Risk prediction of chronic disease using machine learning and rebalancing methods

This research evaluated and compared the performance of SMOTE, Resampling, and SpreadSubsampling rebalancing techniques in six machine learning classifiers on four different data sets

A Thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of Master of Computer and Information Sciences

By

Cheng Li

Supervisor

Dr. Farhaan Mirza

June 2021

School of Engineering, Computer and Mathematical Sciences

Abstract

Chronic diseases cause damage to important organs such as the brain, heart, and liver, which can easily cause disability, affect labor ability and quality of life, and the medical expenses are extremely high, which increases the economic burden of society and families. An effective method is to create predictive models to assess the risk of chronic diseases. Researchers have conducted several projects, but challenges still exist.

The challenge is the imbalance of chronic disease data. When encountering unbalanced chronic diseases data, the classification algorithms will calculate the majority class (non- disease), while the minority class sample (disease) is not calculated. In order to accurately identify the disease and non-disease individuals, this research proposes a multi-combination method to deal with chronic disease data sets with imbalanced categories. The researcher conducted an in-depth analysis of the impact of three rebalancing methods: Synthetic minority oversampling technique (SMOTE), Resampling and SpreadSubsampling on the classifier processing through six classifiers and four data sets. Experimental results show that Random Forest (RF) combined with Resample rebalancing method (RF-RESAMPLE) is the best classifier of our selection of data sets and achieved 94.8770%. The method can assist doctors to identify chronic diseases, and then diagnose and treat patients early to increase their chances of survival.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of student

Acknowledgements

Many thanks to:

My supervisor Dr. Farhaan Mirza, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology. The research area is challenging for me. I did not know it very well at the beginning. He helped me whenever I encountered difficulties and gave me very patient guidance. From the initial research direction to the determination of the research topic, to the literature review and the final thesis writing, he always gave me constructive suggestions and encouragement. Therefore, I want to thank him for his help and guidance for me to complete my academic research thesis, and make me believe that anything can be done as long as I work hard.

My wife Jie Liu who supports me whenever I encounter difficulties in research or life will always be here to give me spiritual support and encouragement. In addition, she spent a lot of time helping me take care of my little son Yiting Li.

Finally, I appreciated my friends, Dr. Xiaojun Gao, Lei Wang, and Dehai Yang. When I ask them for advice, they will give me recommendations and enthusiastic support.

Chapter 1

Introduction

As the pace of life accelerates aging, the number of patients with chronic diseases is increasing every year, and there is a trend for them to be younger. This is not only a personal health problem, but also a major public health problem endangering society. Chronic diseases will seriously affect the health and life of patients and many chronic diseases are incurable, and even affect patients for life. This brings a heavy burden to the patients' families and communities (World Health Organization, 2020).

In 2019, seven of the top 10 leading causes of death worldwide were chronic diseases, accounting for 74% of global deaths. 15 million people between the ages of 30 and 69 die each year from non-communicable diseases and more than 85 per cent of these "premature" deaths occur in low and middle-income countries. Of these, 17.9 million die annually from cardiovascular disease, followed by 9 million from cancer, 3.9 million from respiratory disease and 1.6 million from diabetes (WHO, 2020).

The number of deaths from heart disease has increased by more than two million since 2000 to 8.9 million, accounting for 16% of the world's deaths. The number of deaths from chronic kidney disease rose from 813000 in 2000 to 1.3 million in 2019. The number of deaths from bronchi and lung cancer rose from 1.2 million to 1.8 million. Diabetes mortality has also increased by 80% since 2000 (WHO, 2020).

Chronic disease risk prediction assessment has developed rapidly in the treatment of disease in recent years and has an increasing impact on chronic disease outcomes such as treatment monitoring and clinical diagnosis, etc. Its purpose is to extract hidden and useful information from a large amount of ambiguous disease data and predict future trends. The research of chronic disease status and risk prediction technology is mainly through clinical epidemiological investigations to obtain a large amount of data characteristics, and use risk prediction technology to analyze and research different specific disease statuses, influencing factors and symptom characteristics. This will also greatly improve the management level of chronic disease, improve the effectiveness of chronic disease prevention and control and reduce medical and costs (Shuja,Mittal and Zaman, 2020).

In chronic disease risk prediction assessment, the disease data sets are unbalanced at the time of collection. There are always more people with non-disease than those with disease. For example, most instances that are non- disease are marked as belonging to a group called majority class, while a few instances that are disease are marked as belonging to another group called minority class. The imbalance issue is very common in medical data and the established classifiers often produce high accuracy for most categories and low prediction accuracy for a few categories (Kumari and Singh, 2013; Motka, Parmarl, Kumar and Verma, 2013; Vijayan and ravikumar, 2014).

In fact, there are imbalanced data in many areas, such as medical, fraud phone detection, information retrieval and filtering tasks. In these areas, we are interested in a few categories rather than a majority. Therefore, we have to make fairly high predictions about minorities. When the data are extremely unbalanced, most of the samples are easier to predict, and the performance of a few classes is poor. If the data set is extremely unbalanced, even if the classifier correctly classifies most of the samples and misclassifies all a few samples, the accuracy of the classifier is still very high. In this case, the accuracy cannot reflect the reliable prediction of a few categories. Therefore, a more reasonable evaluation index is needed.

This thesis proposes a risk prediction model based on multiple rebalancing methods and multiple machine learning classifiers. Therefore, we will use Weka machine learning tool for research experiments and compare three rebalancing methods to reduce class imbalance issues on the multiple data sets. In the second part, we use the balanced data sets and a multiple of machine learning classifiers to find the best classifier for predicting chronic diseases.

1.1 Research Motivation

In the risk assessment of chronic diseases, classifier has become an important decision-making tool. Researchers have proposed a variety of classifiers to assess the risk of chronic diseases in the medical field. The common classifiers include Decision Tree, Support Vector Machine (SVM), Naive Bayes, Random Forest, Bayesian Network and K-nearest neighbors (KNN).

However, the chronic risk prediction model is based on medical data sets, and most medical data sets are not uniformly distributed, and one class is often more instances than the other. This leads to an imbalance in the data set. When a classifier is built on an unbalanced data set, the classifier tends to produce high accuracy for most classes and less prediction accuracy for a few classes. So even though the classifier achieves a good result, this result is not accurate.

The first major landmark Symposium on "Class Imbalance Problem" was presented at the American Association for Article Intelligence Conference in early 2000. They further understood the factors that affect the accuracy of minority and majority classes, because in the imbalanced domain, accuracy highly depends on the trade-off between the data and the sampling method is proposed to solve the imbalance issue (Japkowicz & Holtz, 2000).

There are two sampling methods to solve the data imbalance issues: undersampling and oversampling;

The undersampling method can produce more compact data sets, thus reducing the processing events and costs faced by the classifier in the training phase. However, the disadvantage of undersampling is that it discards most of the counterexample data, which weakens the influence of the middle part of the counterexample and leads to a large deviation model (Wasikowski & Chen, 2010).

In order to solve this problem, the oversampling method which does not need to reduce the majority sets has been found by researchers. It deals with class imbalance by copying a few class examples. The sampling techniques that are known to belong to this category are copied to make use of other examples of a few classes. However, the application of random oversampling needs to adjust the important weight of minorities. However, only when the learning algorithm is competent to distinguish the category type, noise and clustering, can these weights be correctly determined and calculated (Wasikowski & Chen, 2010).

Therefore, we will compare the rebalancing methods. This will contribute to the research of chronic diseases. In recent years, the field of machine learning has changed the field of chronic disease research with higher accuracy and optimization performance.

1.2 Research Gaps

Based on our literature review (Chapter 2) we found the following key research gaps:

Research Gap 1:

In the chronic disease risk prediction model, only a few classifiers are compared because different classifiers have different performance results. Therefore, some possibly better classifiers are ignored. See more details in (2.4.1).

Research Gap 2:

Studies classifiers are performed on imbalanced data sets. This will lead to inaccurate results. See more details in (2.4.2).

Research Gap 3:

Due to the imbalance of medical data, researchers used different sampling methods for the imbalanced data, but did not mention to what degree should one balance the original data set. See more details in (2.4.3).

Research Gap 4:

Most articles only use one data set to train and test the classifiers because every classifier has different performance results on multiple data sets. Therefore, one data set is not enough to verify the classifier performance. See more details in

(2.4.4).

Based on the gaps above, our research aims to answer the following questions:

Q 1: Which is the best classifier to predict chronic diseases?

Q 2: Which type of sampling method is most effective for medical data?

Q 3: What degree should one balance the original data set?

Q 4: What is the difference between classifiers' performance on multiple data sets?

1.3 Research Objectives

To solve the research gaps we defined the objectives and actions of study as follows:

Table 1.1 Research Gap

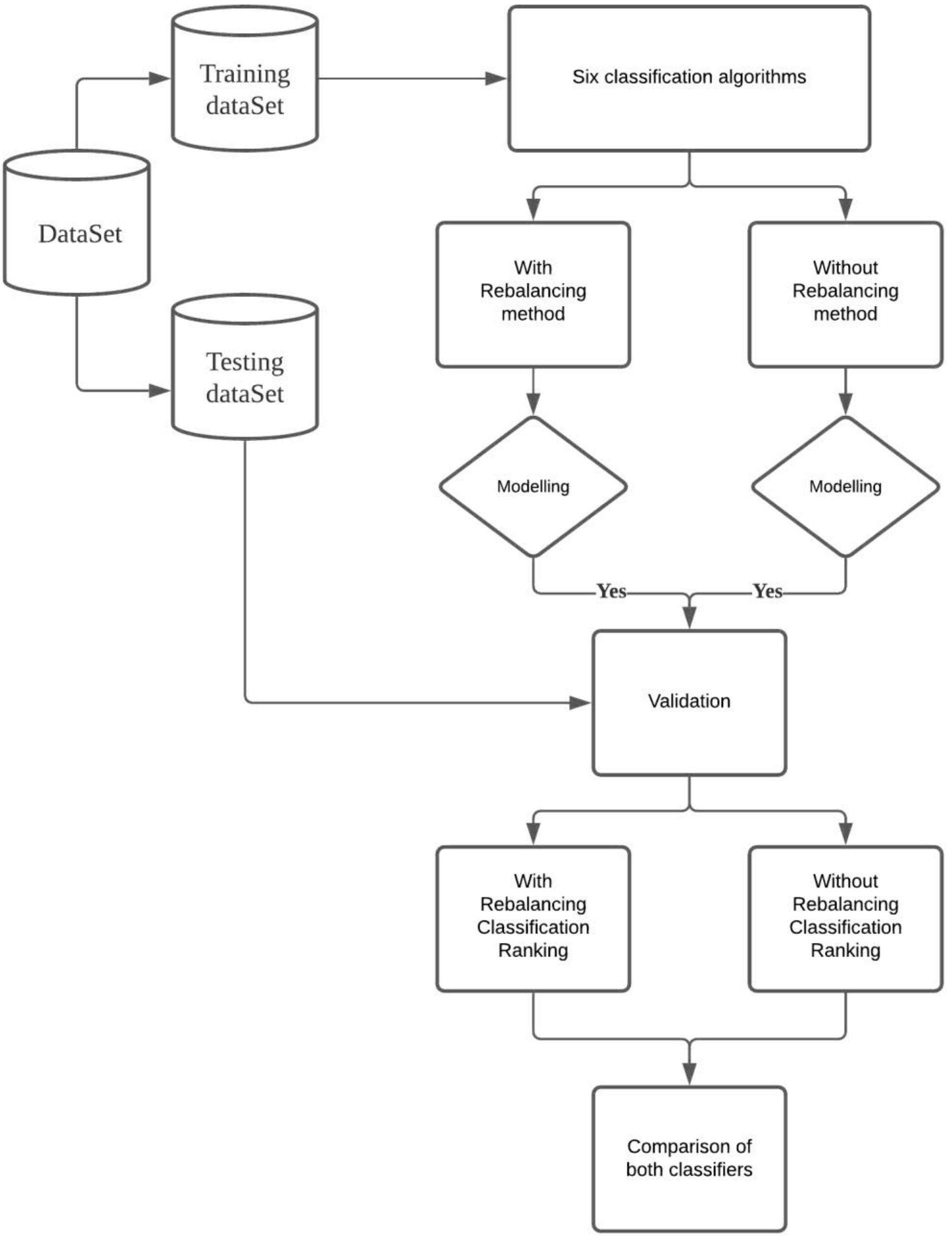
Limited classifiers issue	In this research, we plan to add more classifiers and multiple data sets for comparison.
Data rebalancing	In this research we plan to use three sampling methods: SMOTE, Resampling, and SpreadSubsampling to rebalance the data sets.
Parameters issues	We will announce the details of the adjustment parameters.
limited data set options for research	In the research, we will use multiple data sets to train and test classifiers.

The challenges of this research are to effectively evaluate the risk prediction of chronic diseases and the solution of data imbalance issues. The aim is to find the optimal classifier and expand our understanding of rebalancing technique, and evaluate its usefulness in solving some problems caused by the highly unbalanced distribution of chronic disease risk prediction.

The scope of this research is defined as follows:

- Using rebalancing methods to reduce class imbalance in the data sets
- A variety of classifiers are used for comparison to find the best classifier for risk prediction of chronic diseases
- Multiple data sets are used to train and test the classifiers to verify the fitted chronic disease risk prediction model
- Release the parameters and describe the parameter adjustment method in detail

Fig. 1.1 Summary of the complete process of conducting this research.


Training dataSet
Six classification algorithms
DataSet
With Rebalancing method
Without Rebalancing method
Testing dataSet
Modelling
Modelling
Yes
Yes
Validation
With Rebalancing Classification Ranking
Without Rebalancing Classification Ranking
Comparison of both classifiers

1.4 Research Contributions

Our work in solving the research objectives has delivered the following major contributions:

- We explored the factors of rebalancing the classes in the data set by SMOTE, Resampling, SpreadSubsampling rebalancing methods and obtained useful parameters through experiments

- We compared six machine learning classifiers, and found the best classifier for chronic disease risk prediction

- We verified the chronic disease risk prediction models through multiple data sets for performance assessment

1.5 Research Benefits

- The major benefit of this research is that it could change and improve the treatment and management of disease for governments' medical departments

- The comparison of classifiers will guide machine-learning researchers in their research on classification algorithms and rebalancing methods

- Doctors can use the machine learning risk prediction models to give treatment and recommendations for chronic disease issues

1.6 Thesis Structure

Chapter 1 - Introduction:

Presents background of this research, and the research situation of chronic diseases analysis is briefly summarized, highlighting the main research content of this thesis including research objectives, research contributions, and research benefits.

Chapter 2 – Literature review and research gaps:

This Chapter describes the previous research results of Resampling, SpreadSubsampling and SMOTE rebalancing methods; six classification algorithms such as SVM, Naïve Bayes, KNN, Bayesian Network, Random Forest, and J48 (Decision tree). In addition, issues and existing gaps are explained.

Chapter 3 - Research Methodology:

Discusses the design of the research method and procedures of implementing this research. In addition, an explanation of the theoretical knowledge, such as knowledge and function equation of the six classification algorithms involved in this thesis. Weka machine learning tool and the four chronic disease data sets have been introduced. The theoretical knowledge points of the data mining processing and ROC curve are described in detail.

Chapter 4 – Findings:

The results of the experiment are compared.

Chapter 5 - Validation:

Validates the result of the classifiers to verify the performance of the classifiers in the testing data sets.

Chapter 6 – Conclusion:

Summary and outlook. This thesis makes a summary analysis; summarizes the core content of the implementation work, explains the contribution, limitations and deficiencies, and puts forward suggestions for future research.

Chapter 2

Literature review and research gaps

Introduction

In this chapter, previous work on rebalancing methods; oversampling, undersampling, and resampling will be reviewed. In addition, the six classifiers, and issues and existing gaps will be reviewed and discussed.

Section 2.1: Guideline of the lecture review in chronic disease risk prediction.

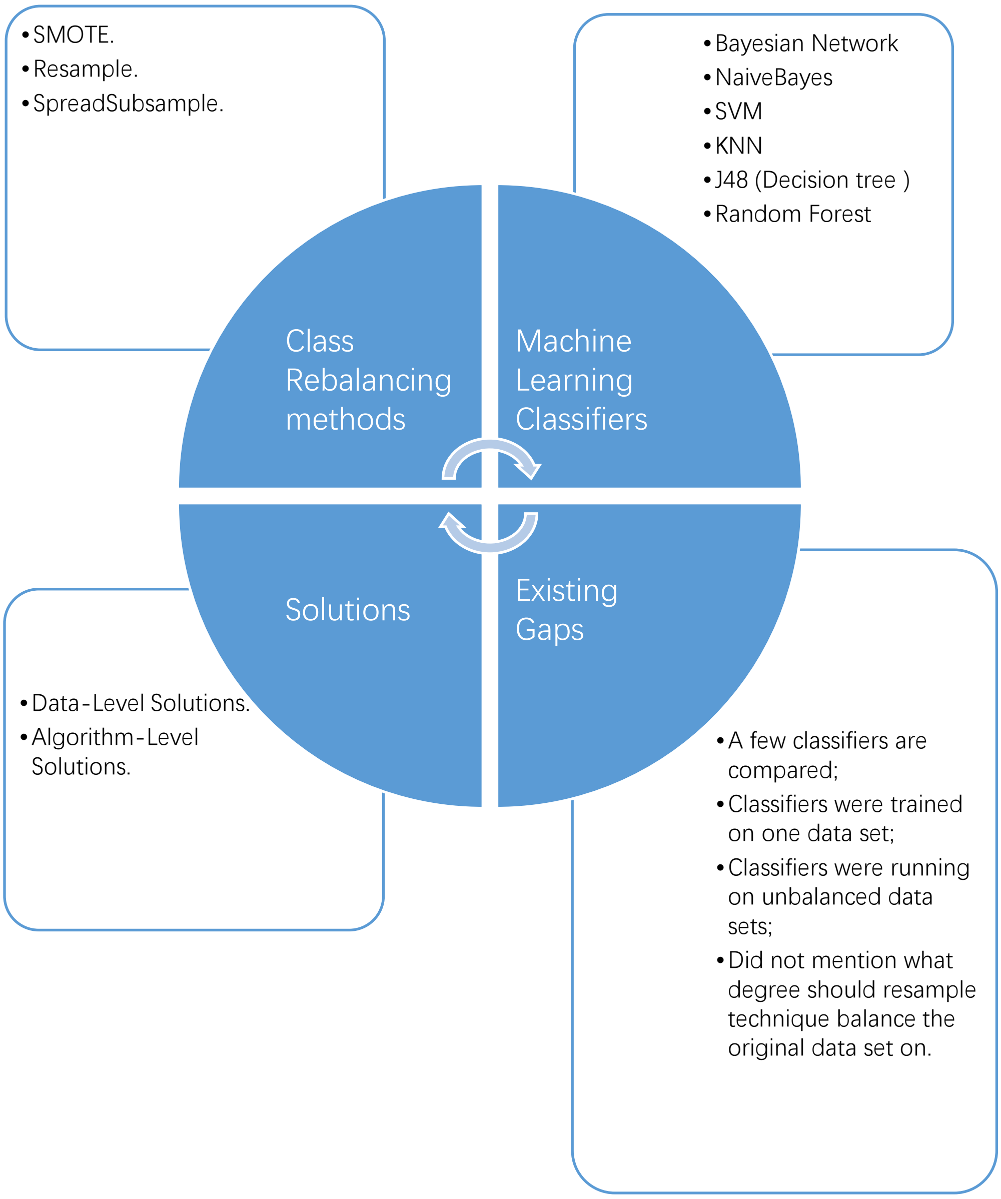
Section 2.2: Reviews three class rebalancing methods: SMOTE, Resampling, and SpreadSubsampleling.

Section 2.3: Reviews the six classifiers: Bayesian Network, NaiveBayes, SVM, KNN, Decision tree, and Random Forest.

Section 2.4: Presents issues and existing gaps in chronic disease risk prediction research.

2.1 Guideline of the lecture review in chronic disease risk prediction

Fig. 1.2 Guideline of the lecture review.



Class Rebalancing methods
Machine Learning Classifiers
Solutions
Existing Gaps
•SMOTE.
•Resample.
•SpreadSubsample.
•Bayesian Network
•NaiveBayes
•SVM
•KNN
•J48 (Decision tree)
•Random Forest
•Data-Level Solutions.
•Algorithm-Level Solutions.
•A few classifiers are compared;
•Classifiers were trained on one data set;
•Classifiers were running on unbalanced data sets;
•Did not mention what degree should resample technique balance the original data set on.

2.2 Class rebalancing methods research

2.2.1 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE (synthetic minority oversampling technique) is an improved scheme based on a random oversampling algorithm. Because random oversampling adopts the strategy of simply copying samples to increase minority samples, it is easy to produce the problem of model overfitting. Smote algorithm is used to analyze the minority samples and add new samples to the data set (Chawla et al., 2002).

Mirza, et al., (2018): The main motivation of this paper was to use SMOTE oversampling technique to develop a Decision tree classifier. In the first stage, SMOTE was used to rebalanced the data set, and in the second stage, a decision tree classifier was used to diagnose diabetes on the balanced data set. This method improves the classifier accuracy of the decision tree to 94.7013%. Research shows that SMOTE oversampling technique can effectively improve the prediction rate of the classifier

Abdoh, et al., (2018) presented the SMOTE technique and Random Forest classifier for early risk prediction of cervical cancer. The main reason is that the data for cervical cancer is unbalanced; the number of patients is far less than that of non-patients, and the cure rate can be improved by determining the risk factors of cervical cancer. When SMOTE technique was used in the Random Forest classifier, the performance of the classifier was improved by 1.7% to 3.5%.

Pandey, and Janghel (2019) claim through the analysis of electrocardiogram (ECG) signal and cardiovascular disease risk prediction to reduce the mortality of patients with heart disease using SMOTE technique to deal with minority groups imbalance phenomenon. The model of risk prediction is established with the classifier of Continuous Neural Network (CNN). The results show that CNN achieves the best performance of 98.3% on the rebalanced data set.

Apostolopoulos, (2020) used SMOTE oversampling technique to rebalance coronary artery disease data and generate a new data set. The main reason was that the imbalanced data set makes the classifier unable to analyze the real relationship. Artificial Neural Network, Decision Tee, and k-nearest neighbors (KNN) classifiers were used to classify the balanced data set. Results show that SMOTE can improve the classifier's ability to data-mining information.

Zheng, (2020) focuses on the prediction of heart disease risk by presented sampling techniques. This paper has compared the classical SMOTE and the combination of SMOTE and various classifiers. They are Borderline - SMOTE, SVM-SMOTE, and Kmeans - SMOTE. The results show that SVM-SMOTE and Borderline-SMOTE have the best performance.

The purpose of the research by Khadija,& Setiawan,(2020) is to predict the prevalence of early liver disease with high accuracy in order to improve the survival rate of patients. Four classifiers: Naive Bayes, KNN, Random Forest, and SVM, were used for comparison. The SMOTE oversampling technique was used to preprocess liver disease data, and then InfoGain feature selection was performed. The accuracy of Random Forest was 77.6%.

2.2.2 Resampling

Yildirim, (2017): The purpose of this study was to solve the imbalance effect in the training set of chronic kidney disease (CKD). SMOTE, SpreadSubsample, and Resample methods were used for comparison. The result found that for accuracy, the resampling method was a better rebalancing method than the others. It is emphasized that the accuracy can be improved by adjusting the parameters.

Mohapatra, and Mohanty, (2018) used feature selection (CFS) and resampling technique to analyze heart disease data. The research was divided into two categories; before sampling and after sampling. The accuracy of Random Forest classifier achieved 96%.

Qaisar, and Subasi, (2018): In this paper, the Electrocardiogram (ECG) signal collected by the event-driven A/D converter (EDADC) was uniformly resampled. The noise signal was extracted by the autoregressive (AR) method and classified by Support Vector Machine, k-nearest value, and Artificial Intelligence Network classifiers. The results show that the best classification (Support Vector Machine) accuracy was 93.73%.

Dharmarajan, (2020) used the rebalancing technology to rebalance the early prediction data set of chronic kidney disease. Logistic Regression, Random Forest, NaiveBayes, Decision Tree, Support Vector Machine were used for modeling. The classification methods were compared and the optimal classification algorithm was selected.

Rao, Makkithaya,(2017) carried out two experiments using Weka machine learning tool, The first experiment was to compare the performance of different classifiers, which were Bayesian, Bayesian Network, decision tree, and KNN, to find the best classifier for health data set. The second experiment compared the different rebalance methods oversampling, undersampling, and SMOTE. Experimental results show that the Bayesian classifier performed best on imbalanced data sets. In the Decision tree constructed by using undersampling data sets, the accuracy of KNN classifier was significantly improved.

Kumari, et al., (2017): In this paper, the prediction model of anti-mycobacterial ChEMBL database was constructed by Random Forest, Decision Tree, and KNN classifiers. Data rebalancing was performed first; SMOTE, SpreadSubsample, and Resample rebalancing methods were used to filter the classes. The results show that Random Forest was the best model and LibSVM had the highest sensitivity when compared with other classifiers. The original author suggested that prediction models are very useful for the analysis of drug candidates for tuberculosis and other diseases.

2.2.3 SpreadSubsampling

Pooja, (2013) compares the supervised and unsupervised balancing techniques. The reason is that with the increasing amount of data the challenge to data mining algorithms, so this research task was to analyze the supervised instance filters including Resample, SpreadSubsample, StructuredRemoveFolds and unsupervised instance filters including RemoveWithValues, ReserverSample and RemovePercentage. The experimental results showed that the resampling filter was the best in terms of accuracy and minimum mean absolute difference.

Drajiti, (2016) compared the undersampling of SpreadSubSample with the oversampling of SMOTE to solve the issue of diabetes data imbalance. The NaiveBayes classifier was used to classify the data. The results show that the over sampling method is more accurate in processing the training set data, while the under sampling method has a lower recall value in the same process.

Junior, et al., (2018) used three traditional machine learning algorithms, Naive Bayes, KNN and Artificial Neural Network, to classify the causes of deaths by lung cancer, and the first 100 features were extracted by the feature selection method. At the same time, undersampling and oversampling data sets were used for training and testing, and compared with individual data sets for verification. Experimental results show that Naive Bayes was the best classifier.

Krishnani, et al., (2019) used three classifiers: Random Forest, Decision Tree, and KNN to predict coronary heart disease then three sampling algorithms were used to analyze the data. The results show that resampling provided the highest accuracy. The spreadsubsample algorithm has the least execution time.

Research by Mishra.et al., (2020) found that with the increasing volume of disease data, effective analysis and processing of data is becoming more difficult. The main reason is the uneven data between disease categories. Therefore, they used three sampling techniques: SpreadSubSampling, Resampling and SMOTE to reduce the class inequality. The result showed that although there was no uniform balance specification, the sampling method for class preprocessing was the ideal choice (Mishra, Mallick,Jena, and Chae,2020).

Rajendran,et al., (2020) used three different kinds of balance techniques, SMOTE, SpreadSubsample and a mixed method of (SMOTE and SpreadSubsample) to predict the incidence rate of breast cancer through four classifiers. Because the unbalanced data are usually biased to most categories, the prediction results are not accurate. Also, a few disease categories are often the most important in breast cancer research. Therefore, the research results show that the hybrid method (smote and spread subsample) is the best way to solve the data imbalance.

2.3 Classifiers research

2.3.1 Bayesian Network

In 1763, Thomas Bayes published a paper called "An essay towards solving a problem in the doctrine of chances". The publication did not have much influence at that time, but during the 20th century it gradually became valued by people. It gradually became apparent that the Bayesian method was not only in line with the way of thinking in people's daily lives, but also in line with the law of people's understanding of nature. After continuous development, it eventually occupied half of the field of statistics, competing with classical statistics. It is a graphic mode to describe the dependency relationship between data variables, and a model for reasoning. Bayesian network provides a convenient framework to express causality, which makes uncertainty reasoning more clear and understandable in logic (Bayes, 1958).

Bayesian Network expresses the conditional independence relationship between each node, and can intuitively get the conditional independence and dependence relationship between attributes from Bayesian Network; in addition, it can be considered that Bayesian network uses another form to express the joint probability distribution of events; according to the network structure and conditional probability table of Bayesian network, each node can be quickly obtained of a basic event (Friedman, Geiger, and Goldszmidt,1997).

2.3.2 Naïve Bayes

Naive Bayes is a classification method based on the Bayesian theorem and independent assumption of feature conditions. Bayesian justification was developed by British mathematician Thomas Bayes (1702-1761) to describe the relationship between two conditional probabilities. It calculates the probability of classification through features and selects cases with high probability for classification. Therefore, it is a machine learning classification method based on probability theory. Because the goal of classification is determined, so it belongs to supervised learning (Rish, 2001).

2.3.3 Support Vector Machines (SVM)

SVM was first proposed by Vladimir n. Vapnik and Alexey ya. Chervonenkis in 1963. The current version of soft margin was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995. SVM is considered to be the most successful and best performing algorithm in machine learning in recent decades. It is a two classification model, which maps the feature vector of the instance to some points in space. The purpose of SVM is to draw a line to distinguish the two types of points, so that if there are new points in the future, the line can also make a good classification. SVM is suitable for small and medium data samples and nonlinear, high-dimensional classification problems (Qu and You, 2012). The LibSVM method is used in this thesis, which is a set of support vector machine libraries developed by Professor Chih-Chung Chang, and Chih Jen Lin of Taiwan in 2001. This set of libraries is fast, and can be used to classify or regress data conveniently (Chang, and Lin, 2011).

2.3.4 K-nearest neighbors (KNN)

K-nearest neighbor algorithm was first proposed by Cover and Hart in 1968. It is a mature method in theory and one of the simplest machine learning algorithms. The idea of this method is very simple and intuitive: In order to determine the class of unknown samples, the distance between the unknown samples and all known samples is calculated by taking all known samples as references, and K known samples which are closest to the unknown samples are selected from them. According to the majority voting rule, the unknown samples are compared with the class of k nearest samples Most of them belong to the same category (Deng, Zhu, Cheng, and Zhang, 2016).

Distance measurement, K value selection and classification decision rules are the three basic elements of the k-nearest neighbor method. According to the selected distance measure (such as Manhattan distance or Euclidean distance), the distance between the test case and each instance point in the training set can be calculated. K nearest neighbor points can be selected according to the K value, and the test cases can be classified according to the classification decision rules.

2.3.5 Decision Tree

Decision Tree algorithm originated from the paper "Experiments in induction" published by E.B.Hunt in 1966, but it is J.R.Quinlan who made Decision Tree the mainstream algorithm of machine learning. Decision Tree is a prediction model that represents a mapping relationship between object attributes and object values. In the tree each node represents an object, each branch path represents a possible attribute value, and each leaf node corresponds to the value of the object represented by the path from the root node to the leaf node. Decision Tree is a frequently used technology in data mining and can be used to analyze data, and also for prediction. Compared with other machine learning classification algorithms, Decision Tree classification algorithm is relatively simple. As long as the training sample set can be represented by feature vectors and categories, Decision Tree classification algorithms can be constructed. The complexity of the predictive classification algorithm is only related to the number of layers of the Decision Tree, and it is linear. The efficiency of data processing is very high, so it is suitable for real-time classification (Quinlan, 1986).

2.3.6 Random Forest

Random Forest is a machine learning algorithm mentioned by Breiman Leo and Adele Cutler of the University of California, Berkeley in 2001. It can be used for classification, clustering and regression. Here is a brief introduction to the application of the algorithm in classification. Random Forest algorithm is used to train multiple Decision Trees, generate models, and then use the classification results of multiple Decision Trees to vote, so as to achieve classification. The Random Forest algorithm only needs two parameters: the number of Decision Trees to be constructed T, and the number of input features to be considered when each node of the Decision Tree is split M (Biau, and Scornet, 2016)

2.4 Issues and Existing Gaps in Chronic disease risk prediction research

Based on the literature review, we found the following gaps:

1. Studies use limited classifiers. (see section 2.4.5)
2. Studies' classifiers are performed on imbalanced data sets. (see section 2.4.5)
3. Studies did not mention what degree should one balance the original data set. (see section 2.2)
4. Classifiers are trained and tested on limited data sets. (see section 2.4.5)

2.4.1 Limited Classifiers

Each classifier has advantages and limitations, so the performance results are different. Therefore, to ascertain which classifier is most suitable for chronic disease prediction is the purpose of this paper. However, if only one or two classifications are verified, some possibly better classifications will be ignored. The solution of this paper is to compare various classifiers and find the most suitable to predict chronic diseases (Uddin, Khan, Hossain, & Moni, 2019).

Evidence 1 - 2.4.5 (Saravananathan, and Velmurugan, 2016).

In this article, the author only compared the classifier performance of J48 (Decision Tree), CART, SVM and KNN.

Evidence 2 - 2.2 (Mirza, Mittal, and Zaman, 2018).

In this article, the author only uses the Decision Tree classifier to analyze the data.

Evidence 3 - 2.4.5 (Rajendran, Jayabalan, and Thiruchelvam, 2020).

The author compared four classifiers: Naive Bayes, Bayesian Network, Random Forest and Decision Tree (C4.5).

Evidence 4 - 2.4.5 (Vembandasamy, Sasipriya, and Deepa, 2015).

The author only uses Naive Bayes classifier to analyze the data set.

Evidence 5 - 2.2 (Abdoh, Rizka, and Maghraby, 2018).

In this article, the author only uses Random Forest classifier to analyze the data.

2.4.2 Imbalanced data sets

In medical data sets, imbalanced data is quite common, mainly reflected in the classes, such as a small number of disease people and the majority of non- disease people. As a result, in risk prediction, the classifier will analyze the majority of non-disease people and ignore the disease people, but the purpose of the prediction is to analyze these disease people. Therefore, unbalanced data can lead to inaccurate classifier accuracy. The solution of this thesis is to rebalance the data sets through three kinds of sampling methods, so as to find the optimal method (Shuja, Mittal,& Zaman,2020).

Evidence 1 - 2.4.5 (Saravananathan, and Velmurugan, 2016).

Classifiers are used to analyze data under an unbalanced data set.

Evidence 2 - 2.4.5 (Abdar, Kalhori, Sutikno, Subroto, and Arji, 2015).

This article did not use data rebalance method for data balancing, and directly trains the classifier on it.

Evidence 3 - 2.4.5 (Vembandasamy, Sasipriya, and Deepa, 2015).

This article does not rebalance the data set, but uses the collected data set of 500 patients

2.4.3 Parameters adjustment detail are not specified

In the literature review, we can see that different sample methods are used to rebalance the data set, and the performance of the classifiers are greatly improved by using rebalance methods. However, the problem is to what degree should one balance the original data set? What parameters can be adjusted to improve the classifier? Therefore, this thesis releases details of parameter adjustment, which will be of great help to disease researchers and medical institutions (Krawczyk, 2016).

Evidence 1 - 2.2 (Mirza, Mittal, and Zaman, 2018).

The parameter adjustment index does not tell us that the performance of the classifier is significantly improved after using the rebalanced data set.

Evidence 2 - 2.2 (Abdoh, Rizka, and Maghraby, 2018).

The rebalancing techniques are compared and verified by classifier, but the parameters are not described in detail.

Evidence 3 - 2.4.5 (Rajendran, Jayabalan, and Thiruchelvam, 2020).

No rebalancing parameters were given.

2.4.4 Limited data sets

A classifier does not necessarily adapt to all data sets, and the performance results on each data set are different, Because the number of attributes and the size of the data set will also affect the results, this thesis will use a variety of data sets to train and test the classifier. The aim is to find the best classifier for chronic diseases (Özdemir, Yavuz, & Dael, 2019).

Evidence 1 - 2.4.5 (Saravananathan, and Velmurugan, 2016).

The experiment used data set of 545 patients for analysis

Evidence 2 - 2.2 (Mirza, Mittal, and Zaman, 2018).

The data set: 734 patients and 11 features

Evidence 3 - 2.4.5 (Abdar, Kalhori, Sutikno, Subroto, and Arji, 2015).

Data set with 270 instances and 13 features

Evidence 4 - 2.4.5 (Vembandasamy, Sasipriya, and Deepa, 2015).

Data set with 500 instances and 11 features

Evidence 5 - 2.4.5 (Abdoh, Rizka, and Maghraby, 2018).

Data set used with 858 instances and 32 features

2.4.5 Existing Gaps

Table 1.2 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Analyzing diabetic data using classification algorithms in data mining. Indian Journal of Science and Technology, 9(43), 1-6.	Saravananathan, K., & Velmurugan, T. (2016).	The relationship between classifier (accuracy and ROC) performance is analyzed. Decision Tree (J48) classifier was the best classifier with accuracy of 67.15%. This paper uses statistical measure to avoid over fitting issue.	1) Only four (J48,CART,SVM,KNN)classifiers are compared; 2) The experiment used data set of 545 patients for analysis; 3) Classifiers are performed on imbalanced data sets.

Table 1.3 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Comparing Performance of Data Mining Algorithms in Prediction Heart Diseases. International Journal of Electrical & Computer Engineering (2088-8708), 5(6).	Abdar, M., Kalhori, S. R. N., Sutikno, T., Subroto, I. M. I., & Arji, G. (2015).	Compared four classifiers and provided a good overall view and analysis. The results show that the best performance of C5.0 is 93.02%, followed by KNN, SVM and Neural Network In the experiment, feature selection conducted to reduce the running time and improve the accuracy of the classifiers.	1) Data set with 270 instances and 13 features; 2) The classifier was running on unbalanced data sets; 3) Classifiers are performed on imbalanced data sets.

Table 1.4 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Decision Support Predictive model for prognosis of diabetes using SMOTE and Decision Tree. International Journal of Applied Engineering Research, 13(11), 9277-9282.	Mirza, S., Mittal, S., & Zaman, M. (2018).	The SMOTE over-sampling method shows good results. Decision Tree classifier performance improved by 2%.	1) Only used Decision Tree classifier; 2) ROC curve analysis is not performed; 3) The Decision Tree classifier only tested on one data set; 4) The data set: 734 patients and 11 features; 5) Did not mention to what degree should SMOTE balance the original data set.

Table 1.5 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Predicting Breast Cancer via Supervised Machine Learning Methods on Class Imbalanced Data.	Rajendran, K., Jayabalan, M., & Thiruchelvam, V. (2020)	The analysis of several oversampling and undersampling methods for class imbalance problem shows that SMOTE and SpreadSubsample hybrid balancing method show good results. Among the four classifiers, Bayesian Network and Naive Bayes have the best performance, reaching 99.1%, Decision Tree 98.4%, and Random Forest 94.8%.	1) Only four classifiers are compared; 2) There is no describing the parameter of the sampling methods; 3) Did not mentioned what degree should undersampling methods balance the original data set to.

Table 1.6 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Heart diseases detection using Naive Bayes algorithm. International Journal of Innovative Science, Engineering & Technology, 2(9), 441-444.	Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015).	This paper presents the application of machine learning in heart disease detection. The Naive Bayes model can reach 86.4198%	1) Data set with 500 instances and 11 features; 2) Only used Naive Bayes classifier is tested; 3) Classifiers were performed on imbalanced data set.

Table 1.7 Gaps Identified

Gaps Identified			
Study	**Author(s) and Date**	**Relevant Findings**	**Gaps in the literature**
Cervical cancer diagnosis using Random Forest classifier with SMOTE and feature reduction techniques. IEEE Access, 6, 59475-59485.	Abdoh, S. F., Rizka, M. A., & Maghraby, F. A. (2018).	SMOTE technique can improve the performance of the Random Forest classifier by 1.7% to 3.5%.	1) Only Random Forest classifier is tested; 2) ROC curve analysis is not performed; 3) Data set used with 858 instances and 32 features; 4) Did not mention to what degree should SMOTE balance the original data set.

2.5 Summary

In summary, the results show that the three sampling methods SMOTE, Resampling, and SpreadSubsampleling have great influence on disease data. In addition, the differences and history of Bayesian Network, Naive Bayes, SVM, KNN, J48 (Decision Tree), and Random Forest are summarized. The rebalancing problems and existing gaps are reviewed and analyzed. Research gap will be the focus of this thesis.

Chapter 3

Research Methodology

This chapter provides the framework of the methods and techniques used in this research. The full text is divided into four parts and summarized as follows: Section 3.1 is the research design. We discuss how the CRISP-DM data mining process is implemented in this research. 3.2 For the background of chronic diseases, the data set has an explanation and analysis. 3.3 In-depth understanding of the methods and tool for data mining. 3.4 Conduct detailed research on data modelling and algorithms. 3.5 We explain the verification part. We verify the high-quality model through the performance ranking of the classifier and the ROC curve. 3.6 We compare the accuracy of excellent classifiers and find the best model for feature risk prediction, thus completing the chapter.

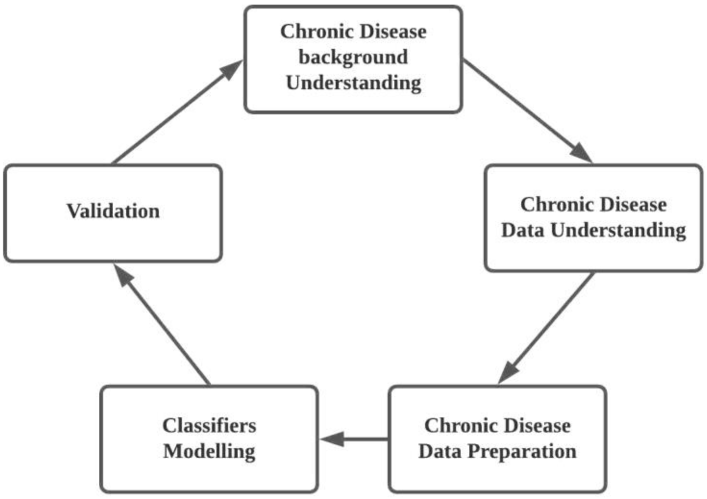
3.1 Research Design

The cross-industry standard process for data mining model is called CRISP-DM for short, which provides a structured approach to planning a data mining project. The model is particularly concerned with the integrity of machine learning work. Not only related to the data control, display processing and operations; it also covers how to effectively address and respond to enterprise requirements (Fayyad et. al. 1996).

There are many methodologies to tackle data mining opportunities, such as CRISP-DM, Knowledge Discovery in Database (KDD), and Sample, Explore, Modify, Model, and Assess (SEMMA). These are designed to improve the success of data mining projects and these methodologies are used in many sectors such as medical, or health care industries.

This research conducted a CRISP-DM data mining process to experiment and analyze chronic disease data. The CRISP-DM process combines Selection-Preprocessing (KDD) or Sample-Explore (SEMMA) stages into the Data Understanding stage. It also incorporates Business Understanding and Deployment stages. An important difference between CRISP-DM and two other methodologies is that transitions between stages in CRISP-DM can be reversed. Especially helps when analysts work with medical data that any misstep can be fixed without having to finish the whole cycle and the process reminds analysts to put the business projects in the core position, which has more advantages than KDD and SEMMA Data methodologies. The purpose of this research is to find the best classifier for chronic diseases and solve the imbalance issue in medical data through the CRISP-DM method. Therefore, we selected four data sets related to chronic diseases and rebalanced them, and then trained the six classifiers to find the best classifier. Finally, we used the best classifier for comparison with different articles. The following sections provide more details.

Fig. 1.3 Chronic Disease Experiment Process:


Chronic Disease background Understanding
Chronic Disease Data Understanding
Chronic Disease Data Preparation
Classifiers Modelling
Validation

3.2 Chronic Disease background understanding

Effective prediction and timely action in the treatment of chronic diseases is the primary task of today's society. The crucial step is how to predict more accurate information to be transmitted to the front-line health professionals and the general public (WHO, 2020). Kidney disease is among the chronic diseases able to be predicted, and this disease can be predicted through the rebalanced algorithm and Decision Tree model. The model has a high accuracy rate of 98.73% (Potharaju, and Sreedevi 2016). They also established a diabetes prediction model through a classifier, which effectively solved the problem of diabetes prediction and rebalancing of medical data (Shuja, Mittal, and Zaman 2020). Therefore, this research follows the lines drawn by these works. The purpose of this research is to find the optimal classifier and solve the imbalance issues in medical data. We used six different classifiers and compared their performance on different data sets. We describe this work from the data phase. This includes data collection, mining process, and modelling, validation and performance comparison. More details are provided in the following sections.

3.3 Chronic Disease Data Understanding

The focus of this stage is to extract information from the original data and understand the background and content of the collected data set in detail, including actions to be taken in the next phase.

3.3.1 Data sets collection

The selection of chronic disease data is very important for this research. Our data collection scope: The data collection on chronic diseases needs to come from different resources. The number and type of attributes may vary in data sets, but they must be related to chronic diseases and medical issues. Furthermore, because we cannot go directly to the hospital to collect the data, we have chosen the public data set for our research. The common data sets used in this research are UCI heart disease data set, Framingham Heart Study (FHS) data set, Acute Liver Failure data set and Surgery Timing data set. We understand that the Surgery Timing data set is not a case of chronic disease, but we consider that the complications of surgery and the time of surgery are also of great significance for the treatment of chronic diseases, so we chose this data set for the experiment.

The details of the data sets are as follows:

3.3.1.1 UCI heart disease data set (Manu, 2019)

The UCI heart disease data set is consolidated by combining three different heart disease data sets. The combination of more than 11 common features makes it the largest UCI heart disease data set available for research. The three data sets are the Cleveland data set (composed of US patient data), Stalog data set (UK patient data), and Hungary data set (mainly Swiss and Hungarian patient data). This data set is a combination of data sets collected from patients from three different places, because we can learn about different patients and ethnicities with heart disease in Europe and the United States through this data set.

Table 1.8 Summary of UCI data set

Age Group	Men	Women	Total
28-39	69	23	92
40-49	207	72	279
50-59	383	99	482
60-69	220	76	296
70-77	30	11	41
Total	909	281	1190

The data set comes with 12 attributes and 1190 instances of data.

Exploratory descriptive analysis of the UCI Heart Disease Data set Data Set

These 12 attributes are introduced as follows:

Fig 1.4 UCI data set exploration

age	Age in years ;
sex	1= male, 0 = female;
chest pain type	type of chest pain categorized into 1 typical, 2 typical angina, 3 non-angina pain, 4 asymptomatic;
resting bp s	Level of blood pressure at resting mode in mm/HG (Numerical)
cholesterol Serum	cholesterol in mg/dl (Numeric);
fasting blood sugar	Blood sugar levels on fasting > 120 mg/dl represents as 1 in case of true and 0 as false;
resting ecg	result of electrocardiogram while at rest are represented in 3 distinct values 0 : Normal 1: Abnormality in;
max heart rate	Maximum heart rate achieved (Numeric);
exercise angina	Angina induced by exercise 0 depicting NO 1 depicting Yes (Nominal);
oldpeak	Exercise induced ST depression in comparison with the state of rest (Numeric);
ST slope	ST segment measured in terms of slope during peak exercise 0: Normal 1: Upsloping 2: Flat 3: Downsloping;
target	Heart Risk 1 means heart disease 0 means normal;

3.3.1.2 Framingham Heart Study (FHS) data set (Aman, 2017)

The Framingham heart study is seen as a model of medical research. After more than 10 years of research, the influence of hypertension and hypercholesterolemia on coronary heart disease has been confirmed, and the importance of risk factor verification has been proposed. It has been proved that the prevention and intervention of hypertension, diabetes and other risk factors have effectively reduced the incidence and mortality of cardiovascular diseases and saved countless patients' lives.

Framingham research is very important for creating living research objects and selecting and tracking participation. We chose this data set for the purpose of its etiological research and long-term data collection. This feature of the FHS data set is very beneficial for us in conducting this research.

Table 1.9 Summary of FHS data set

Age Group	Men	Women	Total	TenYearCHD
32-39	253	303	556	23
40-49	721	940	1661	167
50-59	526	771	1333	263
60-70	284	406	690	191
Total	1784	2420	4240	644

The data set come with 16 attributes and 4240 instances of data.

Exploratory descriptive analysis of the Framingham Heart Study Data Set

These 16 attributes are introduced as follows：

Fig 1.5 Framingham Heart Study data set exploration

sex	1= male, 0 = female;
age	Age in years ;
education	1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = College;
currentSmoker	0 = non smoker; 1 = smoker
cigsPerDay	number of cigarettes smoked per day (estimated average)
BPMeds	0 = not on blood pressure medications; 1 = is on blood pressure medications
prevalentStroke	Prevalent stroke;
prevalentHyp	Prevalent hypertension;
diabetes	(glucose > 200 mg/dL or on treatment) 0 = No; 1 = yes;
totChol in mg/dL:	totChol in mg/dL;
sysBP	Systolic blood pressure, mmHg;
diaBP	Diastolic blood pressure, mmHg;
Body Mass Index (BMI)	calculated as: Weight (kg)/ Height (meter squared);
heartrate Beats/min (Ventricular)	heartrate Beats/min (Ventricular);
glucose serum glucose in mg/Dl	glucose serum glucose in mg/dL;
TenYearCHD	those that did or did not develop CHD (Coronary Heart Disease) during the study period);

3.3.1.3 Acute Liver Failure (ALF) data set (Rahul, 2018)

The JPAC Center for Health Diagnostics and Control has conducted a national survey of Indian adults since 1990. Various demographic and health information was collected through direct interviews, examinations, and blood samples. The data set consists of expected information from 8,785 adults aged 20 or older from surveys conducted in 2008-2009 and 2014-2015. We chose this data set because the survey involved a large number of people, and the age group from 20-85 is involved.

Table 2.1 Summary of Acute Liver Failure data set

Age Group	Male	Female	Total	ALF
20-29	960	673	1633	2
30-39	857	671	1528	7
40-49	752	757	1509	14
50-59	569	572	1141	34
60-69	667	668	1335	87
70-79	476	511	987	143
80-85	349	303	652	177
Total	4630	4155	8785	464

The data set come with 30 attributes and 8785 instances of data.

Exploratory descriptive analysis of the Liver Failure data set

These 30 attributes are introduced as follows：

Fig 1.6 Acute Liver Failure data set exploration

Age	Age in years ;
Gender	M= male, F = female;
Region	east; south
Weght	25.6 – 193 kg
Height	130 - 200 cm
Body Mass Index	kg m/h2
Obesity	1 = obesity, 0 = non-obesity
Waist between	58.5 – 173 cm
Maximum Blood Pressure	72 – 233 mmHg
Minimum Blood Pressure	10-132 mmHg
Good Cholesterol	8 – 160 mg/dL
Bad Cholesterol	27 – 684 mg/dL
Total Cholesterol	72 – 727 mg/dL
Dyslipidemia	0 = No, 1 = Yes
PVD	Peripheral vascular disease 0 = No, 1 = Yes
Physical Activity	1, 2 3 4
Education	0 = non – educated, 1 = educated
Unmarried	0 = No, 1 = Yes
Income	0 = No, 1 = Yes
Source of Care Private	Hospital 58%, clinic 21% Other 21%
PoorVision	0 = No, 1 = Yes
Alcohol Consumption	0 = No, 1 = Yes
HyperTension	0 = No, 1 = Yes
Family HyperTension	0 and 1
Diabetes	0 = No, 1 = Yes
Family Diabetes	0 = No, 1 = Yes 1
Hepatitis	0 = No, 1 = Yes
Family Hepatitis	0 = No, 1 = Yes
Chronic Fatigue	0 = No, 1 = Yes
ALF	Acute Liver Failure 0 = No, 1 = Yes

3.3.1.4 Surgery Timing data set (Mahesh, 2018)

This data set contains 14,635 instances. Age, sex, race, BMI, several comorbidities, timing of surgery predictors (hour, week, month, month), several surgical risk indicators, and outcomes (30-day mortality and hospitalization complications) were provided. This data set does not belong to chronic diseases, but we consider the time of operation and whether there are complications of chronic diseases are critical to the success of the operation.

Table. 2.2 Summary of Surgery Timing data set

Age Group	Male	Female	Total
6-19	11	14	25
20-29	62	52	114
30-39	557	1114	1671
40-49	515	993	1508
50-59	1832	2087	3919
60-69	739	902	1641
70-79	1264	1282	2546
80-90	3053	158	3211
Total	8033	6602	14635

The data set come with 25 attributes and 14630 instances of data.

Exploratory descriptive analysis of the Surgery Timing Data set

These 25 attributes are introduced as follows：

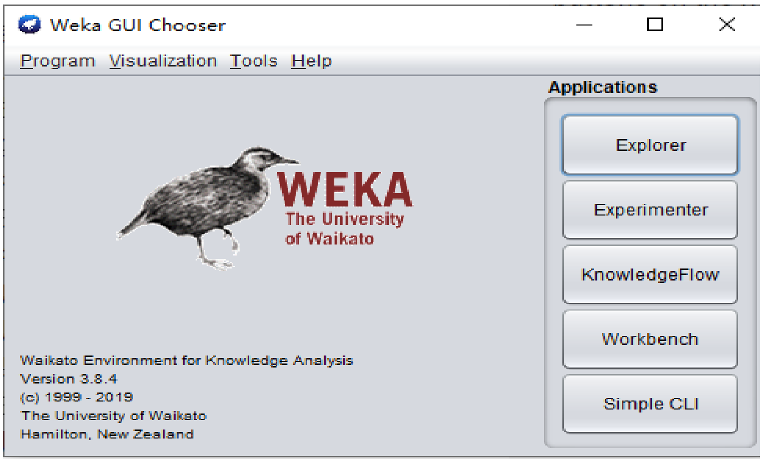
Fig 1.7 Surgery Timing data set exploration

age	Age in years ;
gender Gender	1 = male; 2 = female;
bmi	Body Mass Index kg/m2;
asa_status	American Society of Anesthesiologist Physical Status 1 = I – II 2 = III 3 = IV - VI;
baseline_cancer	Cancer 0 = No; 1 = Yes;
baseline_charlson	Charlson Comorbidity Index;
baseline_cvd	Cardiovascular/Cerebrovascular Disease 0 = No; 1 = Yes;
baseline_dementia	Dementia 0 = No; 1 = Yes;
baseline_diabetes	Diabetes 0 = No; 1 = Yes;
baseline_digestive	Digestive Disease 0 = No; 1 = Yes;
baseline_osteoart	Osteoarthritis 0 = No; 1 = Yes;
baseline_psych	Psychiatric Disorder 0 = No; 1 = Yes;
baseline_pulmonary	Pulmonary Disease 0 = No; 1 = Yes;
ahrq_ccs	United States Agency for Healthcare Research and Quality's Clinical Classifications Software (AHRQ- CCS) Procedure Category;
ccsComplicationRate	Overall incidence of In-Hospital Complications for Each AHRQ-CCS Procedure Category;
ccsMort30Rate	Overall Incidence of 30-day Mortality for Each AHRQ-CCS Procedure Category;
complication_rsi	Risk Stratification Index (In-Hospital Complications);
dow	Day of Week 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday;
hour	Operation Hour;
month	Month of Year 1 = January 2 = February 3 = March 4 = April 5 = May 6 = June 7 = July 8 = August 9 = September 10 = October 11 = November 12 = December;
moonphase	Phase of Moon 1 = new moon 2 = first quarter 3 = full moon 4 = last quarter;
mort30	30-Day Mortality 0 = No; 1 = Yes;
mortality_rsi	Risk Stratification Index (30-Day Mortality);
race	Race 1 = Caucasian 2 = African American 3 = Other;
Complication	In-Hospital Complication 0 = No; 1 = Yes.

3.3.2 Data Mining Tool (Weka)

Weka is an open-source machine learning tool, which was developed at the University of Waikato, New Zealand. The reason for choosing the Weka tool for this thesis is that it has the algorithm efficiency of mining control implementation with a huge number of tasks and can perform effective pre-control on the data, realize clustering processing and other operations, and operate more transparently on the interaction interface (Hall, Frank, Holmes, Pfahringer, Reutemann, & Witten, 2009).

Fig 1.8 Weka GUI Chooser

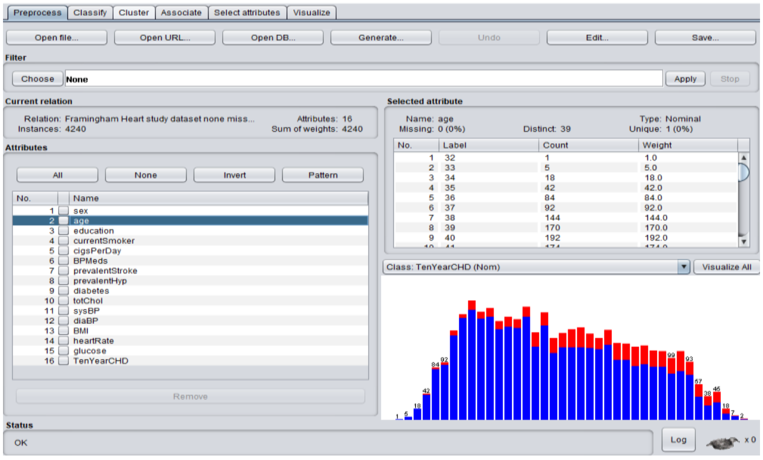


Waikato Environment for Knowledge Analysis
Version 3.8.4
(c) 1999 - 2019
The University of Waikato
Hamilton, New Zealand

The main interface of WEKA is the GUI chooser. It provides multiple main applications: Explorer, Experimenter, KnowledgeFlow, workbench and Simple CLI are the five buttons on the right as shown in the figure above.

The Explorer interface is mainly used in this research, as shown in Figure 1.8. The interface contains six tabs at the top: Preprocess, Classify, Cluster, Associate, Select Attributes, and Visualize. Different tabs have different functions, which basically include all data pre-processing functions and machine learning algorithm implementation. The focus of this thesis is to use the Preprocess, Classify, and Select Attributes tab, enter the Preprocess interface, click the Open File button, import the data set and Implement the relevant processing operations.

Fig. 1.9 Weka Working Dashboard



Filter: None

Current relation

Relation: Framingham Heart study dataset none miss... Attributes: 16
Instances: 4240 Sum of weights: 4240

Attributes

No.		Name
1	☐	sex
2	☐	age
3	☐	education
4	☐	currentSmoker
5	☐	cigsPerDay
6	☐	BPMeds
7	☐	prevalentStroke
8	☐	prevalentHyp
9	☐	diabetes
10	☐	totChol
11	☐	sysBP
12	☐	diaBP
13	☐	BMI
14	☐	heartRate
15	☐	glucose
16	☐	TenYearCHD

Selected attribute

Name: age Type: Nominal
Missing: 0 (0%) Distinct: 39 Unique: 1 (0%)

No.	Label	Count	Weight
1	32	1	1.0
2	33	5	5.0
3	34	18	18.0
4	35	42	42.0
5	36	84	84.0
6	37	92	92.0
7	38	144	144.0
8	39	170	170.0
9	40	192	192.0

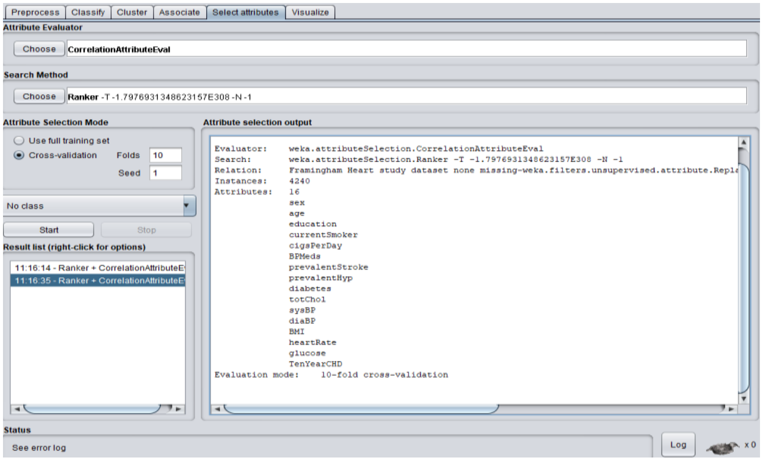
Class: TenYearCHD (Nom)

Status

OK

WEKA conducts use a form of the associated Select Attributes tab in Figure 1.9 to assist with effective attribute selection processing. To implement the attribute selection process, researchers have to set the search mode and the attribute evaluator. The option groups that exist at the top of the form are Attribute Evaluator and Search Method. The search method and the parameters of the evaluator can be set in the form of setting the information of the two option groups to get the best combination of attributes.

Fig. 2.1 Weka Select Attributes



Attribute Evaluator

CorrelationAttributeEval

Search Method

Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

○ Use full training set
● Cross-validation Folds 10
Seed 1

No class

Result list (right-click for options)

11:16:14 - Ranker + CorrelationAttributeE
11:16:35 - Ranker + CorrelationAttributeE

Attribute selection output

```
Evaluator:    weka.attributeSelection.CorrelationAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     Framingham Heart study dataset none missing-weka.filters.unsupervised.attribute.Repla
Instances:    4240
Attributes:   16
              sex
              age
              education
              currentSmoker
              cigsPerDay
              BPMeds
              prevalentStroke
              prevalentHyp
              diabetes
              totChol
              sysBP
              diaBP
              BMI
              heartRate
              glucose
              TenYearCHD
Evaluation mode:    10-fold cross-validation
```

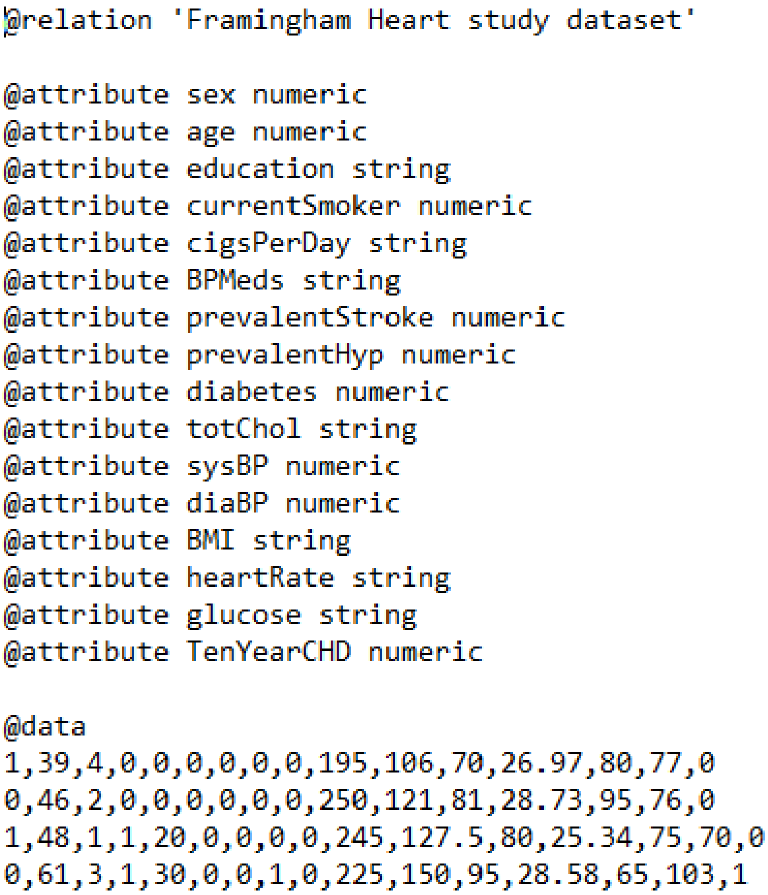
Status

See error log

ARFF file

There are many file types that can be supported by WEKA, and ARFF is a text file for ASCII processing. We will take a data set as an example (FHS data set). The generated ARFF file format is shown in the figure below:

Fig. 2.2 Data set in ARFF file



```
@relation 'Framingham Heart study dataset'

@attribute sex numeric
@attribute age numeric
@attribute education string
@attribute currentSmoker numeric
@attribute cigsPerDay string
@attribute BPMeds string
@attribute prevalentStroke numeric
@attribute prevalentHyp numeric
@attribute diabetes numeric
@attribute totChol string
@attribute sysBP numeric
@attribute diaBP numeric
@attribute BMI string
@attribute heartRate string
@attribute glucose string
@attribute TenYearCHD numeric

@data
1,39,4,0,0,0,0,0,0,195,106,70,26.97,80,77,0
0,46,2,0,0,0,0,0,0,250,121,81,28.73,95,76,0
1,48,1,1,20,0,0,0,0,245,127.5,80,25.34,75,70,0
0,61,3,1,30,0,0,1,0,225,150,95,28.58,65,103,1
```

Line 1 is the relation name, where the author uses the name of the data set.

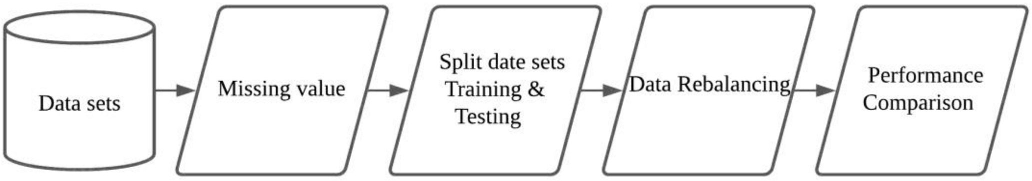
Lines 2-17 are the feature lists, where columns 2-14 are the feature descriptions and column 15 is the feature value range.

@data (line 18) is the description of the data field, and all the data below it is data. Each row contains one piece of data.

3.4 Chronic Disease Data Preparation

Through the below data exploration, the overall process of the mining process in this research is shown in the following Figure 2.3:

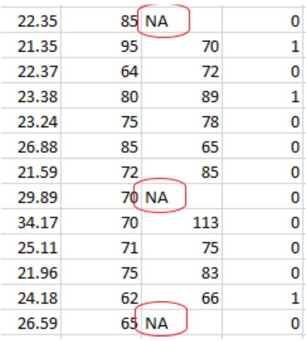
Fig. 2.3 Mining Process



3.4.1 Missing Value

By using Excel's “Data-Filter” function we can view the details of this data set by checking each attribute. As shown in Figure 2.4, the missing value is “NA”, which means "unknown". The reason is that Weka cannot recognize “NA” as a missing value; it needs to be replaced with blank in the CSV file before importing to Weka.

Fig. 2.4 Missing value with FHD data set



22.35	85	NA	0
21.35	95	70	1
22.37	64	72	0
23.38	80	89	1
23.24	75	78	0
26.88	85	65	0
21.59	72	85	0
29.89	70	NA	0
34.17	70	113	0
25.11	71	75	0
21.96	75	83	0
24.18	62	66	1
26.59	65	NA	0

There are multiple ways to deal with missing values in the data sets. For example, if there are few instances with missing values, we can delete them because they will not affect the overall forecast. However, there are not many missing values in the current four data sets so if we delete them directly, the experiment will be inaccurate. Therefore, according to Frank, Hall and Witten (2016), the strategy for dealing with missing values is as follows:

- Numeric attributes: In order to maintain the consistency of the data set; it is common practice to replace the missing numeric attributes. In this research, the strategy is to replace the average value of the existing value of the attribute with the numeric attribute of the missing value.
- Nominal property: Since this property contains binary and 0 or 1, the method here is to replace the missing value with the nominal value that occurs most frequently in this property.

Therefore, Weka-unsupervised – ReplaceMissingValues has been chosen. The missing value instance is replaced by average value.

3.4.2 Split the data sets into Training and Testing

In order to obtain a high-performance classifier based on machine learning, it is necessary to train and test the classifier. The data set uses a training classifier. However, after training, a test data set is needed for fitting test, which should be different from the training set. Therefore, the solution is to divide the data set into two groups before training the model, one for training and the other for testing.

According to O'Meara (2019), a typical measure is to split the data in a random way and use most of the data for training (eg. 70% / 30% ratio), setting the smaller part as a test to keep the training set similar to the test set. By retaining similar data for training and testing, we can minimize the impact of data differences and better understand the characteristics of the model.

Therefore, by choosing Weka – unsupervised – instance – Remove Percentage, in order to maintain a similar distribution and ratio between training and test data sets, this paper performs stratified random sampling on Weka. Set Remove Percentage to 70 percentage, and create a new training file named Training.arff, and create a test data set file Testing.arff. We will take one of the data sets as an example, the results show that the FHS data set are 2968 examples in the training data set after split up; including 2530 no Coronary HD, and 438 Coronary HD yes.

3.4.3 Data Rebalancing

Since the classifications are severely unbalanced, the currently selected training data set will be biased towards the majority class during classification. Next, a method of rebalancing was used to improve performance.

Compare oversampling and undersampling in rebalancing:

According to Li, Wang, and Yue (2005) the use of the hitting technique on oversampling (Synthetic Minority oversampling Technique) keeps the information unchanged, but the chance of a few classification errors increases.

Thus, to generate a subset of the data, one should use a supervision and SpreadSubsample WEKA resampling method, because it can be balanced with a minority proportion of the majority, and the total number of instances can be controlled. Even if we cannot create too few instances to capture, here we still try to use the hits technique to perform some processing on the oversampling to compare the results.

Resampling in supervised filters is better than resampling in unsupervised filters, because as described by Eibe,et al., (2016), it can maintain the distribution of classes in the subsample, and can be configured to bias the classes in a uniform direction. SpreadSubsample: random subsamples will be generated between the rarest and most common categories and control the frequency difference. In this report, the author uses two methods to obtain a balanced data set.

Resampling parameters: For resampling methods, there are some parameters to choose from. The parameter biasToUniformClass represents the ratio of the minority category to the majority category; randomSeed refers to the random number seed used for random sampling. Therefore, we can have multiple options to resample the data set (as shown in Table. 2.3).

SpreadSubsample parameters: Similarly, SpreadSubsample parameters also have some parameters. By setting the maximum number of instances maxCount new equilibrium data set, allows us to control the number of new data sets (Table 2.3).

SMOTE and random sampling tests to practice the balance of the training data set. SMOTE uses the nearest neighbor algorithm (Eibe,et al., 2016) to calculate new eigenvalues and create new instances for minority categories. Here, the default value of neighborNeighbors is 5, which is very beneficial to the calculation. By adjusting the parameters of the number of "percent", we can set the number of new few instances, for example, 200 “percent” means that the original few examples we will create two times (200% = 2) However, due to the limitations of computing resources, it is impossible to set a larger "percent" value. Therefore, we only use a percentage of 100 to 300 for training, which means that a few instances will be twice or three times the original data set.

Table.2.3 Rebalance methods and Parameters values

Methods	Parameters methods	
Resample	biasToUniformClass 0, 1, 0.8, 0.4	default randomSeed
SpreadSubsample	distributionSpread 1,3,5	default maxCount
SMOTE	percentage 100,200,300	default nearestNeighbors

3.4.4 Performance comparison

There are various indicators for measuring the performance of classifiers. The new metric formula created here is mainly used as a performance metric, and other functions (such as time, ROC, area under the curve (AUC), and number of features) are also very important (Shuja, Mittal, & Zaman, 2020).

A new metric formula: Suppose N (HD) is heart disease, and N (nHD) is non-heart disease. Here we can use the four parameters in the confusion indicator, namely N (HD-> HD) = TP: correct classification of heart disease; N (nHD-> nHD) = TN: correctly classified as non-heart disease; N (nHD-> HD) = FP: the number of non-heart diseases classified as heart disease; N (HD-> nHD) = FN: the number of heart diseases classified as non-heart disease. The accuracy, precision and recall rate are shown as three formulas.

Table 2.4 Confusion matrix

Confusion Matrix		True Value	
		Positive	Negative
Predict Value	Positive	TP	FP
	Negative	FN	TN

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN}$$

After analyzing the "detailed accuracy by category" based on the WEKA results, we can create a new indicator formula by the following method.

According to the results of the WEKA confusion matrix, two Precision and Recall can be marked as P1 (0, absent) and P2 (1, present) and R1 (0, absent), R2 (1, present) for calculation, as shown below:

$$\mathrm{P1(absent)} = \frac{TP}{TP+FP} \qquad \mathrm{P2(present)} = \frac{TN}{TN+FN}$$

$$\mathrm{R1(absent)} = \frac{TP}{TP+FN} \qquad \mathrm{R2(present)} = \frac{TN}{TN+FP}$$

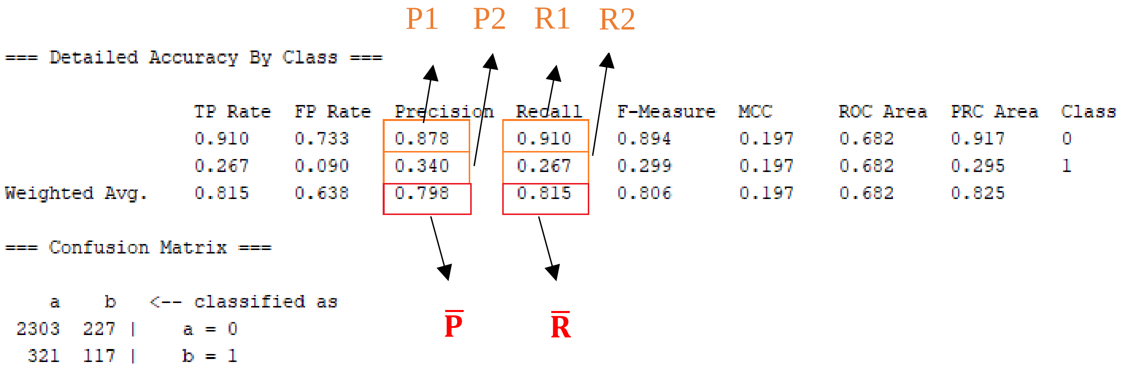
Then we will use the average precision and recall by linear weighted value. The following:

$$\overline{\mathbf{P}} = \mathrm{P1} \times \mathrm{y1} \; + \mathrm{P2} \times \mathrm{y2} \;; \quad \text{while } \; \mathrm{y1} = \frac{\mathrm{TP + FN}}{\mathrm{TP + FP + TN + FN}} \times 100\%; y2 = \frac{TN + FP}{TP + FP + TN + FN} \times 100\% \;; so \;\; y1 + y2 = 1$$

$$\overline{\mathbf{R}} = \mathrm{R1} \times \mathrm{y1} + \mathrm{R2} \times \mathrm{y2}; \quad \text{while } \; \mathrm{y1} = \frac{\mathrm{TP + FN}}{\mathrm{TP + FP + TN + FN}} \times 100\%; y2 = \frac{TN + FP}{TP + FP + TN + FN} \times 100\% \;; so \;\; y1 + y2 = 1$$

$\overline{\mathbf{P}}$ and $\overline{\mathbf{R}}$ results provided by WEKA.

Fig 2.5 Example of FHD data set detailed accuracy by category



```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.910    0.733    0.878      0.910    0.894      0.197    0.682     0.917     0
                 0.267    0.090    0.340      0.267    0.299      0.197    0.682     0.295     1
Weighted Avg.    0.815    0.638    0.798      0.815    0.806      0.197    0.682     0.825

=== Confusion Matrix ===

    a    b   <-- classified as
 2303  227 |    a = 0
  321  117 |    b = 1
```

From Fig 2.5, we can see that:

TP = 2303; TN = 117; FP = 321; FN = 227; Sum = 2968 instance.

P1 = 2303/(2303+321) = 0.877; P2 = 117/(117+227) = 0.340
R1 = 2303/(2303+227) = 0.910; R2 = 117/(117+321) = 0.267

Here, y1 = (2303+227)/2968 * 100% = 85.2%; y2 = (117+321)/2968 * 100% = 14.7% Then, $\overline{\mathbf{P}}$= 0.877 * 85.2% + 0.340*14.7% = 0.797; and $\overline{\mathbf{R}}$= 0.910 * 85.2% + 0.267 * 14.7% = 0.814

3.5 Classifiers algorithms

In this research we chose six classification algorithms, and each classification algorithm has its own characteristics. The performance of multiple conditions are also different, and the impact of performance in different data sets is not the same. Therefore, we conduct in-depth analysis and comparison of them through mathematical formulas. The principle of evaluating the quality of a classification algorithm is to evaluate the classification algorithm by accuracy.

We will use a formula to explain the results we want to obtain in chronic disease data mining and analysis. In this formula, Y is the dependent variable, which is the result we want in this study, and f is the method of converting the input into the result, which is the six classifier algorithms we use. X is the independent variable is the performance of the classifier used, and the ε random error term is caused by many factors that cannot be explained in the chronic disease data.

$$Y = f(X) + \varepsilon$$

In the classifier model, we are more concerned about the estimated accuracy of the target variable Y so we expect that the model is as accurate as possible. In addition, the classifier model f in the form of the model itself, will not get more explanation, as long as the structure can improve our prediction accuracy, and achieve the goal.

$$Y = \boxed{f}(X)$$

In the control task, we describe the relationship between X and Y as clearly as possible. The result is certainly important, but we are also concerned about the specific form of the model, or what kind of discriminant rules are generated by the statistical mining model to help us generate.

$$Y = \boxed{f(X)}$$

The following information is the six classifier algorithms to be used in this experiment, and we will analyze them in depth.

3.5.1 Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised machine learning algorithm. This algorithm has better prediction accuracy, mainly because it can transform low dimensional linear non-separable space into high dimensional linear separable space (Jakkula, 2006). Due to the high prediction accuracy, the algorithm is very popular in medical prediction.

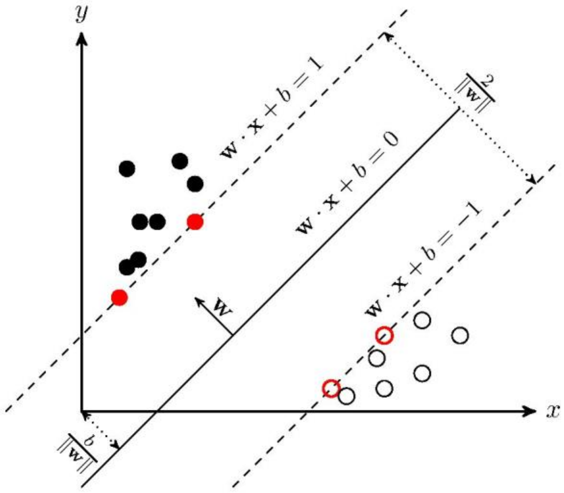
SVM classifier has several advantages in the classification or prediction of dependent variable rows. For example, because SVM classifier leads to the increase or decrease of the support-vector of the sample points, it will not change the effect of the classifier and avoids the occurrence of dimension disaster. The model has good generalization ability and avoids overfitting to a certain extent. It also avoids the local optimality when running the model in the program.

Furthermore, the main reason that we choose the SVM classifier is because this classifier can achieve better results than other algorithms on the medical medium- sized sample training set and has excellent generalization ability. Our selected chronic disease data set is mainly composed of four medium-sized data sets, so we believe that SVM will get good results.

The principle of the formula is as follows:

As shown in the figure (Fig 2.6) below, w * x + B = 0 is the separating hyperplane. The linear separable data sets of such hyperplanes are numerous, but the separable hyperplane with the largest geometric distance is unique.

Fig 2.6 Principle of SVM formula



y
$\frac{2}{\|\mathbf{w}\|}$
$\mathbf{w}\cdot\mathbf{x}+b=1$
$\mathbf{w}\cdot\mathbf{x}+b=0$
$\mathbf{w}\cdot\mathbf{x}+b=-1$
$\mathbf{w}$
x
$\frac{b}{\|\mathbf{w}\|}$

(Zhihua, 2016)

We assume that given a training data set on the feature space,

$$\mathrm{T} = \{(x_1, y_1), (x_2, y_2)\ \ , \dots, (x_N, y_N)\}$$

Where, $x_i \in R^n, y_i \in \{+1, -1\}, i = 1,2, \dots N$, x_i is the i eigenvector, y_i is the class mark, In addition, it is a positive example when it is equal to +1, and negative when is -1. We can also assume that the training data set is linearly separable.

3.5.2 Naïve Bayes

Naive Bayes classification algorithm is based on the probability theory and mathematical statistics knowledge which is suitable for medical big data sets. Its advantages are simple, high classification accuracy and high speed. The classification algorithm based on Naive Bayes mainly uses the deviation theorem to predict the unknown categories, gives the probability of each category, and finally selects the most likely category as the final prediction category of the sample (Webb, Keogh, & Miikkulainen, 2010).

The principle of the formula is as follows:

Naïve Bayes conditional probability (also known as posterior probability) is the probability that event A will occur if another event B has already occurred. The conditional probability is expressed as P (A|B), pronounced as "the probability of A under B".

$$\mathrm{P(A|B)} = \frac{P(AB)}{P(B)}$$

We assume that A is non-disease and B is disease, so the above formula is explained as follows:

P (A|B) - Probability of A under B conditions. That is, the probability of occurrence of event A under the condition that another event B has already occurred.

P (AB) - the probability of events A and B occurring at the same time, that is, the joint probability. Joint probability represents the probability of two events occurring together. The joint probability of A and B is expressed as P (AB).

P (B) - the probability of event B occurring.

Conditional probability: This is the probability of event A occurring under the condition that another event B has already occurred. P (A|B), which is the probability of A under the condition of B.

3.5.3 K- Nearest Neighbour (KNN)

KNN is a relatively mature method in theory first proposed by Cover in 1968, and it is also one of the simplest machine learning algorithms (Cover, 1968). This idea is simple and intuitive: If a sample belongs to a category, and K samples are the most similar in features, then the sample also belongs to this category. This method only determines the category of the samples to be divided according to the category of the latest one or more samples.

If most of them belong to the nearest K samples in a certain class of feature space, then these samples also belong to this class and the features of this class of samples. When the KNN algorithm determines the classification decision, it only determines the classification of the samples to be classified according to the category of the nearest one or more samples. When making classification decisions, KNN method is related to a few adjacent samples because the KNN method mainly depends on the limited samples around, rather than identifying the class domain to determine the class (Guo, Wang, bell, Bi, & Greer, 2003). The KNN method is superior to other methods in the division of overlapping or more overlapping sample sets, because it mainly depends on a limited number of adjacent samples rather than the method to determine the class domain.

K value choosing

When choosing a smaller K value, using smaller adjacent training examples for prediction can reduce the approximation error of learning. Only the training sample close to the input instance can help to predict the result.

The disadvantage is that the learning estimation error increases, and the prediction result is sensitive to the adjacent instance points. If adjacent instance points happen to have noise, the prediction will be wrong. In other words, the decrease of K value means that the whole model becomes more complex and the division is not clear, so over fitting may occur (Guo, Wang, bell, Bi, & Greer, 2003). The disadvantage is that the estimation error increases. If the adjacent instance points have noise, the prediction is wrong. It can be said that the decrease of K value means that the whole model becomes more complex and the division is not clear, so fitting is likely to occur (Guo, Wang, bell, Bi, & Greer, 2003).

Choosing a larger K value is equivalent to using a training instance to predict in a larger neighborhood. Its advantage is that it can reduce the estimation error of learning, but the approximation error will increase, and the prediction of input instance is not accurate. The increase of K value means that the whole model is simple.

Approximation error: the training error of the existing training set.
Estimation error: test error on test set.
The approximation error is concentrated on the training set. If the K value is small, there will be over fitting phenomenon, which is a good prediction for the existing training set, but for the unknown test samples, there will be large bias prediction. The model itself is not the closest to the best model. The smaller the estimation error is, the better the prediction ability of unknown data is. The model itself is closest to the best model (Guo, Wang, bell, Bi and Greer, 2003).

The K value used in our experiment is 1, because when it is 1, the error is the smallest and will give the best accuracy.

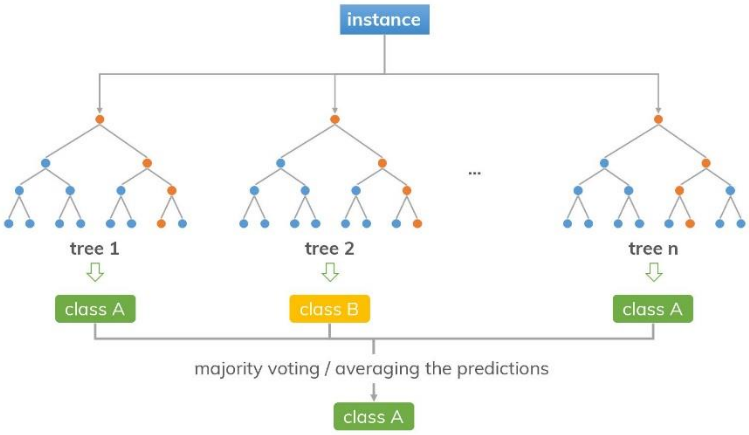
3.5.4 Random Forest

Random Forest is very suitable for medical data sets. The classifier is reflected in that the training sample of each tree is random, and the split attribute set of each node in the tree is also determined by random selection. It can process high-dimensional data without feature selection, and the training speed is fast.

Random Forest algorithm is composed of classification tree, with rows representing random numbers and columns representing variables. They randomly assign values to rows and columns to generate more classification trees. The branch determines the random forest algorithm, and the structure of the branch belongs to the content of recursive control. All information can be accurately classified and controlled to get effective types, but in practical application, it is difficult to achieve. Even if there are usually a large number of nodes behind the model obtained by construction, it also shows over fitting (Esmaily, Tayefi, Doosti, Ghayour-Mobarhan, Nezami, & Amirabadizadeh, 2018).

In practical operation, it should be able to achieve high performance in the control treatment of branch and leaf construction. Random Forest can be used to control this situation. In order to realize the classification construction, it is necessary to vote the decision tree of the forest obtained from multiple decision trees. In the process of decision tree generation and control, each part of the decision tree inevitably shows a strong effect of random execution, and the required content can be obtained by optimizing the segmentation.

Fig 2.7 Random Forest illustration



(Github. 2020)

Each classification tree in the random forest is also a vertical binary tree, and each tree conforms to the principle of top-down recursive segmentation, that is, starting from the root interrupt to divide the training set. The split continues with all the training data, a subset of the left routine, and a subset of the upper right routine. They can only stop splitting if they meet the splitting stop rule (Github. 2020).

Random Forest algorithm can use a classification method for medical data which improves the accuracy of the estimation information. The analysis can only be carried out in the case of unbalanced control or missing data. It can predict explanatory variables thousands of times, and the algorithm is powerful.

3.5.5 Bayesian Network

Bayesian network has a powerful uncertainty reasoning method, which will solve some uncertainty of medical diagnosis. This ability is exactly what we need for this experiment.

As a causal inference model, Bayesian network algorithm is widely used in medical diagnosis, electronic technology, information retrieval and industrial engineering, etc.

The probability graph model is a Bayesian network algorithm, which was first proposed by Judeapall in 1985 (Judeapall, 1985). It is an uncertainty processing model and a model simulating human reasoning causality. The structure is directed acyclic graph (DAG). In the use of Bayesian network, probability reasoning and decision-making methods are used. In the case of incomplete information, invisible random variables can be inferred from observable random variables, and invisible random variables can be more. In general, invisible variables are initially set to random values, and then probabilistic reasoning is performed.

The Bayesian network stipulates that the parent node of node Xi is the condition, and Xi is conditionally independent of any non-Xi child nodes. According to this agreement, the joint probability distribution of a Bayesian network with n nodes is:

$$P(x_1, x_2 \dots, x_n) = \prod_{i=1}^{n} P\,(x_i|\pi(x_i))$$

Among them, $\pi(x_i)$is a combination of the values of the variables in the x_i parent node set $\prod xi$ in the network. If x_i has no parent node, then the set $\prod xi$ is empty, that is $\mathrm{p}\big(x_i \big| \pi(x_i)\big) = \mathrm{p}(x_i)$

According to the above formula, we can write the joint probability distribution of Bayesian network.
The difference between the two expressions lies in the conditional probability part. In Bayesian networks, if its dependent variable is known, some nodes will be conditionally independent of its “dependent” variable, and only the nodes related to the “dependent” variable have conditional probability.

If the number of dependencies of joint distribution is very small, the Bayesian function method can save considerable memory capacity. For example, 10 variables with a value of 0 or 1 are stored in the conditional probability table type. The intuitive idea is that we need to calculate a total of 2 ^ 10 = 1024 values; however, if none of the 10 variables has more than three related "dependent" variables, then the conditional probability table of Bayesian network only needs to calculate 10 * 2 ^ 3 = 80 values at most.

3.5.6 J48 (Decision Tree)

The benefit of the Decision Tree classifier for this chronic disease analysis is that it is easy to interpret the classification results and can handle the association between features without considering whether abnormal samples or data are linearly separable.
Decision Tree calculates the probability that the expected value is greater than or equal to zero by constructing a branch structure on the basis of knowing the occurrence probability of various situations. This is a graphical method that intuitively uses probabilistic analysis (Bhargava, Sharma, Bhargava, and Mathuria, 2013). The calculation process of Decision Tree algorithm is a sigmoid function, which does not need to standardize or normalize the original data. The construction image of decision branch is similar to a branch tree, so it is called Decision Tree. The type of decision tree depends on the type of target variable, which can be divided into two categories:

- Categorical variable Decision Tree: also known as classification tree. When the target variable of a Decision Tree is an attribute class and the output data is a sample class label, it is characterized as a discrete variable Decision Ttree.

- Continuous variable Decision Tree: It is also called a regression tree. When the target variables of the Decision Tree are a series of continuous variables, the value of the output data is expressed as a Decision Tree body of continuous variables.

A Decision Tree is composed of root nodes, decision nodes, leaves, subtrees, etc. The related concepts are introduced as follows:

- Root node: represents the entire group or sample, which can be further divided into two or more homogeneous sets

- Splitting: The process of dividing a node into two or more child nodes

- Decision node: When a child node splits into more child nodes, the node is a decision node

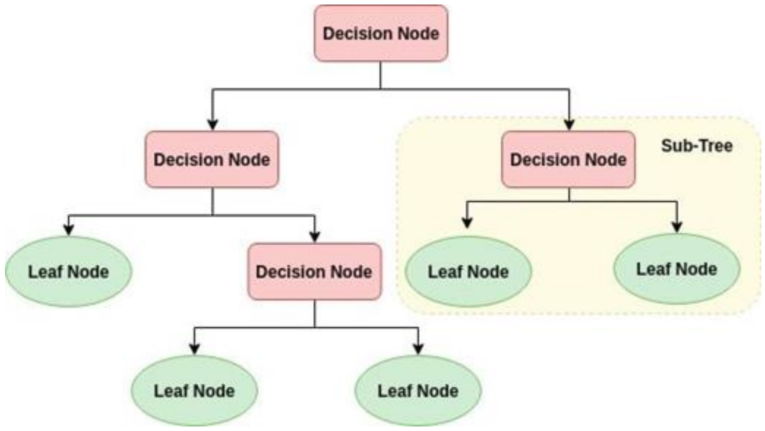
- Terminal node: also known as a leaf node, that is, a node that can no longer be split

- Pruning: represents the process of deleting the child nodes of the decision node, which is the inverse process of splitting

- Branch: also known as a subtree, sub-parts of the entire tree can be called branches or subtrees

- Parent node and child node: Every node except the root node has a parent node. If a node has child nodes, the node is called the parent node of these child nodes, and the child nodes of the same parent node are called siblings

Fig 2.8 Decision Tree illustration



(Avinash, 2018)

The functions and advantages of Decision Trees are as follows:

(1) Extract important features

Decision Tree is a method of mining the most important variables and relationships among multiple acting variables. Through Decision Tree, new functions can be created to predict target variables (Rokach, & Maimon, 2005).

(2) Low data cleaning requirements

Compared with other methods, Decision Trees have lower requirements for data cleaning, because invalid values and missing values have no effect on the decision-making process. In addition, the decision-making process is characterized as a sigmoid function, and there is no need to standardize or normalize the original data (Rokach, & Maimon, 2005).

(3) No requirement for data type

Decision Tree algorithm and its optimization algorithm are suitable for numerical and nominal data (Rokach, & Maimon, 2005).

3.6 Validation

When performing the evaluation process, the models will be built under the test data set. At this stage, it is necessary to ensure that the models can effectively completely achieve the goals. (Mariscal, Marban and Fernandez, 2010).

Therefore, In this part, we use the testing data set to rank the performance of the selected SVM, Naive Bayes, KNN, Random Forest, Bayesian Network and J48 (Decision Tree) classifiers, and verify whether the results are the same as the results in the training data set. We use the 10 fold cross validation method to verify it again in the experiment. Also, we verify the ROC curve of the optimal classifiers generated in each data set.

The following are the arguments of these two methods：

3.6.1 10-fold cross-validation

The 10-fold cross-validation is used to prevent over fitting. It divides the chronic disease data sets into 10 fold for experiments. 10 fold is the appropriate choice to obtain the best error estimate and repeatedly uses random. The generated sub-samples are trained and verified, and the results are verified once each time. K = 10, that the data set is divided into 10 parts. The cycle extracts 1 part as the verification set and the other 9 parts as the training set.

3.6.2 ROC (Receiver Operator Curve)

The ROC value reflects the diagnostic ability of the classifier under different thresholds. The graph is based on the true positive rate and false positive rate (Sarang, 2018). The ROC curve is described as follows:

1. The curve shows the trade-off between specificity and sensitivity (specificity decreases with the increase of sensitivity, and vice versa).

2. The curve inclination in the upper left quadrant of the curve indicates that the test

result is more accurate. Therefore, the compactness of the curves and diagonals indicate that the accuracy of the test results is low.

3. The area under the curve is the measurement accuracy, which is called the area under the curve.

The area under the curve represents the test accuracy. If the area under the curve is large, the test is more accurate. The area under the ROC curve is greater than 0.5, which proves that the diagnostic experiment has a certain diagnostic value. At the same time, the closer the area under the ROC curve is to 1, the better the authenticity of the diagnostic experiment is.

3.6.3 Performance Comparison of Classifiers

The realization and creation of the model is only a part of the whole project. In the follow-up, corresponding reports are given based on performance effects to complete effective mining control (Mariscal, Marban and Fernandez, 2010). We found the best three classifiers in the training set of the four data sets. So 4*3=12, we selected 12 classifiers in the training set to enter the next step. We made the same adjustments on the testing set for the 12 selected classifiers at the same time and through training and testing the two models to compare the errors between them, selected the closest model. The model with the closest accuracy is the best model.

3.7 Summary

In summary, this chapter aims to establish a set of research methods suitable for this thesis. We describe in detail the process and method of data mining. The six classifiers used are theoretically explained. Finally, the verification method and the method of classifier comparison are described.

Chapter 4

Findings and Discussion

This experiment is based on four different versions of chronic disease data sets, through six different classifiers and three rebalancing methods to analyze different degrees of imbalance in the data sets.

The first experiment used the original data set; the distribution of class data was unchanged, and six different classifiers were used for training and performance comparison.

In the second experiment, we used SMOTE, Resampling and SpreadSubsampling rebalancing methods to rebalance the data sets. Then six classifiers were trained and compared on the rebalanced data sets. This gave a total of 72 (6*3*4) 6 = six classifiers, 3 = three rebalancing methods and 4 = four data sets metric results that were calculated from the experiments.
There are nine parameter combinations of different values, therefore 216 experiments can be performed. The summary of ranking and the average score was obtained. The purpose of this experiment is to use these metrics to evaluate the relationship between rebalancing techniques and the performance achieved by the model.

Our experiments mainly verify the improvement of classifier performance with and without a rebalancing method. This method is implemented in two configurations; one configuration being the original data set. In the experiment, we used Bayesian Network, Naive Bayes, Support Vector Machine (LibSVM), KNN, J48 (Decision Tree) and Random Forest classifiers to model on the unbalanced data set, and by using the 10-fold cross-validation technique trained the data set to construct all the classifiers to provide accuracy.

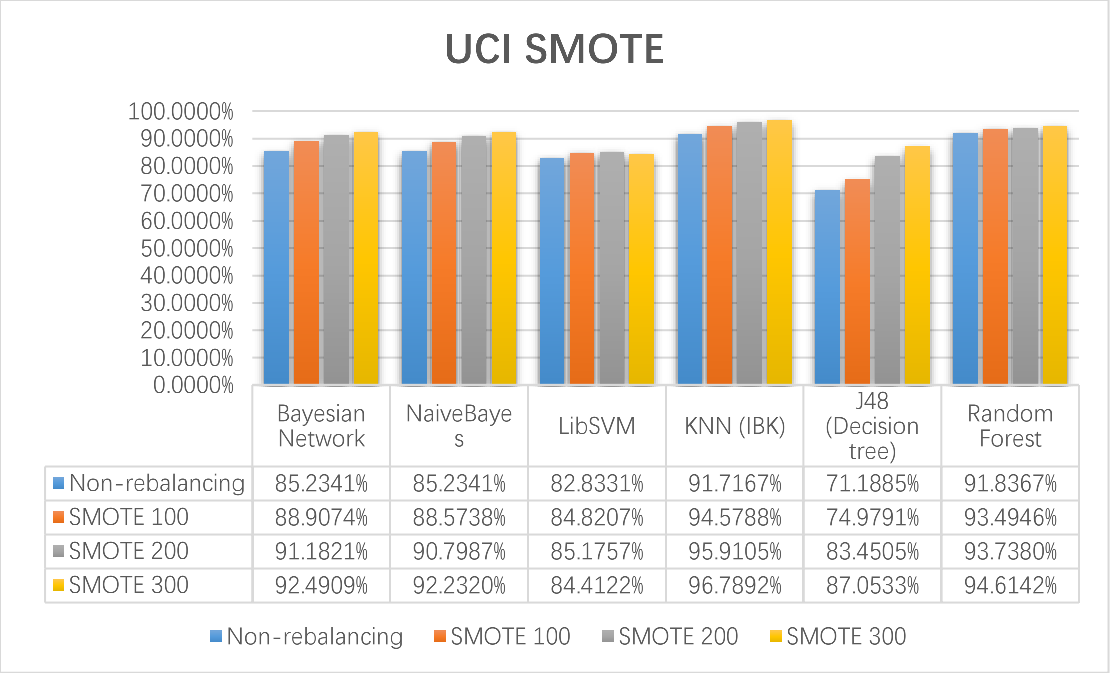
In the second set of experiments, in order to improve the prediction accuracy of the classifier and eliminate the bias of the algorithm on minority classes, we adopted the SMOTE, Resampling, and SpreadSubsamping rebalancing methods to rebalance the data sets. The main principle of SMOTE is to fill several types of synthetic instances. It uses two main parameters: N and k, which represent the percentage of oversampling and the number of nearest neighbors to be considered. In the Weka machine learning tool, we can set the parameters n100 and k5 as the default parameters of the SMOTE method.   The main principle of resampling is to increase the number of instances in the minority class by randomly copying them, thereby increasing the representativeness of the minority class in the sample. The main parameters it uses are: biasTouniformClass and randomSeed as default. In the Weka machine learning tool, we set the parameters biasTouniformClass to 1, 0.8, and 0.4 respectively. The main principle of SpreadSubsampling is to balance the class distribution by randomly eliminating samples of the majority class because the goal is not achieved until the instances of the majority class and the minority class are balanced. For the main parameters we used distributionSpread and maxCount as the default. In the Weka machine learning tool, we set the parameter distributionSpread to 1, 3, and 5 respectively.

The following is a comparison of the results of each classifier models on data sets.

4.1 Classifier Models

4.1.1 SMOTE (Appendix C)

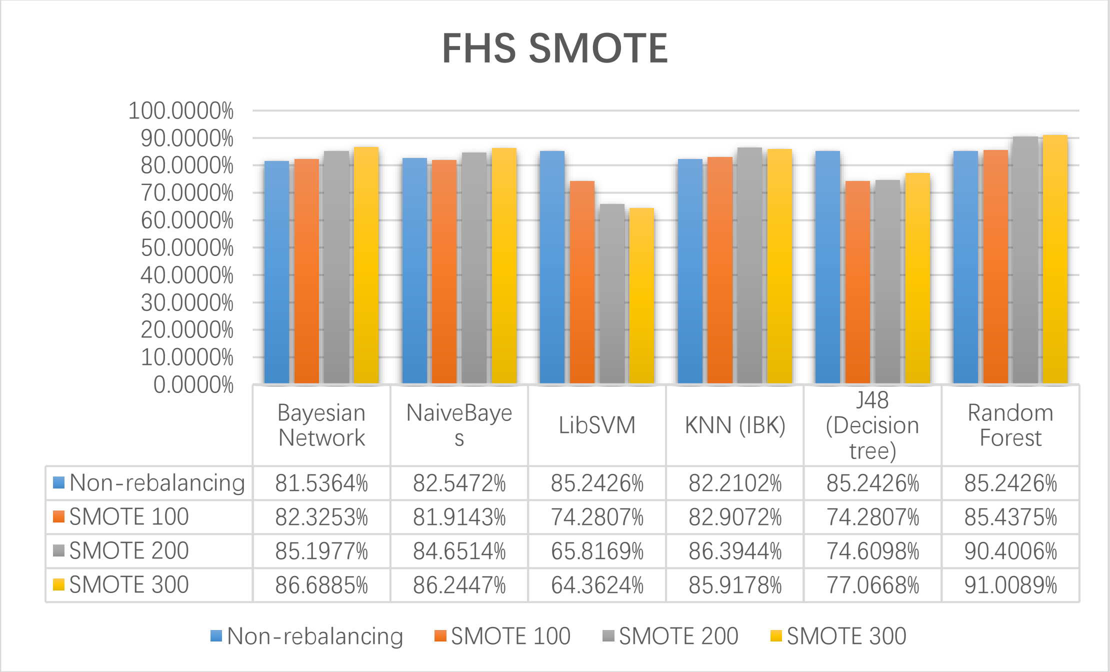
Fig 2.9 UCI Data set - SMOTE



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
■ Non-rebalancing	85.2341%	85.2341%	82.8331%	91.7167%	71.1885%	91.8367%
■ SMOTE 100	88.9074%	88.5738%	84.8207%	94.5788%	74.9791%	93.4946%
■ SMOTE 200	91.1821%	90.7987%	85.1757%	95.9105%	83.4505%	93.7380%
■ SMOTE 300	92.4909%	92.2320%	84.4122%	96.7892%	87.0533%	94.6142%

Firstly, the SMOTE rebalancing method was used on the UCI data set. The results show that the overall situation is that with the improvement of the parameters, the accuracy of the six classifiers has also been improved. We found the three best classifiers to be KNN (IBK) 96.7892%, Random Forest 94.6142% and Bayesian Network 92.4909%. However, they are not the fastest classifiers to improve. In the figure above, the classifier improvement rate of J48 (Decision Tree) is as fast as 15.9%. Meanwhile, KNN increased from 91.7167% to 96.7892%, with a difference of 5.07%. The worst classifier in this data set is SVM (LibSVM). We can see that the accuracy of SVM (LibSVM) is 82.8331% on the imbalanced data set, but it only improved by 1.58% to 84.4122%.

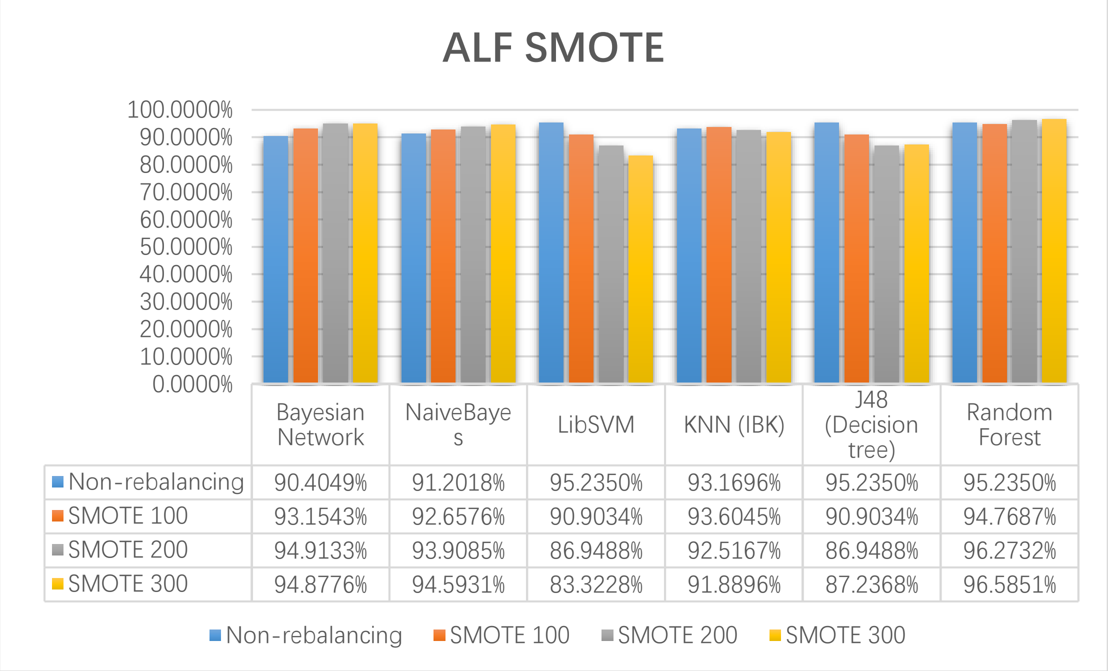
Fig 3.1 FHS Data set - SMOTE



	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
Non-rebalancing	81.5364%	82.5472%	85.2426%	82.2102%	85.2426%	85.2426%
SMOTE 100	82.3253%	81.9143%	74.2807%	82.9072%	74.2807%	85.4375%
SMOTE 200	85.1977%	84.6514%	65.8169%	86.3944%	74.6098%	90.4006%
SMOTE 300	86.6885%	86.2447%	64.3624%	85.9178%	77.0668%	91.0089%

As can be seen from the above figure, the best classifiers are Random Forest 91.0089%, Bayesian Network 86.6885% and Naive Bayes 86.2447%. The overall situation is that with the increase of SMOTE percentage, the accuracy of the classifiers also increased. But we can see that LibSVM and J48 (Decision Tree) are not like this. In other words, with the increase of percentage, the accuracy of LibSVM decreases. The accuracy of LibSVM is 85.2426% on an imbalanced data set, but the accuracy of LibSVM is 64.3624% on SMOTE 300 balanced data set. The J48 (Decision Tree) classifier has a similar situation, especially the accuracy on SMOTE 300 rebalanced data set 77.0668% which is 2.5% higher than SMOTE 200 74.6098%.

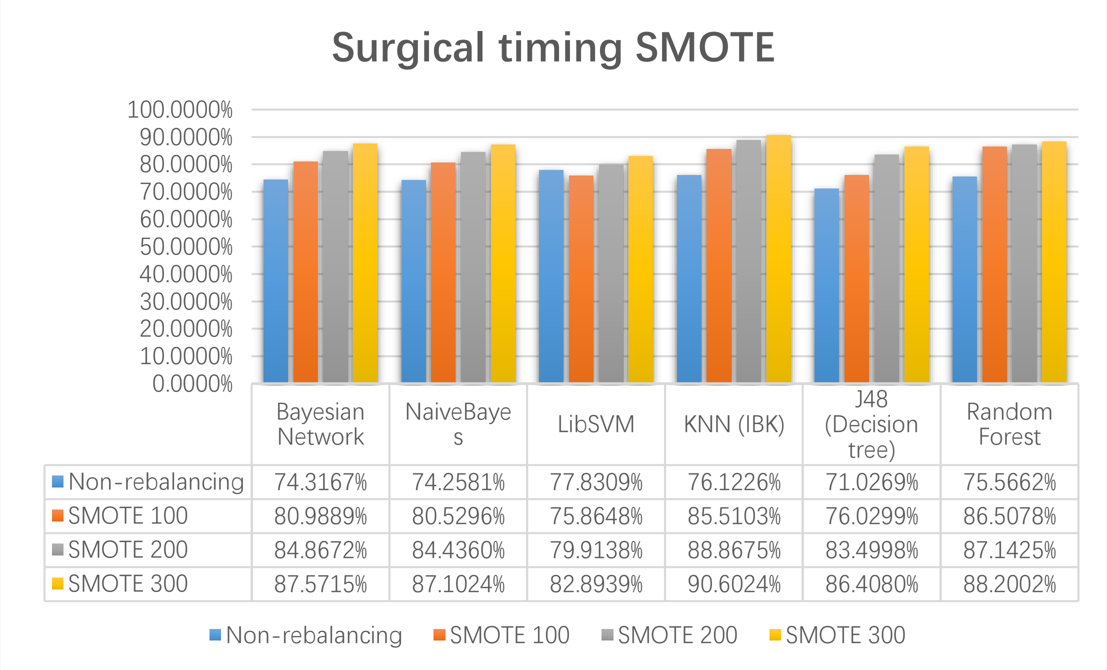
Fig 3.2 ALF Data set - SMOTE



	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
Non-rebalancing	90.4049%	91.2018%	95.2350%	93.1696%	95.2350%	95.2350%
SMOTE 100	93.1543%	92.6576%	90.9034%	93.6045%	90.9034%	94.7687%
SMOTE 200	94.9133%	93.9085%	86.9488%	92.5167%	86.9488%	96.2732%
SMOTE 300	94.8776%	94.5931%	83.3228%	91.8896%	87.2368%	96.5851%

In the ALF data set, we found that the accuracy of SVM (LibSVM) and J48 (Decision Tree) decreases with the adjustment of the parameters. In particular, LibSVM reduced from 95.2350% in the unbalanced data set to 83.3228%. Using the SMOTE 300 data set, the accuracy was reduced by 11.9%. The performance of J48 (Decision Tree) is also 95.2350% on the non - rebalanced data set from the initial experiment. On the 87.2368% SMOTE300 data set, the accuracy is reduced by 7.99%. The best performing classifier is Random Forest 96.5851% on the SMOTE300 data set. With it we see an increase of 1.35% compared to the unbalanced data set. This is followed by J48 (decision tree) 95.2350% and Bayesian network 94.9133%.

Fig 3.3 Surgical Timing Data set - SMOTE

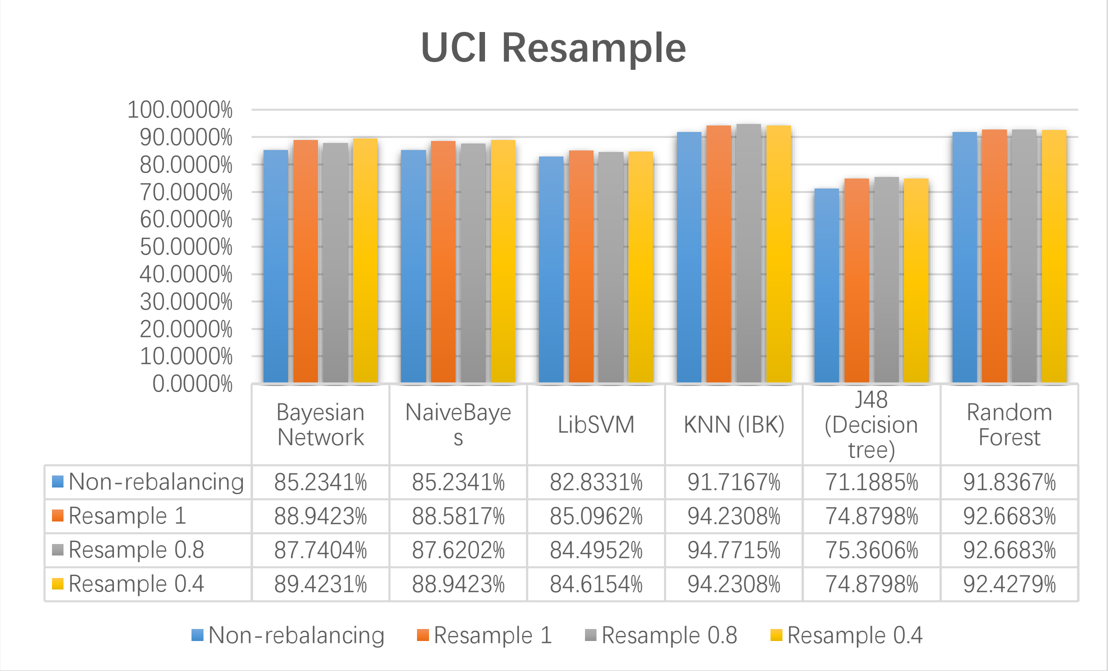


	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
🟦 Non-rebalancing	74.3167%	74.2581%	77.8309%	76.1226%	71.0269%	75.5662%
🟧 SMOTE 100	80.9889%	80.5296%	75.8648%	85.5103%	76.0299%	86.5078%
▨ SMOTE 200	84.8672%	84.4360%	79.9138%	88.8675%	83.4998%	87.1425%
🟨 SMOTE 300	87.5715%	87.1024%	82.8939%	90.6024%	86.4080%	88.2002%

From the surgical timing data set, we can see that the accuracy of the overall classifier increases with the percentage of SMOTE. In particular, the accuracy of J48 (Decision Tree) on the non-rebalanced data set is 71.0269% and the accuracy on the SMOTE300 balanced data set is 86.4080%. It can be seen that the accuracy has increased by 15.38%. Another classifier with higher improvement is KNN (IBK); from the initial experiment 76.1228% on the non-rebalanced data set to 90.6024% on the SMOTE300 data set, the accuracy increased by 14.47%. The best classifier in the data set is KNN (IBK) 90.6024%, followed by Random Forest 88.2002%.

4.1.2 Resampling (Appendix C)

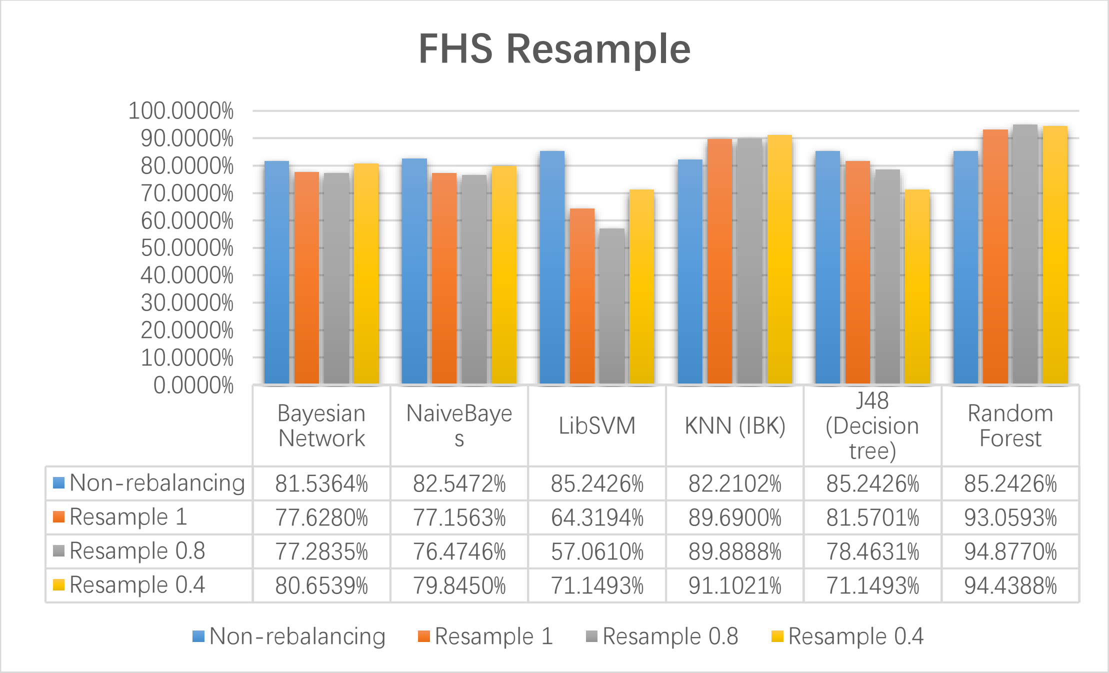
Fig 3.4 UCI Data set - Resampling



	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
🟦 Non-rebalancing	85.2341%	85.2341%	82.8331%	91.7167%	71.1885%	91.8367%
🟧 Resample 1	88.9423%	88.5817%	85.0962%	94.2308%	74.8798%	92.6683%
⬜ Resample 0.8	87.7404%	87.6202%	84.4952%	94.7715%	75.3606%	92.6683%
🟨 Resample 0.4	89.4231%	88.9423%	84.6154%	94.2308%	74.8798%	92.4279%

In the UCI data set, we can see that the accuracy of the overall classifier increases with the adjustment of the parameters. The accuracy of Bayesian Network initially obtained on the unbalanced data set was 85.2341%. After tuning, the accuracy reached 89.4231%. During this period, an increase of 4.189% was achieved. At the same time, the accuracy of KNN (IBK) increased from 91.7167% of the unbalanced data set in the initial experiment to 94.2308% of the Resample0.4 data set. However, the performance of the J48 (Decision Tree) classifier is the worst here. It increased from 71.1885% on the unbalanced data set to 75.3606% on the Resample0.8 data set. Although it has increased by 4.17%, it is comparable to other classifiers. In this experiment, the best classifier is KNN (IBK) 94.2308%, followed by random forest 92.4279%.

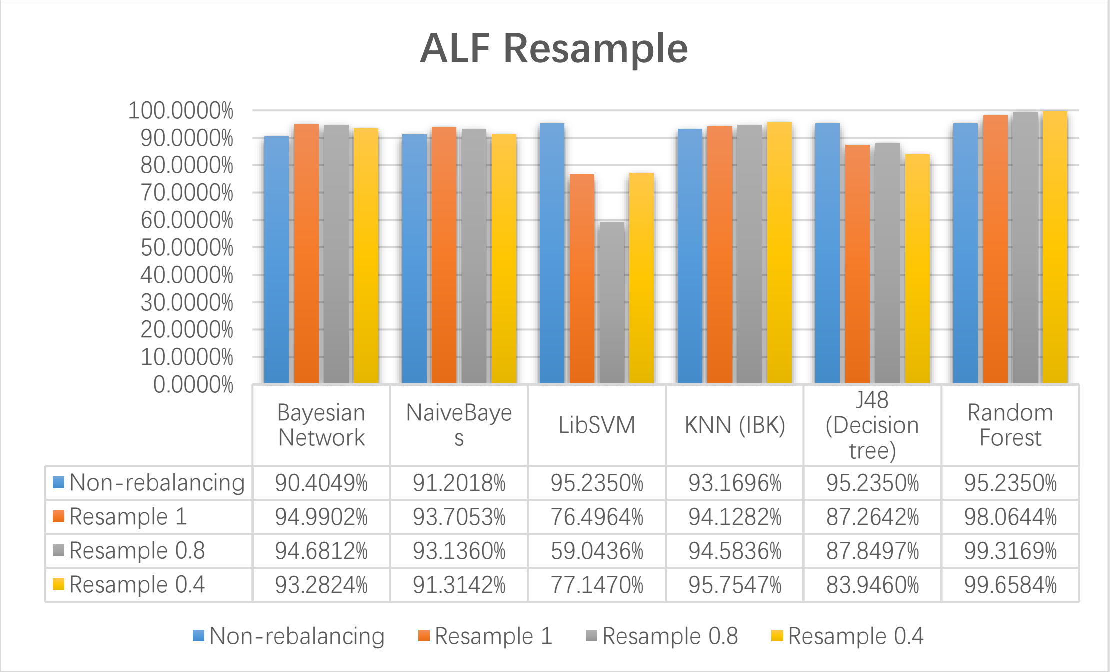
Fig 3.5 FHS Data set - Resampling



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
■ Non-rebalancing	81.5364%	82.5472%	85.2426%	82.2102%	85.2426%	85.2426%
■ Resample 1	77.6280%	77.1563%	64.3194%	89.6900%	81.5701%	93.0593%
■ Resample 0.8	77.2835%	76.4746%	57.0610%	89.8888%	78.4631%	94.8770%
■ Resample 0.4	80.6539%	79.8450%	71.1493%	91.1021%	71.1493%	94.4388%

From the FHS data set, we can see that the accuracy of the classifiers are not stable, and the accuracy of some classifiers is not increased by parameters. For example, the accuracy of SVM (LibSVM) on the non-rebalanced data set is 85.2426%, but the accuracy on the Resample1 data set has dropped to 64.3194%. At the same time, the accuracy has increased to 71.1493% as the parameters continue to be adjusted. We can find that the performance of the SVM (LibSVM) classifier in the FHS data set is not suitable. The accuracy of the J48 (Decision Tree) classifier is reduced as the parameters are adjusted, from 85.2426% on the non-rebalanced data set initially to 71.1493% on the Resample0.4 data set, which reduces the accuracy by 14.09%. In addition, the accuracy of the NaiveBayes classifier has some small pulsations from the 82.5472% on non-rebalanced data set to 76.4746% on Resample0.8 data set which reduces the accuracy by 6.07%. The best classifier in this data set is Random Forest 94.4388%, followed by KNN (IBK) 91.1021%.

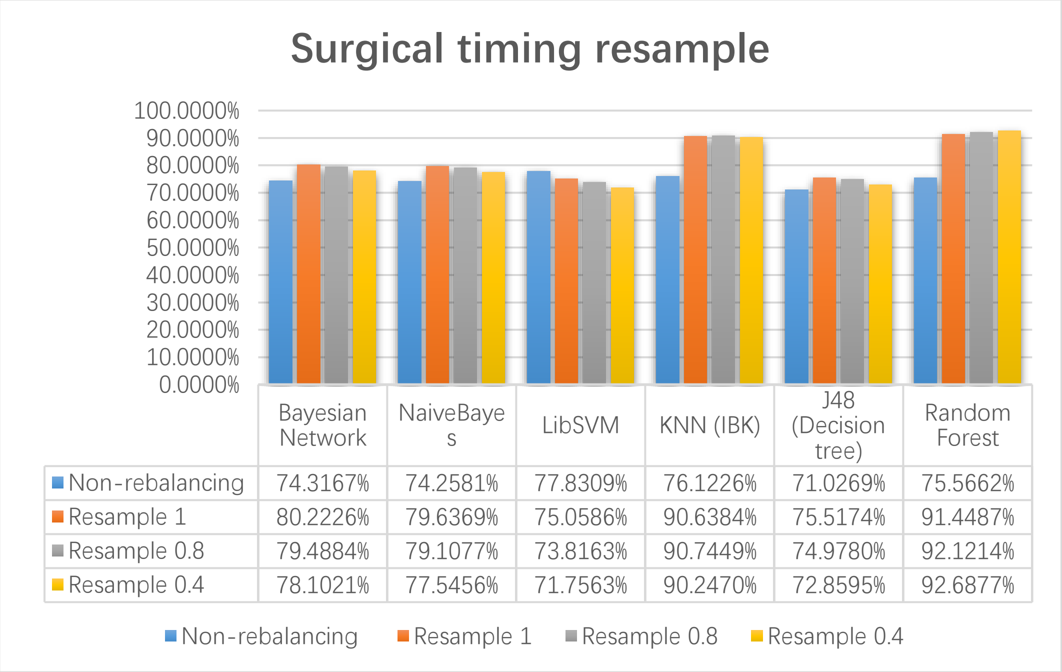
Fig 3.6 ALF Data set – Resampling



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
🟦 Non-rebalancing	90.4049%	91.2018%	95.2350%	93.1696%	95.2350%	95.2350%
🟧 Resample 1	94.9902%	93.7053%	76.4964%	94.1282%	87.2642%	98.0644%
⬜ Resample 0.8	94.6812%	93.1360%	59.0436%	94.5836%	87.8497%	99.3169%
🟨 Resample 0.4	93.2824%	91.3142%	77.1470%	95.7547%	83.9460%	99.6584%

On the ALF data set, we can see that the SVM (LibSVM) classifier is the most unstable. On the non-rebalanced data set, 95.2350% it gradually dropped to 59.0436% and the Resample0.8 data set has a difference of 36.19%. Secondly, the accuracy of the J48 (Decision Tree) classifier decreases with the adjustment of the parameters, from 95.2350% on the non-rebalanced data set to 83.9460% on the Resample0.4 data set. In this experiment, we found the highest accuracy classifier to be Random Forest through tuning parameters to 99.6584%. We believe that its accuracy is the highest in all experiments. The next highest-precision classifiers are Bayesian Network (93.2824%) and NaiveBayes (91.3142%). In our previous experiment, KNN (IBK) was a higher classifier, but the accuracy in this experiment also reached 95.7547%.

Fig 3.7 Surgical Timing Data set - Resampling

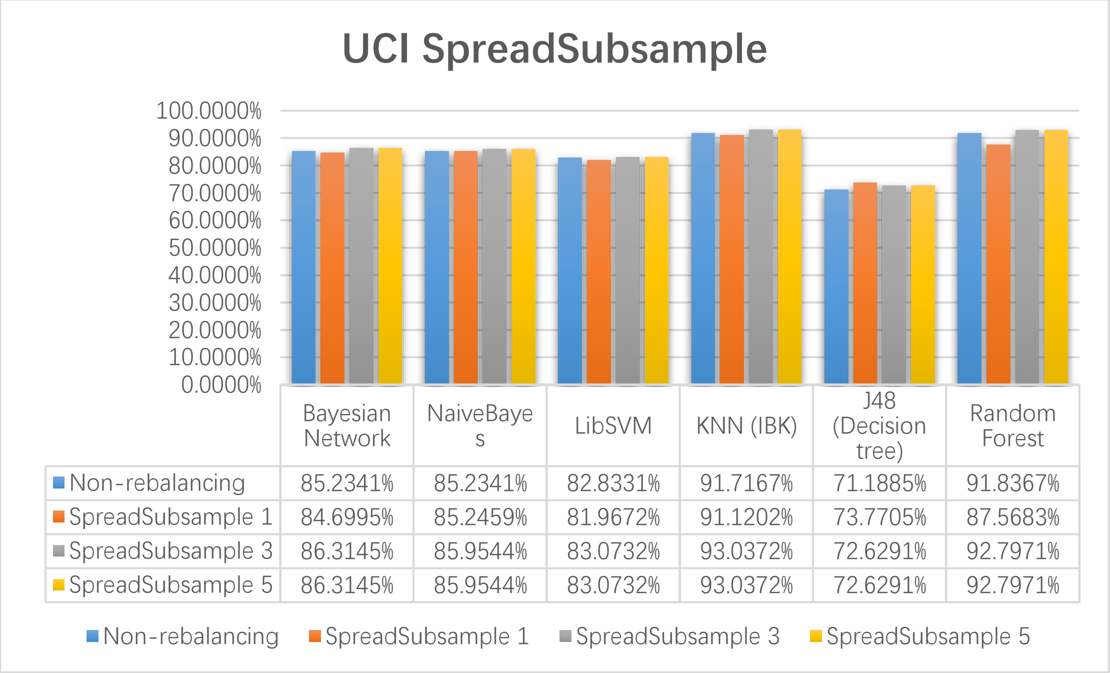


	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
■ Non-rebalancing	74.3167%	74.2581%	77.8309%	76.1226%	71.0269%	75.5662%
■ Resample 1	80.2226%	79.6369%	75.0586%	90.6384%	75.5174%	91.4487%
■ Resample 0.8	79.4884%	79.1077%	73.8163%	90.7449%	74.9780%	92.1214%
■ Resample 0.4	78.1021%	77.5456%	71.7563%	90.2470%	72.8595%	92.6877%

In the operation time data set, we can see that the two classifiers Random Forest and KNN (IBK) perform best. We analyzed the Random Forest classifier from the perspective of accuracy and we can see that the accuracy on the non-rebalanced data set is not very good, only 75.5662%, but with the adjustment of the parameters, it reached 92.6877% on the Resample0.4 data set. The KNN (IBK) classifier is 76.1226% on the non-rebalanced data set, which is also a relatively low accuracy. However, with the adjustment of the parameters, it reaches 90.2470%, which is a difference of 14.12%. The accuracy of the other four classifiers is less Bayesian Network reached 80.2226% on the Resample1 data set, and the remaining classifiers did not reach more than 79%. So the best classifier in this experiment are Random Forest and KNN (IBK).

4.1.3 SpreadSubsampling (Appendix C)

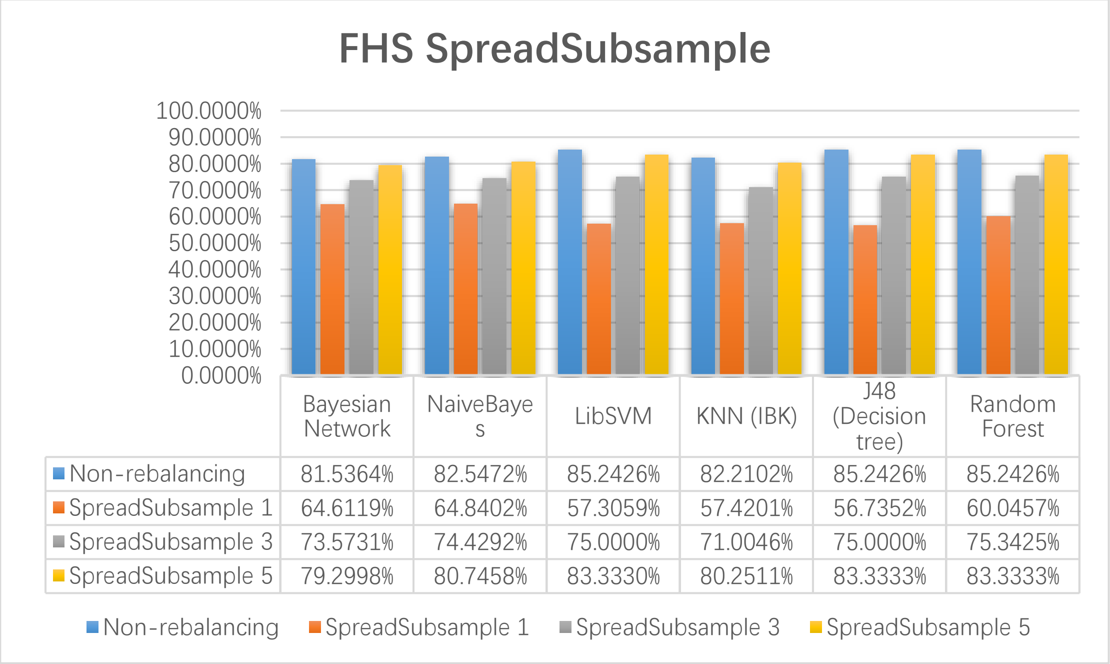
Fig 3.8 UCI Data set - SpreadSubsampling



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
■ Non-rebalancing	85.2341%	85.2341%	82.8331%	91.7167%	71.1885%	91.8367%
■ SpreadSubsample 1	84.6995%	85.2459%	81.9672%	91.1202%	73.7705%	87.5683%
■ SpreadSubsample 3	86.3145%	85.9544%	83.0732%	93.0372%	72.6291%	92.7971%
■ SpreadSubsample 5	86.3145%	85.9544%	83.0732%	93.0372%	72.6291%	92.7971%

In the UCI data set, we can see that the accuracy of J48 (Decision Tree) and SVM (LibSVM) are relatively low among these six separators. In particular, the accuracy of J48 (Decision Tree) on the non-rebalanced data set is only 71.1885%, and after tuning parameters, the accuracy only increased by 1.44% to 72.6291% on the SpreadSubsample3 data set. The accuracy of the SVM (LibSVM) classifier increased from 82.8331% to 83.0732%, which is only an increase of 0.24%. On the other hand, the optimal classifier in this data set is KNN (IBK), which achieved an accuracy of 93.0372% on the SpreadSubsample3 data set. Random Forest achieved an accuracy of 92.7971%.

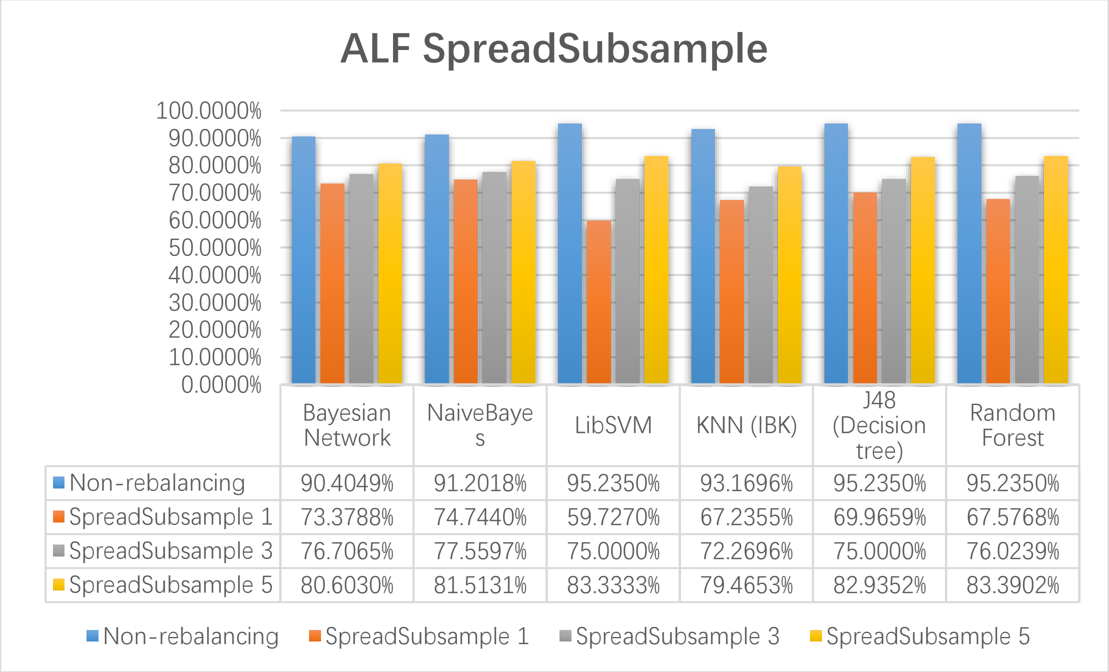
Fig 3.9 FHS Data set - SpreadSubsampling



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
🟦 Non-rebalancing	81.5364%	82.5472%	85.2426%	82.2102%	85.2426%	85.2426%
🟧 SpreadSubsample 1	64.6119%	64.8402%	57.3059%	57.4201%	56.7352%	60.0457%
⬜ SpreadSubsample 3	73.5731%	74.4292%	75.0000%	71.0046%	75.0000%	75.3425%
🟨 SpreadSubsample 5	79.2998%	80.7458%	83.3330%	80.2511%	83.3333%	83.3333%

On the FHS data set, we can see that the overall accuracy of the classifiers is not very good. First of all, let's look at the accuracy of KNN (IBK) on the non-rebalanced data which reached 82.2102%, but with parameter adjustment, the accuracy dropped to 57.4201% on the SpreadSubsample1 data set, which is a difference of 24.7901%. Therefore, it can be seen that the accuracy of the classifier on the data set of SpreadSubsample1 is not very high. SVM (LibSVM) is only 57.3059%, J48 (Decision Tree) 56.7352%, and Random Forest 60.0457%. However, the best classifiers produced in this data set are Random Forest and J48 (Decision Tree), their accuracy is equal at 85.2426%.

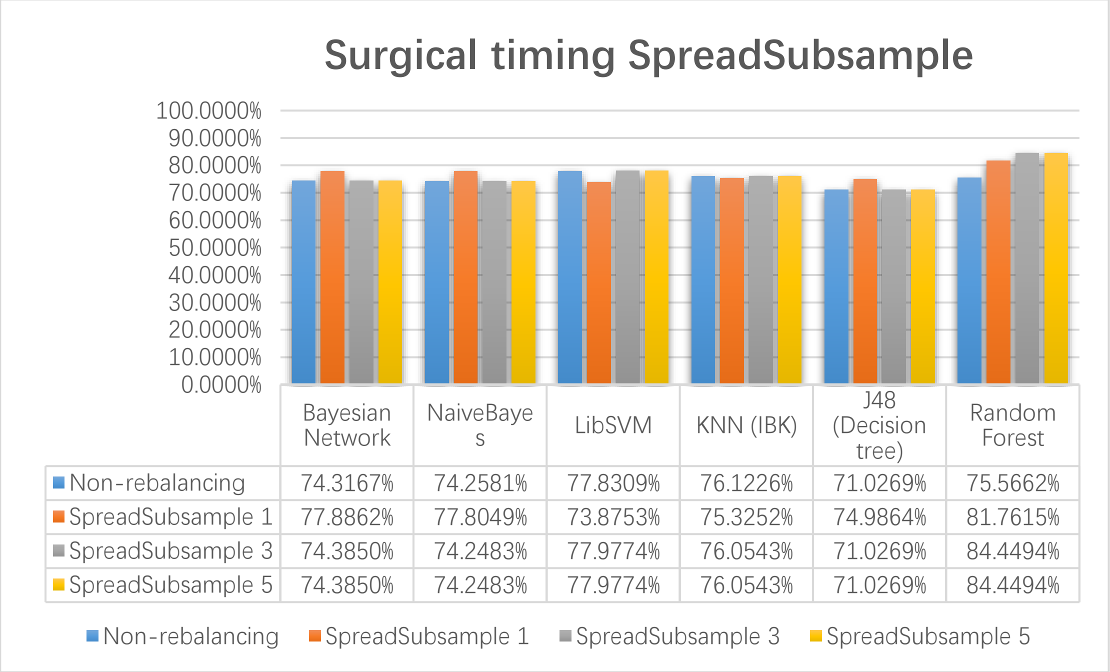
Fig 4.1 ALF Data set - SpreadSubsampling



	Bayesian Network	NaiveBayes	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
Non-rebalancing	90.4049%	91.2018%	95.2350%	93.1696%	95.2350%	95.2350%
SpreadSubsample 1	73.3788%	74.7440%	59.7270%	67.2355%	69.9659%	67.5768%
SpreadSubsample 3	76.7065%	77.5597%	75.0000%	72.2696%	75.0000%	76.0239%
SpreadSubsample 5	80.6030%	81.5131%	83.3333%	79.4653%	82.9352%	83.3902%

In the ALF data set, we found that the six classifiers on the non-rebalanced data set achieved high accuracy. Random Forest, SVM (LibSVM) and J48 (Decision Tree) both achieved accuracy of 95.2350%. In addition, KNN achieved accuracy of 93.1696%, and the accuracy of Bayesian Network and NaiveBayes was 90.4049% and 91.2018% respectively. However, the accuracy of the six classifiers on the SpreadSubsample1 data set are all degraded. Compared with the non-rebalanced data set, the accuracy of LibSVM is reduced by 35.508%, KNN (IBK) 25.9341%, and J48 (Decision Tree) 25.2691%.

Fig 4.2 Surgical Timing Data set - SpreadSubsampling



	Bayesian Network	NaiveBaye s	LibSVM	KNN (IBK)	J48 (Decision tree)	Random Forest
■ Non-rebalancing	74.3167%	74.2581%	77.8309%	76.1226%	71.0269%	75.5662%
■ SpreadSubsample 1	77.8862%	77.8049%	73.8753%	75.3252%	74.9864%	81.7615%
■ SpreadSubsample 3	74.3850%	74.2483%	77.9774%	76.0543%	71.0269%	84.4494%
■ SpreadSubsample 5	74.3850%	74.2483%	77.9774%	76.0543%	71.0269%	84.4494%

On the Surgical timing data set, we found that the Random Forest classifier has the best performance, increasing from 75.5662% to 84.4494%, an increase of 8.8832%. However, the performance of the other five classifiers is not so good. For example, the accuracy of J48 (Decision Tree) on the non-rebalanced data set is only 71.0269%, but after adjusting the parameters, it increased to 74.9564% on the SpreadSubsample1 data set; it then dropped back to 71.0269%. We can see that other classifiers except Random Forest also had this problem.

4.2 Summary

In Summary, this experiment was carried out to ascertain whether oversampling and undersampling techniques including SMOTE, Resampling and SpreadSubsampling achieved highest performance of the classifiers. We found that the use of SMOTE and Resampling methods significantly improved the accuracy of the classifiers. They are reliable sampling methods that can be used for chronic disease data sets. However, SpredSubsampling has a great impact on the performance of the classifier, resulting in unstable accuracy.

Furthermore, we found that the accuracy of the classifier is different on multiple data sets. We had to adjust the parameters of the rebalancing method, but different data sets gave us different results, some greatly improved, some decreased. This once again shows that a multiple data set is very important for the evaluation of classifiers.

Of the classifiers, we found that Random Forest has the best stability and accuracy on the four data sets. It even reached the highest accuracy recorded (99.6584%) on the data set of ALF Resample0.4.

Therefore, we recommend using strategies that include these classifiers or sampling techniques to build a risk assessment model for chronic disease data sets.

Chapter 5

Validation

Introduction

Model validation is the process of predicting the credibility of results. Validation can ascertain whether the predicted value of the model can accurately predict the model. Validation techniques include fitting degree analysis, model calibration and checking the accuracy of observed and predicted values (Blischke, & Murthy, 2011). It is important to obtain a high degree of certainty when developing models for diagnostic or predictive purposes (Polyzotis, Zinkevich, Roy, Breck, & Whang, 2019).

We described in Chapter 3 that this experiment used 70% of the data for training and 30% of data for testing, so we can minimize the impact of data differences and better understand the characteristics of the model. Therefore, we selected the optimal three classifiers of the four data sets, put them on a testing set with the same adjustments and compared the accuracy obtained to find the classifier with the smallest error. The second way was to validate the ROC curve performance of the top three classifiers.

5.1 Classifiers Performance Validation

5.1.1 UCI Data Set

Fig 4.3 UCI Top 3 Classifiers

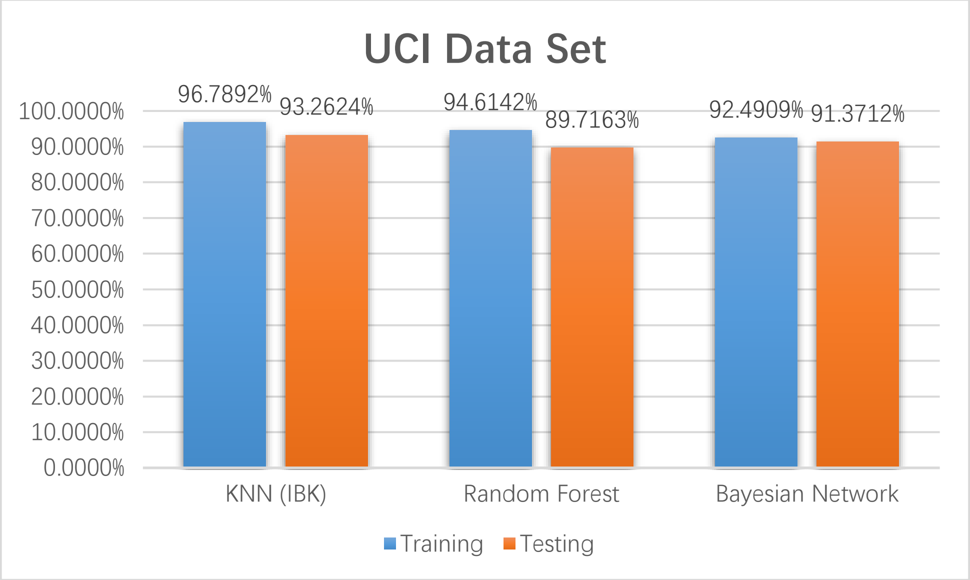


Table 2.5 UCI Data set Training

UCI Data set Training						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
KNN (IBK)	SMOTE 300	96.7892%	0.968	0.968	0.967	0.971
Random Forest	SMOTE 300	94.6142%	0.946	0.946	0.945	0.992
Bayesian Network	SMOTE 300	92.4909%	0.925	0.925	0.925	0.966

Table 2.6 UCI Data set Testing

UCI Data set Testing						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
KNN (IBK)	SMOTE 300	93.2624%	0.933	0.933	0.930	0.932
Random Forest	SMOTE 300	89.7163%	0.896	0.897	0.891	0.969
Bayesian Network	SMOTE 300	91.3712%	0.918	0.914	0.915	0.962

In the UCI data set, we selected three optimal classifiers from the training set. They were KNN (IBK) 96.7892% Random Forest 94.6142% and Bayesian Network 92.4909%. These were the results produced in the SMOTE 300 method. We can see that the accuracy is very high. Then, in Table 2.6, we can see that the accuracy of these three classifiers in the testing set dropped significantly. KNN (IBK) dropped by 3.527%, Bayesian Network dropped by 1.119%, and Random Forest dropped by 4.898%. So in this data set we conclude that the Bayesian Network classifier is the most stable and has the least error.

5.1.2 FHS Data Set

Fig 4.4 FHS Top 3 Classifiers

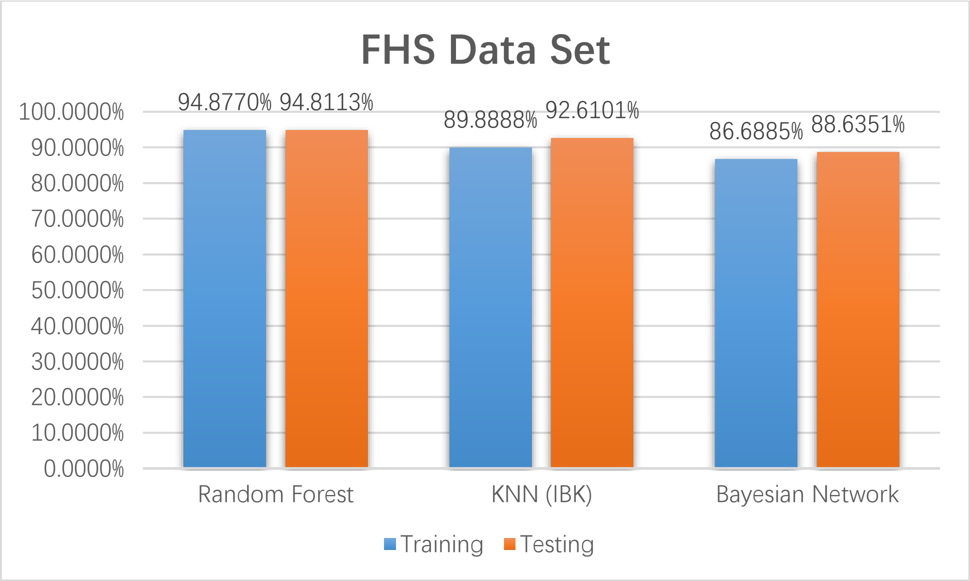


Table 2.7 FHS Data set Training

FHS Data set Training						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.8	94.8770%	0.949	0.949	0.949	0.986
KNN (IBK)	Resample 0.8	89.8888%	0.906	0.899	0.803	0.947
Bayesian Network	SMOTE 300	86.6885%	0.866	0.867	0.866	0.920

Table 2.8 FHS Data set Testing

FHS Data set Testing						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.8	94.8113%	0.951	0.948	0.943	0.919
KNN (IBK)	Resample 0.8	92.6101%	0.923	0.926	0.924	0.863
Bayesian Network	SMOTE 300	88.6351%	0.886	0.886	0.886	0.936

In the FHS data set, our results in the training set are Random Forest 94.8770% obtained in the parameter resample 0.8, KNN (IBK) 89.8888% obtained in the parameter resample 0.8, and Bayesian Network 86.6885% obtained in the parameter SMOTE. We selected these three categories for the testing set, and the results (Table 2.8) show that Random Forest decreased by 0.065% to 94.8113%, KNN (IBK) increased by 2.721% to 92.6101%, and Bayesian Network increased by 1.946%. The results of these three classifiers in the training set and the testing set have relatively few errors. Particularly, Random Forest has a minimum error of only 0.065%. Therefore, Random Forest is the optimal classifier in the FHS data set.

5.1.3 ALF Data Set

Fig 4.5 ALF Top 3 Classifiers

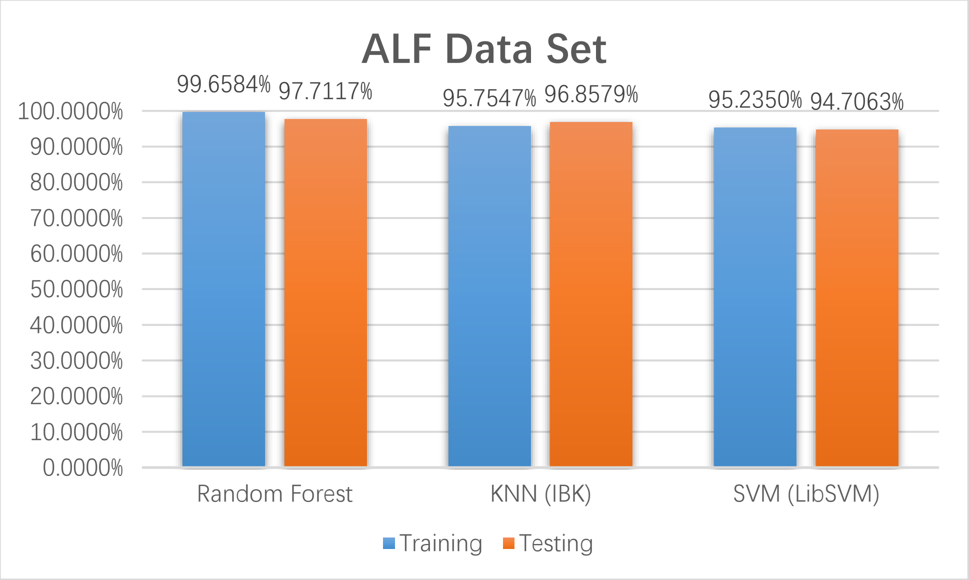


Table 2.9 ALF Data set Training

ALF Data set Training						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.4	99.6584%	0.997	0.997	0.99	0.998
KNN (IBK)	Resample 0.4	95.7547%	0.963	0.958	0.959	0.985
SVM (LibSVM)	Non-Rebalancing	95.2350%	0.952	0.952	0.952	0.500

Table 3.1 ALF Data set Testing

ALF Data set Testing						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.4	97.7117%	0.978	0.977	0.974	0.978
KNN (IBK)	Resample 0.4	96.8579%	0.966	0.969	0.967	0.957
SVM (LibSVM)	Non-Rebalancing	94.7063%	0.947	0.973	0.500	0.900

In the ALF data set, we found that Random Forest achieved the highest 99.6584% obtained in the parameter resample 0.4, KNN, (IBK) reached 95.7545% in the parameter resample 0.4, and for SVM (LibSVM), 95.2350% was achieved in the unbalanced data set. The performance of these three classifiers in the training set is very good so let's look at their performances in the testing set. Random Forest 97.7117% is 1.947% lower than its result in the training set, KNN (IBK) 96.8579% is 1.103% higher than its result in the training set, and SVM (LibSVM) is 0.529% lower than its result in the training set. Therefore, we found that with the lowest error, SVM (LibSVM) is the best classifier in the ALF data set.

5.1.4 Surgical Timing Data Set

Fig 4.6 Surgical Timing Top 3 Classifiers

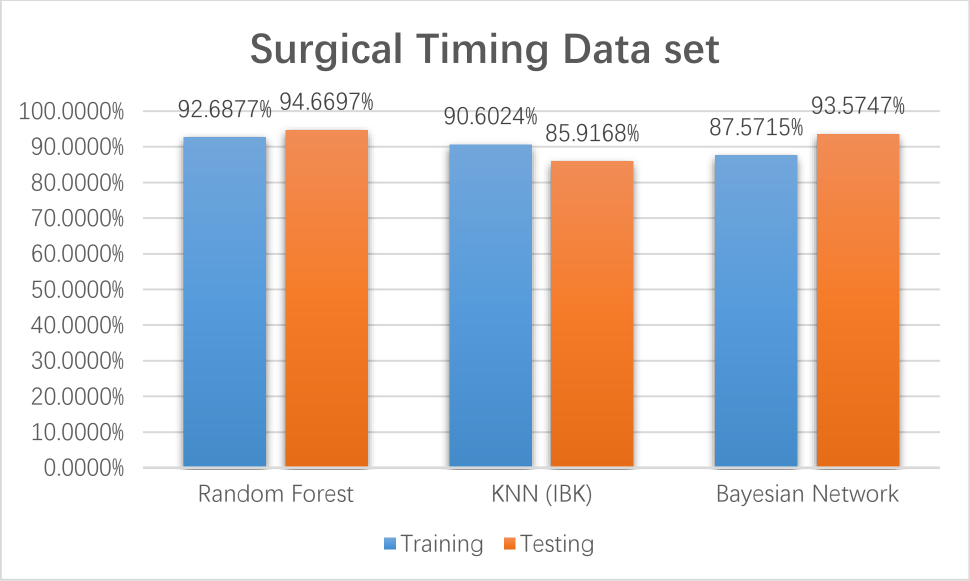


Table 3.2 Surgical Timing Data set Training

Surgical Timing Data set Training						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.4	92.6877%	0.928	0.927	0.927	0.985
KNN (IBK)	SMOTE 300	90.6024%	0.909	0.906	0.903	0.914
Bayesian Network	SMOTE 300	87.5715%	0.876	0.876	0.871	0.954

Table 3.3 Surgical Timing Data set Testing

Surgical Timing Data set Testing						
Classifier	Rebalance method	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.4	94.6697%	0.949	0.947	0.942	0.991
KNN (IBK)	SMOTE 300	85.9168%	0.879	0.859	0.860	0.933
Bayesian Network	SMOTE 300	93.5747%	0.937	0.936	0.936	0.984

In the Surgical Timing data set, we found that the optimal classifiers are Random Forest 92.6877% obtained in the parameter Resample 0.4, KNN (IBK) 90.6024% obtained in the SMOTE 300 parameter, and Bayesian Network 87.5715% in the SMOTE 300 parameter. The result obtained from the testing set is that Random Forest 94.6697% is 1.982% higher than in the training set, KNN(IBK) 85.9168% is 4.687% lower than in the training set, and Bayesian Network 93.5747% is 6.003% higher than in the training set. From this comparison, we found that the Random Forest classifier has the lowest error so is the best classifier in the Surgical Timing data set.

5.2 ROC Curve Validation

We selected the optimal classifiers obtained in the previous step to verify through the ROC curve. Our goal was to finally find the best classifier. The method of ROC curve verification is that the area under the curve represents the test accuracy. If the area under the curve is large, the test is more accurate.

For our experiment, the diagnostic value is low when the area under the ROC curve is between 0.5 and 0.7;

The diagnostic value is moderate when it is between 0.7 and 0.9;

The diagnostic value is higher when it is above 0.9.

The following Table 3.4 shows the top 3 optimal classifiers we have selected from the four data sets.

Table 3.4 Top three classifiers

Classifier	Rebalance method	Data Set	Accuracy	Precision	Recall	F-Measure	ROC Area
Random Forest	Resample 0.8	FHS	94.8770%	0.949	0.949	0.949	0.986
Bayesian Network	SMOTE 300	UCI	92.4909%	0.925	0.925	0.925	0.966
SVM (LibSVM)	Non-Rebalancing	ALF	95.2350%	0.952	0.952	0.952	0.500

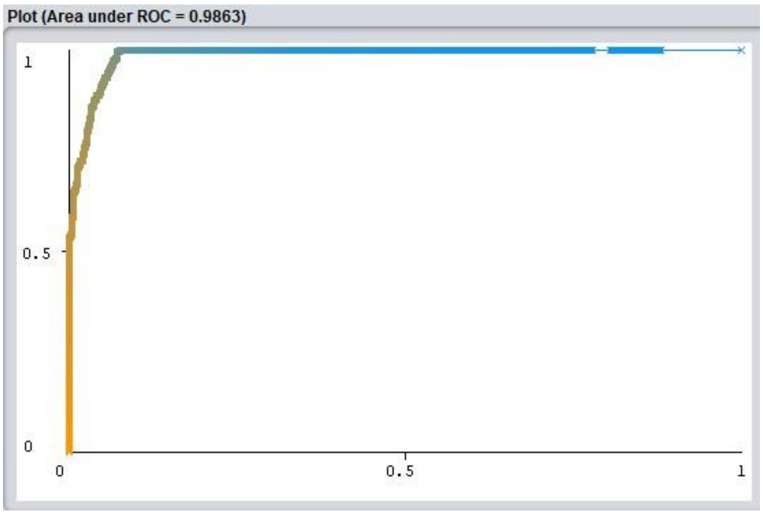
5.2.1 Random Forest (FHS Data Set)

ROC Curve

Random Forest Resample 0.8 94.8770%

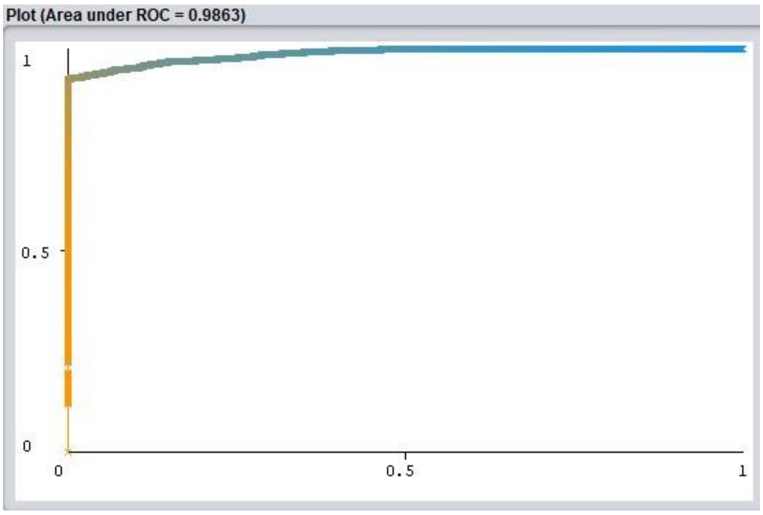
ROC curve for Random Forest classifier with area of curve = 0.9863 (minority class)

Fig 4.7 ROC curve for Random Forest (minority class)



ROC curve for Random Forest classifier with area of curve = 0.9863 (majority class)

Fig 4.8 ROC curve for Random Forest (majority class)



We found that Random Forest classifier has an area of curve of 0.9863 in both minority and majority classes on the Roc Curve. This result is very good and as we mentioned previously, when it is above 0.9 the diagnostic value is higher.

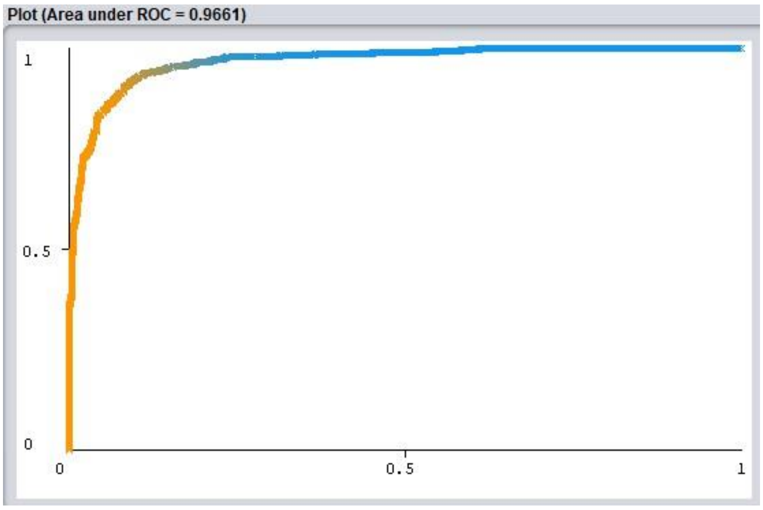
5.2.2 Bayesian Network (UCI Data Set)

ROC Curve

Bayesian Network SMOTE 300 92.4909%

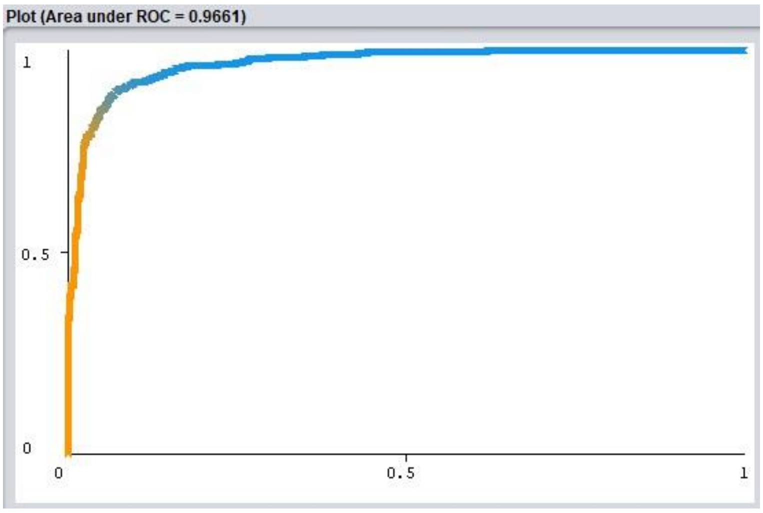
ROC curve for Bayesian Network classifier with area of curve = 0.9661 (minority Class)

Fig 4.9 ROC curve for Bayesian Network (minority class)



ROC curve for Bayesian Network classifier with area of curve = 0.9661 (majority class)

Fig 5.1 ROC curve for Bayesian Network (majority class)



The Roc Curve of Bayesian Network is 0.9661 in both minority and majority Classes. This verification shows that this model is very reliable.

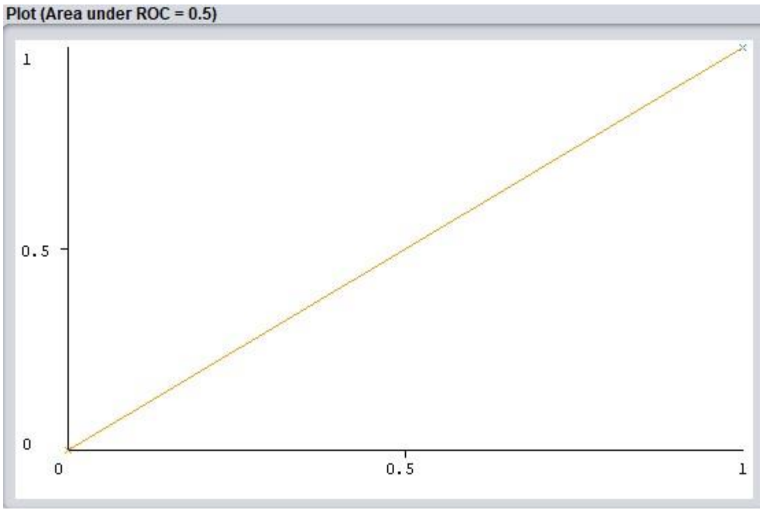
5.2.3 SVM (LibSVM) (ALF Data Set)

ROC Curve

SVM (LibSVM) Non-Rebalancing 95.2350%

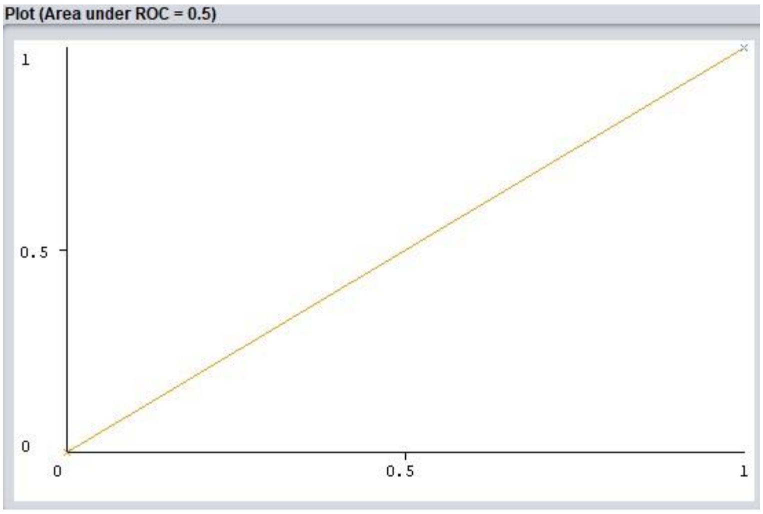
ROC curve for SVM (LibSVM) classifier with area of curve = 0.5 (minority class)

Fig 5.2 ROC curve for SVM (LibSVM) (minority class)



ROC curve for SVM (LibSVM) classifier with area of curve = 0.5 (majority class)

Fig 5.3 ROC curve for SVM (LibSVM) (majority class)



The Roc Curve of SVM (LibSVM) is only 0.5 in both minority and majority classes. We also pointed out previously that the diagnostic value is only moderate when it is between 0.7 and 0.9 so the performance of this model is not very good.

5.3 Summary

In summary, different experiments were conducted. The combination of various algorithms and processing methods constitutes the experimental study. In this study, different experimental combinations T= A * P were processed to select the best classifier. The values of different combinations are: A (algorithm) uses 6 classifiers; P (parameter) uses the SMOTE, Resample and SpreadSubsample of parameter combinations.

We selected the top three classifiers from the four data sets and placed them in a test set for training with the same adjustments, and compared the accuracy of the obtained results. The classifier with the lowest error is the optimal classifier. Then we screened and used Roc Curve to confirm that Random Forest (RF) as found by the resample rebalancing method (RF-RESAMPLE) is the optimal classifier.

Chapter 6

Conclusion

This thesis undertook analysis research on the performance of resampling, oversampling, and undersampling techniques based on SMOTE, Resample, and Spreadsubsample methods in the diagnosis of chronic diseases. Using four chronic disease data sets, six different classifiers were evaluated and compared. Data sets are obtained by evaluating the performance of all different combinations of classifiers and rebalancing techniques. In addition, all the strategies were trained under different conditions, resulting in 72 different combinations (Table 3.5). By comparing these combinations, a more robust strategy was determined.

Table 3.5 72 different combinations

Six classifiers	Three rebalancing Method	Four Data sets
SVM	SMOTE	UCI data set
Naïve Bayes	Resampling	FHS data set
KNN	SpreadSubsampling	ALF data set
Random Forest		Surgery Timing data set
Bayesian Network		
Decision Tree		

The findings of this research restate the importance of techniques designed to help handle highly unbalanced data, such as SMOTE, Resample, and Spreadsubsample. The medical data is very complex, data set is extremely unbalanced especially in the case of disease and non-disease, so these techniques can be used to significantly improve the performance of the classifier.

Furthermore, we trained and tested the performance of six classifiers and discovered the Random Forest classifier can get the best results. Therefore, the conclusion is that

Random Forest (RF) combined with Resample rebalancing method can achieve good results in the classification of chronic diseases. From the results, we can see that the RF-RESAMPLE classifier is the most effective classifier to predict chronic diseases based on the four data sets. On the other hand, it can be concluded that Resample method can improve the performance of the classifier by reducing the class imbalance.

6.1 Contributions

The main contribution of this research is our analyses' of the classifiers and rebalance methods, and finding the best classifier to predict chronic diseases risk. This is very encouraging. The findings of this thesis are significant in at least four main areas.

- We compared the performance of six classifiers; Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbor (KNN), Random Forest, Bayesian Network and J48 (Decision Tree). This has value for machine learning researchers to better understand the performance of the six classifiers.

- In machine learning, different data sets have varying effects on classifier performance. For this reason, we used four data sets to analyze and compare the classifiers and find the optimal classifier. In the end, we found that Random Forest performed best and most effectively of the four data sets.

- The analysis generated by this research is useful for machine learning, class imbalance and chronic diseases. The conclusion of this research confirms the importance of SMOTE, Resample and Spreadsubsample in dealing with highly unbalanced data. Especially in the case of chronic diseases, there are many data attributes and class imbalance. These techniques can be used to significantly improve the performance of the classifier in accurately distinguishing diseases from non-diseases.

- Basically, the purpose of establishing a risk prediction model for chronic diseases should be to help medical researchers and doctors make correct judgments on patients, Therefore, an important development of the risk prediction model is to find the optimal classifier and rebalance method to help determine the most stable strategy. So we hope this research will be used for reference by doctors and medical researchers, ultimately eliminating the suffering caused by chronic diseases to patients.

6.2 Limitations

The limitations of our research need to be recognized.

- The data sets chosen for this thesis are mainly from Europe and India and different ethnicities can have different results. Therefore, the effectiveness of chronic disease risk prediction models should be evaluated in multi ethnic groups.

- The data sets used are open data sets, we did not actually collect the data in the hospital, so we question the applicability of the data collection. Furthermore, there was no feedback from these patients, which may reduce the performance of the results of this research.

- We have compared six classification algorithms and found the best classifier. More classification algorithms can provide better diagnosis for chronic diseases. In addition, the model needs to be tested by doctors, especially chronic disease experts, before it can be applied to hospitals.

- As for the selection of analysis tools, we found in the experimental stage that Weka tool's analysis speed slows down in the face of a large amount of data. And some effective algorithm packages are not available yet. Therefore, more efficient and comprehensive data analysis tools are needed for data analysis.

- For the research of rebalancing methods, we only compared SMOTE, Resample and Spreadsubsample. Using more rebalancing methods for comparison could help to find the best rebalancing methods.

6.3 Future Work

In order to avoid the prejudice caused by the limited scale of chronic disease issues, future research should expand the data scale for data analysis. More data frames need to be included and pre-processed. The possible methods are as follows:

- Collect large data sets of habits and medical, meteorological, environmental data to help identify the causes of chronic diseases

- Can collaborate with a hospital to collect data from patients, obtain feedback from patients and conduct follow-up study on the condition of patients, so as to ensure the integrity of data.

- Considering the large expansion of SMOTE, Resample and Spreadsubsample, it is necessary to create a more systematic framework to help determine which methods are more suitable for chronic disease data sets and the basic characteristics of data.

- Future work can be done to understand why the Resample - Random Forest is superior to other strategies.

- More classifiers could be added for comparison to find the best risk prediction model for chronic diseases.

- The research methods of the thesis can be applied to other disease surveillance or other scientific research fields. The analytical procedures used in this thesis are fully applicable to other areas of disease surveillance and scientific research, and provide a solid theoretical basis for the analysis of the association between different diseases, which could be of great benefit to disease surveillance and prevention, and needs further research in the future.

References

Abdar, M., Kalhori, S. R. N., Sutikno, T., Subroto, I. M. I., & Arji, G. (2015). Comparing Performance of Data Mining Algorithms in Prediction Heart Diseases. International Journal of Electrical & Computer Engineering (2088-8708), 5(6).

Abdoh, S. F., Rizka, M. A., & Maghraby, F. A. (2018). Cervical cancer diagnosis using random forest classifier with SMOTE and feature reduction techniques. IEEE Access, 6, 59475-59485.

Aman Ajmera, Framingham Heart study dataset [online dataset]. Kaggle Inc; Publicadoy actualizado Nov 2017. URL< https://www.kaggle.com/ amanajmera1/framingham-heart-study-dataset >(accessed 9 Mar 2020), Version 1.

Apostolopoulos, I. D. (2020). Investigating the Synthetic Minority class Oversampling Technique (SMOTE) on an imbalanced cardiovascular disease (CVD) dataset. arXiv preprint arXiv:2004.04101.

Avinash, Navlavi. (2018). *Decision Tree Classification in Python*. Retrieved from https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Bashir, S., Qamar, U., & Khan, F. H. (2016). A multicriteria weighted vote-based classifier ensemble for heart disease prediction. Computational Intelligence, 32(4), 615-645.

Bayes, F. R. S. (1958). An essay towards solving a problem in the doctrine of chances. Biometrika, 45(3-4), 296-315.

Beunza, J. J., Puertas, E., García-Ovejero, E., Villalba, G., Condes, E., Koleva, G., ... & Landecho, M. F. (2019). Comparison of machine learning algorithms for clinical event prediction (risk of coronary heart disease). Journal of biomedical informatics, 97, 103257.

Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, 3(6).

Bhatla, N., & Jyoti, K. (2012). An analysis of heart disease prediction using different data mining techniques. International Journal of Engineering, 1(8), 1-4.

Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

Blischke, W. R., & Murthy, D. P. (2011). Reliability: modeling, prediction, and optimization (Vol. 767). John Wiley & Sons.

Bloomfield, A. (2017). Health and Independence Report 2017.

Bravo-Escobar, R., González-Represas, A., Gómez-González, A. M., Montiel-Trujillo, A., Aguilar-Jimenez, R., Carrasco-Ruíz, R., & Salinas-Sánchez, P. (2017). Effectiveness and safety of a home-based cardiac rehabilitation programme of mixed surveillance in patients with ischemic heart disease at moderate cardiovascular risk: A randomised, controlled clinical trial. BMC cardiovascular disorders, 17(1), 66.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

Brownlee, J. (2019). Impact of dataset size on deep learning model skill and performance estimates. Machine Learning Mastery, 6.

Cadwell, B. L., Boyle, J. P., Tierney, E. F., & Thompson, T. J. (2007). A Bayesian approach to assess heart disease mortality among persons with diabetes in the presence of missing data. Health care management science, 10(3), 231-238.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3), 1-27.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Chen, J. H., & Asch, S. M. (2017). Machine learning and prediction in medicine—beyond the peak of inflated expectations. The New England journal of medicine, 376(26), 2507.

CHENG, C. H., & WANG, Y. C. A NOVEL MULTI-COMBINED METHOD FOR HANDLING MEDICAL DATASET WITH IMBALANCED CLASSES PROBLEM.

Cover, T. (1968). Estimation by the nearest neighbor rule. IEEE Transactions on Information Theory, 14(1), 50-55.

Dailey, S. L., & Zhu, Y. (2017). Communicating health at work: Organizational wellness programs as identity bridges. Health Communication, 32(3), 261-268.

Dale, L. P., Whittaker, R., Jiang, Y., Stewart, R., Rolleston, A., & Maddison, R. (2015). Text message and internet support for coronary heart disease self-management: results from the Text4Heart randomized controlled trial. Journal of medical Internet research, 17(10), e237.

Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient kNN classification algorithm for big data. Neurocomputing, 195, 143-148.

Dharmarajan, K. Prediction of Chronic Kidney Disease using Classification techniques.

Drajati, D. P. (2016). Perbandingan Teknik Undersampling dan Oversampling pada

Klasifikasi Data Pasien Diabetes Mellitus (Dm) Dengan Menggunakan Algoritma Naive Bayes Classifier (NBC).

Edmonds, W. A., & Kennedy, T. D. (2016). *An applied guide to research designs: Quantitative, qualitative, and mixed methods*. Sage Publicat

Eibe Frank, Mark A. Hall & Ian H. Witten (2016). The WEKA Workbench. https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

Esmaily, H., Tayefi, M., Doosti, H., Ghayour-Mobarhan, M., Nezami, H., & Amirabadizadeh, A. (2018). A comparison between decision tree and random forest in determining the risk factors associated with type 2 diabetes. Journal of research in health sciences, 18(2), 412.

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal, 2(1), 602-609.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

Ferguson, T., Rowlands, A. V., Olds, T., & Maher, C. (2015). The validity of consumer-level, activity monitors in healthy adults worn in free-living conditions: a cross-sectional study. International Journal of Behavioral Nutrition and Physical Activity, 12(1), 42.

Flick, U. (2018). *An introduction to qualitative research*. Sage Publications Limited.

Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. H. (2004). Data mining in bioinformatics using Weka. Bioinformatics, 20(15), 2479-2481.

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. Machine learning, 29(2-3), 131-163.

Gagnon, M. P., Ngangue, P., Payne-Gagnon, J., & Desmartis, M. (2016). m-Health adoption by healthcare professionals: a systematic review. Journal of the American Medical Informatics Association, 23(1), 212-220.

Github. (2020). *Random Forest*. Retrieved from https://dinhanhthi.com/random-forest/

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg.

Hachesu, P. R., Ahmadi, M., Alizadeh, S. & Sadoughi, F. (2013). Use of data mining techniques to determine and predict length of stay of cardiac patients. Healthcare informatics research, 19(2), 121–129.

Haider, R., Hyun, K., Cheung, N. W., Redfern, J., Thiagalingam, A., & Chow, C. K. (2019). Effect of lifestyle focused text messaging on risk factor modification in patients with diabetes and coronary heart disease: A sub-analysis of the TEXT ME study. Diabetes research and clinical practice, 153, 184-190.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878-887). Springer, Berlin, Heidelberg.

Ibarguren, I., Pérez, J. M., Muguerza, J., Gurrutxaga, I., & Arbelaitz, O. (2015). Coverage-based resampling: Building robust consolidated decision trees. *Knowledge-Based Systems*, *79*, 51-67.

Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University, 37.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent data analysis, 6(5), 429-449.

Jiang, Y., Jiao, N., Nguyen, H. D., Lopez, V., Wu, V. X., Kowitlawakul, Y., ... & Wang, W. (2019). Effect of a mHealth programme on coronary heart disease prevention among working population in Singapore: A single group pretest–post-test design. Journal of advanced nursing, 75(9), 1922-1932.

Jie, Xu. (2018). Research on Medical Health Classification Based on Machine Learning. Retrieved from http://cdmd.cnki.com.cn/Article/CDMD-10459-1018109787.htm

Junior, J. R. F., Koenigkam-Santos, M., Cipriano, F. E. G., Fabro, A. T., & de Azevedo-Marques, P. M. (2018). Radiomics-based features for pattern recognition of lung cancer histopathology and metastases. Computer methods and programs in biomedicine, 159, 23-30.

Kauw, D., Koole, M. A., Winter, M. M., Dohmen, D. A., Tulevski, I. I., Blok, S., ... &

Mulder, B. J. (2019). Advantages of mobile health in the management of adult patients with congenital heart disease. International journal of medical informatics, 132, 104011.

Kawasaki, R. (2016). Development of health parameter model for risk prediction of cvd using svm. Computational and mathematical methods in medicine, 2016.

Khadija, M. A., & Setiawan, N. A. Detecting Liver Disease Diagnosis by Combining SMOTE, Information Gain Attribute Evaluation and Ranker. ITSMART: Jurnal Teknologi dan Informasi, 9(1), 13-17.

Khemphila, A., & Boonjing, V. (2010, October). Comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients. In 2010 international conference on computer information systems and industrial management applications (CISIM) (pp. 193-198). IEEE.

Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232.

Krishnani, D., Kumari, A., Dewangan, A., Singh, A., & Naik, N. S. (2019, October). Prediction of coronary heart disease using supervised machine learning algorithms. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (pp. 367-372). IEEE.

Kumari, M. & Godara, S. (2011). Comparative study of data mining classification methods in cardiovascular disease prediction 1.

Kumari, M., Tiwari, N., Subbarao, N., & Chandra, S. (2017). Evaluation of predictive models based on random forest, decision tree and support vector machine classifiers and virtual screening of anti-mycobacterial compounds. International Journal of Computational Biology and Drug Design, 10(3), 248-263.

Kumari, S., & Singh, A. (2013, January). A data mining approach for the diagnosis of diabetes mellitus. In *2013 7th International Conference on Intelligent Systems and Control (ISCO)* (pp. 373-375). IEEE.

Kusiak, A., Dixon, B., & Shah, S. (2005). Predicting survival time for kidney dialysis patients: a data mining approach. Computers in biology and medicine, 35(4), 311-327.

Lakshmanarao, A., Swathi, Y., & Sundareswar, P. S. S. (2019). Machine learning techniques for heart disease prediction. Forest, 95(99), 97.

Li, Z., Wang, Y., & Yue, B. (2005). Advertisement Image Detection

Liu, Z., Wang, Y., Liu, X., Du, Y., Tang, Z., Wang, K., ... & Tian, J. (2018). Radiomics analysis allows for precise prediction of epilepsy in patients with low-grade gliomas. NeuroImage: Clinical, 19, 271-278.

Long, N. C., Meesad, P., & Unger, H. (2015). A highly accurate firefly based algorithm for heart disease prediction. Expert Systems with Applications, 42(21), 8221-8231.

Lopes, E. L., Beaton, A. Z., Nascimento, B. R., Tompsett, A., Dos Santos, J. P., Perlman, L., ... & Bonisson, L. (2018). Telehealth solutions to enable global collaboration in rheumatic heart disease screening. Journal of telemedicine and telecare, 24(2), 101-109.

Mahesh Babu Mariappan, Surgery Timing dataset [online dataset]. Kaggle Inc; Publicadoy actualizado Dec 2018. URL< https://www.kaggle.com/omnamahshivai/surgical-dataset-binary-classification >(accessed 17 May 2020), Version 1.

Manu Siddhartha, Heart Disease Dataset (Comprehensive) [online dataset]. Kaggle Inc; Publicadoy actualizado Dec 2019. URL< https://www.kaggle.com/sid321axn/heart-statlog-cleveland-hungary-final/>(accessed 3 April 2020), Version 1.

Mariscal, G., Marban, O., & Fernandez, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review, 25(2), 137.

MELLO, R. F., & Ponti, M. A. (2018). Machine Learning: A Practical Approach on the Statistical Learning Theory. Springer.

Meltzer LJ, Hiruma LS, Avis K, Montgomery-Downs H, Valentin J. Comparison of a commercial accelerometer with polysomnography and actigraphy in children and adolescents. Sleep. 2015;38(8):1323–30.

Minichiello, Victor, Rosalie Aroni, and Victor Minichiello. *In-depth interviewing: Researching people*. Longman Cheshire, 1990.

Mirza, S., Mittal, S., & Zaman, M. (2018). Decision Support Predictive model for prognosis of diabetes using SMOTE and Decision tree. International Journal of Applied Engineering Research, 13(11), 9277-9282.

Mishra, S., Mallick, P. K., Jena, L., & Chae, G. S. (2020). Optimization of Skewed Data Using Sampling-Based Preprocessing Approach. Frontiers in Public Health, 8.

Mohapatra, S. K., & Mohanty, M. N. (2018, September). Analysis of resampling method for arrhythmia classification using random forest classifier with selected features. In 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA) (pp. 495-499). IEEE.

Motka, R., Parmarl, V., Kumar, B., & Verma, A. R. (2013, September). Diabetes mellitus forecast using different data mining techniques. In *2013 4th International Conference on Computer and Communication Technology (ICCCT)* (pp. 99-103). IEEE.

Narain, R., Saxena, S., & Goyal, A. K. (2016). Cardiovascular risk prediction: a comparative study of Framingham and quantum neural network based approach. Patient preference and adherence, 10, 1259.

O'Meara, G. (2019). Mining and Classifying Images from an Advertisement Image Remover. Annals of Data Science, 6(2), 279-303

Oliver, Pickup. (2018). Machine-learning will revolutionise heart health. Retrieved from https://www.raconteur.net/healthcare/cardiovascular-health-2018/machine-learning-heart-health

Osborne, J. W. (Ed.). (2008). Best practices in quantitative methods. Sage.

Özçift, A. (2011). Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis. Computers in biology and medicine, 41(5), 265-271.

Özdemir, A., Yavuz, U., & Dael, F. A. (2019). Performance evaluation of different classification techniques using different datasets. International Journal of Electrical and Computer Engineering, 9(5), 3584.

Pandey, S. K., & Janghel, R. R. (2019). Automatic detection of arrhythmia from imbalanced ECG database using CNN model with SMOTE. Australasian physical & engineering sciences in medicine, 42(4), 1129-1139.

Pearl, J. (1985, August). Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA, USA (pp. 15-17).

Polyzotis, N., Zinkevich, M., Roy, S., Breck, E., & Whang, S. (2019) . Data validation for machine learning. Proceedings of Machine Learning and Systems, 1, 334-347.

Pooja, S. R. (2013). A comparative study of instance reduction techniques. In Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM.

Potharaju, S. P., & Sreedevi, M. (2016). An improved prediction of kidney disease using SMOTE. Indian Journal of Science and Technology, 9(31), 1-7.

Qaisar, S. M., & Subasi, A. (2018, July). An adaptive rate ECG acquisition and analysis for efficient diagnosis of the cardiovascular diseases. In 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP) (pp. 177-181). IEEE.

Qian, Yun, Yanchun Liang, Mu Li, Guoxiang Feng, and Xiaohu Shi. "A resampling ensemble algorithm for classification of imbalance problems." *Neurocomputing* 143 (2014): 57-67.

Qiu, Longfei, Keke Gai, and Meikang Qiu. "Optimal big data sharing approach for tele-health in cloud computing." 2016 IEEE International Conference on Smart Cloud (SmartCloud). IEEE, 2016.

Qu, X., Bai, L., & You, H. (2012). Fast Processing in Support Vector Machine for Large-Scaled Data Set. In Green Communications and Networks (pp. 171-176). Springer, Dordrecht.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Rahul Kumar, Acute Live Failure dataset [online dataset]. Kaggle Inc; Publicadoy actualizado Dec 2018. URL< https://www.kaggle.com/rahul121/acute-liver-failure>(accessed 17 April 2020).

Rajendran, K., Jayabalan, M., & Thiruchelvam, V. Predicting Breast Cancer via Supervised Machine Learning Methods on Class Imbalanced Data.

Rao, R. R., & Makkithaya, K. (2017). Learning from a class imbalanced public health dataset: A cost-based comparison of classifier performance. International Journal of Electrical and Computer Engineering, 7(4), 2215.

Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

Rohit, Walimbe. (2017). Handling imbalanced dataset in supervised learning using family of SMOTE algorithm. Retrieved from https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family

Rokach, L., & Maimon, O. (2005). Decision trees. In Data mining and knowledge discovery handbook (pp. 165-192). Springer, Boston, MA.

Sarang, Narkhede. (2018) *Understanding AUC - ROC Curve*. Retrieved from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Saravananathan, K., & Velmurugan, T. (2016). Analyzing diabetic data using classification algorithms in data mining. Indian Journal of Science and Technology, 9(43), 1-6.

Sasaki, J. E., Hickey, A., Mavilia, M., Tedesco, J., John, D., Keadle, S. K., & Freedson, P. S. (2015). Validation of the Fitbit wireless activity tracker for prediction of energy expenditure. Journal of Physical Activity and Health, 12(2), 149-154.

Scuse, D., & Reutemann, P. (2007). Weka experimenter tutorial for version 3-5-5. *University of Waikato*.

Shafique, U., Majeed, F., Qaiser, H., & Mustafa, I. U. (2015). Data mining in healthcare for heart diseases. International Journal of Innovation and Applied Studies, 10(4), 1312.

Sherwood, L. (2015). Human physiology: from cells to systems. Cengage learning.

Shinde, A., Kale, S., Samant, R., Naik, A., & Ghorpade, S. (2017). Heart Disease Prediction System using Multilayered Feed Forward Neural Network and Back Propagation Neural Network. International Journal of Computer Applications, 166(7), 32-36.

Shuja, M., Mittal, S., & Zaman, M. (2020). Effective prediction of type ii diabetes mellitus using data mining classifiers and SMOTE. In *Advances in Computing and Intelligent Systems* (pp. 195-211). Springer, Singapore.

Simpson, L. A., Eng, J. J., Klassen, T. D., Lim, S. B., Louie, D. R., Parappilly, B., ... & Zbogar, D. (2015). Capturing step counts at slow walking speeds in older adults: comparison of ankle and waist placement of measuring device. Journal of rehabilitation medicine, 47(9), 830-835.

Stephenson, T. A. (2000). An introduction to Bayesian network theory and usage (No. REP_WORK). IDIAP.

Storm, F. A., Heller, B. W., & Mazzà, C. (2015). Step detection and activity recognition accuracy of seven physical activity monitors. PloS one, 10(3).

Subbalakshmi, G., Ramesh, K., & Rao, M. C. (2011). Decision support in heart disease prediction system using naive bayes. Indian Journal of Computer Science and Engineering (IJCSE), 2(2), 170-176.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, *23*(04), 687-719.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). Introduction to data mining, Pearson education. *Inc., New Delhi*.

Tayefi, M., Tajfard, M., Saffar, S., Hanachi, P., Amirabadizadeh, A. R., Esmaeily, H., ... & Ghayour-Mobarhan, M. (2017). hs-CRP is strongly associated with coronary heart disease (CHD): A data mining approach using decision tree algorithm. Computer methods and programs in biomedicine, 141, 105-109.

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC medical informatics and decision making, 19(1), 1-16.

Unnikrishnan, P., Kumar, D. K., Poosapadi Arjunan, S., Kumar, H., Mitchell, P. &

Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. International Journal of Innovative Science, Engineering & Technology, 2(9), 441-444.

Vijayan, V. V., & Anjali, C. (2015, April). Decision support systems for predicting diabetes mellitus—A review. In 2015 Global Conference on Communication Technologies (GCCT) (pp. 98-103). IEEE.

Vijayan, V., & Ravikumar, A. (2014). Study of data mining algorithms for prediction and diagnosis of diabetes mellitus. *International journal of computer applications*, *95*(17).

Voss, C., Gardner, R. F., Dean, P. H., & Harris, K. C. (2017). Validity of commercial activity trackers in children with congenital heart disease. Canadian Journal of Cardiology, 33(6), 799-805.

Wan, X., Liu, J., Cheung, W. K., & Tong, T. (2014). Learning to improve medical decision making from imbalanced data without a priori cost. *BMC medical informatics and decision making*, *14*(1), 111.

Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. Encyclopedia of machine learning, 15, 713-714.

Wiharto, W., Kusnanto, H., & Herianto, H. (2016). Interpretation of clinical data based on C4. 5 algorithm for the diagnosis of coronary heart disease. Healthcare informatics research, 22(3), 186-195.

William, Malsam.(2018). *What Does Y=f(x) Mean? How to Use This Powerful Six Sigma Formula*. Retrueved from https://www.projectmanager.com/blog/yfx-six-sigma-formula

Word Health Organization. (2020). *The top 10 causes of death*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

World Health Organization. (2014). Global status report on noncommunicable diseases 2014 (No. WHO/NMH/NVI/15.1). World Health Organization.

Yang, P., Xu, L., Zhou, B. B., Zhang, Z., & Zomaya, A. Y. (2009, December). A particle swarm based hybrid system for imbalanced medical data sampling. In *BMC genomics* (Vol. 10, No. S3, p. S34). BioMed Central.

Yildirim, P. (2017, July). Chronic kidney disease prediction on imbalanced data by multilayer perceptron: Chronic kidney disease prediction. In 2017 IEEE 41st Annual

Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 193-198). IEEE

Yue, Qian (2018). Research on the diagnosis of heart disease based on data mining technology. Retrieved from http://cdmd.cnki.com.cn/Article/CDMD-10708-1018204606.htm

Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M. M., & Alamri, A. (2015). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. IEEE Systems Journal, 11(1), 88-95.

Zheng, X. (2020). SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction (Doctoral dissertation, UCLA).

Zhihua Zhou, (2016) *Machine Learning*. Retrieved from Qing hua da xue chu ban she.

Appendix A

Abbreviations

CVD Cardiovascular Disease

NCD non-communicable disease

CHD Congenital heart disease

T2DM Type 2 Diabetes Mellitus

SMS short message service

CAD coronary artery disease

WHO World Health Organization

HBP high blood pressure

HF heart failure

AHCD Adult Congenital Heart Disease

BP blood pressure

BMI body mass index

ICC intra-class correlation coefficient

EE energy consumption

TST total sleep time

SVM support vector machine

FT function tree

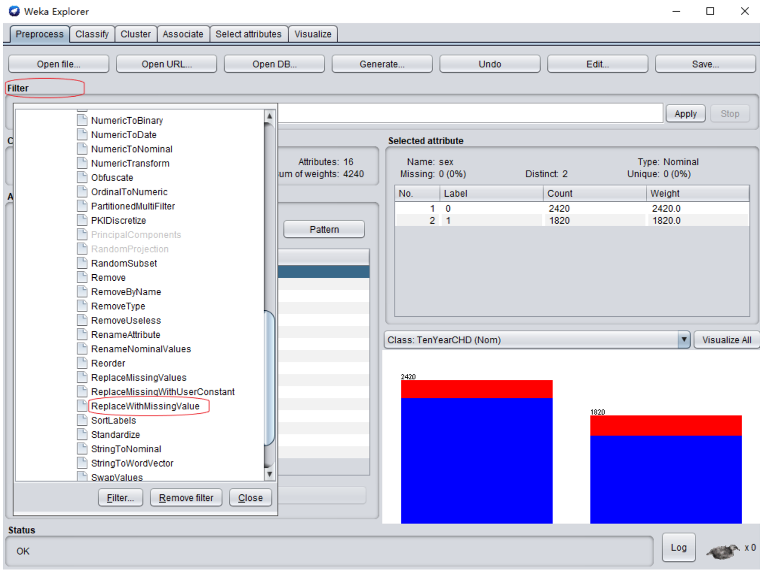
KNN k nearest neighbor

ROC Curve Receiver Operating Characteristic Curve

Weka Waikato Environment for Knowledge Analysis

Appendix B

Missing value replacement



NumericToBinary
NumericToDate
NumericToNominal
NumericTransform
Obfuscate
OrdinalToNumeric
PartitionedMultiFilter
PKIDiscretize
PrincipalComponents
RandomProjection
RandomSubset
Remove
RemoveByName
RemoveType
RemoveUseless
RenameAttribute
RenameNominalValues
Reorder
ReplaceMissingValues
ReplaceMissingWithUserConstant
ReplaceWithMissingValue
SortLabels
Standardize
StringToNominal
StringToWordVector
SwapValues

Attributes: 16
um of weights: 4240

Selected attribute

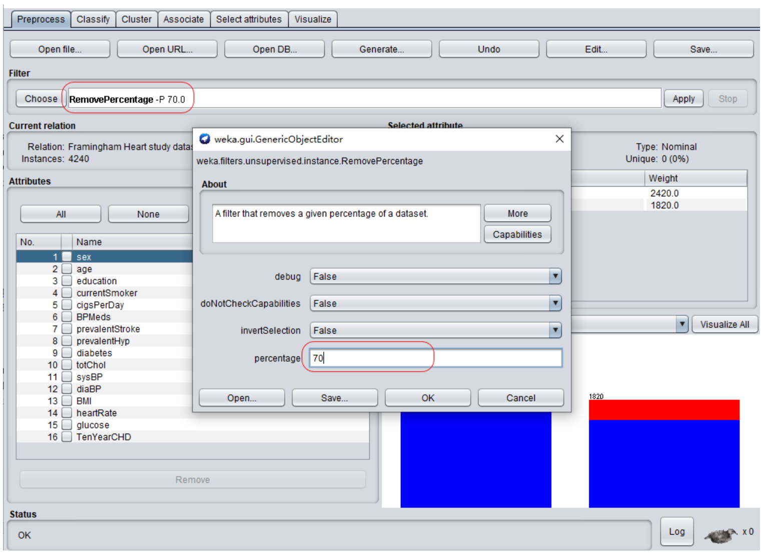
Name: sex
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	0	2420	2420.0
2	1	1820	1820.0

Class: TenYearCHD (Nom)

Status
OK

Data set Splitting



Filter
RemovePercentage -P 70.0

Current relation
Relation: Framingham Heart study data
Instances: 4240

Attributes

No.	Name
1	sex
2	age
3	education
4	currentSmoker
5	cigsPerDay
6	BPMeds
7	prevalentStroke
8	prevalentHyp
9	diabetes
10	totChol
11	sysBP
12	diaBP
13	BMI
14	heartRate
15	glucose
16	TenYearCHD

Selected attribute
Type: Nominal
Unique: 0 (0%)

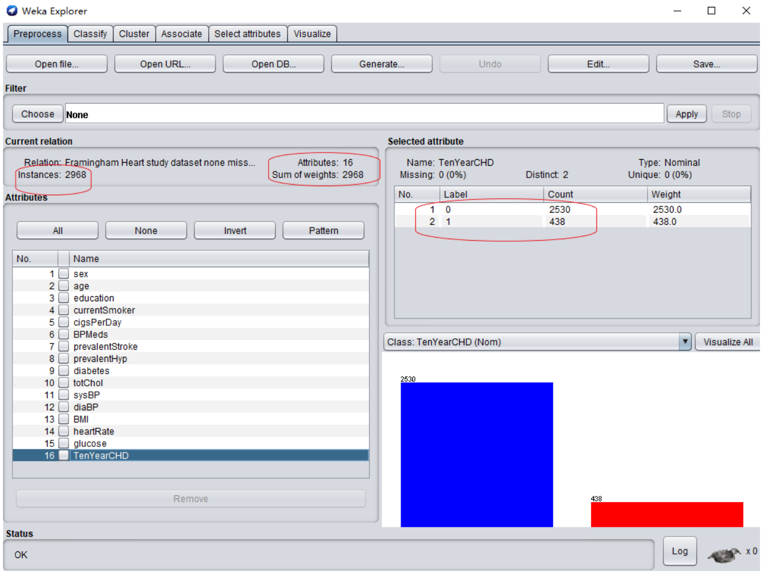
Weight
2420.0
1820.0

weka.gui.GenericObjectEditor
weka.filters.unsupervised.instance.RemovePercentage

About
A filter that removes a given percentage of a dataset.

debug: False
doNotCheckCapabilities: False
invertSelection: False
percentage: 70

Status
OK



Filter

None

Current relation

Relation: Framingham Heart study dataset none miss...
Instances: 2968
Attributes: 16
Sum of weights: 2968

Selected attribute

Name: TenYearCHD
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	0	2530	2530.0
2	1	438	438.0

Attributes

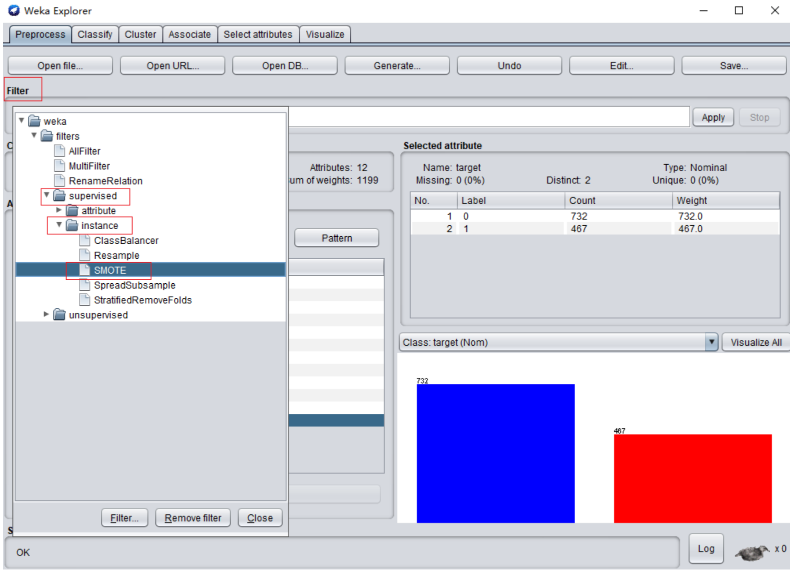
No.	Name
1	sex
2	age
3	education
4	currentSmoker
5	cigsPerDay
6	BPMeds
7	prevalentStroke
8	prevalentHyp
9	diabetes
10	totChol
11	sysBP
12	diaBP
13	BMI
14	heartRate
15	glucose
16	TenYearCHD

Class: TenYearCHD (Nom)

Status

OK

Data Rebalancing With SMOTE:



Filter

- weka
  - filters
    - AllFilter
    - MultiFilter
    - RenameRelation
    - supervised
      - attribute
      - instance
        - ClassBalancer
        - Resample
        - SMOTE
        - SpreadSubsample
        - StratifiedRemoveFolds
    - unsupervised

Attributes: 12
um of weights: 1199

Selected attribute

Name: target
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

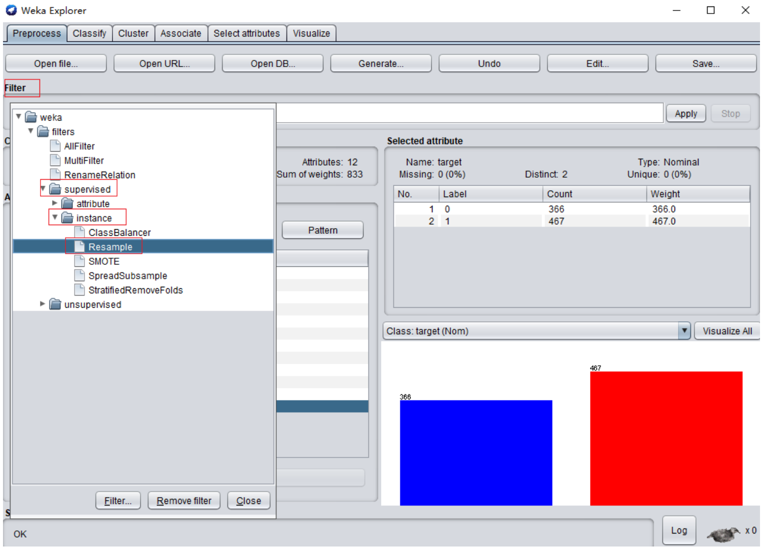
No.	Label	Count	Weight
1	0	732	732.0
2	1	467	467.0

Class: target (Nom)

OK

Filter – Supervised – Instance - SMOTE

Data Rebalancing With Resample:



Attributes: 12
Sum of weights: 833

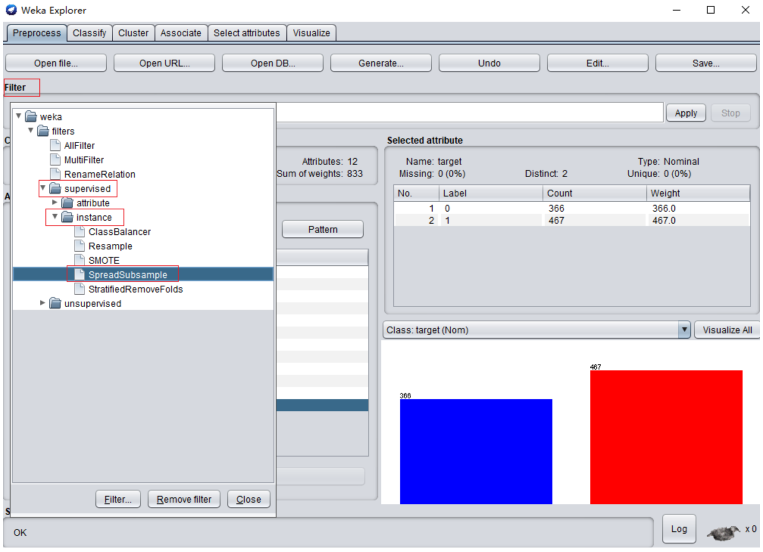
Selected attribute

Name: target
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	0	366	366.0
2	1	467	467.0

Filter – Supervised – Instance - Resample

Data Rebalancing With SpreadSubsample:



Attributes: 12
Sum of weights: 833

Selected attribute

Name: target
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

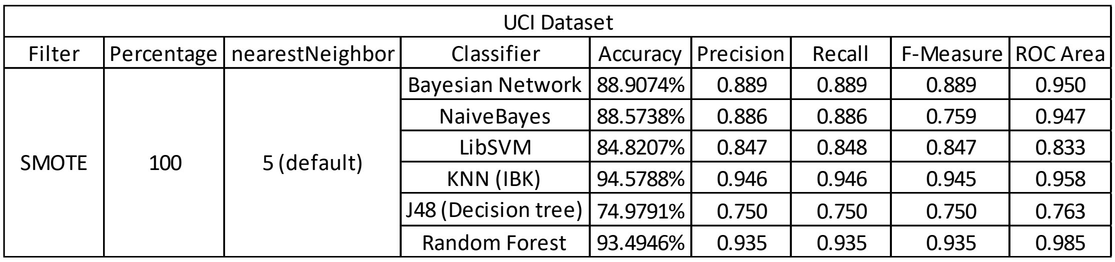
No.	Label	Count	Weight
1	0	366	366.0
2	1	467	467.0

Filter – Supervised – Instance – SpreadSubsample

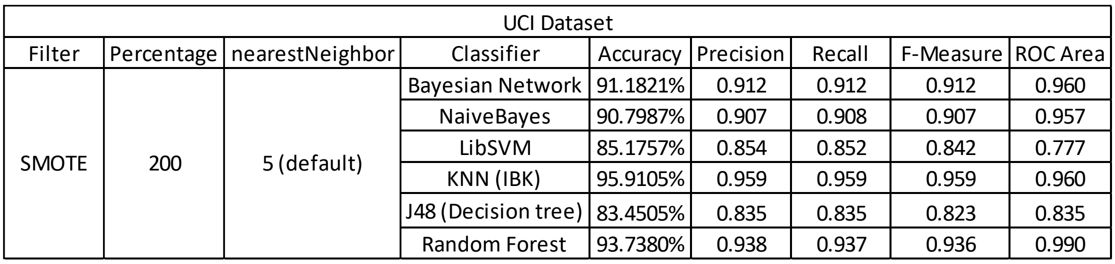
Appendix C

SMOTE

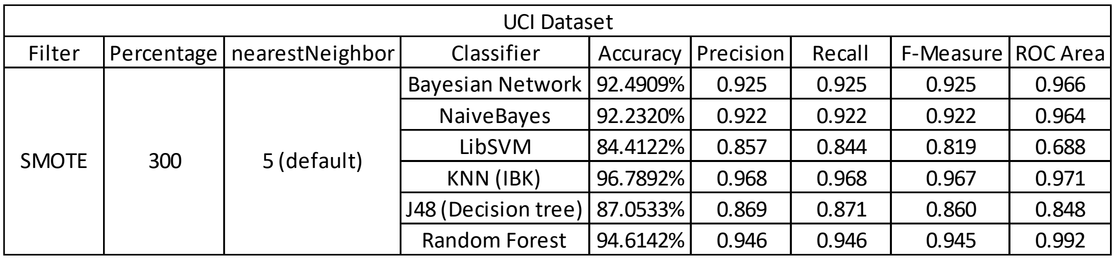
UCI Data set



UCI Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	100	5 (default)	Bayesian Network	88.9074%	0.889	0.889	0.889	0.950
			NaiveBayes	88.5738%	0.886	0.886	0.759	0.947
			LibSVM	84.8207%	0.847	0.848	0.847	0.833
			KNN (IBK)	94.5788%	0.946	0.946	0.945	0.958
			J48 (Decision tree)	74.9791%	0.750	0.750	0.750	0.763
			Random Forest	93.4946%	0.935	0.935	0.935	0.985

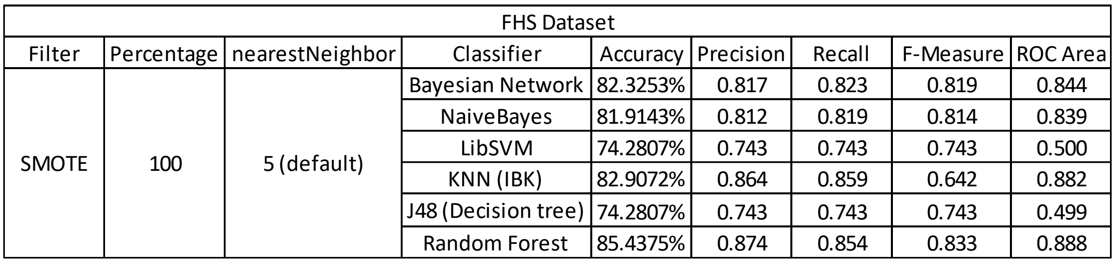


UCI Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	200	5 (default)	Bayesian Network	91.1821%	0.912	0.912	0.912	0.960
			NaiveBayes	90.7987%	0.907	0.908	0.907	0.957
			LibSVM	85.1757%	0.854	0.852	0.842	0.777
			KNN (IBK)	95.9105%	0.959	0.959	0.959	0.960
			J48 (Decision tree)	83.4505%	0.835	0.835	0.823	0.835
			Random Forest	93.7380%	0.938	0.937	0.936	0.990



UCI Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	300	5 (default)	Bayesian Network	92.4909%	0.925	0.925	0.925	0.966
			NaiveBayes	92.2320%	0.922	0.922	0.922	0.964
			LibSVM	84.4122%	0.857	0.844	0.819	0.688
			KNN (IBK)	96.7892%	0.968	0.968	0.967	0.971
			J48 (Decision tree)	87.0533%	0.869	0.871	0.860	0.848
			Random Forest	94.6142%	0.946	0.946	0.945	0.992

FHS Data set



FHS Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	100	5 (default)	Bayesian Network	82.3253%	0.817	0.823	0.819	0.844
			NaiveBayes	81.9143%	0.812	0.819	0.814	0.839
			LibSVM	74.2807%	0.743	0.743	0.743	0.500
			KNN (IBK)	82.9072%	0.864	0.859	0.642	0.882
			J48 (Decision tree)	74.2807%	0.743	0.743	0.743	0.499
			Random Forest	85.4375%	0.874	0.854	0.833	0.888

FHS Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	200	5 (default)	Bayesian Network	85.1977%	0.850	0.852	0.851	0.895
			NaiveBayes	84.6514%	0.845	0.847	0.845	0.893
			LibSVM	65.8169%	0.658	0.658	0.658	0.500
			KNN (IBK)	86.3944%	0.876	0.864	0.866	0.911
			J48 (Decision tree)	74.6098%	0.738	0.746	0.737	0.770
			Random Forest	90.4006%	0.911	0.904	0.900	0.927

FHS Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	300	5 (default)	Bayesian Network	86.6885%	0.866	0.867	0.866	0.920
			NaiveBayes	86.2447%	0.862	0.862	0.862	0.919
			LibSVM	64.3624%	0.768	0.644	0.549	0.565
			KNN (IBK)	85.9178%	0.874	0.859	0.860	0.933
			J48 (Decision tree)	77.0668%	0.771	0.771	0.771	0.831
			Random Forest	91.0089%	0.913	0.910	0.909	0.944

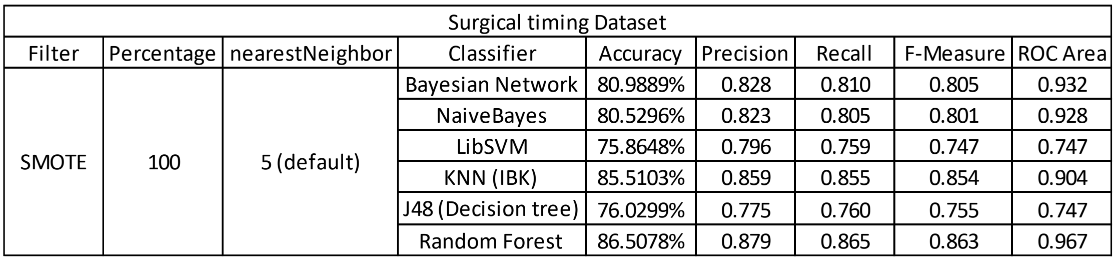
ALF Data set

ALF Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	100	5 (default)	Bayesian Network	93.1543%	0.939	0.932	0.935	0.934
			NaiveBayes	92.6576%	0.936	0.927	0.930	0.931
			LibSVM	90.9034%	0.909	0.909	0.909	0.500
			KNN (IBK)	93.6045%	0.945	0.936	0.940	0.914
			J48 (Decision tree)	90.9034%	0.909	0.909	0.909	0.498
			Random Forest	94.7687%	0.951	0.948	0.938	0.931

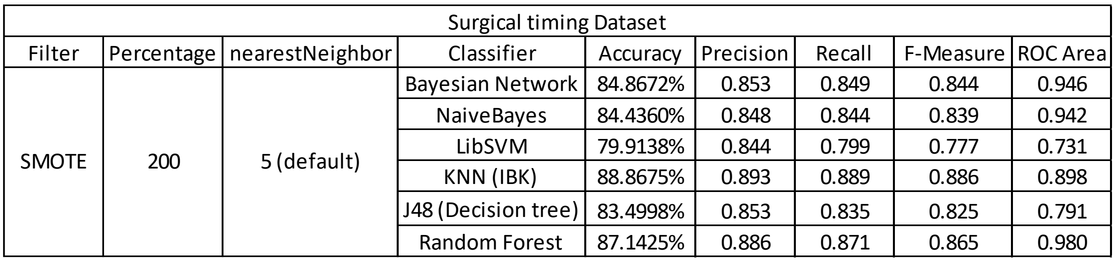
ALF Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	200	5 (default)	Bayesian Network	94.9133%	0.945	0.943	0.944	0.959
			NaiveBayes	93.9085%	0.942	0.938	0.939	0.957
			LibSVM	86.9488%	0.869	0.869	0.869	0.500
			KNN (IBK)	92.5167%	0.941	0.925	0.930	0.936
			J48 (Decision tree)	86.9488%	0.869	0.869	0.869	0.499
			Random Forest	96.2732%	0.964	0.963	0.960	0.957

ALF Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	300	5 (default)	Bayesian Network	94.8776%	0.949	0.949	0.949	0.969
			NaiveBayes	94.5931%	0.947	0.946	0.946	0.969
			LibSVM	83.3228%	0.833	0.833	0.833	0.500
			KNN (IBK)	91.8896%	0.936	0.919	0.923	0.950
			J48 (Decision tree)	87.2368%	0.859	0.872	0.860	0.792
			Random Forest	96.5851%	0.967	0.966	0.964	0.969

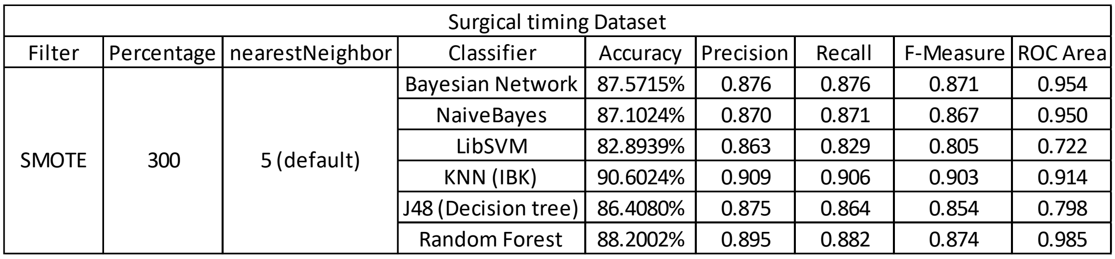
Surgical Timing Data set



Surgical timing Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	100	5 (default)	Bayesian Network	80.9889%	0.828	0.810	0.805	0.932
			NaiveBayes	80.5296%	0.823	0.805	0.801	0.928
			LibSVM	75.8648%	0.796	0.759	0.747	0.747
			KNN (IBK)	85.5103%	0.859	0.855	0.854	0.904
			J48 (Decision tree)	76.0299%	0.775	0.760	0.755	0.747
			Random Forest	86.5078%	0.879	0.865	0.863	0.967



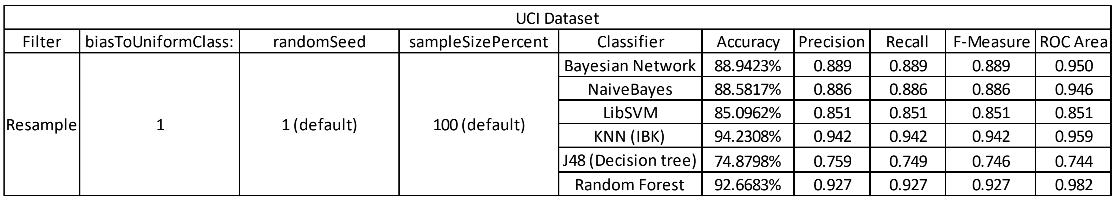
Surgical timing Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	200	5 (default)	Bayesian Network	84.8672%	0.853	0.849	0.844	0.946
			NaiveBayes	84.4360%	0.848	0.844	0.839	0.942
			LibSVM	79.9138%	0.844	0.799	0.777	0.731
			KNN (IBK)	88.8675%	0.893	0.889	0.886	0.898
			J48 (Decision tree)	83.4998%	0.853	0.835	0.825	0.791
			Random Forest	87.1425%	0.886	0.871	0.865	0.980



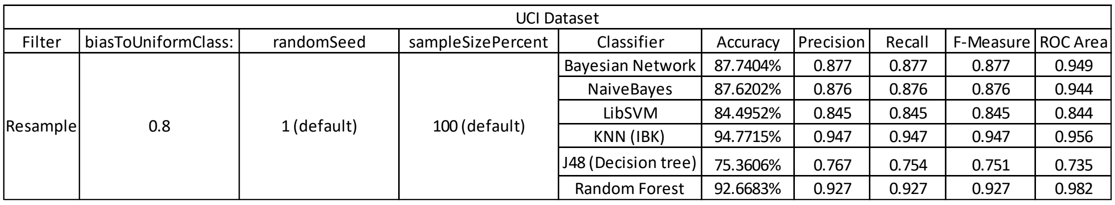
Surgical timing Dataset								
Filter	Percentage	nearestNeighbor	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SMOTE	300	5 (default)	Bayesian Network	87.5715%	0.876	0.876	0.871	0.954
			NaiveBayes	87.1024%	0.870	0.871	0.867	0.950
			LibSVM	82.8939%	0.863	0.829	0.805	0.722
			KNN (IBK)	90.6024%	0.909	0.906	0.903	0.914
			J48 (Decision tree)	86.4080%	0.875	0.864	0.854	0.798
			Random Forest	88.2002%	0.895	0.882	0.874	0.985

Resampling

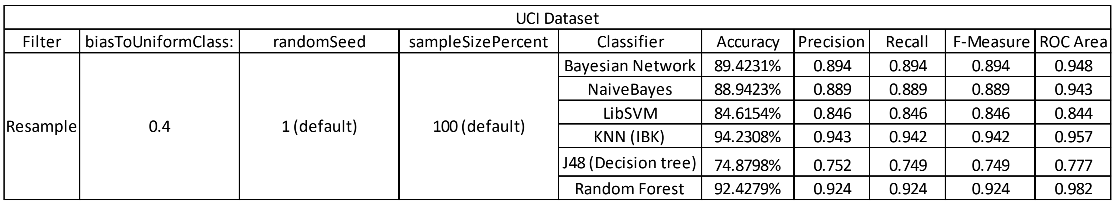
UCI Data set



UCI Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	1	1 (default)	100 (default)	Bayesian Network	88.9423%	0.889	0.889	0.889	0.950
				NaiveBayes	88.5817%	0.886	0.886	0.886	0.946
				LibSVM	85.0962%	0.851	0.851	0.851	0.851
				KNN (IBK)	94.2308%	0.942	0.942	0.942	0.959
				J48 (Decision tree)	74.8798%	0.759	0.749	0.746	0.744
				Random Forest	92.6683%	0.927	0.927	0.927	0.982

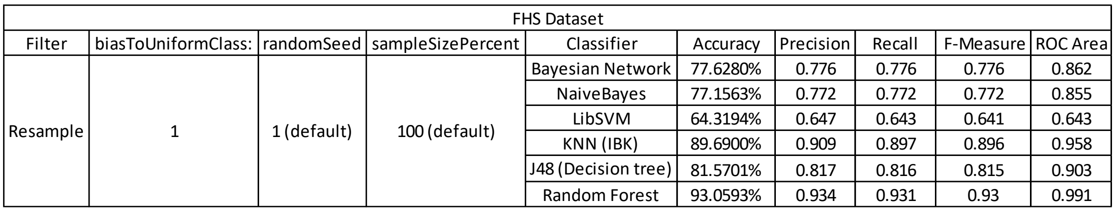


UCI Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.8	1 (default)	100 (default)	Bayesian Network	87.7404%	0.877	0.877	0.877	0.949
				NaiveBayes	87.6202%	0.876	0.876	0.876	0.944
				LibSVM	84.4952%	0.845	0.845	0.845	0.844
				KNN (IBK)	94.7715%	0.947	0.947	0.947	0.956
				J48 (Decision tree)	75.3606%	0.767	0.754	0.751	0.735
				Random Forest	92.6683%	0.927	0.927	0.927	0.982

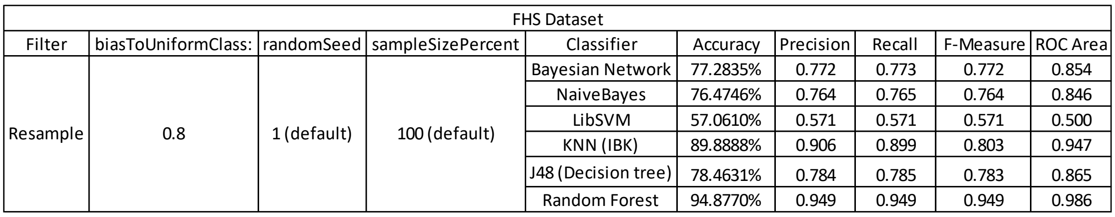


UCI Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.4	1 (default)	100 (default)	Bayesian Network	89.4231%	0.894	0.894	0.894	0.948
				NaiveBayes	88.9423%	0.889	0.889	0.889	0.943
				LibSVM	84.6154%	0.846	0.846	0.846	0.844
				KNN (IBK)	94.2308%	0.943	0.942	0.942	0.957
				J48 (Decision tree)	74.8798%	0.752	0.749	0.749	0.777
				Random Forest	92.4279%	0.924	0.924	0.924	0.982

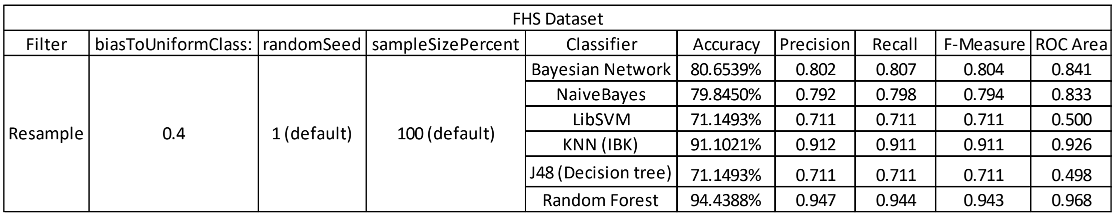
FHS Data set



FHS Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	1	1 (default)	100 (default)	Bayesian Network	77.6280%	0.776	0.776	0.776	0.862
				NaiveBayes	77.1563%	0.772	0.772	0.772	0.855
				LibSVM	64.3194%	0.647	0.643	0.641	0.643
				KNN (IBK)	89.6900%	0.909	0.897	0.896	0.958
				J48 (Decision tree)	81.5701%	0.817	0.816	0.815	0.903
				Random Forest	93.0593%	0.934	0.931	0.93	0.991

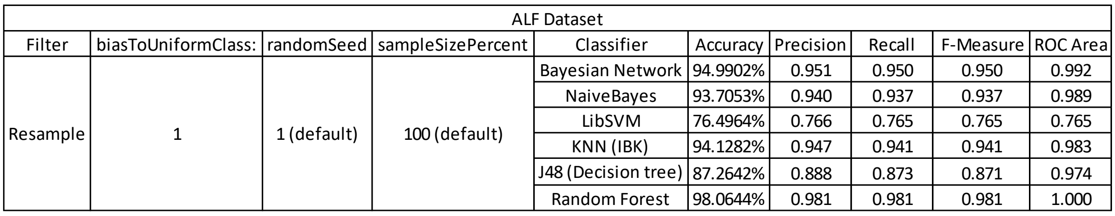


FHS Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.8	1 (default)	100 (default)	Bayesian Network	77.2835%	0.772	0.773	0.772	0.854
				NaiveBayes	76.4746%	0.764	0.765	0.764	0.846
				LibSVM	57.0610%	0.571	0.571	0.571	0.500
				KNN (IBK)	89.8888%	0.906	0.899	0.803	0.947
				J48 (Decision tree)	78.4631%	0.784	0.785	0.783	0.865
				Random Forest	94.8770%	0.949	0.949	0.949	0.986

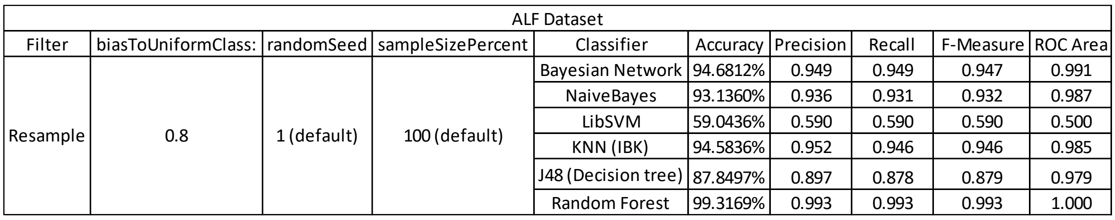


FHS Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.4	1 (default)	100 (default)	Bayesian Network	80.6539%	0.802	0.807	0.804	0.841
				NaiveBayes	79.8450%	0.792	0.798	0.794	0.833
				LibSVM	71.1493%	0.711	0.711	0.711	0.500
				KNN (IBK)	91.1021%	0.912	0.911	0.911	0.926
				J48 (Decision tree)	71.1493%	0.711	0.711	0.711	0.498
				Random Forest	94.4388%	0.947	0.944	0.943	0.968

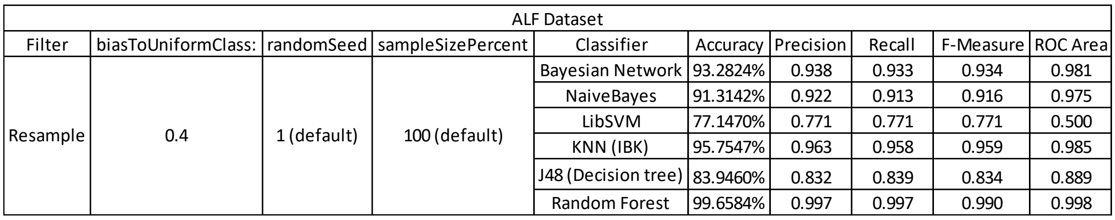
ALF Data set



ALF Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	1	1 (default)	100 (default)	Bayesian Network	94.9902%	0.951	0.950	0.950	0.992
				NaiveBayes	93.7053%	0.940	0.937	0.937	0.989
				LibSVM	76.4964%	0.766	0.765	0.765	0.765
				KNN (IBK)	94.1282%	0.947	0.941	0.941	0.983
				J48 (Decision tree)	87.2642%	0.888	0.873	0.871	0.974
				Random Forest	98.0644%	0.981	0.981	0.981	1.000

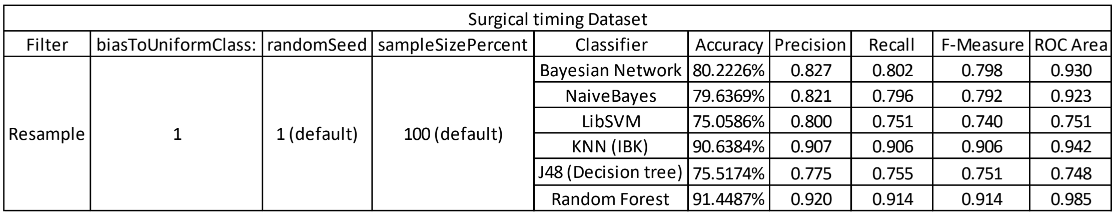


ALF Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.8	1 (default)	100 (default)	Bayesian Network	94.6812%	0.949	0.949	0.947	0.991
				NaiveBayes	93.1360%	0.936	0.931	0.932	0.987
				LibSVM	59.0436%	0.590	0.590	0.590	0.500
				KNN (IBK)	94.5836%	0.952	0.946	0.946	0.985
				J48 (Decision tree)	87.8497%	0.897	0.878	0.879	0.979
				Random Forest	99.3169%	0.993	0.993	0.993	1.000

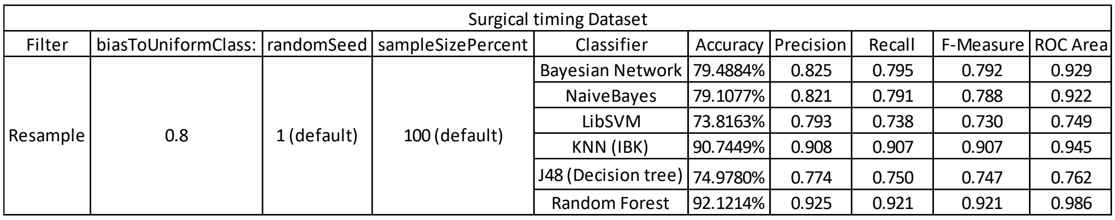


ALF Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.4	1 (default)	100 (default)	Bayesian Network	93.2824%	0.938	0.933	0.934	0.981
				NaiveBayes	91.3142%	0.922	0.913	0.916	0.975
				LibSVM	77.1470%	0.771	0.771	0.771	0.500
				KNN (IBK)	95.7547%	0.963	0.958	0.959	0.985
				J48 (Decision tree)	83.9460%	0.832	0.839	0.834	0.889
				Random Forest	99.6584%	0.997	0.997	0.990	0.998

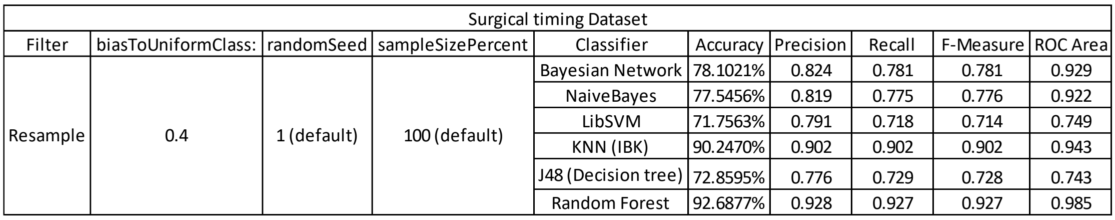
Surgical Timing Data set



Surgical timing Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	1	1 (default)	100 (default)	Bayesian Network	80.2226%	0.827	0.802	0.798	0.930
				NaiveBayes	79.6369%	0.821	0.796	0.792	0.923
				LibSVM	75.0586%	0.800	0.751	0.740	0.751
				KNN (IBK)	90.6384%	0.907	0.906	0.906	0.942
				J48 (Decision tree)	75.5174%	0.775	0.755	0.751	0.748
				Random Forest	91.4487%	0.920	0.914	0.914	0.985



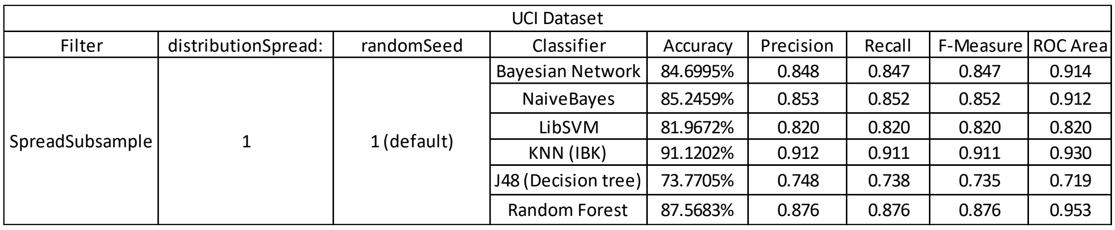
Surgical timing Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.8	1 (default)	100 (default)	Bayesian Network	79.4884%	0.825	0.795	0.792	0.929
				NaiveBayes	79.1077%	0.821	0.791	0.788	0.922
				LibSVM	73.8163%	0.793	0.738	0.730	0.749
				KNN (IBK)	90.7449%	0.908	0.907	0.907	0.945
				J48 (Decision tree)	74.9780%	0.774	0.750	0.747	0.762
				Random Forest	92.1214%	0.925	0.921	0.921	0.986



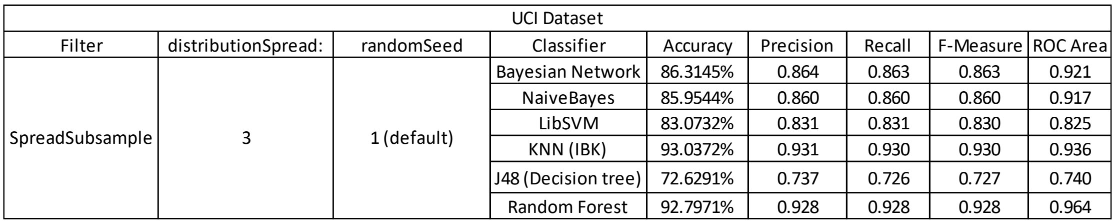
Surgical timing Dataset									
Filter	biasToUniformClass:	randomSeed	sampleSizePercent	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
Resample	0.4	1 (default)	100 (default)	Bayesian Network	78.1021%	0.824	0.781	0.781	0.929
				NaiveBayes	77.5456%	0.819	0.775	0.776	0.922
				LibSVM	71.7563%	0.791	0.718	0.714	0.749
				KNN (IBK)	90.2470%	0.902	0.902	0.902	0.943
				J48 (Decision tree)	72.8595%	0.776	0.729	0.728	0.743
				Random Forest	92.6877%	0.928	0.927	0.927	0.985

SpreadSubsampling

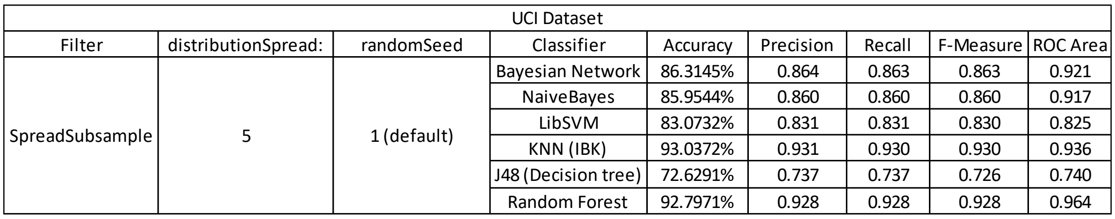
UCI data set



UCI Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	1	1 (default)	Bayesian Network	84.6995%	0.848	0.847	0.847	0.914
			NaiveBayes	85.2459%	0.853	0.852	0.852	0.912
			LibSVM	81.9672%	0.820	0.820	0.820	0.820
			KNN (IBK)	91.1202%	0.912	0.911	0.911	0.930
			J48 (Decision tree)	73.7705%	0.748	0.738	0.735	0.719
			Random Forest	87.5683%	0.876	0.876	0.876	0.953

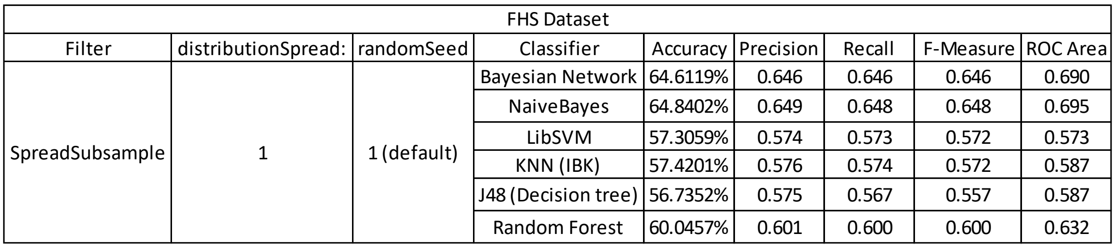


UCI Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	3	1 (default)	Bayesian Network	86.3145%	0.864	0.863	0.863	0.921
			NaiveBayes	85.9544%	0.860	0.860	0.860	0.917
			LibSVM	83.0732%	0.831	0.831	0.830	0.825
			KNN (IBK)	93.0372%	0.931	0.930	0.930	0.936
			J48 (Decision tree)	72.6291%	0.737	0.726	0.727	0.740
			Random Forest	92.7971%	0.928	0.928	0.928	0.964

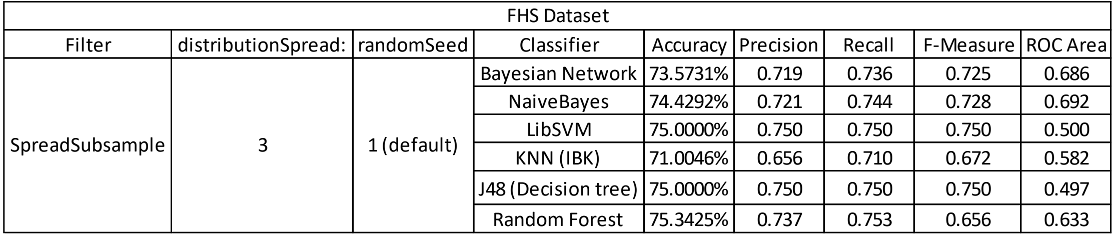


UCI Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	5	1 (default)	Bayesian Network	86.3145%	0.864	0.863	0.863	0.921
			NaiveBayes	85.9544%	0.860	0.860	0.860	0.917
			LibSVM	83.0732%	0.831	0.831	0.830	0.825
			KNN (IBK)	93.0372%	0.931	0.930	0.930	0.936
			J48 (Decision tree)	72.6291%	0.737	0.737	0.726	0.740
			Random Forest	92.7971%	0.928	0.928	0.928	0.964

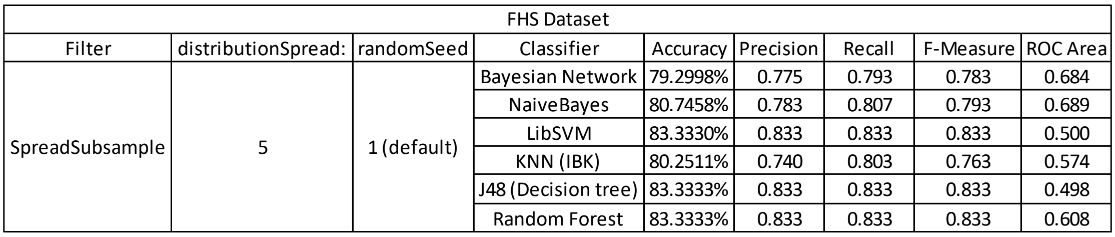
FHS Data set



FHS Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	1	1 (default)	Bayesian Network	64.6119%	0.646	0.646	0.646	0.690
			NaiveBayes	64.8402%	0.649	0.648	0.648	0.695
			LibSVM	57.3059%	0.574	0.573	0.572	0.573
			KNN (IBK)	57.4201%	0.576	0.574	0.572	0.587
			J48 (Decision tree)	56.7352%	0.575	0.567	0.557	0.587
			Random Forest	60.0457%	0.601	0.600	0.600	0.632

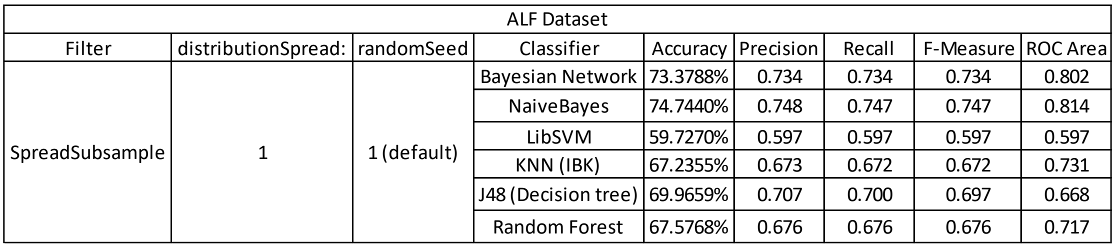


FHS Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	3	1 (default)	Bayesian Network	73.5731%	0.719	0.736	0.725	0.686
			NaiveBayes	74.4292%	0.721	0.744	0.728	0.692
			LibSVM	75.0000%	0.750	0.750	0.750	0.500
			KNN (IBK)	71.0046%	0.656	0.710	0.672	0.582
			J48 (Decision tree)	75.0000%	0.750	0.750	0.750	0.497
			Random Forest	75.3425%	0.737	0.753	0.656	0.633



FHS Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	5	1 (default)	Bayesian Network	79.2998%	0.775	0.793	0.783	0.684
			NaiveBayes	80.7458%	0.783	0.807	0.793	0.689
			LibSVM	83.3330%	0.833	0.833	0.833	0.500
			KNN (IBK)	80.2511%	0.740	0.803	0.763	0.574
			J48 (Decision tree)	83.3333%	0.833	0.833	0.833	0.498
			Random Forest	83.3333%	0.833	0.833	0.833	0.608

ALF Data set



ALF Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	1	1 (default)	Bayesian Network	73.3788%	0.734	0.734	0.734	0.802
			NaiveBayes	74.7440%	0.748	0.747	0.747	0.814
			LibSVM	59.7270%	0.597	0.597	0.597	0.597
			KNN (IBK)	67.2355%	0.673	0.672	0.672	0.731
			J48 (Decision tree)	69.9659%	0.707	0.700	0.697	0.668
			Random Forest	67.5768%	0.676	0.676	0.676	0.717

ALF Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	3	1 (default)	Bayesian Network	76.7065%	0.772	0.767	0.769	0.801
			NaiveBayes	77.5597%	0.776	0.776	0.776	0.810
			LibSVM	75.0000%	0.750	0.750	0.750	0.500
			KNN (IBK)	72.2696%	0.702	0.723	0.710	0.696
			J48 (Decision tree)	75.0000%	0.750	0.750	0.750	0.496
			Random Forest	76.0239%	0.778	0.760	0.670	0.719

ALF Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	5	1 (default)	Bayesian Network	80.6030%	0.817	0.806	0.811	0.804
			NaiveBayes	81.5131%	0.817	0.815	0.816	0.813
			LibSVM	83.3333%	0.833	0.833	0.833	0.500
			KNN (IBK)	79.4653%	0.766	0.795	0.778	0.668
			J48 (Decision tree)	82.9352%	0.768	0.829	0.772	0.543
			Random Forest	83.3902%	0.806	0.834	0.760	0.714

Surgical Timing Data set

Surgical timing Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	1	1 (default)	Bayesian Network	77.8862%	0.806	0.779	0.774	0.900
			NaiveBayes	77.8049%	0.804	0.778	0.773	0.896
			LibSVM	73.8753%	0.806	0.739	0.724	0.739
			KNN (IBK)	75.3252%	0.755	0.753	0.753	0.838
			J48 (Decision tree)	74.9864%	0.770	0.750	0.745	0.740
			Random Forest	81.7615%	0.833	0.818	0.816	0.912

Surgical timing Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	3	1 (default)	Bayesian Network	74.3850%	0.812	0.744	0.748	0.905
			NaiveBayes	74.2483%	0.811	0.742	0.747	0.901
			LibSVM	77.9774%	0.785	0.780	0.782	0.771
			KNN (IBK)	76.0543%	0.756	0.761	0.756	0.835
			J48 (Decision tree)	71.0269%	0.781	0.710	0.715	0.740
			Random Forest	84.4494%	0.843	0.844	0.843	0.923

Surgical timing Dataset								
Filter	distributionSpread:	randomSeed	Classifier	Accuracy	Precision	Recall	F-Measure	ROC Area
SpreadSubsample	5	1 (default)	Bayesian Network	74.3850%	0.812	0.744	0.748	0.905
			NaiveBayes	74.2483%	0.811	0.742	0.747	0.901
			LibSVM	77.9774%	0.785	0.780	0.782	0.771
			KNN (IBK)	76.0543%	0.756	0.761	0.756	0.835
			J48 (Decision tree)	71.0269%	0.781	0.710	0.715	0.740
			Random Forest	84.4494%	0.843	0.844	0.843	0.923