td>6.2</td>
<td>9.8</td>
<td>29.5</td>
<td>200</td>
</tr>
<tr>
<td>11</td>
<td>LE</td>
<td>93</td>
<td>6.628</td>
<td>1.518</td>
<td>4.06</td>
<td>5.61</td>
<td>6.81</td>
<td>7.65</td>
<td>9.93</td>
</tr>
<tr>
<td>12</td>
<td>NEU</td>
<td>93</td>
<td>51.959</td>
<td>9.838</td>
<td>28</td>
<td>45.4</td>
<td>50.9</td>
<td>59</td>
<td>73.3</td>
</tr>
<tr>
<td>13</td>
<td>LY</td>
<td>93</td>
<td>35.455</td>
<td>8.991</td>
<td>18.4</td>
<td>27.8</td>
<td>35.5</td>
<td>42.2</td>
<td>57.7</td>
</tr>
<tr>
<td>14</td>
<td>CRP</td>
<td>92</td>
<td>5.359</td>
<td>3.477</td>
<td>0.8</td>
<td>3.8</td>
<td>4.4</td>
<td>5.025</td>
<td>24.5</td>
</tr>
<tr>
<td>15</td>
<td>E</td>
<td>93</td>
<td>4.323</td>
<td>0.412</td>
<td>2.63</td>
<td>4.08</td>
<td>4.36</td>
<td>4.57</td>
<td>5.37</td>
</tr>
<tr>
<td>16</td>
<td>HB</td>
<td>93</td>
<td>134.710</td>
<td>12.458</td>
<td>91</td>
<td>127</td>
<td>136</td>
<td>141</td>
<td>167</td>
</tr>
<tr>
<td>17</td>
<td>HTC</td>
<td>93</td>
<td>0.393</td>
<td>0.032</td>
<td>0.28</td>
<td>0.37</td>
<td>0.4</td>
<td>0.41</td>
<td>0.47</td>
</tr>
<tr>
<td>18</td>
<td>Clear</td>
<td>92</td>
<td>1.692</td>
<td>0.450</td>
<td>0.72</td>
<td>1.4075</td>
<td>1.635</td>
<td>2.0125</td>
<td>3.21</td>
</tr>
<tr>
<td>19</td>
<td>RF</td>
<td>93</td>
<td>27.892</td>
<td>82.864</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>677</td>
</tr>
<tr>
<td>20</td>
<td>VITB12</td>
<td>93</td>
<td>284.335</td>
<td>158.786</td>
<td>97.8</td>
<td>197</td>
<td>247</td>
<td>306</td>
<td>885.6</td>
</tr>
<tr>
<td>21</td>
<td>FOLNA</td>
<td>93</td>
<td>20.718</td>
<td>8.521</td>
<td>6.5</td>
<td>15.1</td>
<td>19.5</td>
<td>25</td>
<td>43.9</td>
</tr>
<tr>
<td>22</td>
<td>CORTIS</td>
<td>93</td>
<td>374.596</td>
<td>122.387</td>
<td>180.3</td>
<td>278.9</td>
<td>362.6</td>
<td>433</td>
<td>812.1</td>
</tr>
<tr>
<td>23</td>
<td>TSH</td>
<td>92</td>
<td>2.040</td>
<td>2.609</td>
<td>0.024</td>
<td>0.95575</td>
<td>1.47</td>
<td>2.1975</td>
<td>22.7</td>
</tr>
<tr>
<td>24</td>
<td>FT3</td>
<td>93</td>
<td>5.457</td>
<td>0.528</td>
<td>4.35</td>
<td>5.17</td>
<td>5.42</td>
<td>5.72</td>
<td>7.96</td>
</tr>
<tr>
<td>25</td>
<td>FT4</td>
<td>93</td>
<td>14.014</td>
<td>2.210</td>
<td>8.92</td>
<td>12.3</td>
<td>13.8</td>
<td>15.9</td>
<td>18.9</td>
</tr>
<tr>
<td>26</td>
<td>ANA</td>
<td>93</td>
<td>29.082</td>
<td>34.982</td>
<td>7</td>
<td>14</td>
<td>18.9</td>
<td>32</td>
<td>300</td>
</tr>
<tr>
<td>27</td>
<td>IGE</td>
<td>93</td>
<td>135.913</td>
<td>245.589</td>
<td>2</td>
<td>16.3</td>
<td>57</td>
<td>143</td>
<td>1782</td>
</tr>
<tr>
<td>28</td>
<td>TARGET</td>
<td>93</td>
<td>0.194</td>
<td>0.397</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4.1: Statistical Analysis of data

The Count column provides the information about the total number of records of each variable. Only three biomarkers, 'CRP', 'Clear' and 'TSH' has missing value out of 37 variables and is given in table 4.1.
CHAPTER 4. IMPLEMENTATION AND RESULTS

<table>
<thead>
<tr>
<th>Variable</th>
<th>Missing Count</th>
<th>Missing Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>1</td>
<td>0.010753</td>
</tr>
<tr>
<td>TSH</td>
<td>1</td>
<td>0.010753</td>
</tr>
<tr>
<td>CRP</td>
<td>1</td>
<td>0.010753</td>
</tr>
</tbody>
</table>

Table 4.1: Missing Value Analysis

Figure 4.2 and 4.3 shows the distribution of the numeric features with respect to the target variable. BMI, CMV, CORTIS, Chol, Clear, E, FOLNA, FT3, FT4, HB, HTC, LE, LY, MMS, NEU, VITB12, age, skinf and wh are normally distributed with the mortality risk. Biomarkers such as ANA, CRP, HPA, HbA1c, IGE, TSH are positively skewed whereas, EBV is negatively skewed.

Figure 4.2: Distribution plot of numeric features with target
The frequency plot of categorical variables; sex, hypert, DM, CVD, COPB, aller_d, d_aller, analg, neo, Psy is plotted as shown in figure 4.4. The ‘Target’ variable is highly biased as per the information provided by the bar graph. Only 20% of the values are ‘1’ and rest of the records have ‘0’ values. The number of patients having high mortality risk is almost $\left(\frac{1}{4}\right)^{th}$ of the number of patients without any mortality risk. The balancing of the target feature will be taken care of in Data Preparation section.

All the categorical variables are binary and have ‘Yes’ or ‘No’ values except DM which has 3 distinct values as shown in figure 4.4.
Pearson correlation coefficient is used in the experiment to interpret the linear association between the numeric-continuous variables. The correlation coefficient ranges between -1 to 1, the greater the absolute value the stronger the linear relationship. The correlation heatmap matrix as shown in figure 4.5, gives the strength of relationship between the features. The result deduced from the matrix is as under:

- All the biomarker features have very little correlation with mortality risk.
- LY and NEU are highly negatively correlated.
- E has a strong positive correlation with HB and HTC.
- HTC is strongly positively correlated with HB.
- LY and NEU are highly negatively correlated.
- skinf has a moderate positive correlation with BMI.
Figure 4.6 gives information about the relationship strength and magnitude of relations between independent features (biomarkers) and Target (mortality risk). The biomarkers those are lying on the left side of the axis have a negative correlation with the mortality risk i.e. the increase in the value of these variables will decrease the chances of mortality whereas, the biomarkers those are lying on the right side of the axis has a positive correlation i.e. the increase in the value of these variables will also increase the chances of mortality. Further, the height of the bar graph from the center of the axis represents the magnitude of the strength of the correlation of each feature with mortality risk.
4.3 Data preparation

The first step after analyzing the data is to remove the issues identified in the dataset in order to make it fit for the modeling. The mean value of each of the variable having missing values is calculated and later the records having missing values' are imputed with the mean of ‘CRP’, ‘Clear’ and ‘TSH’ respectively.

4.3.1 SMOTE Algorithm: Balanced dataset

Standard classifier algorithms like Logistic Regression have a tendency to produce results biased towards classes which have a higher number of instances. Due to this characteristic, classifiers often ignore minority class features treating them as noise. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.
SMOTE algorithm is implemented on the data set, which is an over-sampling technique to balance the data set by creating synthetic records. Initially, the number of records belonging to mortality risk as ‘1’ is significantly lower than those belonging to class ‘0’ as shown in table 4.2. The number of samples containing ‘1’ value is increased to 50% after running SMOTE oversampling algorithm.

<table>
<thead>
<tr>
<th>Target: Mortality Risk</th>
<th>Imbalanced Data set</th>
<th>Balanced Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>75</td>
</tr>
<tr>
<td>0</td>
<td>75</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 4.2: SMOTE oversampling

### 4.3.2 Normalization and Standardization: Z-score

The distribution of the features as shown in figure 4.2 and 4.3 provides information about the skewness of variables which is removed by feature scaling of the predictor variables. Z-score of each value of numeric variables is calculated and replaced with the actual value, finally building model using the normalized values.

### 4.3.3 One-Hot Encoding: Categorical variables

After generating the balanced and normalized dataset, the only issue left is to get rid of categorical variables, as regression and support vector models only work on numeric variables and can’t handle features having string values. All the 10 categorical variables are nominal i.e. their values don’t follow any particular natural order. To deal with such nominal variables in classification, one-hot encoding is done. One-hot encoding involves the generation of dummy variables having binary values for each possible value of that nominal feature variable. Number of features will be increased to 48 after implementing one-hot encoding on 10 categorical variables (‘sex’, ‘hypert’, ‘DM’, ‘CVD’, ‘COPB’, ‘aller_d’, ‘dr_aller’, ‘analg’, ‘neo’ and ‘Psy’). Dummy variables generated from categorical variables is shown in table 4.3.
CHAPTER 4. IMPLEMENTATION AND RESULTS

<table>
<thead>
<tr>
<th>Categorical Variables</th>
<th>Dummy Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex</td>
<td>sex_F, sex_M</td>
</tr>
<tr>
<td>hypert</td>
<td>hypert_high, hypert_low</td>
</tr>
<tr>
<td>DM</td>
<td>DM_IGT, DM_no, DM_yes</td>
</tr>
<tr>
<td>CVD</td>
<td>CVD_no, CVD_yes</td>
</tr>
<tr>
<td>COPB</td>
<td>COPB_yes, COPB_no</td>
</tr>
<tr>
<td>aller_d</td>
<td>aller_d_no, aller_d_yes</td>
</tr>
<tr>
<td>dr_aller</td>
<td>dr_aller_no, dr_aller_yes</td>
</tr>
<tr>
<td>analg</td>
<td>analg_no, analg_yes</td>
</tr>
<tr>
<td>neo</td>
<td>neo_no, neo_yes</td>
</tr>
<tr>
<td>Psy</td>
<td>Psy_no, Psy_yes</td>
</tr>
</tbody>
</table>

Table 4.3: One-hot Encoding

4.4 Modelling

In this stage, classification models are built to predict the mortality risk of the patient using LR and SVM algorithms. The balanced data set generated after adding dummy variables is used as training data to build the models. Before training the model, input data is first normalized. Also correlation and multicollinearity analysis is performed for the better fit of the model. All the variables are very weakly correlated with each other and hence none are dropped while training the model.

In total, four supervised machine learning models are built, one for LR and other three using SVM implementing linear, radial basis and polynomial kernels. Each model is fitted by running 10 iterations where each iteration will give accuracy of the model, resulting into 10 accuracies per model.

Similarly, models are generated on another set of data which have undergone all data pre-processing techniques except SMOTE algorithm (to balance the data) and impact analysis of balanced and unbalanced data is also done as part of the experiment.
4.4.1 Logistic regression: Balanced Target values

Firstly, Logistic Regression model is built which is a binary classification regression model. In total 37 variables are used in training the model out of which 10 are categorical variables which are converted into binary vectors using one-hot encoding as described in section 4.3.2.

Stratified 10 k-fold validation technique is implemented to split data into training and test set. In python, Scikit-learn has inbuilt packages containing functions for stratified k-fold cross validation\(^3\), LR\(^4\) modeling and SVM\(^5\) modeling. All the instances of the data is fed into the ‘StratifiedKFold’ function with the number of splits provided as 10. This will randomly generate 10 folds, out of which 9 folds are used for training the model and one sample is kept for testing the model. These folds are generated in such a way that it has an equal ratio of target variable in each fold, say if there are 60-40 ratio of survivors and non-survivors in first fold then there should be 60-40 ratio in other folds as well.

![ROC Curve: Logistic regression](image)

**Figure 4.7: ROC curve: Logistic Regression**

CHAPTER 4. IMPLEMENTATION AND RESULTS

Once the model is trained, it is used to predict the target value for test data set. In this way, the score of the model is generated which gives the prediction accuracy of the model. In total 10 scores are generated and mean of all the scores gives the average accuracy of the LR classifier. The minimum and maximum accuracy obtained by LR model are 69% and 93% respectively. The average mean accuracy of the LR classifier is 82% as shown in figure 4.7.

4.4.2 Support vector machine: Balanced Target values

Similarly, Support Vector Machine models are built using 10 k-fold stratified sampling to generate training and test data set. In the following section, three types of SVM models are explored, SVR with linear kernel, radial kernel and polynomial kernel tuned with different values of tuning parameter ‘C’ and ‘γ’ as shown in table 4.4.

<table>
<thead>
<tr>
<th>SVM kernels</th>
<th>C</th>
<th>γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
<tr>
<td>Radial basis function</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
<tr>
<td>Polynomial</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
</tbody>
</table>

Table 4.4: SVM kernels and tuning parameters

The minimum and maximum accuracy obtained by the model build using linear SVM algorithm is 71% and 94% with average classifier accuracy as 86%. It can be deduced from ROC curve that almost half of the folds achieved accuracy above 90% as illustrated by figure 4.8. The results show that the performance of model build with linear SVM is slightly higher than logistic regression model.
CHAPTER 4. IMPLEMENTATION AND RESULTS

Figure 4.8: ROC curve: Support Vector Machine - Linear

Figure 4.9: ROC curve: Support Vector Machine - Radial
The ROC curves are also plotted separately for SVM radial basis function and polynomial kernels. Figure 4.9 and 4.10 illustrates the average classifier accuracy of both the models having information of the accuracy per iteration. The mean accuracy of obtained by both SVM radial and SVM polynomial model is 95%.

![ROC Curve: Support Vector Machine - Polynomial](image)

Figure 4.10: ROC curve: Support Vector Machine - Polynomial

### 4.4.3 Logistic regression: Imbalanced Target values

The set of experiments performed in section 4.4.1 and 4.4.2 are repeated on the dataset which has biased values of Target (mortality risk) variable. Raw data has the percentage of ‘0’ and ‘1’ values of the target as 80% and 20% respectively. The biasing in the data may decrease the overall performance of the classifier because minority classes are sometimes ignored.
Figure 4.11 shows that the accuracy obtained by logistic regression classifier is very low, 54%. In few iterations of 10-k fold, the model performed worst and is below than the base line of ROC curve.

4.4.4 Support vector machine: Imbalanced Target values

The model trained with the SVM using biased values also performed worst. SVM with the radial and polynomial kernels give prediction accuracy of 50-50% of binary dependent variable as shown in figure 4.13 and 4.14.

<table>
<thead>
<tr>
<th>SVM kernels</th>
<th>C</th>
<th>γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
<tr>
<td>Radial basis function</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
<tr>
<td>Polynomial</td>
<td>1,2,3,4,5,6,7,8,9,10</td>
<td>0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5</td>
</tr>
</tbody>
</table>

Table 4.5: SVM kernels and tuning parameters

The values used for tuning the SVM radial bias kernel model and SVM polynomial kernel model is given in table 4.5.
CHAPTER 4. IMPLEMENTATION AND RESULTS

Figure 4.12: ROC curve: Support Vector Machine - Linear

Figure 4.13: ROC curve: Support Vector Machine - Radial
The results of the experiment performed on both the data set i.e. one having equally distributed values of mortality risk and another with biased values of mortality risk shows the difference in the classification accuracy of the classifiers which will be discussed in next chapter ‘Discussion and findings’.

4.5 Evaluation

In order to evaluate the performance of the models, ROC-AUC curve is needed which is generated in modeling section. The basic requirement to plot ROC curve is to compute True Positive Rate (TPR) and False Positive Rate (FPR). Sklearn has inbuilt roc_curve() and auc() functions which returns TPR and FPR as output.

Since stratified k-fold validation technique is used to split train-test data, an average of accuracy obtained after each iteration is considered to compare the models. Thereafter, box plots are generated using a range of accuracies obtained in 10 iterations of each model to analyze the performance of the models. Similarly, confusion

matrix is also generated after each iteration i.e. for each model 10 confusion matrix are generated along with 10 accuracies.

4.6 Summary of implementation

This chapter details the practical implementation of the experiments proposed to answer the research question and provides results of each intermediate stage.

The exploratory data analysis is done on the biomarkers dataset to understand the distribution of predictor variables with respect to the target variable. A heatmap matrix is generated to analyze the strength of the relationship between each variable. All the variables are weakly correlated with each other and hence none of the predictor variables is dropped.

The data-preprocessing techniques such as missing value imputation, balancing of the values of target variable using SMOTE algorithm, normalization of variables using Z-score and generation of dummy variables are used to make data fit for modeling phase. Further, four mortality risk prediction models are build using LR, SVM linear kernel, SVM radial basis kernel and SVM polynomial kernel on training-test data set obtained from stratified K-fold cross validation techniques resulting into 10 accuracies per model. An ROC curve is generated and average classifier accuracy is calculated using AUC.

Finally, in order to evaluate the performance of all the models, specificity, sensitivity and average classification accuracy of each model will be compared. A full analysis of the results obtained from this section is explained in chapter 5 and will be used to evaluate the hypothesis.
Chapter 5

Discussion and findings

This chapter provides a detailed analysis of the results of the experiments performed in the previous chapter. In essence, experiments are carried out to build four separate supervised machine learning models. The performance of all the models will be evaluated based on the accuracies obtained after training each model on the pre-processed dataset. Also, the same experiment is performed on the imbalanced dataset with biased values of the target variable. Further, results obtained on both the datasets will also be compared in the analysis.

The statistical significance of the experimental result will be discussed later in this chapter and critical evaluation of the strengths and limitations of the experiment is concluded.

5.1 Discussion

The goal of the research is to perform an experiment to evaluate the performance of two supervised machine learning algorithms. Initially the classifiers have been trained on a set of samples which have biased dependent variable values. Before that data preprocessing steps such as missing value imputation, feature standardization, generation of dummy variables as illustrated in section 4.3 has been applied on the biomarkers dataset excluding data oversampling technique. It is found that data balancing has a major impact on the prediction accuracy of the classifier.
5.1.1 Comparison of average accuracies of imbalanced and balanced target data

The class imbalance is the most common problem in the binary classification predictive models. Figure 5.1 presents the performance results of all the classifiers on the imbalanced dataset where the dependent variable i.e. risk of mortality is highly biased with only \((1/4)^{th}\) of the instances having ‘1’ value.

Firstly, the analysis has been conducted on the results obtained on the imbalanced dataset. It can be clearly interpreted from the box-whisker plot of the accuracies that logistic regression algorithm based model has the highest accuracy than support vector machine based algorithm models. The maximum accuracy obtained by logistic regression model is lower than 70%. SVM build with radial basis function and polynomial kernels are worthless as learning is minimal for both the models during the training phase of modeling. The poor classification accuracy is due to the presence of biased value of the target variable.

![Figure 5.1: Model comparison using box-whisker plot: Unbalanced dataset](image)

Due to its prevalence, oversampling approach (SMOTE algorithm) has been taken
to create a balanced dataset by generating synthetic data. Balancing of the imbalanced dataset comes under data pre-processing activities. Also, it is advisable to use alternative metrics like recall, precision, sensitivity, specificity, true positive rate, false positive rate instead of general accuracy to measure the performance of the model.

The accompanying table 5.1 shows the mean accuracies obtained from both imbalanced and balanced dataset of each model.

<table>
<thead>
<tr>
<th>Supervised ML models</th>
<th>AUROC imbalanced data</th>
<th>AUROC balanced data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>54%</td>
<td>82%</td>
</tr>
<tr>
<td>SVM Linear</td>
<td>52%</td>
<td>86%</td>
</tr>
<tr>
<td>SVM radial basis function</td>
<td>50%</td>
<td>95%</td>
</tr>
<tr>
<td>SVM polynomial</td>
<td>50%</td>
<td>95%</td>
</tr>
</tbody>
</table>

Table 5.1: AUROC on imbalanced and balanced target (mortality risk)

5.1.2 Comparison of classifiers performance

Taking this further, another experiment has been performed on all the four models after applying SMOTE algorithm. The new dataset has 150 instances out of which...
75 instances have the mortality risk value as true and rest with zero mortality risk. Figure 5.2 shows the accuracies box-whisker plot of balanced dataset models.

The most noticeable change here is the significant increase in the average accuracy of the classifiers. The result generated on evenly distributed target variable values has yielded the best approach in terms of giving good classification accuracy. As depicted from the graph, SVM models have higher prediction accuracy as compared to logistic regression model. On top of that SVM with the radial basis and polynomial kernels have tendency to classify instances with 100% accuracy as well. The final results show that SVM models have outstanding performance on the mortality risk prediction. The only task left is to calculate the statistical significance of the results which is discussed in the following section.

5.1.3 Statistical significance and hypothesis evaluation

This section will focus on the statistical significance of the results of the experiments performed in order to answer the research question. The hypothesis to be accepted or rejected is listed below:

“H0 The accuracy of the model built with Support Vector Machine outperforms Logistic Regression for prediction of the mortality in elderly using bio-markers of the patient, with p-value <0.01.”

A simple statistical test, ‘Wilcoxon Signed-Rank Test’ is performed, on the accuracies scores obtained for each model i.e 10 accuracies per model. The cut-off chosen for determining the significance of the results is ‘0.01’. To test the statistical significance of the experimental results the accuracies obtained by each model per stratified 10 k-fold cross validation iteration is used.

<table>
<thead>
<tr>
<th></th>
<th>LR-SVM linear</th>
<th>LR-SVM radial</th>
<th>LR-SVM polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>p-value</td>
<td>0.23</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
</tr>
</tbody>
</table>

Table 5.2: Statistical significance of experimental results

As depicted from the table 5.2 all the results are statistically significant excluding
LR-SVM linear models. Support vector machine models with radial basis function and polynomial kernels performed better than logistic regression model in predicting the risk of mortality among elderly as the results of statistical test is significant. Further, combining all the experimental results and ‘Wilcoxon Signed-Rank Test’ results, the hypothesis can be accepted and it is concluded that the predictive models build with support vector machine have better performance as compared to logistic regression.

5.2 Strength and limitations of results

The contribution of supervised machine learning techniques in clinical diagnosis is studied as part of the research study. This experiment has taken two machine learning algorithms (LR and SVM), similar in a way that they are used for the classification of instances but also different as LR based classification models are preferred where classes are linearly separable or have a single decision surface whereas support vector machine models are better for non-linearly separable classification problems.

The training of both the models belonging to different families on the same set of data can be considered as one of the strength of the research. Additionally, three SVM models are build using different values of kernels- linear, radial basis function and polynomial in order to get efficient optimization of the model. The ‘Kernel-trick’ provided opportunity to compare LR model with a number of SVM models and hence the results obtained are more significant than the results obtained by comparing with just one linear SVM model.

Also, tuning of models is taken care of while performing experiments with SVM. Models are tuned using 10 distinct values of tuning parameters ‘C’ and ‘γ’ which helped in achieving higher accuracy. Figure 5.2 clearly shows that SVM models with tuning parameters (radial and polynomial vector) are better than the linear model.

This study also focused on the impact of balancing the imbalanced dependent parameter in the dataset. SMOTE algorithm is implemented to over-sample the data in order to get rid of biasing in the results. A significant improvement is achieved on the accuracy of all the models after applying SMOTE algorithm. SVM radial and
polynomial which were initially worst predictive models turned out to be the best models when trained on the balanced dataset.

Data pre-processing techniques namely feature scaling using z-score and dummy variable generation using one-hot encoding are studied thoroughly during the research and thereafter applied on the data to improve the results. Multiple iterations using stratified 10 k-fold validation is used in modeling. The average accuracy of the classifiers is taken which is the mean accuracy of the accuracies obtained at each fold.

Lastly, the models are very simple and computationally light and may be altered to accommodate more features and instances. The use of the models can be further extended from mortality risk prediction to other chronic disease diagnosis or prognosis.

To move further to the limitations, the results are based on experiments performed over small-sized dataset with only 93 instances against 37 biomarkers predictor variables for training and testing the model. In general higher the number of predictor features, higher the number of samples used in the training of the model. The behavior of the models may vary if a relevant size of the data is used for the experiment which can be explored further as part of the future study.

Also only stratified k-fold validation technique is used in modeling due to time constraints. Other validation techniques like leave-one-out cross validation, random train-test split can also be applied and the results can be compared.

The experimental results and statistical test of ‘Wilcoxon Signed-Rank Test’ supports that SVM radial basis kernels and SVM polynomial kernels have better performance than LR model when trained on same data sample. The statistical test doesn’t comes out to be significant i.e. $>0.01$ for LR and SVM linear models and needs future research.

Another limitation of this research is its unique focus on the patients of a particular hospital and might have biasing on the biomarkers traits based on the population. Also, the time span during which the patients are monitored is small (3 years). In order to examine the organ functionality or biological changes in an individual and provide prognostic results the observation duration should be increased to get more stable value of biomarkers and may impact the result of predictive models.
5.3 Summary of analysis

The breakdown and evaluation of the overall experiment is discussed in this chapter. All the four models are built on two set of data, one having biased values of target variable and another one with balanced values.

The SVM models outperformed LR models in the prediction of mortality risk. The ‘Wilcoxon Signed-Rank Test’ is used to calculate the statistical significance of each result with p-value $<0.01$. The results of SVM with radial basis and polynomial kernel are statistically significant. There is no proof on the significance of the results obtained after comparing LR and SVM linear kernel models which can be explored further as future research.

The strength and the limitation of the results is elaborated focusing on the data preprocessing techniques which are used to improve the performance of the models. The major limitation is the small-sized imbalanced data used for the experiments. The next chapter will give the detailed summary of the research, contribution and impact of the research and the areas of future research.
Chapter 6

Conclusion

6.1 Research Overview

This research study investigates multiple supervised machine learning techniques used in the bioinformatics to analyze and interpret biological data for medical diagnosis or prognosis at an early stage for elderly people. At the initial stage, literature review is conducted, summarizing existing state-of-art on mortality prediction, biomarkers and machine learning.

An attempt has been made to perform experiments using two supervised classification techniques to build mortality risk predictive models on the data sample collected from a hospital in a European country. One of the models is trained using Logistic Regression (LR) algorithm which is preferred for linearly separable classification problems. LR is the most widely used model for clinical purposes when the dependent variable is binary in nature. Another model is built using Support Vector Machine (SVM) algorithm which is suitable for both linearly and non-linearly separable classes. The main benefit of SVM modeling is that it works efficiently on both small and high dimensional data spaces. Also, SVM models are capable of providing good results on small sized data and are best suited for the research.

Each technique can predict the mortality risk using biomarker features in their own particular way. The main goal of the study is to measure the classifier accuracy and compare the performance of both the models, further concluding that which is
CHAPTER 6. CONCLUSION

the outperforming model.

6.2 Problem Definition

The gaps and limitations identified in the existing literature is used as motivation for the thesis. Biomarkers such as hypertension, hypercholesterolemia and stress are not explored in the research studies (Gruenewald et al., 2006), (Rosero-Bixby & Dow, 2012) and hence are included in the dataset while performing experiments. As suggested by Liu, Delen and Song, better promising methods such as Support Vector Machines can be used with biomarkers to improve the prediction accuracy of the classifier (Liu, 2007), (Delen, Walker, & Kadam, 2017), (Song et al., 2004). The support vector machine with polynomial kernels and radial basis function can also be implemented (Song, Mitnitski, Cox, & Rockwood, 2004) for the optimization of the predictive models. Logistic regression is the most commonly used algorithm for the identification of diseases, hence a comparison is made on the prediction accuracy of both logistic regression and support vector machine models. The work is done to empirically determine which of the two classifiers has better performance resulting into research question:

“Can accuracy of models build with Support Vector Machine (SVM) for the prediction of mortality risk in elderly using biomarkers of the patients outperforms models build with Logistic Regression (LR) ?”

6.3 Contributions and impact

This research explores the application of supervised machine learning techniques in clinical studies. The population of elderly is classified under two broader categories, one with mortality risk and another without any risk using machine learning modeling. Machine learning algorithms are collaborated with biomarkers to build the mortality risk predictive models. In total, four models are build and average mean accuracy of each classifier is measured to identify the performance of each model. To begin
with, support vector machine based models provided a greater accuracy than logistic regression models making it more suitable for real world diagnosis of mortality risk.

Secondly, a number of biological markers are used in this research. The correlation between each biomarker and mortality risk is measured which gives an insight into the strength of that particular biomarker explaining the mortality risk. The direction of the relationship between mortality risk and biomarkers is also calculated which proves the importance of each biomarker and gives an option to include or exclude a biomarker in the modeling phase. Researchers can use these identified biomarkers in further researches on mortality diagnosis.

Data pre-processing machine learning techniques i.e. SMOTE algorithm, Z-score standardization, one-hot encoding, stratified K-fold validation are put into effect to optimize the results obtained from each model. Initially, the models are build over imbalanced values of mortality risk and later same models are trained over balanced dataset to study the impact and importance of the balanced dataset in modeling. SVM with radial basis and polynomial kernels achieved higher accuracy tuned with different values of ‘C’ and ‘γ’, and future work can be done focusing on tuning of the models to build more accurate predictive models.

6.4 Future Work & recommendations

This project only focused on two algorithms: logistic regression and support vector machine, however performance of models built over neural networks, classification trees and random forests (Maroco et al., 2011) can be further compared to find the best model in terms of learning time, prediction accuracy and size of data available. As a result of experiments, it is found that tuning of support vector machine models using ‘C’ and ‘γ’ has a huge impact on the performance of the models. In the on going research, only 10 distinct values of hyper-parameters and tuning parameters are used to tune the models due to the time constraint. Hence, future work can be done on the model tuning to enhance the prediction accuracy of SVM models.

The dataset used in the experiment is very small in size and models are trained
on limited biological markers. In order to build a more realistic model a maximum number of biomarkers and genetic traits could be included in the experiment. Also, further research can be conducted on monitoring and capture more patient’s information which will help in building a more generic model. The models are defined for elderly population only provides no sufficient evidence on the risk of mortality on younger population (<50 years). The distribution of age group taken for training can be widen using data having youth population biomarkers and future research can be done to study the relation between the biomarkers and the risk of mortality in youths. The research can prove beneficial in making accurate prognosis as soon as an individual comes under mortality risk zone.

Researchers might seek to implement leave-one-out (LOO) or percentage split techniques for generating training and testing data. LOO is good for a very small dataset and is based on the concept of training entire dataset and leaving one instance for training. Feature selection is not included in the current research study because all the predictor variables are weakly correlated with each other and also the dataset contains limited information. Excluding any instance or variable may result in the loss of information and may degrade the performance of the models. Hence, another area for future research would be to employ feature engineering or feature selection techniques like random forest classifier to identify the clinically relevant biological markers. Further taking feature selection to next level, backward and forward logistic regression models could be build as an extension of this research.

Eventually, another approach that can be explored for predicting mortality without using Machine Learning is defeasible reasoning (Longo, Kane, & Hederman, 2012; Longo & Dondio, 2014). Here, instead of inducing a model from data, a knowledge base, built with the expertise of doctors, can be elicited for the prediction of mortality (Longo & Hederman, 2013). This knowledge base can be built using arguments (Longo, 2016), which are pieces of evidence and reasons believed to be useful for mortality prediction. These arguments can interact and contradict each other (Rizzo, Dondio, Delany, & Longo, 2016). However, through the use of semantics, approaches for the resolution of contradictions, justifiable and more rationale inference can be made.


REFERENCES


references


Appendix A

Mortality risk prediction code: Balanced dataset

```python

""
Mortality risk prediction model using Logistic Regression and Support Vector Machine (linear, radial and polynomial)
""

#libraries for dataframe
import pandas as pd
from pandas import DataFrame
import numpy as np

#libraries for plots
import matplotlib
matplotlib.use('TkAgg')
import seaborn as sns
import matplotlib.pyplot as plt

#libraries for preprocessing and validation
from sklearn import preprocessing
```
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import StratifiedKFold

#libraries for models
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

#libraries for evaluation
from sklearn.metrics import roc_curve, auc
from scipy import stats

from itertools import cycle
from scipy import interp

class Analysis():
    def __init__(self, df):
        
        
            :param df: raw dataset
            
                self.df = df

                #balanced dataset
                self.X_bal = None
                self.y_bal = None

                #training and test dataset
                self.X_train = None
                self.X_test = None
self.y_train = None
self.y_test = None

#accuracies of each model per k-fold iteration
self.auc_lr = None
self.auc_svm = None
self.auc_svmrbf = None
self.auc_svmpoly = None

def exp_analysis(self):
    
    ""
    Exploratory data analysis
    :return:
    ""

    # Statistical analysis of data
    print ("\n\nStatistical analysis of data:")
    statistical_analysis = self.df.describe()
    pd.DataFrame(statistical_analysis).to_csv('/Users/priyanka/Thesis/stats_anlaysis2')

    #print number of rows and columns
    print ("\n\nNumber of rows and columns:")
    print self.df.shape

    #print column names of the dataframe
    print self.df.columns

    #print categorical features
    print ("\n\nCategorical features:")
categorical_features = self.df.select_dtypes(include=[
    object]).columns.values
print categorical_features

# print numeric features
print ("\n\nNumeric features:")
numeric_features = self.df.select_dtypes(exclude=[object]).columns.values
print numeric_features

def missing_values(self):
    ""
    Missing data % analysis
    :return:
    """
    print ("\n\nMissing value analysis:")
total = self.df.isnull().sum().sort_values(ascending=False)
percent = (self.df.isnull().sum() / self.df.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Missing', 'Percent'])
print missing_data
pd.DataFrame(missing_data).to_csv('/Users/priyanka/Thesis/Missing_value2.csv')

print ("\nImpute missing values:")
self.df['Clear'] = self.df['Clear'].fillna(self.df['Clear'].mean())
self.df['TSH'] = self.df['TSH'].fillna(self.df['TSH'].mean())

self.df['CRP'] = self.df['CRP'].fillna(self.df['CRP'].mean())

total = self.df.isnull().sum().sort_values(ascending=False)

percent = (self.df.isnull().sum() / self.df.isnull().count()).sort_values(ascending=False)

missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

print missing_data

pd.DataFrame(missing_data).to_csv('/Users/priyanka/Thesis/Impute_missing2.csv')

def plots(self):
    """
    Plots graph of numeric variables
    :return:
    """

df_num = self.df[self.df.dtypes[(self.df.dtypes == "float64") | (self.df.dtypes == "int64")].index]

df_num.hist(figsize=[8, 8])

plt.tight_layout()

plt.show()

sv_lab = 'no\_risk'

nsv_lab = 'risk'

sns.set(color_codes=True)
```python
fig, axes = plt.subplots(nrows=3, ncols=5, figsize=(12,12))
ax = sns.distplot(df_num[df_num['TARGET'] == 1].ANA,
                  bins=20, label=sv_lab, ax=axes[0][0])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].ANA,
                  bins=20, label=nsv_lab, ax=axes[0][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].BMI,
                  bins=20, label=sv_lab, ax=axes[0][1])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].BMI,
                  bins=20, label=nsv_lab, ax=axes[0][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].CMV,
                  bins=20, label=sv_lab, ax=axes[0][2])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].CMV,
                  bins=20, label=nsv_lab, ax=axes[0][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].CORTIS,
                  bins=20, label=sv_lab, ax=axes[0][3])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].CORTIS,
                  bins=20, label=nsv_lab, ax=axes[0][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].CRP,
                  bins=20, label=sv_lab, ax=axes[0][4])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].CRP,
                  bins=20, label=nsv_lab, ax=axes[0][4])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].Chol,
                  bins=20, label=sv_lab, ax=axes[1][0])
```
ax = sns.distplot(df_num[df_num['TARGET'] == 0].Chol, bins=20, label=nsv_lab, ax=axes[1][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].Clear, bins=20, label=sv_lab, ax=axes[1][1])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].Clear, bins=20, label=nsv_lab, ax=axes[1][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].E, bins=20, label=sv_lab, ax=axes[1][2])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].E, bins=20, label=nsv_lab, ax=axes[1][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].EBV, bins=20, label=sv_lab, ax=axes[1][3])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].EBV, bins=20, label=nsv_lab, ax=axes[1][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].FOLNA, bins=20, label=sv_lab, ax=axes[1][4])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].FOLNA, bins=20, label=nsv_lab, ax=axes[1][4])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].FT3, bins=20, label=sv_lab, ax=axes[2][0])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].FT3, bins=20, label=nsv_lab, ax=axes[2][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].FT4, bins=20, label=sv_lab, ax=axes[2][1])

80
ax = sns.distplot(df_num[df_num['TARGET'] == 0].FT4, bins=20, label=nsv_lab, ax=axes[2][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].HB, bins=20, label=sv_lab, ax=axes[2][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 0].HB, bins=20, label=nsv_lab, ax=axes[2][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].HPA, bins=20, label=sv_lab, ax=axes[2][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 0].HPA, bins=20, label=nsv_lab, ax=axes[2][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].HTC, bins=20, label=sv_lab, ax=axes[2][4])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 0].HTC, bins=20, label=nsv_lab, ax=axes[2][4])
ax.legend()
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(nrows=3, ncols=5, figsize=(12, 12))
ax = sns.distplot(df_num[df_num['TARGET'] == 1].HbA1c, bins=20, label=sv_lab, ax=axes[0][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 0].HbA1c, bins=20, label=nsv_lab, ax=axes[0][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].IGE, bins=20, label=sv_lab, ax=axes[0][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 0].IGE, bins=20, label=nsv_lab, ax=axes[0][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].LE, bins=20, label=sv_lab, ax=axes[0][2])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].LE, bins=20, label=nsv_lab, ax=axes[0][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].LY, bins=20, label=sv_lab, ax=axes[0][3])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].LY, bins=20, label=nsv_lab, ax=axes[0][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].MMS, bins=20, label=sv_lab, ax=axes[0][4])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].MMS, bins=20, label=nsv_lab, ax=axes[0][4])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].NEU, bins=20, label=sv_lab, ax=axes[1][0])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].NEU, bins=20, label=nsv_lab, ax=axes[1][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].RF, bins=20, label=sv_lab, ax=axes[1][1])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].RF, bins=20, label=nsv_lab, ax=axes[1][1])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].TSH, bins=20, label=sv_lab, ax=axes[1][2])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].TSH,  
               bins=20, label=nsv_lab, ax=axes[1][2])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].VITB12,  
               bins=20, label=sv_lab, ax=axes[1][3])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].VITB12,  
               bins=20, label=nsv_lab, ax=axes[1][3])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].age,  
               bins=20, label=sv_lab, ax=axes[1][4])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].age,  
               bins=20, label=nsv_lab, ax=axes[1][4])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].skinf,  
               bins=20, label=sv_lab, ax=axes[2][0])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].skinf,  
               bins=20, label=nsv_lab, ax=axes[2][0])
ax.legend()
ax = sns.distplot(df_num[df_num['TARGET'] == 1].w_h,  
               bins=20, label=sv_lab, ax=axes[2][1])
ax = sns.distplot(df_num[df_num['TARGET'] == 0].w_h,  
               bins=20, label=nsv_lab, ax=axes[2][1])
ax.legend()
plt.tight_layout()
plt.show()
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```python

""

color = ['blue', 'green']
fig, axes = plt.subplots(3, 4, figsize=(16, 10))
sns.countplot('TARGET', data=self.df, ax=axes[0, 0], alpha = 0.5)
sns.countplot('sex', hue='TARGET', data=self.df, ax=axes[0, 1], alpha = 0.5)
sns.countplot('hypert', hue='TARGET', data=self.df, ax=axes[0, 2], alpha = 0.5)
sns.countplot('DM', hue='TARGET', data=self.df, ax=axes[0, 3], alpha = 0.5)
sns.countplot('CVD', hue='TARGET', data=self.df, ax=axes[1, 0], alpha = 0.5)
sns.countplot('COPB', hue='TARGET', data=self.df, ax=axes[1, 1], alpha = 0.5)
sns.countplot('aller_d', hue='TARGET', data=self.df, ax=axes[1, 2], alpha = 0.5)
sns.countplot('dr_aller', hue='TARGET', data=self.df, ax=axes[1, 3], alpha = 0.5)
sns.countplot('analg', hue='TARGET', data=self.df, ax=axes[2, 0], alpha = 0.5)
sns.countplot('neo', hue='TARGET', data=self.df, ax=axes[2, 1], alpha = 0.5)
sns.countplot('Psy', hue='TARGET', data=self.df, ax=axes[2, 2], alpha = 0.5)
plt.tight_layout()
plt.show()

def correlation_matrix(self, target_var):
    """
```
heatmap matrix to represent correlation between price and other features

:param target_var: price

:return:

""

fig, ax = plt.subplots(figsize=(8, 6))
correlation = self.df.select_dtypes(include=['float64', 'int64']).iloc[:, 1:].corr()
sns.heatmap(correlation, ax=ax, vmax=1, square=True)
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.title('Correlation matrix')
plt.tight_layout()
plt.show()

k = 28 # number of variables for heatmap
f, ax = plt.subplots(figsize=(9,9))
corrmat = self.df.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corrmat, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
cols = corrmat.nlargest(k, target_var)[target_var].index

cm = np.corrcoef(self.df[cols].values.T)
sns.set(font_scale=1.0)
hm = sns.heatmap(cm, mask = mask, cbar=True, annot=True,
    square=True, fmt='.2f', annot_kws={'size': 7},
    yticklabels=cols.values,
    xticklabels=cols.values)
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.title('Annotated heatmap matrix')
plt.tight_layout()
plt.show()

corr = self.df.corr()[target_var]
correlation = (corr[np.argsort(corr, axis=0)::-1])
plt.figure(figsize=(10, 10))
correlation.plot(kind="barh", fontsize=10, color = 'r')
plt.title('Positive and Negative correlation with Target Mortality Risk')
plt.tight_layout()
plt.show()

def scatter_plot(self):
    
    scatterplot of numeric variables with target variable
    :param cols: numeric variables
    :param filename1: Filename to save scatter plot
    :return:
    
    sns.set(style="ticks", color_codes=True, font_scale=0.8)
features_mean = ['age', 'HbA1c', 'Chol', 'BMI', 'w_h', 'skinf', 'MMS', 'CMV', 'EBV', 'TARGET']
sns.pairplot(self.df[features_mean], hue='TARGET',
plot_kws={'s': 10}, markers=['o', 's'], palette='husl', size=1.0)
plt.tight_layout()
plt.show()
def feature_standard(self):
    
    """
    calculate z-score of each numeric variable
    :return:
    """
    self.df[cols] = preprocessing.scale(self.df[cols])

def one_hot_encoding(self):
    
    """
    Create lists for categorical variables (columns) to generate the dummy variables
    :return:
    """
    col = ['sex', 'hypert', 'DM', 'CVD', 'COPB', 'aller_d', 'dr_aller', 'analgl', 'neo', 'Psy']
    print ("\n\nColumns to be hot-encoded:")
    print col
    self.df = pd.get_dummies(self.df, columns=col)
    print self.df.columns

def feature_selection(self):
    
    """
    plot graph of features according to their importance
    :return:
    """
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```python
""
names = self.df.columns.values
rfc = RandomForestClassifier()
Y = self.df['TARGET']
X = self.df.drop('TARGET', 1)
rfc.fit(X, Y)

# Print the results
print("Features sorted by their score:"
print(sorted(zip(map(lambda x: round(x, 4), rfc.feature_importances_), names), reverse=True))
importance = rfc.feature_importances_
sorted_importances = np.argsort(importance)
features = np.arange(len(names)-1)

# Plot the data
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(111)
ax.barh(features, importance[sorted_importances], align='center', color = 'green', alpha = 0.5)
plt.yticks(features, names[sorted_importances])
plt.xlabel("Relative Importance")
plt.title("Variable Importance")
plt.tight_layout()
plt.show()

def imbalanced_data(self):
    ""
    Apply SMOTE algorithm to balance the target feature
    :return:
```
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

""
print ("\nDistribution of Target variable")
print self.df.TARGET.value_counts()
y = self.df['TARGET']
x = self.df.drop('TARGET', 1)
oversampler = SMOTE(random_state=15)
self.X_bal, self.y_bal = oversampler.fit_sample(x, y)

def lr_model(self):
    ""
    logistic regression model
    :return:
    ""
    seed = 7
    model_lr = LogisticRegression()
    skf = StratifiedKFold(n_splits=10, random_state=seed)
    skf.get_n_splits(self.X_bal, self.y_bal)
    #ROC curve
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
    colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow',
                     'blue', 'darkorange', 'red', 'brown', 'green', '
                     grey'])
    lw = 2
    i = 0
    self.auc_lr = {}
    print ("\n")
    for (train_index, test_index), color in zip (skf.split(
        self.X_bal, self.y_bal), colors):
        ...
X_trainN, X_testN = self.X_bal[train_index], self.X_bal[test_index]
y_trainN, y_testN = self.y_bal[train_index], self.y_bal[test_index]
model_lr.fit(X_trainN, y_trainN)
pred_lr = model_lr.predict(X_testN)

# Compute ROC curve and area the curve
fpr, tpr, thresholds = roc_curve(self.y_bal[test_index], pred_lr)
mean_tpr += interp(mean_fpr, fpr, tpr)
mean_tpr[0] = 0.0
roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=lw, color=color, label='ROC fold %d (area = %0.2f)' % (i + 1, roc_auc))
self.auc_lr[i] = roc_auc
i += 1

print self.auc_lr
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
mean_tpr /= skf.get_n_splits(self.X_bal, self.y_bal)
mean_tpr[-1] = 1.0
mean_auc_lr = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--', label='Mean ROC (area = %0.2f)' % mean_auc_lr)
print 'Average LR classifier accuracy = %0.2f' % mean_auc_lr
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Logistic regression')
plt.legend(loc="lower right")
plt.show()

#return mean_auc_lr

def svm_model(self):
    
    ""
    support vector machine linear model
    :return:
    ""
    
    parameters = {'kernel' : ['linear'], 'C': [1,2,3,4,5,6,7,8,9,10], 'gamma': [0.01,0.02,0.03,0.04,0.05,0.10,0.2,0.3,0.4,0.5]}
    seed = 7
    model_svm = svm.SVC()
    grid = GridSearchCV(model_svm, parameters)
    skf = StratifiedKFold(n_splits=10, random_state=seed)
    skf.get_n_splits(self.X_bal, self.y_bal)
    print ("
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
    colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow',
                    'blue', 'darkorange', 'red', 'brown', 'green', '
                    grey'])
    lw = 2
    i = 0
    self.auc_svm = {"}
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```
print("\n")
for (train_index, test_index), color in zip (skf.split(
    self.X_bal, self.y_bal), colors):
    X_trainN, X_testN = self.X_bal[train_index], self
    .X_bal[test_index]
    y_trainN, y_testN = self.y_bal[train_index], self
    .y_bal[test_index]
    grid.fit(X_trainN, y_trainN)
    pred_svm = grid.predict(X_testN)
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(self.y_bal[
        test_index], pred_svm)
    mean_tpr += interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=lw, color=color,
        label='ROC fold %d (area = %0.2f)' %
        (i + 1, roc_auc))
    self.auc_svm[i] = roc_auc
    i += 1

print self.auc_svm
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k
    )
mean_tpr /= skf.get_n_splits(self.X_bal, self.y_bal)
mean_tpr[-1] = 1.0
mean_auc_svm = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
    label='Mean ROC (area = %0.2f)' %
    mean_auc_svm, lw=lw)
```

print 'Average SVM Linear classifier accuracy = %0.2f' % mean_auc_svm
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Support Vector Machine - Linear')
plt.legend(loc="lower right")
plt.show()

def svm_model_rbf(self):
    ""
    support vector machine radial basis function model
    :return:
    ""
    parameters = {'kernel': ['rbf'], 'C': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'gamma': [0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.2, 0.3, 0.4, 0.5]}
    seed = 7
    model_svmrbf = svm.SVC()
    grid = GridSearchCV(model_svmrbf, parameters)
    skf = StratifiedKFold(n_splits=10, random_state=seed)
    skf.get_n_splits(self.X_bal, self.y_bal)
    print ("\n")
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
    colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow', 'blue', 'darkorange', 'red', 'brown', 'green', 'grey'])
lw = 2
i = 0
self.auc_svmrbf = {}
print ("\n")
for (train_index, test_index), color in zip (skf.split(self.X_bal, self.y_bal), colors):
    X_trainN, X_testN = self.X_bal[train_index], self.X_bal[test_index]
    y_trainN, y_testN = self.y_bal[train_index], self.y_bal[test_index]
    grid.fit(X_trainN, y_trainN)
pred_svm = grid.predict(X_testN)
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(self.y_bal[test_index], pred_svm)
    mean_tpr += interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=lw, color=color,
             label='ROC fold %d (area = %0.2f)' % (i + 1, roc_auc))
    self.auc_svmrbf[i] = roc_auc
    i += 1
print self.auc_svmrbf
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
mean_tpr /= skf.get_n_splits(self.X_bal, self.y_bal)
mean_tpr[-1] = 1.0
mean_auc_svmrbf = auc(mean_fpr, mean_tpr)
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```python
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
         label='Mean\ROC\(\text{area} = \%0.2f\)' %
         mean_auc_svmrbf, lw=lw)

print 'Average\SVM\Radial\classifier\accuracy = \%0.2f' %
      mean_auc_svmrbf
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False\Positive\Rate')
plt.ylabel('True\Positive\Rate')
plt.title('ROC\Curve: Support\Vector\Machine\Radial')
plt.legend(loc="lower\right")
plt.show()

def svm_model_poly(self):
    
    """
    support vector machine polynomial model
    :return:
    """

    parameters = {'kernel': ['poly'], 'C': [1, 2, 3, 4, 5,
                                          6, 7, 8, 9, 10], 'gamma':
               [0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.2, 0.3,
                0.4, 0.5]}

    seed = 7
    model_svm_poly = svm.SVC()
    grid = GridSearchCV(model_svm_poly, parameters)
    skf = StratifiedKFold(n_splits=10, random_state=seed)
    skf.get_n_splits(self.X_bal, self.y_bal)
    print ("\n")
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
```
colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow',
               'blue', 'darkorange', 'red', 'brown', 'green',
               'grey'])
lw = 2
i = 0
self.auc_svmpoly = {}
print ("\n")
for (train_index, test_index), color in zip (skf.split(
    self.X_bal, self.y_bal), colors):
    X_trainN, X_testN = self.X_bal[train_index], self
    .X_bal[test_index]
    y_trainN, y_testN = self.y_bal[train_index], self
    .y_bal[test_index]
    grid.fit(X_trainN, y_trainN)
pred_svm = grid.predict(X_testN)
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(self.y_bal[
        test_index], pred_svm)
    mean_tpr += interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=lw, color=color,
             label='ROC fold %d (area = %0.2f)' % (i + 1, roc_auc))
    self.auc_svmpoly[i] = roc_auc
    i += 1

print self.auc_svmpoly
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```python
mean_tpr /= skf.get_n_splits(self.X_bal, self.y_bal)
mean_tpr[-1] = 1.0
mean_auc_svmpoly = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
          label='Mean ROC (area = %0.2f)' % mean_auc_svmpoly, lw=lw)
print 'Average SVM Poly classifier accuracy = %0.2f' % mean_auc_svmpoly
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Support Vector Machine - Poly')
plt.legend(loc="lower right")
plt.show()
#return mean_auc_svmpoly

def compare(self):
    
    ""
    model comparison
    :return:
    ""
    np.random.seed(10)
    LR = [self.auc_lr.get(0), self.auc_lr.get(1), self.
          \(\rightarrow\) auc_lr.get(2), self.auc_lr.get(3), self.auc_lr.get
          \(\rightarrow\) (4),
          self.auc_lr.get(5), self.auc_lr.get(6), self.
          \(\rightarrow\) auc_lr.get(7), self.auc_lr.get(8), self.
          \(\rightarrow\) auc_lr.get(9)]
```

97
SVM = [self.auc_svm.get(0), self.auc_svm.get(1), self.
    → auc_svm.get(2), self.auc_svm.get(3), self.auc_svm.
    → get(4),
        self.auc_svm.get(5), self.auc_svm.get(6), self.
        → auc_svm.get(7), self.auc_svm.get(8), self
        → .auc_svm.get(9)]

SVM_RBF = [self.auc_svmrbf.get(0), self.auc_svmrbf.get
    → (1), self.auc_svmrbf.get(2), self.auc_svmrbf.get
    → (3), self.auc_svmrbf.get(4),
        self.auc_svmrbf.get(5), self.auc_svmrbf.
        → get(6), self.auc_svmrbf.get(7),
        → self.auc_svmrbf.get(8), self.
        → auc_svmrbf.get(9)]

SVM_POLY = [self.auc_svmpoly.get(0), self.auc_svmpoly.
    → get(1), self.auc_svmpoly.get(2), self.auc_svmpoly.
    → get(3), self.auc_svmpoly.get(4),
        self.auc_svmpoly.get(5), self.
        → auc_svmpoly.get(6), self.
        → auc_svmpoly.get(7), self.
        → auc_svmpoly.get(8), self.
        → auc_svmpoly.get(9)]

data_to_plot = [LR, SVM, SVM_RBF, SVM_POLY]
fig = plt.figure(1, figsize=(9, 6))
ax = fig.add_subplot(111)
bp = ax.boxplot(data_to_plot, patch_artist=True)
ax.set_xticklabels(['LR', 'SVM_LINEAR', 'SVM_RBF', '
    → SVM_POLY'])
plt.tight_layout()
plt.show()

t_statistic, p_value = stats.wilcoxon(LR, SVM)
print "\nLR - SVM\n"
print t_statistic
print p_value

t_statistic, p_value = stats.wilcoxon(LR, SVM_RBF)
print "\nLR - SVM_radial\n"
print t_statistic
print p_value

t_statistic, p_value = stats.wilcoxon(LR, SVM_POLY)
print "\nLR - SVM_polynomial\n"
print t_statistic
print p_value

def main():
    # Read data from csv file and store in dataframe 'df'
    df = DataFrame.from_csv("/Users/priyanka/Thesis/mortality_data.csv")
a = Analysis(df)
a.exp_analysis()
a.missing_values()
a.plots()
a.correlation_matrix("TARGET")
a.plots_cat()
a.scatter_plot()
a.feature_standard()
a.one_hot_encoding()
a.feature_selection()
APPENDIX A. MORTALITY RISK PREDICTION CODE: BALANCED DATASET

```python
a.imbalanced_data()
a.lr_model()
a.svm_model()
a.svm_model_rbf()
a.svm_model_poly()
a.compare()

if __name__ == '__main__':
    main()
```
Appendix B

Mortality risk prediction code: Imbalanced dataset

```python
# libraries for dataframe
import pandas as pd
from pandas import DataFrame
import numpy as np

# libraries for plots
import matplotlib
matplotlib.use('TkAgg')
import seaborn as sns
import matplotlib.pyplot as plt

# libraries for preprocessing and validation
from sklearn import preprocessing
from sklearn.model_selection import StratifiedKFold

# libraries for models
from sklearn import svm
from sklearn.linear_model import LogisticRegression
```
from sklearn.model_selection import GridSearchCV

#libraries for evaluation
from sklearn.metrics import roc_curve, auc
from scipy import stats
from itertools import cycle
from scipy import interp

class Analysis():

    def __init__(self, df):
        
        
        :param df: raw dataset
        
        self.df = df
        self.X = None
        self.Y = None

        self.auc_lr = None
        self.auc_svm = None
        self.auc_svmrbf = None
        self.auc_svmpoly = None

    def exp_analysis(self):
        
        
        Exploratory data analysis
        :return:
        

APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET

```python
# Statistical analysis of data
print("\n\nStatistical analysis of data:")
statistical_analysis = self.df.describe()
pd.DataFrame(statistical_analysis).to_csv('/Users/priyanka/Thesis/stats_anlaysis2')

# To print number of rows and columns
print("\n\nNumber of rows and columns:")
print(self.df.shape)

# column names of the dataframe
print(self.df.columns)

# categorical features
print("\n\nCategorical features:")
categorical_features = self.df.select_dtypes(include=[object]).columns.values
print(categorical_features)

# numeric features
print("\n\nNumeric features:")
umeric_features = self.df.select_dtypes(exclude=[object]).columns.values
print(numeric_features)

def missing_values(self):
    
    """
    Missing data % analysis
    :return:
    """
```

print ("Missing value analysis:")
total = self.df.isnull().sum().sort_values(ascending=False)
percent = (self.df.isnull().sum() / self.df.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Missing', 'Percent'])
print missing_data
pd.DataFrame(missing_data).to_csv('/Users/priyanka/Thesis/Missing_value2.csv')

print ("Impute missing values:")
self.df['Clear'] = self.df['Clear'].fillna(self.df['Clear'].mean())
self.df['TSH'] = self.df['TSH'].fillna(self.df['TSH'].mean())
self.df['CRP'] = self.df['CRP'].fillna(self.df['CRP'].mean())
total = self.df.isnull().sum().sort_values(ascending=False)
percent = (self.df.isnull().sum() / self.df.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
print missing_data
pd.DataFrame(missing_data).to_csv('/Users/priyanka/Thesis/Impute_missing2.csv')

def feature_standard(self):
    ""

104
calculate z-score of each numeric variable
:return:

```
cols = ['age', 'HbA1c', 'Chol', 'BMI', 'w_h', 'skinf',
        MMS', 'CMV', 'EBV', 'HPA', 'LE', 'NEU', 'LY', 'CRP
        E', 'HB', 'HTC', 'Clear', 'RF', 'VITB12',
        'FOLNA', 'CORTIS', 'TSH', 'FT3',
        FT4', 'ANA', 'IGE']
self.df[cols] = preprocessing.scale(self.df[cols])
```

def one_hot_encoding(self):
    
    Create lists for categorical variables(columns) to
genenerate the dummy variables
:return:

```
# Create lists for categorical variables(columns) for
which you want to create the dummy variables
col = ['sex', 'hypert', 'DM', 'CVD', 'COPB', 'aller_d',
        'dr_aller', 'analg', 'neo', 'Psy']
print ("\n\nColumns to be hot-encoded:")
print col
self.df = pd.get_dummies(self.df, columns=col)
print self.df.columns
self.Y = self.df['TARGET']
self.X = self.df.drop('TARGET', 1)
```

def lr_model(self):
    
    """
APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET

```python
logistic regression model
:return:
"

seed = 7
model_lr = LogisticRegression()
skf = StratifiedKFold(n_splits=10, random_state=seed)
skf.get_n_splits(self.X, self.Y)

#ROC curve
mean_tpr = 0.0
mean_fpr = np.linspace(0, 1, 100)

colors = cycle(["cyan", 'indigo', 'seagreen', 'yellow',
                 'blue', 'darkorange', 'red', 'brown', 'green', '
                 grey'])
lw = 2
i = 0

self.auc_lr = {}
for (train_index, test_index), color in zip (skf.split(self.X, self.Y), colors):
    X_train, X_test = self.X.iloc[train_index], self.
                      X.iloc[test_index]
    y_train, y_test = self.Y.iloc[train_index], self.
                      Y.iloc[test_index]
    model_lr.fit(X_train, y_train)
    pred_lr = model_lr.predict(X_test)
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(self.Y.iloc[test_index], pred_lr)
    mean_tpr += interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc += auc(fpr, tpr)
```

106
```python
plt.plot(fpr, tpr, lw=lw, color=color,
         label='ROC fold %d (area = %0.2f)' % (i + 1, roc_auc))
self.auc_lr[i] = roc_auc
i += 1

print self.auc_lr
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
mean_tpr /= skf.get_n_splits(self.X, self.Y)
mean_tpr[-1] = 1.0
mean_auc_lr = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
         label='Mean ROC (area = %0.2f)' % mean_auc_lr, lw=lw)
print 'Average LR classifier accuracy = %0.2f' % mean_auc_lr
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Logistic regression')
plt.legend(loc="lower right")
plt.show()

def svm_model(self):
    
    """
    support vector machine linear model
    :return:
    """
```
parameters = {'kernel': ['linear'], 'C': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'gamma': [0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.2, 0.3, 0.4, 0.5]}

seed = 7
model_svm = svm.SVC()
grid = GridSearchCV(model_svm, parameters)
skf = StratifiedKFold(n_splits=10, random_state=seed)
skf.get_n_splits(self.X, self.Y)
print ("
mean_tpr = 0.0
mean_fpr = np.linspace(0, 1, 100)
colors = cycle(
    ['cyan', 'indigo', 'seagreen', 'yellow', 'blue',
     'darkorange', 'red', 'brown', 'green', '
     grey'])

lw = 2
i = 0
self.auc_svm = {}
print ("
for (train_index, test_index), color in zip(skf.split(
    self.X, self.Y), colors):
    X_trainN, X_testN = self.X.iloc[train_index],
    self.X.iloc[test_index]
y_trainN, y_testN = self.Y.iloc[train_index],
    self.Y.iloc[test_index]
grid.fit(X_trainN, y_trainN)
pred_svm = grid.predict(X_testN)
# Compute ROC curve and area the curve
fpr, tpr, thresholds = roc_curve(self.Y.iloc[test_index], pred_svm)
mean_tpr += interp(mean_fpr, fpr, tpr)
mean_tpr[0] = 0.0
roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=lw, color=color,
         label='ROC fold %d (area = %0.2f)' % (i + 1, roc_auc))
self.auc_svm[i] = roc_auc
i += 1

print self.auc_svm
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
mean_tpr /= skf.get_n_splits(self.X, self.Y)
mean_tpr[-1] = 1.0
mean_auc_svm = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
         label='Mean ROC (area = %0.2f)' % mean_auc_svm, lw=lw)
print 'Average SVM Linear classifier accuracy = %0.2f' % mean_auc_svm
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Support Vector Machine - Linear')
plt.legend(loc="lower right")
plt.show()
def svm_model_rbf(self):
    
    """
    support vector machine radial basis function model
    :return:
    """

    parameters = {'kernel': ['rbf'], 'C': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'gamma': [0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.2, 0.3, 0.4, 0.5]}

    seed = 7
    model_svmrbf = svm.SVC()
    grid = GridSearchCV(model_svmrbf, parameters)
    skf = StratifiedKFold(n_splits=10, random_state=seed)
    skf.get_n_splits(self.X, self.Y)
    print ("
"
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
    colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow', 'blue', 'darkorange', 'red', 'brown', 'green', 'grey'])
    lw = 2
    i = 0
    self.auc_svmrbf = {}
    print ("
"
    for (train_index, test_index), color in zip (skf.split(self.X, self.Y), colors):
        X_trainN, X_testN = self.X.iloc[train_index], self.X.iloc[test_index]
        y_trainN, y_testN = self.Y.iloc[train_index], self.Y.iloc[test_index]

grid.fit(X_trainN, y_trainN)
pred_svm = grid.predict(X_testN)

# Compute ROC curve and area the curve
fpr, tpr, thresholds = roc_curve(self.Y.iloc[test_index], pred_svm)
mean_tpr += interp(mean_fpr, fpr, tpr)
mean_tpr[0] = 0.0
roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=lw, color=color,
label='ROC fold %0.2f' % (i + 1, roc_auc))
self.auc_svmrbf[i] = roc_auc
i += 1

print self.auc_svmrbf
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k')
mean_tpr /= skf.get_n_splits(self.X, self.Y)
mean_tpr[-1] = 1.0
mean_auc_svmrbf = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
label='Mean ROC %0.2f' % mean_auc_svmrbf, lw=lw)

print 'Average SVM Radial classifier accuracy %0.2f' % mean_auc_svmrbf
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Support Vector Machine - Radial')
plt.legend(loc="lower right")
plt.show()

def svm_model_poly(self):
    ""
    support vector machine polynomial model
    :return:
    ""

    parameters = {'kernel': ['poly'], 'C': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'gamma': [0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.2, 0.3, 0.4, 0.5]}

    seed = 7
    model_svmpoly = svm.SVC()
    grid = GridSearchCV(model_svmpoly, parameters)
    SKF = StratifiedKFold(n_splits=10, random_state=seed)
    SKF.get_n_splits(self.X, self.Y)
    print ("\n")
    mean_tpr = 0.0
    mean_fpr = np.linspace(0, 1, 100)
    colors = cycle(['cyan', 'indigo', 'seagreen', 'yellow', 'blue', 'darkorange', 'red', 'brown', 'green', 'grey'])
    lw = 2
    i = 0

    self.auc_svmpoly = {}
    print ("\n")
    for (train_index, test_index), color in zip (SKF.split(self.X, self.Y), colors):
        ...
APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET

```python
X_trainN, X_testN = self.X.iloc[train_index],
                   self.X.iloc[test_index]
y_trainN, y_testN = self.Y.iloc[train_index],
                   self.Y.iloc[test_index]
grid.fit(X_trainN, y_trainN)
pred_svm = grid.predict(X_testN)

# Compute ROC curve and area the curve
fpr, tpr, thresholds = roc_curve(self.Y.iloc[test_index], pred_svm)
mean_tpr += interp(mean_fpr, fpr, tpr)
mean_tpr[0] = 0.0
roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=lw, color=color,
         label='ROC fold %d (area = %0.2f)
               % (i + 1, roc_auc))

self.auc_svmpoly[i] = roc_auc
i += 1

print self.auc_svmpoly
plt.plot([0, 1], [0, 1], linestyle='--', lw=lw, color='k
         -- ')
mean_tpr /= skf.get_n_splits(self.X, self.Y)
mean_tpr[-1] = 1.0
mean_auc_svmpoly = auc(mean_fpr, mean_tpr)
plt.plot(mean_fpr, mean_tpr, color='g', linestyle='--',
         label='Mean ROC (area = %0.2f) %
               mean_auc_svmpoly, lw=lw)
print 'Average SVM Poly classifier accuracy = %0.2f' %
      mean_auc_svmpoly
plt.xlim([-0.05, 1.05])
```

113
APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET

```python
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve: Support Vector Machine - Poly')
plt.legend(loc="lower right")
plt.show()

def compare(self):
    ""
    model comparison
    :return:
    ""
    np.random.seed(10)

    LR = [self.auc_lr.get(0), self.auc_lr.get(1), self.auc_lr.get(2), self.auc_lr.get(3), self.auc_lr.get(4),
          self.auc_lr.get(5), self.auc_lr.get(6), self.auc_lr.get(7), self.auc_lr.get(8), self.auc_lr.get(9)]

    SVM = [self.auc_svm.get(0), self.auc_svm.get(1), self.auc_svm.get(2), self.auc_svm.get(3), self.auc_svm.get(4),
           self.auc_svm.get(5), self.auc_svm.get(6), self.auc_svm.get(7), self.auc_svm.get(8), self.auc_svm.get(9)]

    SVM_RBF = [self.auc_svmrbf.get(0), self.auc_svmrbf.get(1), self.auc_svmrbf.get(2), self.auc_svmrbf.get(3),
               self.auc_svmrbf.get(4), self.auc_svmrbf.get(5), self.auc_svmrbf.get(6), self.auc_svmrbf.get(7),
               self.auc_svmrbf.get(8), self.auc_svmrbf.get(9)]
```

114
APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET

(3), self.auc_svmrbf.get(4),
    self.auc_svmrbf.get(5), self.auc_svmrbf.
    get(6), self.auc_svmrbf.get(7),
    self.auc_svmrbf.get(8), self.
    self.auc_svmrbf.get(9))]

SVM_POLY = [self.auc_svmpoly.get(0), self.auc_svmpoly.
    get(1), self.auc_svmpoly.get(2), self.auc_svmpoly.
    get(3), self.auc_svmpoly.get(4),
    self.auc_svmpoly.get(5), self.
    self.auc_svmpoly.get(6), self.
    self.auc_svmpoly.get(7), self.
    self.auc_svmpoly.get(8), self.
    self.auc_svmpoly.get(9)]

data_to_plot = [LR, SVM, SVM_RBF, SVM_POLY]
fig = plt.figure(1, figsize=(9, 6))
ax = fig.add_subplot(111)
bp = ax.boxplot(data_to_plot, patch_artist=True)
ax.set_xticklabels(['LR', 'SVM_LINEAR', 'SVM_RBF', 'SVM_POLY'])
plt.tight_layout()
plt.show()

t_statistic, p_value = stats.wilcoxon(LR, SVM)
print "\n\nLR - SVM"
print t_statistic
print p_value

t_statistic, p_value = stats.wilcoxon(LR, SVM_RBF)
print "\n\nLR - SVM radial"
```python
print t_statistic
print p_value
t_statistic, p_value = stats.wilcoxon(LR, SVM_POLY)
print "\nLR SVM polynomial"
print t_statistic
print p_value

def main():
    # Read data from csv file and store in dataframe 'df'
    df = DataFrame.from_csv("/Users/priyanka/Thesis/mortality_data.csv")
a = Analysis(df)
a.exp_analysis()
a.missing_values()
a.feature_standard()
a.one_hot_encoding()
a.lr_model()
a.svm_model()
a.svm_model_rbf()
a.svm_model_poly()
a.compare()

if __name__ == '__main__':
    main()
```

APPENDIX B. MORTALITY RISK PREDICTION CODE: IMBALANCED DATASET