Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

Identification of Influential Features for Diabetes Prediction at an Early Stage Using Machine Learning

Author
Utsha Das
Roll No. 2004103020, Session: 2020-2021
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Supervised by
Prof. Dr. Boshir Ahmed
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Identification of Influential Features for Diabetes Prediction at an Early Stage Using Machine Learning"*** *submitted by* ***Utsha Das, Roll:2004103020*** *in partial fulfillment of the requirement for the award of the degree of Masters of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidates' own work carried out by them under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Prof. Dr. Boshir Ahmed
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Prof. Dr. Muhammad Shahin Uddin
Professor
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Identification of Influential Features for Diabetes Prediction at an Early Stage Using Machine Learning"*** *submitted by* ***Utsha Das, Roll:2004103020*** *in partial fulfillment of the requirement for the award of the degree of Masters of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidates' own work carried out by them under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Prof. Dr. Boshir Ahmed
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Prof. Dr. Muhammad Shahin Uddin
Professor
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Identification of Influential Features for Diabetes Prediction at an Early Stage Using Machine Learning"*** *submitted by* ***Utsha Das, Roll:2004103020*** *in partial fulfillment of the requirement for the award of the degree of Masters of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidates' own work carried out by them under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Prof. Dr. Boshir Ahmed
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Prof. Dr. Muhammad Shahin Uddin
Professor
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902

ACKNOWLEDGEMENT

I would like to express my gratitude to Almighty God for providing me with the opportunity and confidence to complete my thesis work. I am eternally thankful to my supervisor, Prof. Dr. Boshir Ahmed from the Department of Computer Science & Engineering at Rajshahi University of Engineering & Technology, for providing technical guidance, direction, and continuous support throughout the journey. His confidence-raising inspirational and motivational talks helped me a lot. He guided me with his vast amount of experience and knowledge which is a great experience for any student. Without his continuous support and mentoring, this work may not come to light.

I also extend my thanks to Professor Dr. Md. Al Mamun, the Head of the Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology for his aspiring words and kindness. All the teachers from Computer Science & Engineering of Rajshahi University of Engineering & Technology also deserve my sincere gratitude for their valuable suggestions and inspiration.

Finally, I express my appreciation to my family, friends, and well-wishers for their constant support and encouragement.

Utsha Das

ABSTRACT

One of the most prevalent illnesses, diabetes does not directly result in patient mortality. But, it increases the risk of death. Any disease that may be predicted in its early stages can lessen its fatal effects while also enhancing the quality of the healthcare system. For the early stage prediction of diabetes or such type of non-communicable diseases, we need the proper set of influential features. This research has developed a machine learning based disease prediction model to identify the influential features for diabetes prediction and give a near-perfect classification accuracy. This model includes Min-Max normalization for data normalization, Isolation Forest (iForest) for outlier removal, Synthetic Minority Oversampling Technique (SMOTE) for oversampling, Random Forest based Recursive Feature Elimination (RFE-RF) test, Chi-Square test, and Minimum Redundancy Maximum Relevancy (mRMR) test based feature selection methods for identifying the influential features, and Support Vector Machine (SVM) for the classification. The results clarify that the proposed model outperforms the other models and previous studies. The selected five features using the Chi-Square test can classify the test cases with a classification accuracy of 99.58% when the Support Vector Machine is used as the classifier. Finally, the SHAP is used to explain the importance of each selected feature in classification. These selected features and the classifier model can be used for the early stage prediction of diabetes. The proposed approach is also applied to another diabetes dataset. The proposed diabetes prediction model also outperforms this time.

Keywords: Communicable Disease, Diabetes, Feature Selection, Imbalanced Dataset, Insulin, Kernel Function, Outlier Samples.

PUBLICATIONS

Published Conference Papers:

1. Utsha Das and Boshir Ahmed, "Identification of Influential Features for Diabetes Prediction at Early Stage," *2022 4th International Conference on Electrical, Computer Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, 2022*, pp. 1-4, IEEE, 2022.

Journal Under Review:

1. Utsha Das and Boshir Ahmed, "A Machine Learning Approach for Predicting Diabetes in the Early Stage Using the Influential Features,"*IEEE Access*, IEEE. [Under Review].

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acronyms

χ^2 Chi-Square. ix, 17

4IR 4th Industrial Revolution. 5

AI Artificial Intelligence. 5, 19, 20

ANN Artificial Neural Network. x, xv, 22, 31, 32

CNN Convolutional Neural Network. 5

DBSCAN Density-Based Spatial Clustering of Applications with Noise. ix, 11, 13

DT Decision Tree. x, 29–31

FS Feature Selection. 7

iForest Isolation Forest. ix, 4, 11, 49

KNN K Nearest Neighbor. x, 3, 28, 59

LIME Local Interpretable Model-Agnostic Explanations. x, 36, 37

ML Machine Learning. viii, 3–7, 9, 13, 19, 20, 22, 25, 42

mRMR Minimum Redundancy and Maximum Relevancy. ix, 17, 18

NB Naive Bayes. x, 3, 22, 29, 59

OR Outlier Removal. 7

RBF Radial Basis Function. 4, 34, 35

RF Random Forest. x, 31

RFE-RF Recursive Feature Elimination based on Random Forest. ix, 17, 18

SCI Solving Class Imbalance. 7

SDG Sustainable Development Goal. 5

SHAP SHapley Additive exPlanations. x, 36, 59, 69

SMOTE Synthetic Minority Oversampling Technique. ix, 4, 16

SVM Support Vector Machine. x, xv, 3, 4, 22, 32–34, 59

Chapter 1

Introduction

1.1 Introduction

Diabetes which is a non-communicable disease is very prevalent around the world. High blood glucose levels resulting from deficiencies in insulin secretion, activity, or both define this metabolic disease. In 2019, it was anticipated that 8.8% of the world's population had diabetes, making it a serious public health issue that impacts millions of people [1]. Research has shown that in 2045, 645 million people worldwide will be affected by diabetes [2]. The situation is worse in developing countries like ours. There will be 228 million diabetics in developing nations by the year 2030 [3]. The Diabetes Association of Bangladesh claims that the number of diabetes patients in Bangladesh is more than 13 million. When considering the number of persons with diabetes, Bangladesh is ranked eighth among the top 10 nations worldwide [4].

There are three primary forms of diabetes. Type 1: The immune system targets the insulin-producing cells in the pancreas in this particular kind of diabetes, which is an autoimmune illness. Because of this, the body is unable to manufacture insulin, which raises blood glucose levels. Type 2: When the body experiences insulin resistance or is unable to create enough insulin to fulfill the body's requirements, it develops. It is the most prevalent kind of diabetes. Type 2 diabetes is frequently linked to being overweight, a poor diet, and inactivity. Another one is Gestational Diabetes. Gestational diabetes develops during pregnancy and often goes away once the baby is born. Women who have gestational diabetes run a higher risk of getting type 2 diabetes in the future if it is not appropriately managed.

1.2 Motivation and Challenges

Diabetes is not directly responsible for death in humans, but the risk of premature death is twice as high in patients with diabetes [5]. Patients suffering from diabetes are more likely to have problems such as foot ulcers, renal disease, heart disease, and stroke [6]. Due to long-term uncontrolled diabetes, complications like hyperosmolar hyperglycemia, and diabetic ketoacidosis may occur [5]. Diabetes-related complications account for 46.2% of all fatalities globally [7]. The cost of treating diabetes is much higher than other diseases. In 2017, approximately US$ 727 billion was spent on diabetes-related treatment worldwide [8].

The damage caused by diabetes can be reduced to a great extent through early diagnosis and proper diet and exercise. Concerned about the increasing risk of diabetes, researchers are investigating how to diagnose diabetes at an early stage using machine learning, a part of artificial intelligence. As a result of long-term research, machine learning is showing good results in diagnosing diabetes [9] [10]. But, due to the nature of data, missing values, class imbalance issues, and outlier samples, in many cases, machine learning cannot provide accurate results or cause complications in diabetes prediction. If these problems are solved, it will be possible to diagnose diabetes more effectively.

1.3 Justification of the Study

As the diabetes is a growing concern in word health system and the complexity and costs of diabetes is high in range. Proper treatment policy is needed in this regard. Early stage prediction of diabetes can be help in growing the concern of the patients. Patients can control the diabetes by taking proper foods and physical exercise.

For early stage prediction of diabetes, we actually need a proper set of influential features which can reduce the costs and complexities of diabetes prediction. Also, we need a Machine Learning model which can accurately identify the diabetes patients from non-patients. In this study, if we can identify those influential features for diabetes and build up the Machine Learning for accurate classification diabetes patients, the treatment policy of diabetes will be enriched. As a result, the premature death due to diabetes will be minimized and the treatment costs of diabetes will be reduced. As a result the human being will be benefited.

1.4 Research Questions

We'll answer the following queries in light of the issues mentioned and covered in the earlier sections:

(i) How to identify the influential features for diabetes prediction?

(ii) How to develop a Machine Learning (ML) model to obtain a satisfactory result for diabetic classification?

(iii) How the results can be justified?

1.5 Research Objectives

The goals of this study can be formulated based on the research questions provided in the preceding sections:

1.5.1 How to identify the influential features for diabetes prediction?

The proposed work has addressed this issue by utilizing three well known feature selections methods. The feature selection methods rank the features in the descending order according to their calculated scores. The top ranked features from each feature selections are studied in the next steps of study to identify the influetial features for diabetes prediction.

1.5.2 How to develop a Machine Learning (ML) model to obtain a satisfactory result for diabetic classification?

Three different Machine Learning (ML) models namely K Nearest Neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM) have been utilized here as the classifications models. This models are used as the binary classifier models in Machine Learning (ML) studies. Theses models have it's own tuning parameters for the best of models with the training data. For K Nearest Neighbor (KNN), the best K value for training data is identified during the cross-validation. The value of $laplace$ is tuned in the case of Naive Bayes (NB). For Support Vector Machine (SVM), there are different types of kernel functions and the tuning parameters for each kernel.

In this study, we have used Radial Basis Function (RBF) kernel for the Support Vector Machine (SVM). The Cost(C) and Gamma(γ) values are tuned in a broad range to identified the best classification model during the cross validation of the classifier models.

Class imbalanced dataset can miss-lead the classification modes. In this study, we have used a well-known over-sampling technique named Synthetic Minority Oversampling Technique (SMOTE) to solve the class imbalance issue. Our aim is to achieve a near perfect classification accuracy.

1.5.3 How the results can be justified?

To justify the results, a comparison has been made with the performance of among the Machine Learning (ML) models and also compare the results with previous studies on the same dataset. We have used a number of performance matrices to verify the classification models.

1.6 Research Contributions

The major contributions of this research are listed as follows:

(i) Five features have been identified as the influential features for diabetes prediction at an early stage.

(ii) The combination of the Isolation Forest (iForest) and Synthetic Minority Oversampling Technique (SMOTE) have not ever been used in a same study of diabetes prediction. We have used these methods in our study.

(iii) A near perfect classification model has been developed to classify the patients and non-patients samples.

(iv) The contribution of each features to the classification of diabetes patients and non-patients is discussed with a mathematical explanation.

The primary contributions of this research are described in detail throughout the later chapters, and the process of discovery is also explained. For instance, in Chapter 4, the application of feature selection methods and the selected feature are discussed. The application details of classifications models is discussed in the Chapter 5. This approach is intended to enhance readers' understanding of the research and make the reading experience more enjoyable.

1.7 Benefits, Ethics, Sustainability, and Delimitations

The selected features from this is marked the influential for diabetes predictions at an early stage. The feature considered in this study are the outcomes physical or pathological tests. No feature is connected to complex hormonal tests.

Public will be able to predict being the diabetic patient in early stage. So, the cost and time for diabetes prediction will be reduced in practical cases. Preventive measure against the diabetes will be taken in early stage. The life risk of the general people will be minimized. As a result the health care system will be benefited.

The influence of the 4th Industrial Revolution (4IR) have the touch the health care system. The Machine Learning (ML), Convolutional Neural Network (CNN), Artificial Intelligence (AI), and Nanotechnology are used in diseases prediction and drug discovery. Also, the good health and well being are listed in the Sustainable Development Goal (SDG) list.

As we mentioned before, this study will bring a positive change in diabetes treatment. The public will be benefited from our research. The cost of diabetes and diabetes related health care system will be reduced. This will contribute the overall economy of the nation by reducing the cost of diabetes treatment.

1.8 Thesis Organization

This thesis book has a total of 5 chapters including this chapter. The later chapters are listed as follows:

(a) **Chapter 2 – Background Study and Literature Review:** This chapter includes different machine learning methods that we have studied for this research. Different algorithms or techniques for each operation are discussed here. Previous studies related to early-stage diabetes prediction using machine learning, outlier removal, and over-sampling are also included in this chapter.

(b) **Chapter 3 – Methodology and Experimental Analysis:** The methodology for early stage diabetes prediction is discussed at be beginning of this chapter. The experiments done in this research are discussed serially according to the proposed diabetes prediction model. The outputs of the experiments are also tabulated in this chapter.

(c) **Chapter 4 – Result Analysis:** This chapter represents the result analysis pf this research. The comparative studies of the outputs of different methods are discussed in this chapter. The model explanation is also included in this chapter.

(d) **Chapter 5 – Conclusion and Future Scopes:** The research summary, the impact of this research in the academic sector and industry, limitations, and future scopes are the topics of this chapter.

1.9 Conclusion

The inaugural chapter delves into the challenges and complexities in diabetes studies. The research questions are formulated based on the presented challenges and objectives are determined. The thesis presents significant contributions to the field by proposing a Machine Learning (ML) model for diabetes prediction. Firstly, importance of feature selection for diabetes prediction is mentioned. Secondly, the importance of improving the classification accuracy of the classification models is discussed. The chapter concludes with an overview of the thesis structure and organization.

Chapter 2

Background Study and Literature Review

2.1 Introduction

This chapter provides a thorough background analysis and literature review of Data Pre-processing, Outlier Removal (OR), Solving Class Imbalance (SCI), Feature Selection (FS), and Machine Learning (ML). The chapter begins with an overview of Data Pre-processing. Different Outlier Removal (OR) methods are also discussed in this chapter. In the next, methods related to Solving Class Imbalance (SCI) issue are discussed. Then, Feature Selection (FS) methods are discussed. Machine Learning (ML) is discussed broadly in this chapter. Machine Learning research related to diabetes prediction, removing outlier samples, and over-sampling are discussed finally.

2.2 Data Pre-Processing

Data Pre-processing is the initial task of machine learning researches. Sometimes, the collected data are not well organised as the researchers need. Dataset may contain missing values, improper data transformation, unwanted data format. So, the collected data is needed to data cleaning, data normalization, and encoding.

2.2.1 Data Cleaning

Data Cleaning is the procedure of removing unwanted data and solving missing value problem. Data cleaning is categorised into two broad categories: Solving Missing Value problem and Solving Noisy Data.

2.2.1.1 Solving Missing Values

Missing Values Problem can be solved by ignoring the tuple and filling.

1. Ignoring the Tuple: When a sample have missing values for large number of attributes or a attribute have missing values for quite large number of samples, the sample or the attribute can be ignored then.

2. Filling: Missing values can be filled with the mean or median value of the corresponding attribute. Sometimes, linear regression model is used to fill up the missing values.

2.2.1.2 Solving Noisy Data

Sometimes, the data collection methods or the tools used to collect the data is faulty. These generate irregular or faulty data. And, this faulty data is needed to be solve. Binning, Regression are used to solve noisy data.

1. Binning: The data are sorted and divided into fixed size segments. These segments are called bins. The data are replaced by the mean value of the bin or the boundary value.

2. Regression: Noisy data can be made smooth by a regression model. A linear or multiple regression model is fit to the dataset and calculate the smooth value for that specific data point.

2.2.2 Data Normalization

Data normalization is a very important part of data pre-processing. Data normalization is used to scale the features in a similar range. By normalizing the features, it is possible to increase the prediction model's accuracy and stability. Min-Max normalization and $Z-$Score normalization are well-known normalization methods.

2.2.2.1 Min-Max Normalization

Min-Max normalization is a very renowned normalization technique in machine learning. The Min-Max normalization transforms the data in the scale of [0,1]. The highest value of a specific feature is replaced by the value 1 and the lowest by the value 0. The mathematical equation for Min-Max normalization is

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2.1}$$

Here, x_i is the value of the attribute for the sample i. x_{max} and x_{min} are the maximum value and the minimum value respectively of that attribute. x'_i is the new value of x_i after normalization.

2.2.2.2 $Z-$Score Normalization

$Z-$Score normalization method is a standard deviation based data normalization method. It actually calculates the number of standard deviations away from the mean. It tries to ensure that the mean and the standard deviation of the data will be 0 and 1 respectively. The $Z-$Score calculation is as

$$x'_i = \frac{x - \mu}{\sigma} \tag{2.2}$$

Here, x'_i is the $Z-$Score of the data point x. μ and σ are the mean and standard deviation respectively. In normal distribution,

2.2.3 Encoding

Dataset may contain categorical data. Most of the cases, these data are in text format. But, Machine Learning (ML) models need data in numerical format. So, the encoding technique is used to convert categorical data into numerical data. One hot encoding, label encoding are most popular encoding method.

2.2.3.1 One Hot Encoding

When the attribute is nominal categorical data, the one hot encoding method is used for data conversion. After applying the one hot encoding, the number of total attributes is increased. For example, Table 2.1 represents the weather of a certain area. The values are Sunny, Rainy, Cold, and Cloudy. After applying the one hot encoding, the data will be like in Table 2.2.

Day	Weather_Condition
Day1	Sunny
Day2	Rainy
Day3	Cloudy
Day4	Cold

Table 2.1: Weather Condition for a Certain Area.

Day	Sunny	Rainy	Cold	Cloudy
Day1	1	0	0	0
Day2	0	1	0	0
Day3	0	0	0	1
Day4	0	0	1	0

Table 2.2: After Applying One Hot Encoding on Table 2.1.

2.2.3.2 Label Encoding

Label Encoding is on the ordinal categorical data. The value is replaced by a numeric value of a certain range. For example, Table 2.3 represents the cancer stage of some patients. The "Early" stage denotes the least critical and the "Distant" stage denotes the most critical stage of the patient. Table 2.4 represents the data of the 2.3 after label encoding.

Patient	Cancer_Stage
Patient1	Early
Patient2	Localized
Patient3	Distant
Patient4	Regional

Table 2.3: Cancer Stages.

Patient	Cancer_Stage
Patient1	1
Patient2	2
Patient3	4
Patient4	3

Table 2.4: Cancer Stages after Label Coding.

2.3 Outlier Removal

One of the most challenging problems in machine learning is dealing with outlier samples. In computer science, the process of identifying data points that deviate from the norm is known as outlier detection. Inter Quartile Range(IQR), $Z-$Score, Isolation Forest (iForest), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) are well-known outlier detection and removal methods.

2.3.1 Inter Quartile Range(IQR)

Inter Quartile Range(IQR) is a statistical method for removing outlier samples. The dataset is sorted and divided into quartiles. 1^{st} quartile(Q_1) is the 25 percentile of the data. 2^{nd} quartile(Q_2) and 3^{rd} quartile(Q_3) consist of the 50 percentile and 75 percentile of the data respectively. IQR is calculated as

$$IQR = Q_3 - Q_1 \tag{2.3}$$

The data points which are smaller than (Q_1-1.5IQR) or greater than (Q_3+1.5IQR) are classified as the outlier samples.

2.3.2 $Z-$Score

$Z-$Score value can also be used for outlier detection. $Z-$Score value is calculated as

$$Z_i = \frac{x_i - \mu}{\sigma} \tag{2.4}$$

Here, Z_i is the $Z-$Score of the data point x_i. μ and σ are the mean and standard deviation (sd) respectively. In a normal distribution, about 68% of the data points are in the first standard deviation, 95% are in the second standard deviation, and 99.7% of the data points are in the third standard deviation. So, the data points that have a $Z-$Score value of more than 3 are outliers. Figure 2.1 represents the relation between data distribution and standard deviation.

2.3.3 Isolation Forest (iForest)

Isolation Forest (iForest) [11], an unsupervised machine learning approach, is developed on decision trees and is mostly used to identify outlier data. The main idea of iForest is the random split of the data points and building up trees. First, an attribute is selected randomly

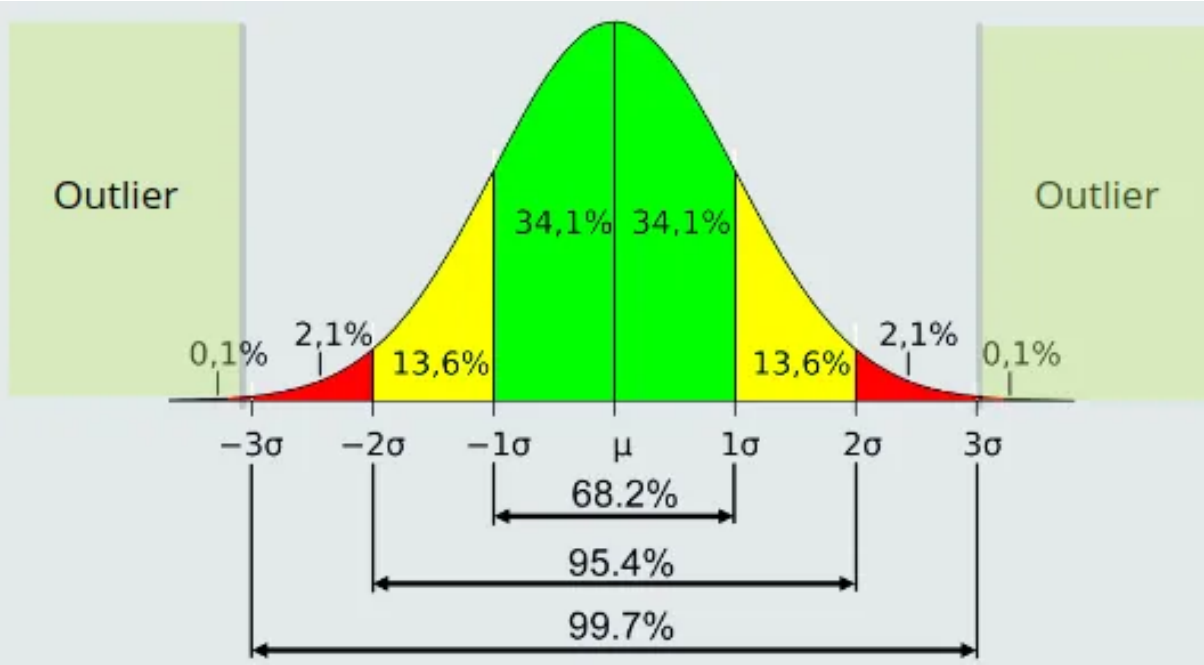

Figure 2.1: Illustration of Normal Distribution with Data Percentile.

as the splitting feature of the data points. Then an arbitrary threshold value is selected within the range of the selected attribute. This threshold value is used to split the samples into two groups. The samples which have a value greater than the threshold are in one group and others are in another group. This random splitting process is repeated on the newly separated groups recursively until no further split on the new group or fulfills certain criteria.

The tree generated by this process is used to detect the outliers. Data points that are isolated after a few splits, i.e. those with lower-level values are outlier data points. The more accurate the tree construction, the easier and more effective it is to find outlier samples. The anomaly score is calculated for each sample in Isolation Forest. The anomaly score is calculated as

$$S(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2.5}$$

Here, n denotes the sample size, $c(n)$ denotes the average cost of an unsuccessful search in Binary Search Tree (BST), and $h(x)$ is the cost of a sample x. In iForest, cost means the path length of a node from the root of the BST.

2.3.4 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [12] is unsupervised Machine Learning (ML) algorithm used clustering as well as the outlier detection. DBSCAN identifies those data points into a single cluster which are densely clustered. For clustering the data points, DBSSCAN uses two parameters which are epsilon (ϵ) and $MinimumPoints$. The epsilon (ϵ) works as the size of the radius of the circle considered around a data point. And, $MinimumPoints$ is the required number of data points needed to be inside the circle of a data point classified as the core data point. For example, Figure 2.2 represents some sample data points which are needed to be clustered.

Figure 2.2: Sample Data Points Which are not Clustered.

If ϵ is the radius and $MinimumPoints$=3, then the cluster generated by DBSCAN is like Figure 2.3. The red-colored data points are core points as there are at least 3 data points in their circle. The yellow-colored data points are border points as they can't fulfill the condition of the core points. They have at least one data point inside the circle. But the number of data points inside the circle is less than 3. The purple-colored data points are outliers as there are no data points inside the circles.

2.4 Solving Class Imbalance Issue

Class imbalance data results in inaccurate accuracy for classification. Due to class imbalance, the trained model becomes biased. So it is very urgent to solve the class imbalance problem.

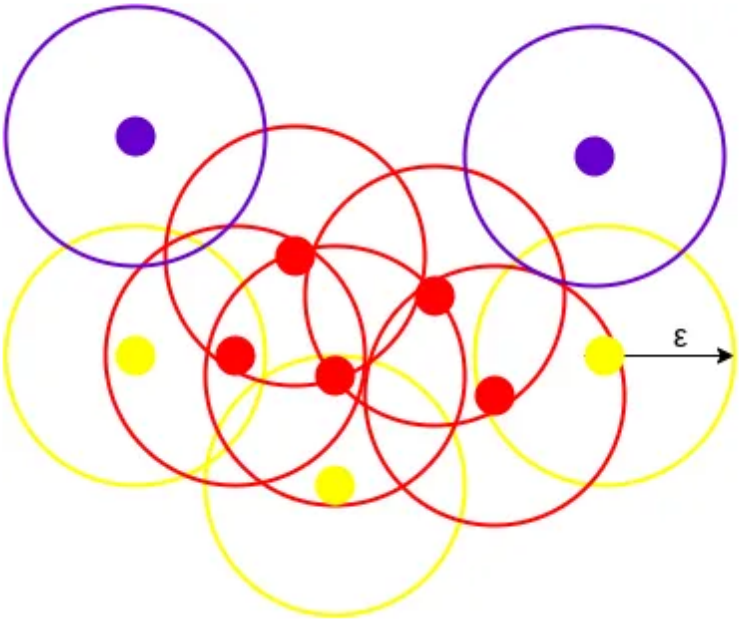

Figure 2.3: Cluster Generated by DBSCAN.

2.4.1 What Is Class Imbalanced Data?

The term imbalanced refer to the disparity encountered in the dependent variable. Therefore, the class imbalanced problem for classification is one in which the dependent variable has an imbalanced proportion of classes. In other words, a dataset that exhibits an unequal distribution between its classes is considered to be imbalanced. For example: Consider a dataset with 40000 samples. This dataset consists of candidates who applied for admission in RUET. Apparently, RUET accepts a very low number of students. The independent variable represents if a candidate has been shortlisted (1) or not shortlisted (0). After analyzing the data, it was found 90% did not get shortlisted and only 10% got lucky. This is a perfect case of imbalanced classification. For better understanding, here are also some real-life examples.

- A test id done to detect cancer in residents of a chosen area may find the number of cancer affected people significantly less than unaffected people.
- In credit card fraud detection, fraudulent transactions will be much lower than legitimate transactions.
- Manufacturing operating under six sigma principle may encounter 10 in a million defected products.

In real life, the class imbalanced situation is frequent. So, it is important to deal with such problems for researchers.

2.4.2 Effects of Class Imbalance on the Accuracy of Machine Learning Algorithms

Class imbalance leads to a reduction in accuracy of Machine learning algorithms. Some reasons for this are given below:

- Machine learning algorithms struggle with accuracy because of the unequal distribution in dependent variable.
- This causes the performance of existing classifiers to get biased towards the majority class.
- The algorithms are accuracy driven i.e. they aim to minimize the overall error to which the minority class contributes very little.
- Machine learnings assume that the data set has balanced class distributions.

To minimize this problem, extra operations are needed to be performed. The methods for balancing class is known as sampling. Basically, there are four types of methods to solve this problem.

2.4.3 Methods for Solving Class Imbalance Problem

These methods have acquired higher importance after many researches have proved that balanced data results in improved overall classification performance compared to an imbalanced dataset. There are four types of popular methods for balancing class imbalance data:

1. Under-sampling
2. Over-sampling
3. Synthetic Data Generation
4. Cost-Sensitive Learning

2.4.4 Under-sampling

The method works with majority class. It reduces the number of observations from majority class to make the dataset balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.

Randomly under-sampling method randomly chooses observations from majority class which is eliminated until the dataset gets balanced. Informative under-sampling follows a pre-specified selection criterion to remove the observations from the majority class.

2.4.5 Over-sampling

This method works with minority class. It replicates the observations from minority class to balance the data. It is also known as up-sampling. Similar to under-sampling, this method also can be divided into two types: Random Oversampling and Informative Oversampling.

Randomly oversampling balance the data by randomly oversampling the minority class. Informative oversampling uses a pre-specified criterion and synthetically generates minority class observations.

2.4.6 Synthetic Data Generation

In simple words, instead of replicating and adding from the minority class, it overcomes imbalances by generating artificial data. It is also a type of oversampling technique. In regards to synthetic data generation, synthetic minority oversampling technique is a powerful and widely used method. It creates artificial data based on feature space (rather data space) similarities from minority samples. We can also say, it generates a random set of minority class observations to shift the classifier learning bias towards minority class. Synthetic Minority Oversampling Technique (SMOTE) is a well-known synthetic data generation method.

2.4.6.1 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is an oversampling technique that generates synthetic data points using the existing data points from the minority class rather than oversampling by replacement. For generating synthetic data points, SMOTE uses the feature space rather than the sample space of the existing data points from the minority class. SMOTE follows a simple but very fruitful use of K-Nearest Neighbors. SMOTE selects a random value of K depending on the amount of oversampling

required. These K neighbors are taken into account while creating synthetic data points for a particular minority class data point. First, a data point from the minority class is chosen, and the K neighbors are found, in order to create synthetic data points. Then, the vector difference between the selected data point and one of its neighbors is multiplied by a random value between [0,1] and added to the selected data point. The newly generated data point is actually a synthetic data point [13]. In this way, oversampling is performed in SMOTE.

2.4.7 Cost-Sensitive Learning

It is another commonly used method to handle classification problems with imbalanced data. It's an interesting method. In simple words, this method evaluates the cost associated with misclassification observations. It does not create a balanced data distribution. Instead, it highlights the imbalanced learning problem by using cost matrices which describe the cost for misclassification in a particular scenario. Recent researches have shown that cost-sensitive learning has many time outperformed sampling methods. Therefore, this method likely alternative to sampling methods.

This is a short description of the method, we have used for the classification task. After classification, we have proceeded to identify the up-regulated and down-regulated genes. In the next section, we will discuss the methods for identifying the up-regulated and down-regulated gene.

2.5 Feature Selection

Feature selection methods are basically used in machine learning for dimensionality reduction. After feature selection, the feature space of the data remains it's original form. Feature selection methods just find out the most relevant features which are important for classification. Chi-Square (χ^2) test, Minimum Redundancy and Maximum Relevancy (mRMR) test, Recursive Feature Elimination based on Random Forest (RFE-RF) test are some well-known feature selection methods used in machine learning specially for diseases prediction models.

2.5.1 Chi-Square (χ^2) Test

The Chi-squared test [14] is used to determine whether there is a statistically significant difference between the actual outcomes and the results that were perceived in one or more areas

of a probability chart. Groups of measures that are typically mutually exclusive are created. The test statistic calculated from the measurements confirms chi-square frequency correlations, which is consistent with the null hypothesis that there are no differences between the groups in the population. Assessing how appropriate the perceived rates would be if the null hypothesis were true is the main objective of the investigation. The Chi-Square is calculated as in equation 2.6.

$$\chi^2 = \frac{(O_i - E_i)^2}{E_i} \tag{2.6}$$

Here, χ^2 is the Chi-Square, O_i and E_i denote the observed and expected values respectively.

2.5.2 Minimum Redundancy and Maximum Relevancy (mRMR) Test

The mRMR test [15] chooses a selection of characteristics from a dataset that optimizes relevance to the target variable while reducing redundancy among the chosen features. This problem is solved by mRMR by taking relevance and redundancy into account simultaneously. Redundancy assesses the similarity or overlap between characteristics, while relevance gauges how useful a feature is in forecasting the target variable. In order to determine which attributes are most valuable, the algorithm attempts to strike a compromise between these two factors. The merit value for each feature is determined by mRMR based on the relevance and redundancy scores. The algorithm then iteratively chooses the characteristics with the highest merit value, making sure that the chosen features share as little information as possible with one another. This procedure is carried out again until the required number of characteristics is attained.

2.5.3 Recursive Feature Elimination based on Random Forest (RFE-RF) Test

RFE is a method that recursively chooses features by training a model on the entire feature set and then removing the features that are deemed to be the least significant based on their importance rankings. A predetermined number of characteristics is attained by repeating this method. The feature importance or coefficients of the underlying model are often examined to determine the importance rankings.

Random Forest (RF) is an ensemble learning technique that blends various decision trees to generate predictions. A random subset of characteristics and data samples are used to train each decision tree in the forest. The resilience and versatility of Random Forest are well recognized.

RFE-RF [16] combines the RFE and the RF. First, it trains the Random Forest model using all features. Then, it ranks the features based on Random Forest Model. Next, it eliminates the least scored feature. This procedure is repeated recursively until a desired number of features is selected.

2.6 Machine Learning

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) which tries to build an intelligent computer program that can learn from data and predict or classify the unknown based on the learning from the previous. It is a learning process like a human learns from nature. When a baby is born, he/she is unable to behave with others. Gradually, he/she learns from his/her surroundings and starts reacting with others. Sometimes, he/she gets feedback from others and uses this feedback in his/her learning. Day by day, the brain of that human develops. As he learns more, his/ her ability increases.

Machine Learning (ML) methods follow the same learning process. The machine is hardware, it doesn't have any intelligence to predict. The Machine Learning (ML) is developing an intelligent machine, actually an intelligent computer program. The Machine Learning (ML) algorithms learn from the given data and try to find out the pattern lies in the given data. These learned patterns are used in future predictions.

Machine Learning (ML) is quite different from the usual computer program. A Computer program is developed by programmers following some instructions. The computer program inputs data and executes the instructions. After execution, the program outputs some data. But, Machine Learning (ML) models not only input the data but also take the output as input. The Machine Learning (ML) models consider both the input data and output data for learning and developing itself. This learning is used for future predictions. Machine Learning (ML) is not a single study field. It is a combination of Computer Science, Mathematics, Statistics, and Data Science.

2.6.1 Relationship Between Machine Learning and Other Fields

Machine Learning is not a stand-alone branch of knowledge. It is actually a combination of different fields of science and technology [17]. After the evolution of computer science and its application in different real-life fields, the concept Artificial Intelligence (AI) became urgent.

The Artificial Intelligence (AI), and its branches were developed then. Machine Learning (ML) is a sub-field of Artificial Intelligence (AI). The Machine Learning (ML) inherits the concept of data structure, algorithms, computational complexity, and programming from Computer Science. The concept of probability, and likelihood are from the statistics. Most of the Machine Learning (ML) models are based on mathematical models. So the concept of linear algebra, calculus, optimization, and matrix from mathematics are used in Machine Learning (ML). Not only these, the concept from pattern recognition, data mining, and neurocomputing are also used in Machine Learning (ML). Figure 2.4 represents the relation of Machine Learning (ML) with other fields.

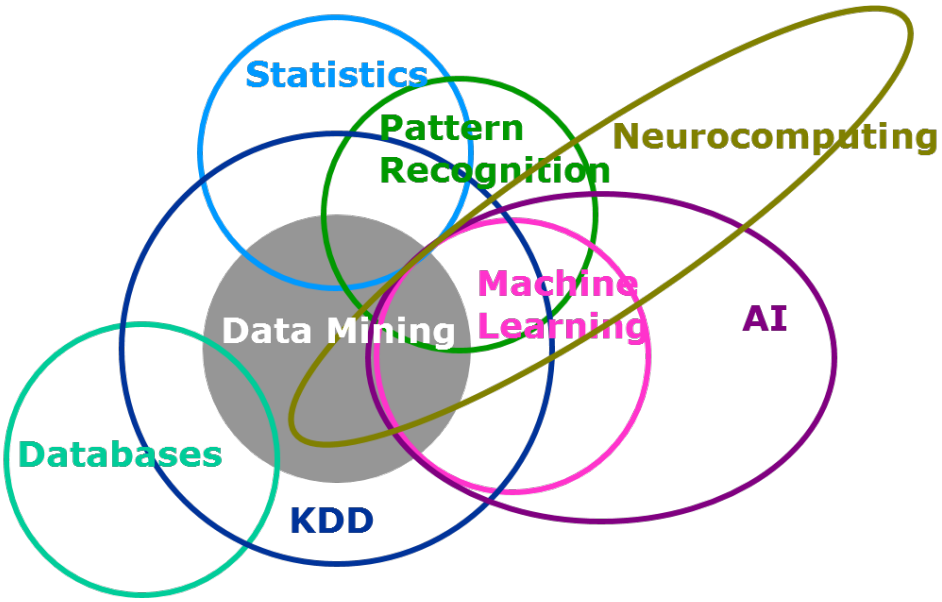

Figure 2.4: Relationship between Machine Learning and Other Fields.

2.6.2 Types of Learning

The machine learning models are classified into three groups based on the nature of learning. They are

1. Unsupervised Learning
2. Supervised Learning
3. Semi-supervised Learning

2.6.2.1 Unsupervised Learning

Unsupervised learning is a machine learning technique that operates without the assistance of labeled training data. Unsupervised learning may be used to detect patterns, trends, and correlations in data without knowing the target class labels beforehand. This approach may be used to group data points that are similar, identify themes, and show the underlying structure of the data. The most prominent approach for unsupervised data categorization is clustering, which involves arranging the data into groups based on similarity. Another technique for better seeing and comprehending the links between the data is dimensionality reduction, which requires converting the data into a lower-dimensional space. Unsupervised learning has advantages since it liberates us from constraints and preconceived conceptions while exploring data, but it also has disadvantages because it is dependent on the correctness and importance of the underlying patterns uncovered in the data. Unsupervised learning is depicted in Figure 2.5.

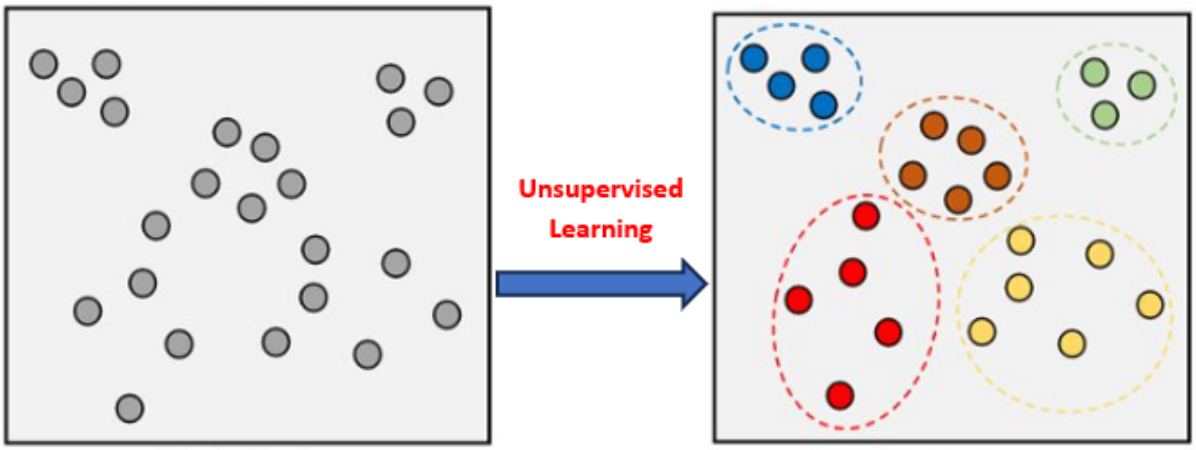

Figure 2.5: Illustration of Unsupervised Learning.

2.6.2.2 Supervised Learning

Supervised learning is a type of machine learning in which the algorithm is trained with labeled data [17]. The purpose of supervised learning in disease prediction is to construct a model that can reliably predict the class label of a given sample, either positive or negative. The method begins with a training dataset made up of input training samples and the class labels assigned to them. This dataset is used to train the model to understand how class labels and features are related to one another. Throughout the training phase, the model updates its parameters to minimize the gap between its predictions and the actual class labels in the training

data. After the model has been trained, it can predict the class label of an unseen sample. For classification, typical supervised learning algorithms include Support Vector Machine (SVM), Naive Bayes (NB), and Artificial Neural Network (ANN). Figure 2.6 depicts the supervised learning process.

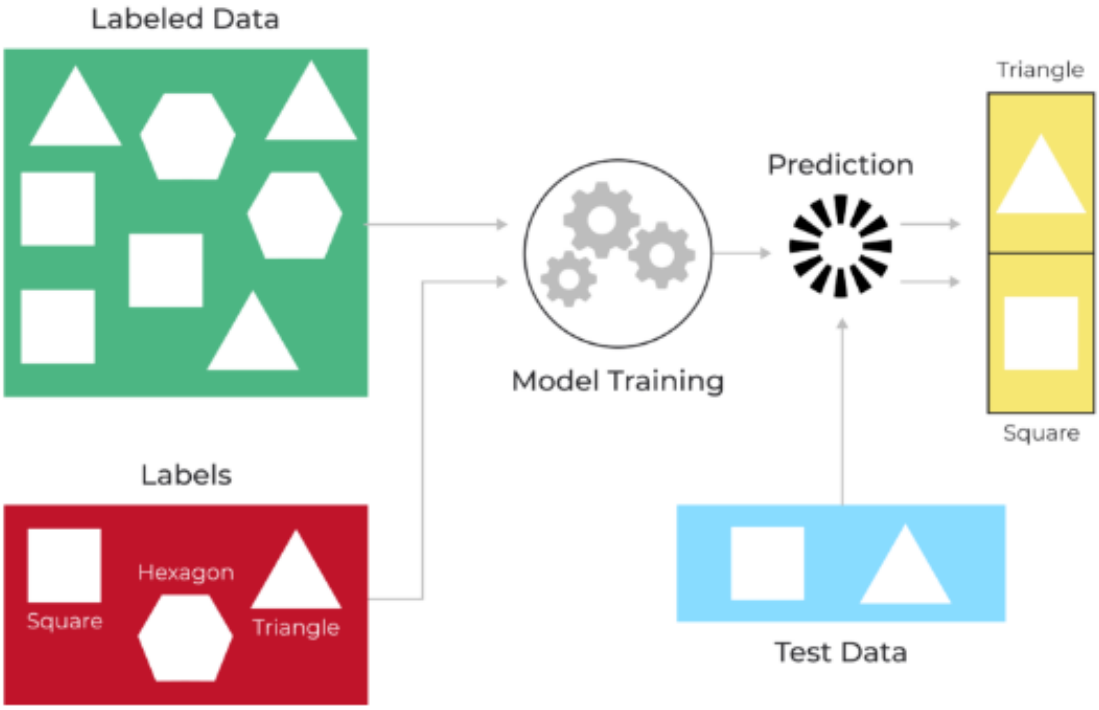

Figure 2.6: Illustration of Supervised Learning.

2.6.2.3 Semi-Supervised Learning

Semi-Supervised Learning is a special type of Machine Learning (ML) which combines both the concept of Unsupervised Learning and Supervised Learning. That means the training dataset consists of both class-labeled data and class-unlabeled data. But, the volume of class-unlabeled data is much more than class-labeled data. These algorithms take advantage of both Unsupervised Learning and Supervised Learning. Figure 2.7 depicts the Semi-Supervised Learning model. 2.5. Semi-Supervised Learning models are basically used for text classification, speech analysis, and DNA sequence analysis.

2.6.3 Types of Machine Learning Task

Machine learning tasks are categorized into two broad categories. They are:

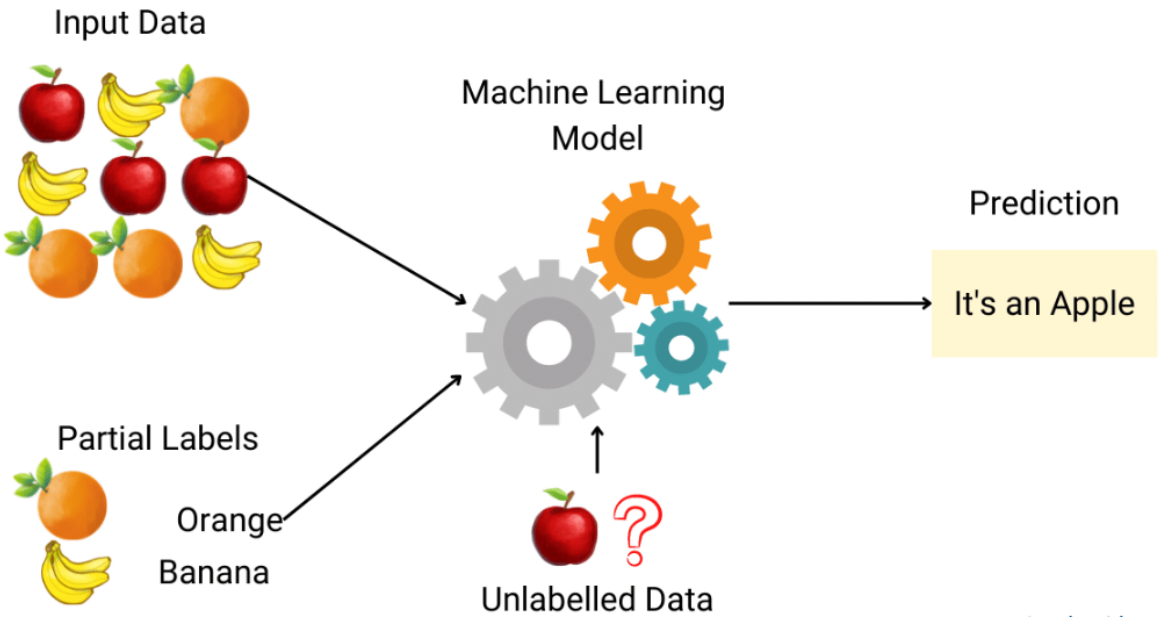

Figure 2.7: Illustration of Semi-Supervised Learning.

1. Classification
2. Regression

2.6.3.1 Classification

Classification is a machine-learning task which is to predict the class label of a given input sample. According to the number of class labels present in a dataset, classification models are categorized into two types, namely, Binary Classification and Multi-class Classification.

1. **Binary Classification**: Binary Classification is used when the number of class labels is two. For example, a disease dataset contains two types of samples, namely, patient and non-patient. The classifier model has to classify the test samples as either patient or non-patient. The model's accuracy is calculated by comparing the predicted class label with the actual class label. Figure 2.8 illustrates the binary classification. The classifier model class the incoming messages as either INBOX or SPAM FOLDER.

2. **Multi-Class Classification**: When the number of class labels is more than two, the classification technique is called multi-class classification. Sometimes, a disease dataset may contain more than two class labels. For example, a cancer dataset contains patients with

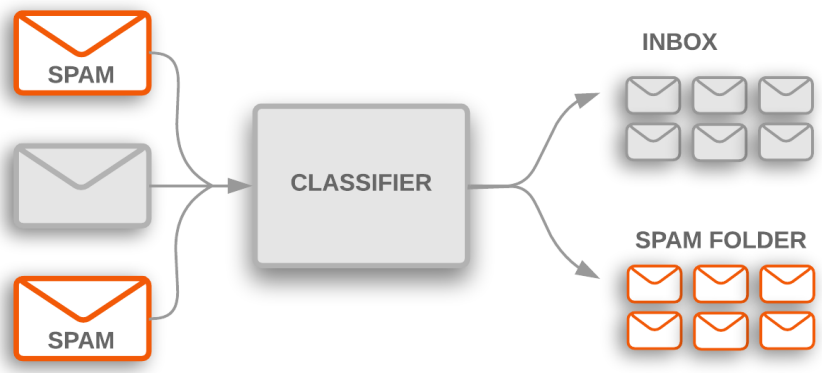

Figure 2.8: Illustration of Binary-Classification.

three different stages of cancer. Here, multi-class classifier in needed to classify patients. Figure 2.9 depicts multi-class classifcation.

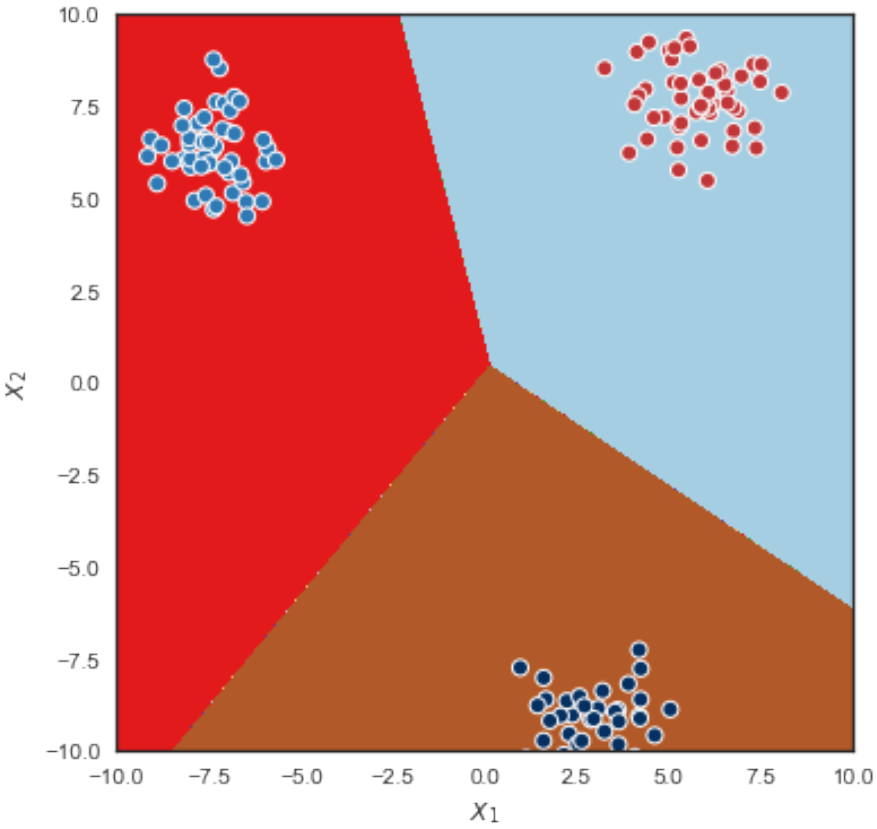

Figure 2.9: Illustration of Multi-Class Classification.

2.6.3.2 Regression

Regression is another type of method used for prediction in Machine Learning (ML). Regression is used to predict continuous or real value instead of discrete value as in classification. Linear Regression and Logistic Regression are popular type of regrssion methods.

1. **Linear Regression**: It is one of the easiest methods for continuous or real value prediction. Linear regression tries to find out the linear relationship between the independent variable and one or more dependent variables. Figure 2.10 depicts the linear regression.

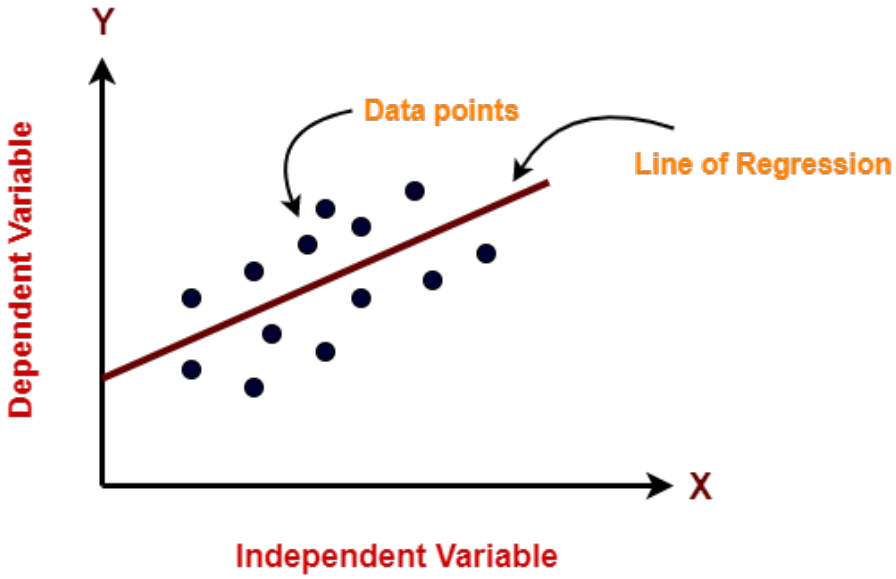

Figure 2.10: Illustration of Linear Regression.

2. **Logistic Regression**: Logistic Regression also finds the relationship between the independent variable and the dependent variable. But, the outcome from Logistic Regression is a probabilistic value that denotes the probability of occurring a class. Logistic Regression is basically used for classification. Figure 2.11 depicts the Logistic Regression.

2.6.4 Cross-Validation

Cross-validation is a model assessment technique used to evaluate a machine learning algorithm's performance in making a prediction on new datasets that it has not been trained on. This is done by portioning a dataset and using a subset to train the algorithm and the remaining data for testing. Because cross-validation does not use all of the data to build a model, it is a commonly used method to prevent over-fitting during training. Each round of cross-validation

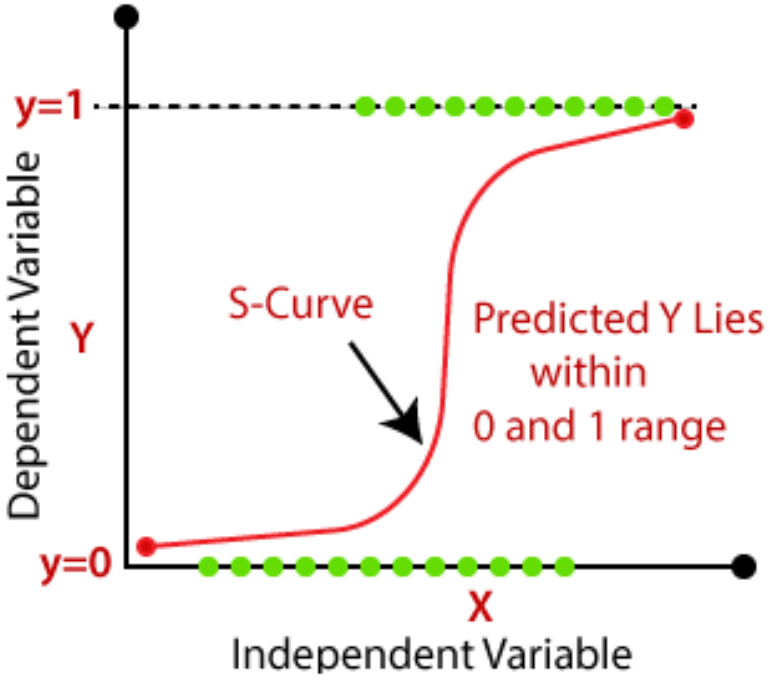

Figure 2.11: Illustration of Logistic Regression.

involves randomly partitioning the original dataset into a training set and a testing set. The training set is then used to train a supervised learning algorithm and the testing is used to evaluate its performance. This process is repeated k- times and average cross-validation error is used as a performance indicator. Common cross-validation techniques include:

- **K-fold**: Partitions data into k randomly chosen subsets (or folds) of roughly equal size. One subset is used to validate the model trained using the remaining subsets. This process is repeated k times such subset is used exactly once for validation.
- **Holdout:** Partitions data into exactly two subsets (or folds) of specified ratio for training and validation.
- **Leave-one-out**: Partitions data using K-fold approach where k is equal to the total number of observations in the data. As know as leave-one-out cross-validation.
- **Repeated random sub-sampling**: Performs Monte Carlo repetitions of randomly portioning data and aggregating results over all the runs.
- **Stratify**: Partitions data such that both training and test sets have roughly the same class proportions in the response or targets.

- **Re substitution**: Does not partition the data; uses the training data for validation. Often produces overly optimistic estimates for performance and must be avoided if there is sufficient data.

Cross-validation can be a computationally intensive operation since training and validation is done several times. Because each partition set is independent, this analysis can be performed in parallel to speed up the process. Among the common cross-validation methods that we have discussed in the above, the independent dataset test, cross-validation test, and jackknife test are commonly used methods in statistical prediction. Considering the computational time for cross-validation some researchers adopted k-fold cross-validation [18] [19].

2.6.5 k-Fold Cross-Validation

Cross-validation is a resampling technique used to assess machine learning models on a small data sample. The number of groups into which a certain data sample is to be divided is indicated by the procedure's sole parameter, k. As a result, the method is frequently referred to as k-fold cross-validation. When a particular value for k is selected, it may be substituted for k in the reference to be model; for example, k=10 would become 10-fold cross-validation.

In applied machine learning, cross-validation is generally used to assess a model's proficiency with untried data. To put it another way, this is to use a small sample to estimate how the model will generally perform when used to generate predictions on data that was not utilized during the model's training.

It is a well-liked technique since it is easy to comprehend and typically yields a less biased or too optimistic assessment of the model skill than other techniques, including a straightforward train/test split. The general procedure as follows:

1. Randomly shuffle the dataset
2. Create k groups from the dataset.
3. For every distinct group:
   (a) Consider the collection as a test or holdout dataset
   (b) As a training dataset, use the remaining groupings
   (c) Adapt a model to the training data, then assess it against the test set

(d) Keep the evaluation result and throw away the model

4. Using a sample of the model assessment ratings, summarize the model's competence.

It's significant that every observation in the data sample is given a unique group and remains there throughout the process. This indicates that each sample has the chance to be used k times to train the model and k times in the hold-out set [20]. It is crucial that all data preparation before model fitting take place on the loop's cross-validation assigned training dataset rather than the larger dataset. This also holds true for any hyperparameter adjustment. A model estimate that is too optimistic might come from not carrying out these activities inside the loop and data leakage. The mean of the scores for the model skill serves as a common summary of the outcomes of a k-fold cross-validation run. The standard deviation or standard error are appropriate measures of the skill scores' variation. This is also excellent practice. Selecting the right k value is important. An incorrect impression of the model's competence is produced by a poorly set k-value.

2.6.6 Classification Models

Classification is a machine-learning process of assigning a class label to a data point. Researchers have developed many classifier models. Some well-known and frequently utilized models are discussed in the next sub-sections.

2.6.6.1 K Nearest Neighbor (KNN)

In machine learning, K-Nearest Neighbors [21] is a well-known supervised classification technique. For the categorization of test data, KNN examines just K samples. During the classification model's training phase, the ideal value of K is found. The classifier model examines the K nearest neighbors based on Euclidean distance and counts the class label of the neighbors when assigning the class label of a test data point. The category with the largest neighbor count is the one to which the classifier assigns the new data points.

Figure 2.12 illustrates the KNN classification. Before applying KNN, the blue-colored data is unclassified. After applying the KNN, the data point is classified as category A. As the nearest neighbors of the data point are in category A, the data point is classified as category A.

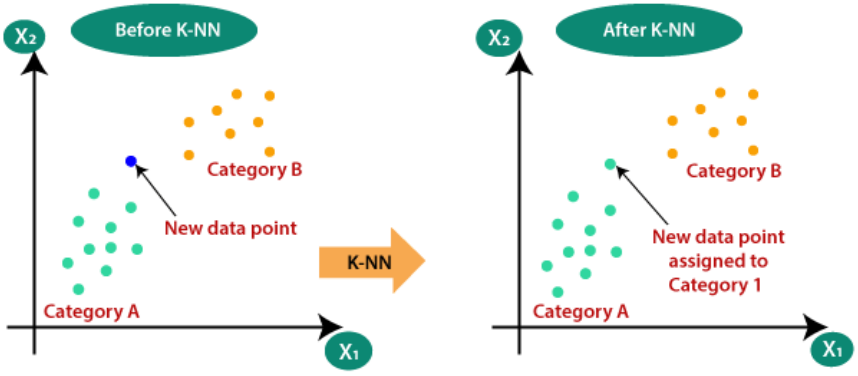

Figure 2.12: Illustration of KNN Classification.

2.6.6.2 Naive Bayes (NB)

Naive Bayes classifier or Bayesian Classifier based on Bayes theorem is a supervised learning model used in data mining and machine learning [22]. Naive Bayes classifier uses the likelihood, prior and evidence probability to determine the probability of being a specific class labeled data.

Suppose, T is an n-dimensional training data set associated with their class label $C_1, ..., C_m$. Each data point X is represented in n-dimensional vector space as $(x_1, x_2, ..., x_n)$. The NB Classier assigns class label C_i to the data point X if the posterior probability of class C_i is the highest for X. According to Bayes theorem, the posterior probability is defined as

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2.7}$$

As $P(X)$ is the same for all classes. So, $P(X|C_i)P(C_i)$ is needed to be maximized.

$$P(C_i|X) = P(X|C_i)P(C_i) \tag{2.8}$$

To reduce the computational complexity, NB assumes that the attributes are conditionally independent. Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{2.9}$$

2.6.6.3 Decision Tree (DT)

Decision Tree (DT) [23] is a supervised machine learning which can be used for both classification and regression. But, Decision Tree (DT) is basically used for classification purposes. It is a tree-like structure where the internal nodes are the features, the branches are the decision

rules, and the leaf nodes are the outcomes. Figure 2.13 illustrates a fruit classifier model using Decision Tree (DT). The root/parent nodes are the decision-making features, the branches are decision rules, and the leaf nodes are the name of the fruit.

Which feature will be the root node or which feature will be selected as the decision-making feature is decided by an attribute selection measure (ASM) score. This ASM score can be calculated by some mathematical models like Information Gain, and Gini Index.

Equation 2.10 represents the equation for Information Gain.

$$InformationGain = Entropy(Parent) - WeightedAverage * Entropy(Child) \quad (2.10)$$

Equation 2.11 represents the Gini Index. Where P is the probability of being in a class.

$$Gini = 1 - \sum_{j} P_j^2 \quad (2.11)$$

Tree pruning is used in Decision Tree (DT) for removing unwanted branches. So that the optimal tree can be achieved.

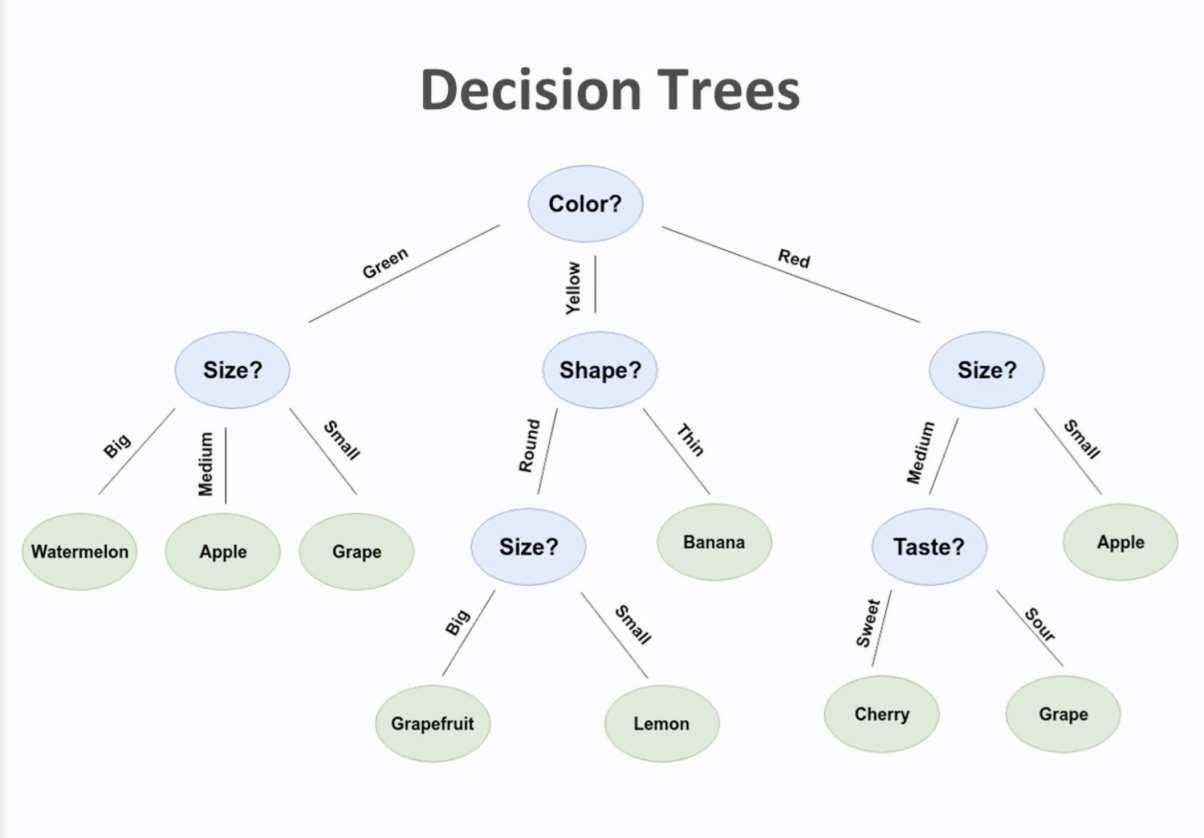

Figure 2.13: Illustration of Decision Tree Classification.

2.6.6.4 Random Forest (RF)

Random Forest (RF) [24] is a Decision Tree (DT) based supervised machine learning algorithm. It is also used for both classification and regression. Random Forest (RF) consists of not only a single decision tree but also many decision trees. It actually follows the property of ensemble learning. Figure 2.14 illustrate the building of a Random Forest (RF). The training

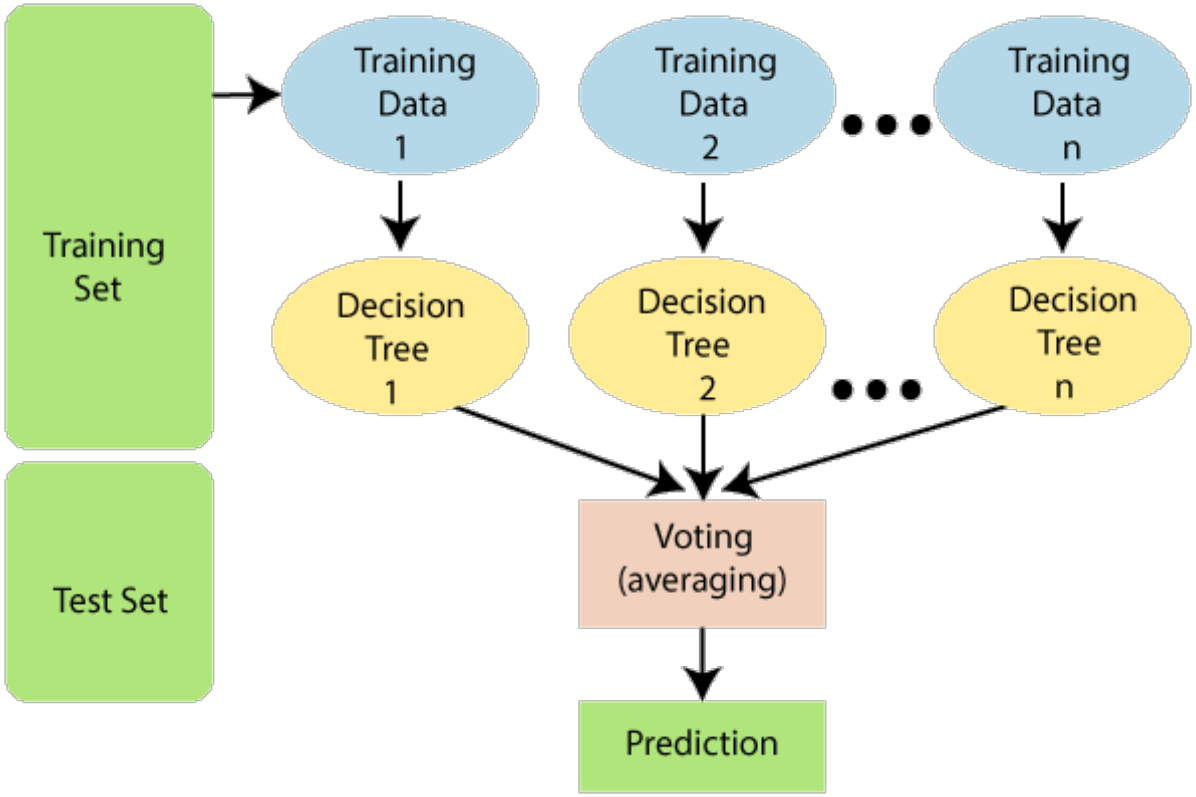

Figure 2.14: Illustration of Random Forest Classifier.

dataset is divided into n subsets. Each subset is used to train a decision tree. The test data is tested using the decision trees. Then the voting is done for the decision trees. The random forest takes the average of the decision trees based on the majority votes of prediction. That's why the random forest algorithm is not biased.

2.6.6.5 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a derived concept from the human brain. Machine learning researchers have tried to implement the concept of biological neurons in computer science. Figure 2.15 illustrates a simple Artificial Neural Network (ANN).

The attribute values are input to the input layer of the Artificial Neural Network (ANN). The

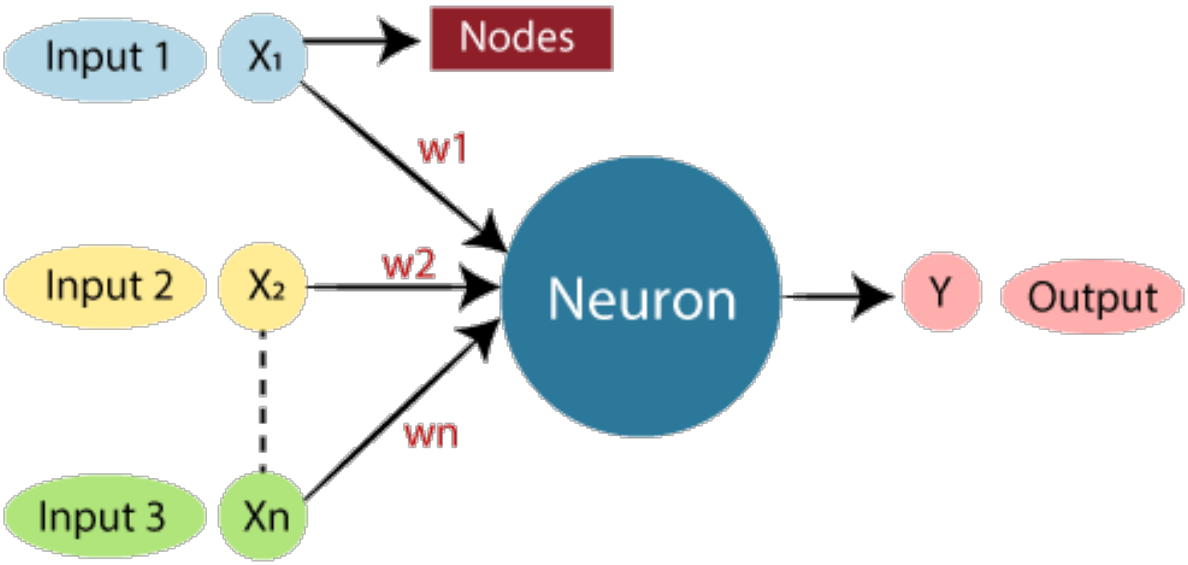

Figure 2.15: Illustration of Artificial Neural Network (ANN).

edges of the Artificial Neural Network (ANN) contain the weights of the corresponding path. The weighted sum of the inputs with biased value b is input to the neuron. Neuron compares the weighted sum with the threshold value and makes the decision for the input data sample. The input to the neuron is like in equation 2.12.

$$\sum_{i=1}^{n} x_i * w_i + b \tag{2.12}$$

2.6.6.6 Support Vector Machine (SVM)

The Support Vector Machine [25] is a supervised machine learning model mostly used for binary classification. SVM seeks to identify the ideal decision boundary that can discretize the data points of classes with the greatest margin..

Suppose, $x_1, x_2, ..., x_n$ are data points and $y_i\{-1, +1\}$ are class label of x_i. w is the weight vector of the classification model. The decision boundary described in equation 2.13 can classify the data points correctly.

$$y_i(w^T x_i + b) \geq 1 \tag{2.13}$$

This decision boundary can be found by solving the following constrained optimization problem.

$$\begin{aligned} Minimize \quad & \frac{1}{2}||w||^2 \\ Subject\ to \quad & y_i(w^T x_i + b) \geq 1 \end{aligned} \tag{2.14}$$

Here, b is the bias. Figure 2.16 depicts the Support Vector Machine (SVM).

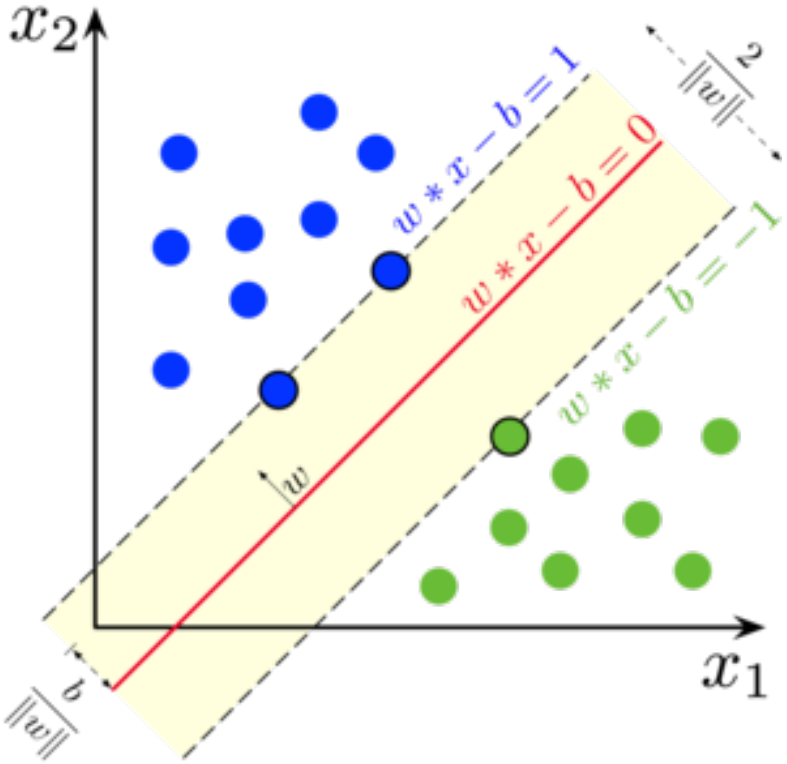

Figure 2.16: Illustration of Support Vector Machine (SVM).

The decision function becomes

$$f(x_i) = sign(w^T x_i + b) \tag{2.15}$$

Noisy data points lead to miss classification. To solve miss-classification due to noisy data, ξ is introduced. Figure 2.17 illustrate Support Vector Machine (SVM) with ξ. Another parameter C is introduced to balance the training accuracy and generalization ability. A boundary line may not always be able to distinguish between data points due to the complexity of the data. For a hyperplane to be able to divide the data points, data must be transformed to a higher dimension. The kernel function aids in this data translation into a higher dimension. The optimization problem becomes

$$\begin{aligned} Minimize_{\xi,w,b} \quad & w^T.w + C\sum_{i=1}^{l}\xi_i \\ Subject\ to \quad & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \quad i = 1, ..., l \\ & \xi_i \geq 0 \quad i = 1, ..., l \end{aligned} \tag{2.16}$$

Here, ϕ works as a data transformation function. We have to find $\alpha_1, ..., \alpha_n$ such that

$$\begin{aligned} Maximize_\alpha \ & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ Sub.\ to & \sum_{i=1}^{n} y_i \alpha_i = 0,\ 0 < \alpha_i < C \ for\ all\ i = 1, 2, ..., n \end{aligned} \tag{2.17}$$

$\phi(x_i)^T \phi(x_j)$ actually denotes the kernel function $k(x_i, x_j)$. The optimal classification function becomes

$$f(x) = \sum \alpha_i y_i k(x_i, x_j) + b \tag{2.18}$$

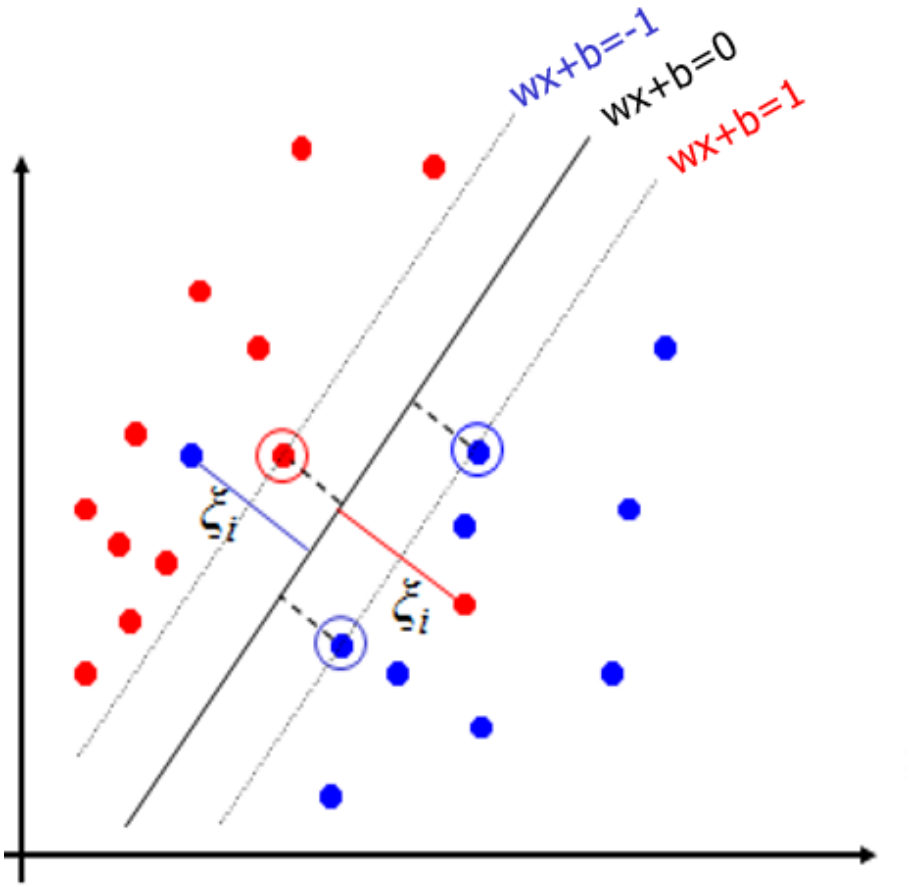

Figure 2.17: Illustration of Support Vector Machine (SVM) with ξ Variable.

As we mentioned before, sometimes, the hyperplane can't distinguish the classes in the original feature space. The kernel function is needed to transform the data in a higher dimension where the hyperplane can separate the class effectively. Linear Kernel, Polynomial Kernel, and Radial Basis Function (RBF) are some common kernel functions used in SVM.

1. **Linear Kernel**: The linear kernel is the most basic kernel function in SVM. The equation is like in 2.19. For complex data, choosing the liner kernel as kernel function is the worst

decision.

$$K(x_i, x_j) =< x_i, x_j > \tag{2.19}$$

2. **Polynomial Kernel**: For complex data, a Polynomial kernel can be used. Equation 2.20 defines the Polynomial kernel.

$$K(x_i, x_j) = (< x_i, x_j > +c)^d \tag{2.20}$$

Here, d is the order of the kernel and c is a constant that allows to trade off the influence of the higher order and lower order terms.

3. **Radial Basis Function (RBF)**: It is the mostly used kernel function for the SVM. Equation 2.21 defines the Radial Basis Function (RBF) kernel function.

$$K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{2.21}$$

Here, $||x_i - x_j||^2$ is the squared Euclidean distance between two data point x_i and x_j.

σ works as the width of the support region. When the value of the σ is small, the SVM considers the Support Vectors (SVs) and the data points, closest to the SVs, as the support region. The classification accuracy of these data points is high in number. So, the training error is less. But the testing dataset may not contain a large number of data points close to the support region. So, the testing error will be high in number. This situation is called the model over-fitting.

On the other hand, the large value of σ denotes a large support region. The data points which are not very close to the SVs are also considered in the support region. Thus, they are classified wrongly as they are different from SVs. The Training error is high in number. Similarly, when the model is applied to the testing dataset, the testing error is also high in number. Because the testing dataset may contain a large number of data points different from the SVs. This situation is called under-fitting. We need to trade off between the over-fitting and under-fitting to get the best classification model.

Radial Basis Function (RBF) has also another parameter, Cost (c). c works as the balancing parameter between the model complexity and the empirical error. Actually, the empirical error is related to the training error. When the value of the c is large, a large number of training samples are considered as SVs. The training error is small. But, the

model generalization is not proper here. So, the testing error becomes high. This is called over-fitting.

On the other hand, when the value of c is small, a small number of training data points are considered as SVs. So, the model is not generalized. So the testing error becomes high. This is called under-fitting. By trading off between over-fitting and under-fitting, we need to select the best value of c.

The best values of c and σ are chosen by tuning the model during cross-validation.

2.6.7 Model Explanation

In recent days, Model Explanation has become a buzzword in the field of Data Science and Machine Learning. Model Explanation is an AI-based method to explain the outcomes of ML models. It is utilized for describing model behavior, making better judgments, transparency, and trustworthiness. Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) are mostly used ML model explanation methods.

2.6.7.1 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) [26] is a game theory-based AI model that is used to explain the importance of each feature for prediction. The model explanation is provided by the SHAP as in Equation 2.22.

$$f(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{2.22}$$

Here, f is the explanation model. $x' \epsilon \{0,1\}^M$ is the coalition vector. M is the maximum coalition size and $\phi_i \epsilon \mathbb{R}$ is the feature attribution for a feature i.

The SHAP value has additivity, local accuracy, missingness, and consistency properties.

1. **Additivity**: The SHAP assigns a score for each feature which denotes the contribution of that particular feature for prediction. These scores can be summed up to denote the final contribution of the features cumulatively.

2. **Local Accuracy**: The SHAP values offer a precise and local interpretation of the model's forecast for a particular input.

3. **Missingness**: The SHAP assigns a zero value for a missing feature for a particular prediction. That means SHAP is responsive to the missing value issue.

4. **Consistency**: SHAP provides consistency values that are not changed due to the change of parameters.

2.6.7.2 Local Interpretable Model-Agnostic Explanations (LIME)

Local Interpretable Model-Agnostic Explanations (LIME) [27] is another method for model interpretability. The LIME determines the contribution of each feature for predictions. The specialty of LIME is that LIME is more robust to the input sample. If the changes occur in the input of the sample, the LIME changes model interpretability and observes the changes in output. Other models consider the dataset for model explainability. But the LIME can consider a single data point.

2.7 Performance Evaluation Matrices

Evaluation metrics are used to evaluate the correctness of the classification metrics. In this research, we have considered six performance metrics namely accuracy, sensitivity, specificity, Mathew's correlation coefficient (MCC), F1 score, and Area Under the Curve (AUC). These can be calculated from the confusion matrix. The confusion matrix has four outcomes namely true positive (TP), false positive (FP), false negative (FN), and true negative (TN). TP outcomes are classified as positive and their actual class labels are also positive. If the actual class label is negative and classified as positive, these are called FP. Similarly, a positive class labeled test data can be classified as negative, these cases are called FN. When the negative class labeled test data is classified correctly, then it is called TN. The performance metrics can be defined using these outcomes.

Accuracy (acc) denotes the correctness of classification.

$$\begin{aligned} acc &= \frac{TP + TN}{Total} \\ &= \frac{TP + TN}{TP + FP + FN + TN} \end{aligned} \tag{2.23}$$

Sensitivity (sen) denotes the model's ability to correctly classify the positive class labeled data. It is defined in terms of TP and FN.

$$sen = \frac{TP}{TP + FN} \tag{2.24}$$

Specificity (spe) is a similar measurement to sensitivity. Specificity measures the model's

correctness to classify the negative class data. Specificity is defined as in equation 2.25.

$$spe = \frac{TN}{TN + FP} \tag{2.25}$$

Matthew's Correlation Coefficient (MCC) is a statistic used to rate the accuracy of binary classification. MCC calculates the discrepancy between the expected and actual classes.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.26}$$

Accuracy is not always a good metric for classification model evaluation where the dataset is class imbalanced. In those cases, the F1 score can be a good choice as it considers both precision and recall at a time.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2.27}$$

Another performance indicator, Area Under the Curve (AUC), is utilized as the summary of the Receiver Operator Characteristic (ROC) and indicates the model's capability to discriminate between positive class and negative class. The true positive rate (TPR) is shown against the false positive rate (FPR) on the ROC probability curve.

$$TPR = \frac{TP}{TP + FN} \tag{2.28}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.29}$$

The value of AUC ranges [0,1]. AUC=1 denotes that the models can classify the samples perfectly. On the other hand, AUC=0 denotes that the model predicates the positive class as negative and the negative class as positive.

2.8 Literature Review

This Section represents the previous studies related to machine learning models for diabetes prediction, outlier detection methods, and oversampling methods for class imbalance issues.

2.8.1 Machine Learning Based Diabetes Prediction Models

Diabetes does not kill people directly, but it carries a higher risk of stroke. Diabetes cannot be fully cured. A healthy diet and regular exercise can help to manage diabetes effectively

[28]. If it is possible to predict diabetes in the early stage, the complications of diabetes can be reduced to a great extent. Early prediction helps patients take preventive measures against diabetes.

Using machine learning, early diabetes prediction models can be developed by analyzing the individual's current risk factors. In this regard, machine learning researchers have investigated in the past and developed early-stage prediction models. Patil, Joshi, and Toshniwal [29] used K-means clustering for outlier removal and the C4.5 classifier on the Pima Indian dataset. The best classification accuracy was attained by them, coming in at 92.38%. Another research [30], collected a large dataset consisting of 13,647,408 samples from different ethnic groups in Kuwait. They used logistic regression, K-Nearest Neighbors (KNN), Multi-factor Dimensionality Reduction (MDR), and Support Vector Machine (SVM) as classifier models. Wu, Yang, Huang, He, and Wang [31] developed a prediction model consisting of an improved version of K-means clustering and logistic regression. The outlier samples were detected by improved K-means clustering. Then, logistic regression was used as the classifier. They found out that they outperform the previous models. Alam and his team [32] used data mining and machine learning for early-stage diabetes prediction. They discovered a significant correlation between body mass index (BMI) and blood glucose levels and diabetes. As a prediction model, they employed K-means clustering, Random Forest, and Artificial Neural Networks (ANN). They concluded that ANN outperforms others and achieved 75.7% classification accuracy. Fitriyani, Syafrudin, Alfian, and Rhee [33] studied a well-known dataset named Dr. John Schorling's diabetes dataset. They used iForest for outlier removal and the synthetic minority oversampling technique Tomek link (SMOTETomek) for oversampling. For prediction, they developed an ensemble-based learning model combining Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Decision Tree as a first-level learner, and Logistic Regression (LR) as a second-level learner. They identified 9 features as important features and achieved precision: 94.49%, recall: 98.62%, F1: 96.32%, AUC: 0.99, and Accuracy: 96.74%. Islam, Ferdousi, Rahman, and Bushra [34] collected data from the Sylhet Diabetes Hospital, Bangladesh. Different machine-learning models were examined, and they came to the conclusion that Random Forest performed better than the others when it came to the data they had gathered. Islam, Rahman, Abedin, Ahammed, Ali, Ahmed, and Maniruzzaman [35] studied the National Health and Nutrition Examination Surveys (NHANES) diabetes dataset. They used Logistic Regression, Naïve Bayes, J48, Multi-layer Perceptron, and Random Forest (RF) as classifier models

and the best classification accuracy, AUC were 95.9%, and 0.946 respectively. Deberneh and Kim [36] developed a diabetes prediction model combining feature selection and classification. They used the dataset of Electronic Medical Record, Hanaro Medical Foundation, Seoul, South Korea. Using ANOVA tests, chi-squared tests, and RFE, they selected 12 features as influential features. For classification, they utilized Logistic Regression, Random Forest, SVM, and eXtreme Gradient Boosting (XGBoost) and achieved 77.87% classification as the best. Syed and Khan [37] solved the class imbalance problem using SMOTE for oversampling. They used the Chi-Square test and Binary Logistic Regression for feature selection and selected 10 features. For classification purposes, Decision Tree (DT), Support Vector Machine, Decision Jungle, Logistic Regression, Boosted DT, and Neural Network were used. Their best classification accuracy, precision, recall, AUC, and F1 score were 82.1%, 77.6%; 89.0%; 0.86, and 82.9% respectively. Hasan, Alam, Das, Hossain, and Hasan [38] developed a robust machine learning model for handling outliers, missing values, and feature selection. They used a weighted ensemble ML model consisting of KNN, DT, RF, NB, AdaBoost, and XGBoost. The developed model achieved sensitivity, specificity, and AUC as 0.789, 0.934, and 0.950. The AUC was 2.0% higher than the previous. Another Research [39] developed a fused ML method for diabetes prediction. They developed a classification framework consisting of SVM and ANN. The output of the classification works as the input of a fuzzy membership function which actually determines the class of the test sample. They achieved 94.87% classification accuracy. Tripathi and Kumar [40] proposed an early-stage diabetes prediction using Linear Discriminant Analysis (LDA), KNN, SVM, and RF. They used Pima Indian Diabetes Database. They achieved the highest accuracy of 87.66% with RF. Another research [41] collected diabetes data from an Indian district namely 'Bandipora'. They applied six ML algorithms namely RF, MLP, SVM, GB, DT, and LR on this dataset and found that RF outperforms others.

Dutta, Paul, and Ghosh [42] made unique research for analyzing the impact of features on diabetes. They studied different datasets and tried to identify the important factors on different datasets. Another research [43], focused on feature selection and dimensionality reduction, developed a diabetes prediction model. They used backward and forward feature selection methods for feature selection and Principle Component Analysis (PCA) for dimensionality reduction. SVM and RF were utilized as the classifier. The RF outperforms the SVM and achieved 83% accuracy. Laila, Mahboob, Khan, Khan, and Taekeun [44] developed an ensemble approach for an early-stage diabetes prediction model consisting of AdaBoost, Bagging, and RF. Their

highest classification accuracy was 97%.

2.8.2 Outlier Detection Method

One of the most challenging problems in machine learning is dealing with outlier samples. In computer science, the process of identifying data points that deviate from the norm is known as outlier detection. The Isolation Forest (iForest) [11] is a decision tree-based outlier detection method. Some previous research has elicited that the iForest can contribute to improving the accuracy of the classifier model by removing the outlier data points.

Ijaz, Attique, and Son [45] developed a Random Forest-based cervical cancer forecasting method. For outlier identification, they employed iForest and density-based spatial clustering of applications with noise (DBSCAN), and for oversampling, they used SMOTE and SMOTE-Tomek. They discovered that the performance of the iForest with SMOTE and SMOTETomek is superior to the DBSCAN with SMOTE and SMOTETomek. A deep learning-based heart disease prediction model was put out in another study [46] for the UCI machine learning heart disease dataset. They found that the isolation forest-based outlier removal method improved the classification accuracy by removing outlier samples. Rezaei, Woodward, Ramirez, and Munroe [47] developed a cardiovascular diseases prediction model using Isolation Forest for outlier detection, SMOTE for oversampling, and ensemble learning for classification. They studied the UK Biobank ECG repository focusing on the binary classification of Atrial Fibrillation and Ventricular Arrhythmia. Their proposed method was more effective than others in improving classification accuracy.

2.8.3 Oversampling Method for Solving Class Imbalance Issue

Class imbalance is another obstacle in machine learning which occurs when the number of positive cases and negative are not equal for binary classification problems. The classification model becomes biased to a specific class if the dataset is class imbalanced and the test result is not accurate. So, one crucial area of machine learning research is resolving the class imbalance issue. Over- and under-sampling are methods that can be used to address this issue. Data duplication from the minority class is the result of oversampling. As a potential remedy for the class imbalance issue, Synthetic Minority Oversampling Technique (SMOTE) [13] may help to raise the prediction models' classification accuracy.

Using Random Forest as a classifier, Density-based Spatial Clustering of Applications with Noise (DBSCAN) as an outlier removal technique, and Synthetic Minority Oversampling Technique (SMOTE) as the oversampling technique, Ijaz, Alan, Syafrudin, and Rhee [48] developed a type 2 diabetes and hypertension prediction model. They demonstrated how the SMOTE may address the class imbalance problem and increase the classifier model's accuracy. Another research [49] developed a model to identify diesel brands using tree-based feature selection, SMOTE for oversampling, and XGBoost-based ensemble learning for classification. According to their results, the classification gained by the combination of Tree-SMOTE-XGBoost is 19.33% higher than the accuracy gained by the XGBoost. Sridhar and Sanagavarapu [50] proposed a machine failure prediction model using SMOTE for oversampling and Random Forest for classification. They found that the AUC score increased by 7.83% if SMOTE was used as the oversampling technique.

2.9 Research Scopes

There are several scopes in diabetes prediction using Machine Learning (ML). As diabetes is a non-communicable disease, early-stage prediction of diabetes can save millions of lives. A proper set of features is needed in this regard. There are several datasets for diabetes. Missing data, outlier samples, and class imbalance are very common problems among these datasets. There are many methods for solving missing data, outlier removal, and solving class imbalance issues in ML. But which method is able to provide the best result in diabetes prediction is not up to the mark. We can study this field to develop a perfect model for diabetes prediction considering these problems and their solution.

2.10 Conclusion

In conclusion, we can say that ML researchers are doing their best to fining out a perfect disease prediction model for early-stage diabetes prediction. They have incorporated the knowledge of data mining, mathematics, and ML for building diabetes prediction models. Data pre-processing is an important part of ML studies, sometimes, it was not addressed properly in ML-based diabetes studies. Sometimes they consider the feature selection, and sometimes may not. Some of them consider ensemble-based classification models for improving the classifica-

tion accuracy of the prediction models. Perfect classification accuracy is rare in these studies. We are focusing on these issues carefully in our research.

Chapter 3

Methodology and Experimental Analysis

3.1 Introduction

This chapter explores the experimental analysis done in this research. It begins with the data collection and data description. Then, the proposed diabetes prediction model is discussed. After that, the experiments done in this research are discussed broadly. The experimental analysis begins with data pre-processing. Then, the outlier removal, feature selection, and solving of class imbalance issues are done. After each step, we have applied the classifier models to observe the gradual progress of classification and the impacts of the step for classification.

3.2 Description of the Used Dataset

We have investigated a well-known type 2 diabetes dataset that Dr. J. Schorling [51] presented. The dataset contains information on 403 of the 1046 people who took part in a research to find out how common diabetes, obesity, and other cardiovascular risk factors are among African Americans in central Virginia. The dataset contains 19 attributes namely id, chol, stab.glu, hdl, ratio, glyhb, location, age, gender, height, weight, frame, bp.1s, bp.1d, bp.2s, bp.2d, waist, hip, time.ppn. The attribute 'id' has no medical significance. It is just used to number the samples. bp.2s, bp.2d have no value for 262 samples. That means these attributes have a missing value for 65% samples. These two attributes have not been considered in the final dataset. Besides this, 29 samples have a lot of missing values for different attributes. These 29 samples have not also been considered in the final dataset. After ignoring the samples having missing value problems, the final dataset contains 374 samples and 16 attributes. Table 3.1 describes the final dataset used in this research and also represents data distributions. The

Table 3.1: Dataset Details: Attribute name and statistical description of the attributes.

Symbol	Attribute Name	Data Type	Mean	Standard Deviation	Minimum	Maximum
chol	Total cholesterol	Numeric	207.604	44.757	78.000	443.000
stab.glu	Stabilized glucose	Numeric	107.684	54.139	48.000	385.000
hdl	High density lipoprotein	Numeric	50.414	17.463	12.000	120.000
ratio	Cholesterol/hdl ratio	Numeric	4.528	1.757	1.500	19.299
glyhb	Glycosylated hemoglobin	Numeric	5.606	2.219	2.680	16.110
location	Location	Nominal	-	-	-	-
age	Age	Numeric	46.898	16.615	19.000	92.000
gender	Gender	Nominal	-	-	-	-
height	Height	Numeric	66.000	3.920	52.000	76.000
weight	Weight	Numeric	177.957	40.604	99.000	325.000
frame	A factor	Nominal	-	-	-	-
bp.1s	First systolic blood pressure	Numeric	137.396	23.185	90.000	250.000
bp.1d	First diastolic blood pressure	Numeric	83.393	13.559	48.000	124.000
waist	Waist	Numeric	37.957	5.785	26.000	56.000
hip	Hip	Numeric	43.093	5.649	30.000	64.000
time.ppn	Postprandial time when labs were drawn	Numeric	335.589	309.270	5.000	1560.000

dataset contains 58 positive samples and 316 negative samples.

3.3 Description of Proposed Diabetes Prediction Model

This section describes the proposed diabetes prediction model. The proposed prediction model combines data pre-processing, removing outlier samples, solving class imbalance issues, and classification. The missing value problem and data normalization are performed at the data pre-processing stage. For normalization, Min-Max normalization is used. The Isolation Forest is used to remove the outlier samples. After removing outliers, feature selection is performed. For feature selection, three well-known feature selection methods namely the Chi-Square test, the Minimum Redundancy and Maximum Relevancy (mRMR) test, and the Random Forest based Recursive Feature Elimination (RFE-RF) test have been used. Then, the class imbalance issue is solved by oversampling. For oversampling, the Synthetic Minority Oversampling Technique (SMOTE) is used. After oversampling, the classifier models are trained with the training samples. The Support Vector Machine, the K-Nearest Neighbors (KNN), and the Naive Bayes (NB) are utilized here as the classifier models. Then, the test samples are tested separately for each classifier model. In the result analysis stage, the classifier models are compared among themselves. The best one is also compared with the previous models. Figure 3.1 represents the

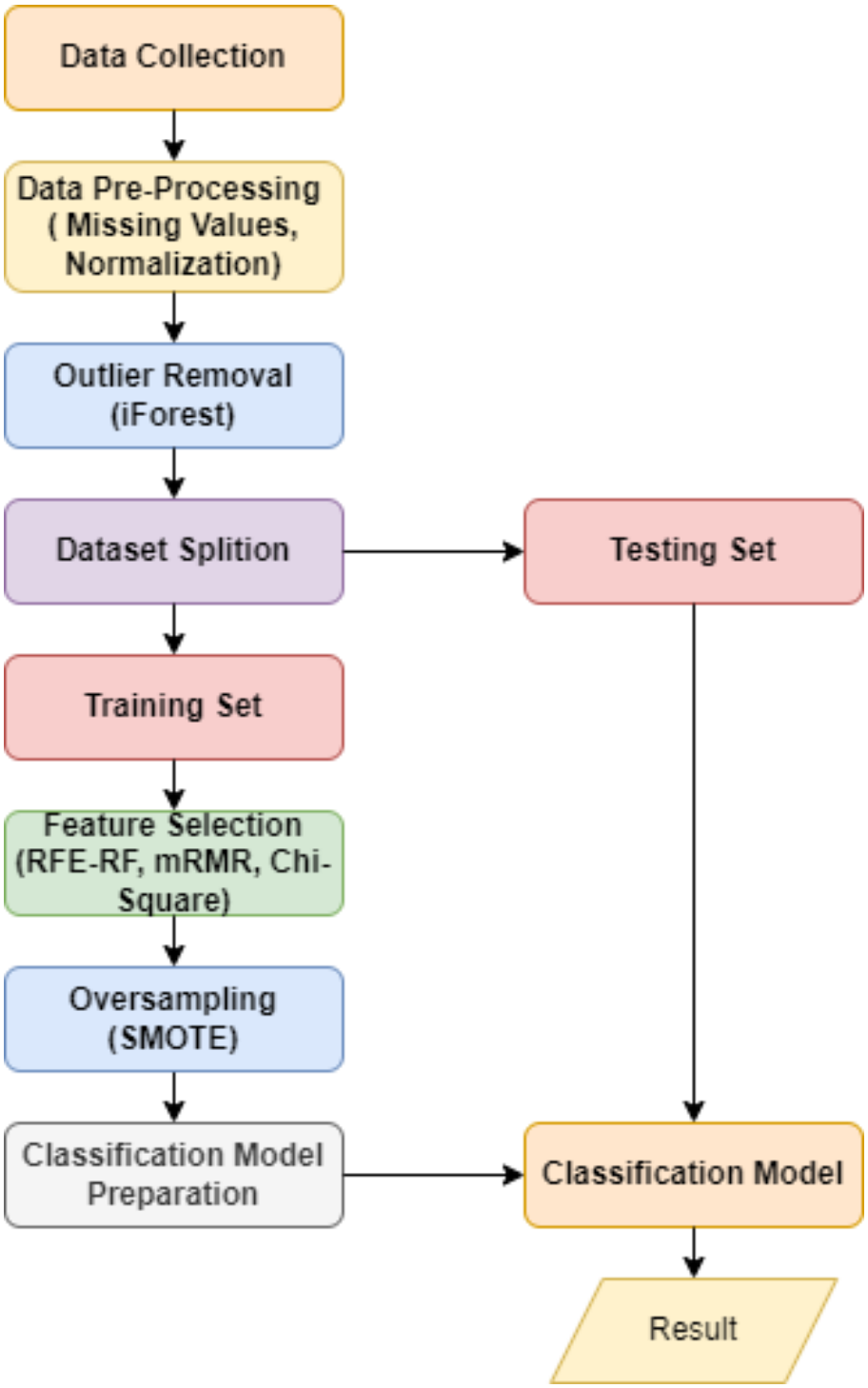

Figure 3.1: The Proposed Diabetes Prediction Model Workflow Diagram

details of the proposed diabetes prediction model.

3.4 Data Pre-Processing

The first step of the experimental analysis is data pre-processing. Actually, we have handled here three issues: encoding, missing values, and data normalization.

3.4.1 Encoding

The attributes ‘location’, ‘gender’, and ‘frame’ are nominal attributes. We have encoded them using label encoding as described in section 2.2.3.2.

The attribute ‘location’ has 2 values namely ‘Buckingham’ and ‘Louisa’. We have replaced ‘Buckingham’ by 1 and ‘Louisa’ by 2. Similarly, ‘gender’ has two values: ‘female’ and ‘male’. We have replaced ‘female’ by 0 and ’male’ by 1. The attribute ‘frame’ has 3 values: ‘small’, ‘medium’, and ‘large’. We have replaced ‘small’ by 0, ‘medium’ by 1, and ‘large’ by 2.

3.4.2 Handling Missing Values

As we have discussed in 3.2, two attributes namely bp.2s, bp.2d have missing value problems for 262 samples. These cover a huge amount of samples, about 65% samples of the total samples. The regression of these values might be faulty as the existing values are little in number. We have also concerned medical persons in this regard. According to their suggestion, we have proceeded for ignoring these attributes. These two attributes have not been considered in the final dataset. In addition to that, 29 samples have a lot of missing values for different attributes. So, these 29 samples have not also been considered in the final dataset. After ignoring the samples having missing value problems, the final dataset contains 374 samples and 16 attributes.

3.4.3 Data Normalization

For data normalization, we have used Min-Max normalization. The mathematical explanation of Min-Max normalization has been described in section 2.2.2.1. Table 3.2 describes the dataset after Min-Max normalization. The minimum value of an attribute is replaced by 0 and the maximum value is replaced by 1. Intermediary values are converted according to the equation 2.1.

Table 3.2: Dataset Details after Min-Max Normalization: Attribute name and statistical description of the attributes.

Symbol	Attribute Name	Data Type	Mean	Standard Deviation	Minimum	Maximum
chol	Total cholesterol	Numeric	0.355080	0.122621	0.000000	1.000000
stab.glu	Stabilized glucose	Numeric	0.177105	0.160649	0.000000	1.000000
hdl	High density lipoprotein	Numeric	0.355689	0.161697	0.000000	1.000000
ratio	Cholesterol/hdl ratio	Numeric	0.170117	0.098701	0.000000	1.000000
glyhb	Glycosylated hemoglobin	Numeric	0.217898	0.165201	0.000000	1.000000
location	Location	Nominal	-	-	-	-
age	Age	Numeric	0.382170	0.227610	0.000000	1.000000
gender	Gender	Nominal	-	-	-	-
height	Height	Numeric	0.583333	0.163338	0.000000	1.000000
weight	Weight	Numeric	0.349368	0.179665	0.000000	1.000000
frame	A factor	Nominal	-	-	-	-
bp.1s	First systolic blood pressure	Numeric	0.296223	0.144905	0.000000	1.000000
bp.1d	First diastolic blood pressure	Numeric	0.465698	0.178414	0.000000	1.000000
waist	Waist	Numeric	0.398574	0.192821	0.000000	1.000000
hip	Hip	Numeric	0.385105	0.166153	0.000000	1.000000
time.ppn	Postprandial time when labs were drawn	Numeric	0.212597	0.198887	0.000000	1.000000

Table 3.3: Classification Model's Performance on the Pre-processed Dataset using all the Features.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
KNN(K=5)	0.88	0.33	0.98	0.46	0.47	0.66
NB	0.97	1	0.97	0.91	0.92	0.98
SVM(RBF)	**0.962**	**0.7625**	**1**	**0.854**	**0.8649**	**0.8813**

3.4.4 Classification Result after Normalization

After the pre-processing, we have randomly selected 80% of the total sample as the training set and the rest 20% as the testing set. The training set consists of 299 samples among which 46 samples are positive cases. And, the testing set consists of 75 samples. Then, we apply the Support Vector Machine, Naive Bayes, and K-Nearest Neighbors classifier on the training dataset. For cross-validation, we use ten-fold cross-validation in all cases. We use the radial basis function (RBF) as the kernel function for the SVM classifier. After the model training, we evaluate the models using the testing dataset. Table 3.3 represents the results of the classifiers on the pre-processed dataset.

3.5 Removing Outlier Samples

After data pre-processing, we have moved to the outlier removal. For outlier removal, we have utilized a well-known outlier removal model method namely Isolation Forest (iForest). The methodology of iForest is discussed in section 2.3.3. Figure 3.2 plots the samples after

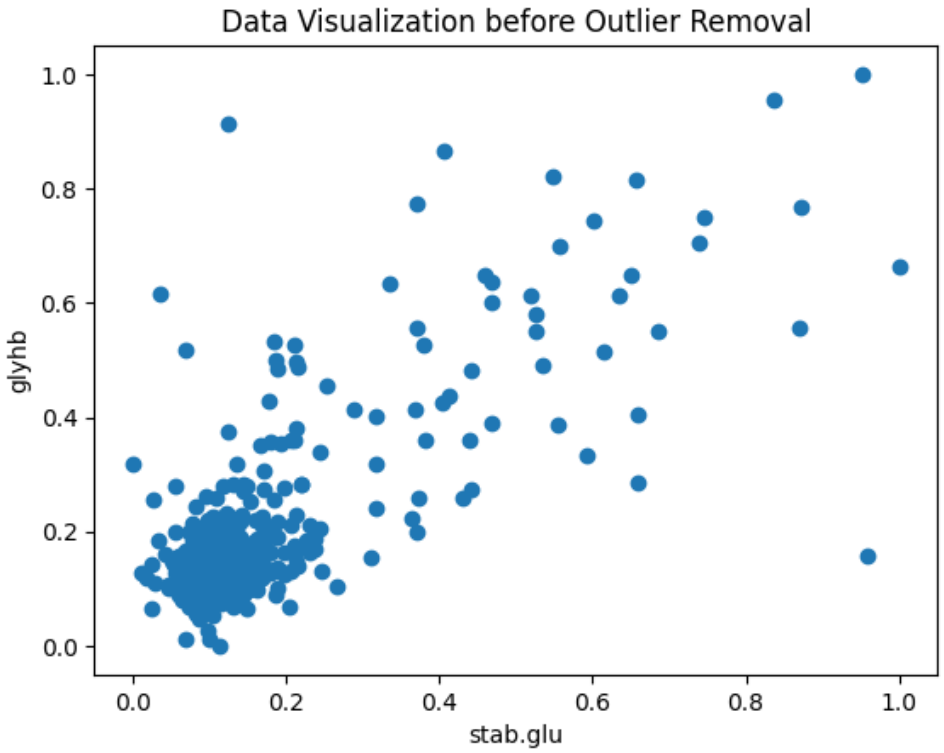

Figure 3.2: All Samples are Plotted in Respect to stab.glu and glyhb.

pre-processing with respect to the attribute 'stab.glu' and 'glyhb'.

We have used the iForest parameters as n_estimators=100, max_samples='auto', contamination=0.05. Figure 3.3 plots the samples after removing the outlier samples.

Table 3.4 describes the dataset after removing the outlier samples.

3.5.1 Splitting the Dataset & Classification after Outlier Removal

After removing the outlier samples using iForest, the dataset consists of 355 samples. We separate 80% data as training data and 20% data as testing data by random selection. Now the training set and testing set consists of 284 and 71 samples respectively. Then, we apply the classifier models to the normalized and outlier-removed data. The results of the classifier models are tabulated in Table 3.5.

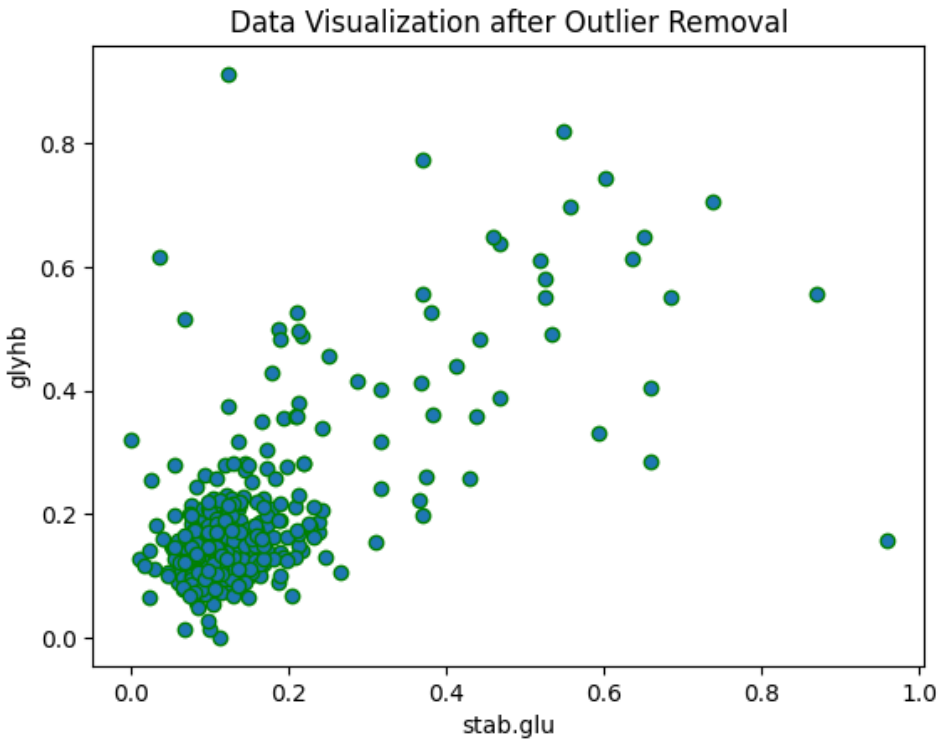

Figure 3.3: Samples are Plotted after Outlier Removal in Respect to stab.glu and glyhb.

Table 3.4: Dataset Details after Outlier Removal: Attribute name and statistical description of the attributes.

Symbol	Attribute Name	Data Type	Mean	Standard Deviation	Minimum	Maximum
chol	Total cholesterol	Numeric	0.349327	0.110344	0.109589	0.723288
stab.glu	Stabilized glucose	Numeric	0.160931	0.131800	0.000000	0.958457
hdl	High density lipoprotein	Numeric	0.356468	0.155016	0.018519	1.000000
ratio	Cholesterol/hdl ratio	Numeric	0.164884	0.084923	0.000000	0.511236
glyhb	Glycosylated hemoglobin	Numeric	0.201143	0.137978	0.000000	0.912882
location	Location	Nominal	-	-	-	-
age	Age	Numeric	0.373374	0.223266	0.000000	1.000000
gender	Gender	Nominal	-	-	-	-
height	Height	Numeric	0.581103	0.161799	0.000000	1.000000
weight	Weight	Numeric	0.343026	0.168623	0.000000	1.000000
frame	A factor	Nominal	-	-	-	-
bp.1s	First systolic blood pressure	Numeric	0.291004	0.141632	0.000000	1.000000
bp.1d	First diastolic blood pressure	Numeric	0.464344	0.178481	0.000000	1.000000
waist	Waist	Numeric	0.391737	0.184865	0.000000	0.900000
hip	Hip	Numeric	0.381939	0.161011	0.000000	1.000000
time.ppn	Postprandial time when labs were drawn	Numeric	0.212427	0.198063	0.000000	1.000000

3.6 Feature Selection

The influential features are selected using the Chi-Square test, the mRMR test, and the RFE-RF test. In all cases, we consider the top five features according to the feature ranking resulting

Table 3.5: Classification Model's Performance on the Outlier Removed Dataset using all the Features.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
KNN(K=5)	0.9	0.22	1	0.44	0.37	0.61
NB	0.972	1	0.97	0.89	0.9	0.984
SVM(RBF)	**0.9782**	**0.85**	**0.9968**	**0.8983**	**0.9073**	**0.9234**

Table 3.6: Selected Feature's Name from the Chi-Square Test and their Corresponding Chi-Square Score

Feature Name	Chi-Square Score
Glycosylated hemoglobin	0.942140826
Stabilized glucose	0.648116177
Age	0.276345332
Cholesterol/hdl ratio	0.253842938
Waist	0.204076578

Table 3.7: Selected Feature's Name from the mRMR Test and their Corresponding mRMR Score

Feature Name	Chi-Square Score
Glycosylated hemoglobin	0.6569431
Stabilized glucose	0.09282437
Age	0.005962992
Cholesterol/hdl ratio	0.004957832
First diastolic blood pressure	0.00004741571

from the feature selection methods. The name of the selected features and the corresponding Chi-Square scores are tabulated in Table 3.6. Figure 3.4 represents the bar chart of the feature name and score of selected features from the Chi-Square test.

Table 3.7 tabulates the selected feature names and the mRMR scores. Figure 3.5 represents the bar chart of the selected features from the mRMR test.

The selected features and the RFE-RFE scores are tabulated in Table 3.8. Figure 3.6 represents the bar chart of the selected features from the RFE-RF test.

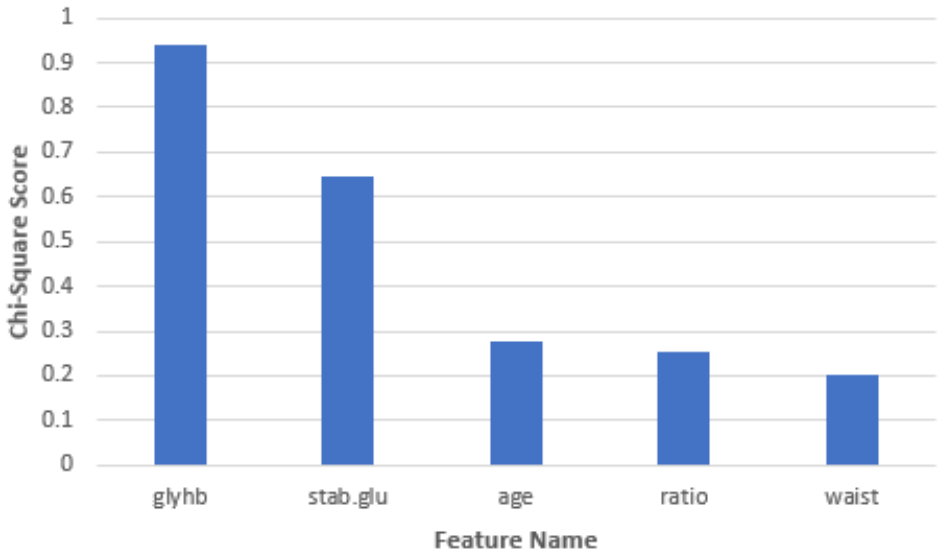

Figure 3.4: Bar Chart of the Chi-Square Score Values of the Top Five Features Selected from the Chi-Square Test

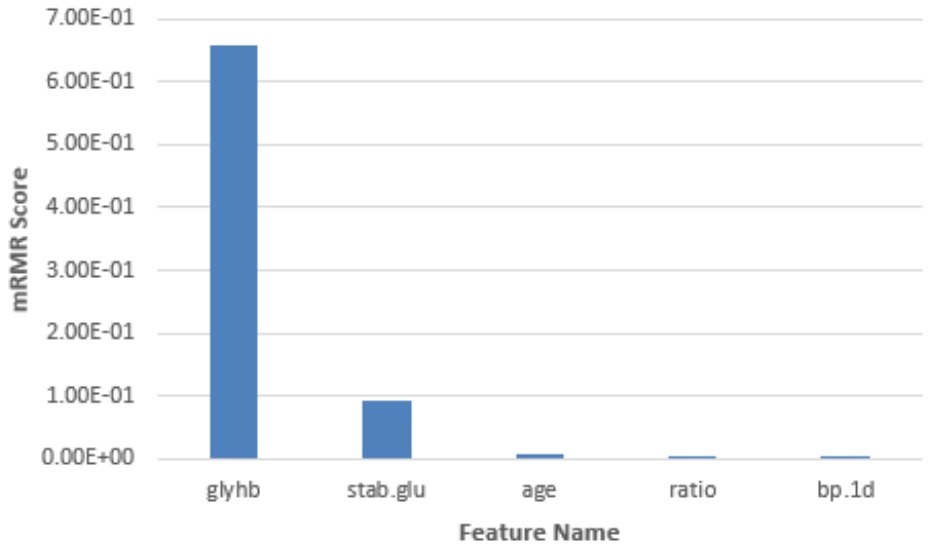

Figure 3.5: Bar Chart of the mRMR Score Values of the Top Five Features Selected from the mRMR Test

3.6.1 Classification after Feature Selection

After feature selection, new datasets are compiled. Now the training datasets consist of 284 samples with 5 features and the testing dataset consists of 71 samples with 5 features. The KNN is applied to the datasets with the selected features from different feature selection methods. The KNN classifier model is programmed in **R** programming language using the **caret** package. The ten-fold cross-validation is used during the training of the classifier model. The best K value

Table 3.8: Selected Feature's Name from the RFE-RF Test and their Corresponding RFE-RF Score

Feature Name	Chi-Square Score
Glycosylated hemoglobin	41.55193166
Stabilized glucose	10.90469737
Age	2.606353872
Waist	2.215980719
Frame	1.327181895

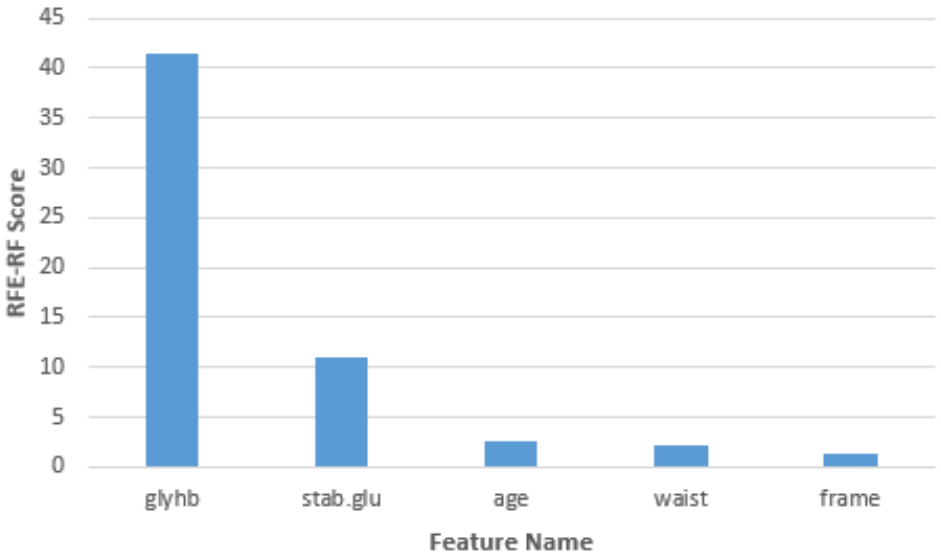

Figure 3.6: Bar Chart of the RFE-RF Score Values of the Top Five Features Selected from the REF-RF Test

is selected during cross-validation and considered in the testing phase. The performance of the KNN classifier for different feature selection methods is tabulated in Table 3.9.

We apply the Naive Bayes classifier to the data set consisting of the selected features from different feature selection methods. We use the Laplace smoothing with the value 3 in the Naive Bayes classifier. We also program the Naive Bayes classifier in **R** programming language using the **e1071** package. The results of Naive classifiers are recorded in Table 3.9.

Next, we apply the Support Vector Machine as the classifier on the training datasets with selected features from different feature selection methods. We use the radial basis function (RBF) as the kernel function in SVM. For cross-validation, we follow the ten-fold cross-validation method here. During the model tuning, we consider the cost and gamma value in the range between $[-2^8, 2^8]$. We also consider the class weighting factor during the model tuning. The

Table 3.9: Performance of the Classifiers for Each Feature Selection Technique.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
Chi-square Test						
KNN(K=5)	0.9577465	0.7777778	0.983871	0.8014133	0.8235294	0.8808
NB	0.971831	1	0.9677419	0.8898252	0.9	0.9839
SVM(RBF)	**0.9950704**	**0.9777778**	**0.9975806**	**0.9776788**	**0.9803922**	**0.9876792**
mRMR Test						
KNN(K=5)	0.9577465	0.7777778	0.983871	0.8014133	0.8235294	0.8808
NB	0.971831	1	0.9677419	0.8898252	0.9	0.9839
SVM(RBF)	**0.9943662**	**0.9777778**	**0.9967742**	**0.974552**	**0.9777778**	**0.987276**
RFE-RF Test						
KNN(K=5)	0.915493	0.444444	0.983871	0.5569598	0.5714286	0.7142
NB	0.971831	1	0.9677419	0.8898252	0.9	0.9839
SVM(RBF)	**0.993662**	**0.9666667**	**0.9975806**	**0.9695377**	**0.9722222**	**0.9821237**

best model from the model tuning is selected as the final SVM classifier. We program the SVM classifier in **R** programming language using the **e1071** package. The performance of the SVM classifier on the different selected feature sets is tabulated in Table 3.9.

3.7 Solving Class Imbalance Issue

The current training datasets are class imbalanced. The number of positive class samples and negative class samples are 249 and 35 samples respectively. We have applied the SMOTE for oversampling to solve the class imbalance issue on the feature-selected training datasets. After solving the class imbalanced issue, the positive class and negative class contain an equal number of samples and it is 249. We also apply the SMOTE on the training dataset which contains all features. Table 3.10

Figure 3.7 plots the newly generated dataset after Over-Sampling.

3.7.1 Classification after Solving Class Imbalance Dataset

After solving the class imbalance issue using the SMOTE, we have applied the classifiers on the class-balanced training dataset for building the classifier model. The performance of the

Table 3.10: Dataset Details after Over-Sampling: Attribute name and statistical description of the attributes.

Symbol	Attribute Name	Data Type	Mean	Standard Deviation	Minimum	Maximum
chol	Total cholesterol	Numeric	0.364967	0.116052	0.109589	0.723288
stab.glu	Stabilized glucose	Numeric	0.256200	0.189106	0.000000	0.869436
hdl	High density lipoprotein	Numeric	0.332348	0.145777	0.018519	1.000000
ratio	Cholesterol/hdl ratio	Numeric	0.185438	0.095211	0.033708	0.511236
glyhb	Glycosylated hemoglobin	Numeric	0.331246	0.204154	0.000000	0.912882
location	Location	Nominal	-	-	-	-
age	Age	Numeric	0.446086	0.207618	0.000000	1.000000
gender	Gender	Nominal	-	-	-	-
height	Height	Numeric	0.573544	0.150191	0.000000	1.000000
weight	Weight	Numeric	0.356923	0.155539	0.000000	1.000000
frame	A factor	Nominal	-	-	-	-
bp.1s	First systolic blood pressure	Numeric	0.321222	0.137933	0.050000	1.000000
bp.1d	First diastolic blood pressure	Numeric	0.473869	0.163778	0.026316	0.973684
waist	Waist	Numeric	0.429987	0.184626	0.000000	0.900000
hip	Hip	Numeric	0.402847	0.147792	0.058824	1.000000
time.ppn	Postprandial time when labs were drawn	Numeric	0.217855	0.194281	0.000000	1.000000

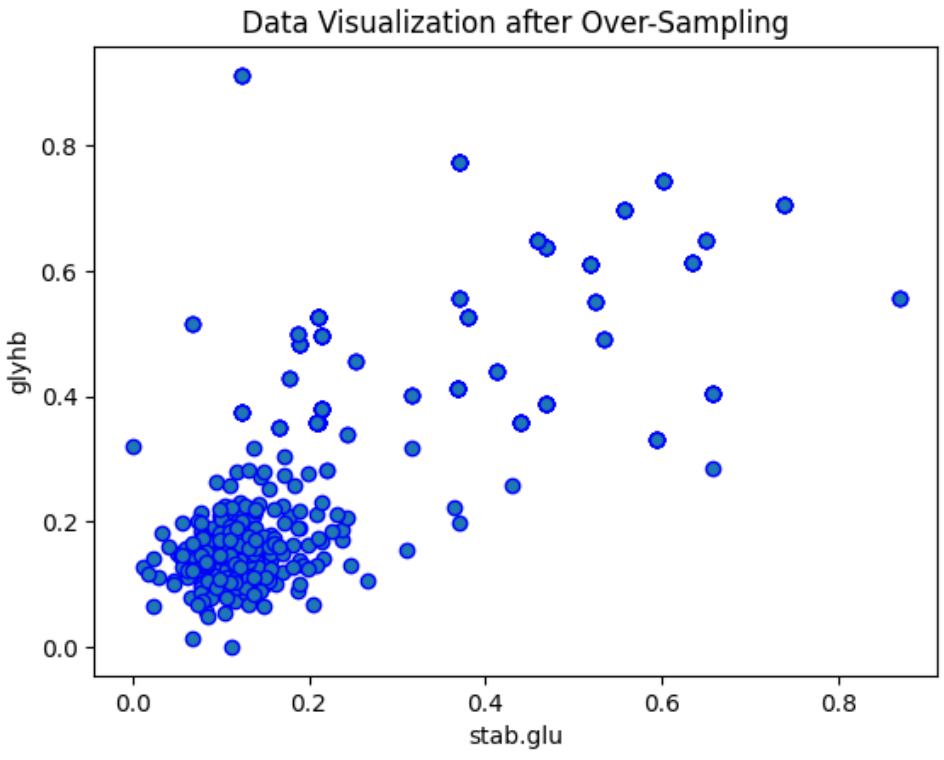

Figure 3.7: All Samples are Plotted after Over-Sampling in Respect to stab.glu and glyhb.

classifier models on the different feature-selected datasets and all feature-contained datasets are recorded in Table 3.11.

Table 3.11: Performance of the Classifiers after Oversampling.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
All Features						
KNN(K=5)	0.8591549	0.6666667	0.8870968	0.4763824	0.5454545	0.7769
NB	0.9577465	1	0.9516129	0.8448134	0.8571429	0.9758
SVM(RBF)	**0.9823944**	**0.8611111**	**1**	**0.9184793**	**0.9246324**	**0.9305556**
Chi-square Test						
KNN(K=5)	0.9577465	1	0.9516129	0.8448134	0.8571429	0.9758
NB	0.971831	1	0.9677419	0.8898252	0.9	0.9839
SVM(RBF)	**0.9957746**	**0.9777778**	**0.9983871**	**0.9801413**	**0.9823529**	**0.9880824**
mRMR Test						
KNN(K=5)	0.9577465	1	0.9516129	0.8448134	0.8571429	0.9758
NB	0.9577465	1	0.9516129	0.8448134	0.8571429	0.9758
SVM(RBF)	**0.9953052**	**0.962963**	**1**	**0.9784322**	**0.9803922**	**0.9814815**
RFE-RF Test						
KNN(K=5)	0.9577465	1	0.9516129	0.8448134	0.8571429	0.9758
NB	0.943662	1	0.9354839	0.8047625	0.8181818	0.9677
SVM(RBF)	**0.9906103**	**0.9259259**	**1**	**0.9560093**	**0.9583333**	**0.962963**

3.8 Model Validation on Another Dataset

To validate the model and observe the performance of the model for other diabetes datasets, we have applied our proposed model to a dataset collected at Sylhet Diabetes Hospital, Bangladesh.

3.8.1 Dataset Description and Pre-Processing

The dataset [52] consists of 520 samples and 17 attributes. The attribute names are Age, Sex, Polyuria, Polydipsia, Sudden Weight Loss, Weakness, Polyphagia, Genital Thrush, Visual Blurring, Itching, Irritability, Delayed healing, Partial Stiffness, Alopecia, Obesity, Class. The attribute 'Age'is a numeric attribute. Others are nominal attributes and values are in 'yes', 'no'. The dataset doesn't have the missing value problem.

The attribute 'Age'has been normalized with the Min-Max normalization. The 'yes'and 'no'values are encoded to 1 and 0 respectively.

Table 3.12: Classification Model's Performance for Validation Dataset after Removing the Outliers.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
KNN(K=5)	0.8989899	0.8644068	0.95	0.801273	0.9107143	0.9072
NB	0.8585859	0.8474576	0.875	0.7134453	0.877193	0.8612
SVM(RBF)	0.97979798	0.966101695	1	0.959216791	0.982758621	0.983050847

Table 3.13: Classification Model's Performance for Validation Dataset after Feature Selection.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
KNN(K=5)	0.8484848	0.8135593	0.9	0.7011896	0.8648649	0.8568
NB	0.8585859	0.8474576	0.875	0.7134453	0.877193	0.8612
SVM(RBF)	0.9505051	0.9423729	0.9625	0.8992575	0.9577867	0.9524364

3.8.2 Outlier Removal and Splitting the Dataset

According to the diabetes prediction model, the next step is to remove the outlier samples. The outlier samples are detected and removed by iForest. After outlier removal, the dataset contains 494 samples.

As the dataset doesn't contain a training dataset and testing dataset separately, we have to split the dataset into training and testing datasets. We have randomly selected 80% of the total samples as the training dataset and 20% as the testing dataset. The training set contains 395 samples and the testing set contains 99 samples. Among the training samples, 237 samples are positive cases. The rest are negative cases.

3.8.2.1 Classification after Removing the Outliers

After removing the outlier samples, we have applied the classification models. We have considered all the features this time. For cross-validation, we have ten-fold cross-validation. The results of the KNN, NB, and SVM are tabulated in Table 3.12.

3.8.3 Feature Selection for Validation Dataset

We have applied the χ^2 test for feature selection. We have compiled the new training set and testing set with the selected feature. Then, we have applied the classifiers on the new training dataset and tested the testing dataset. The test result is recorded in Table 3.13.

Table 3.14: Classification Model's Performance for Validation Dataset after Over-Sampling.

Classifiers Name	Acc	Sen	Spe	MCC	F1	AUC
KNN(K=5)	0.8383838	0.7627119	0.95	0.7003569	0.8490566	0.8564
NB	0.8585859	0.8474576	0.875	0.7134453	0.877193	0.8612
SVM(RBF)	0.95959596	0.949152542	0.975	0.917566588	0.965517241	0.962076271

3.8.4 Over-Sampling for Validation Dataset

After feature selection, we have over-sampled the training dataset. We have used the SMOTE method for over-sampling. Now the training dataset contains 474 samples among which 237 samples are positive cases and 237 samples are negative cases. Then, we have applied the classification models on the training dataset and built the classification models. For cross-validation (CV), we have followed the ten-fold CV. After that, we have tested the test dataset with classifiers. The test results are tabulated in Table 3.14.

3.9 Conclusion

For early-stage disease prediction, we need the proper set of features. Due to the constraints like missing values, and outliers in the datasets, the feature selection methods might be faulty. That's why we have handled these issues carefully and tried to find the set of features that are actually needed to predict diabetes at an early stage. Most of the selected features from the test dataset are the result of physical observation or simple pathological tests. No features are related to complex hormonal tests. So, these selected features help to predict diabetes in the early stage.

Chapter 4

Result Analysis

4.1 Introduction

This chapter discusses the results of the different experiments done in the Experimental Analysis Chapter. Comparative analysis of the results is also discussed in this chapter. We have tried to explore why the models are performing so, and why the results have been improved or deteriorated. The similarities among the results of the feature selection models are also discussed. Finally, the classification model is explained using an AI-Based method namely SHapley Additive exPlanations (SHAP). First, we have discussed the result of the classifiers after data pre-processing in section 4.2.

4.2 Classification Results after Pre-Processing

This section discusses the results of the classifiers after data pre-processing. Table 3.3 represents the results of the classifiers on pre-processed data. We have used three well-known classifiers namely K Nearest Neighbor (KNN), Naive Bayes (NB), and Support Vector Machine (SVM). From Table 3.3, it is clear that the classification accuracy of the NB is better than others. The NB has achieved 97.00% classification accuracy. The sensitivity, specificity, MCC, F1 Score, and AUC are 100%, 97%, 0.91, 0.92, and 0.98. The SVM also has achieved good classification results. The accuracy, sensitivity, specificity, MCC, F1 Score, and AUC are 96.2%, 76.25%, 1%, 0.854, 0.8649, and 0.8813. The classification accuracy gained by NB is 0.8% greater than the accuracy gained by the SVM.

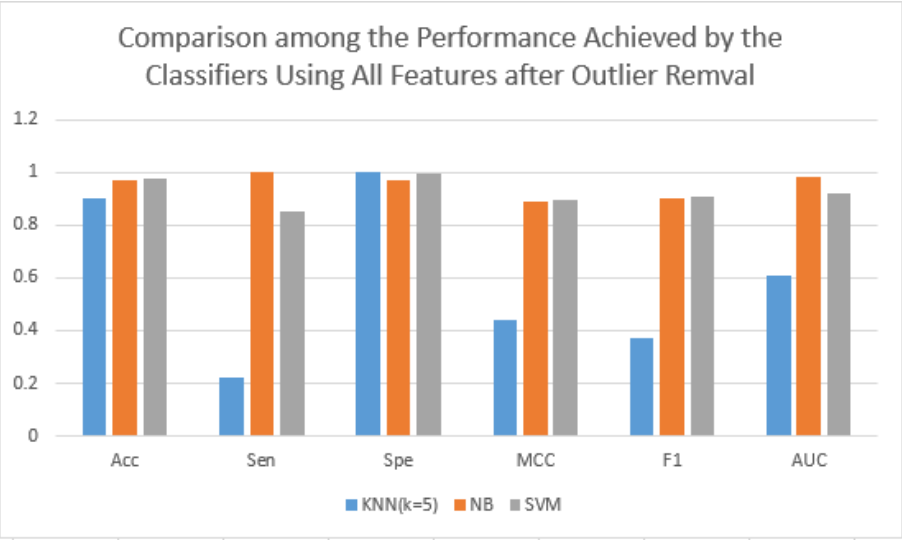

Figure 4.1: Comparison among the Results Achieved by the Classifiers Using All Features.

4.3 Classification Result after Outlier Removal

This section discusses the results gained after removing the outlier samples from the pre-processed dataset. These results have been recorded in Table 3.5. The SVM has achieved 97.82% classification accuracy. The sensitivity, specificity, MCC, F1 Score, and AUC are as 85%, 99.68%, 0.8983, 0.9073, and 0.9234. The KNN has achieved the accuracy, sensitivity, specificity, MCC, F1 Score, and AUC as 90%, 22%, 100%, 0.44, 0.37, and 0.61. Figure 4.1 illustrates the comparison among the results of the classifiers using all features.

4.3.1 Comparison between the Classification Results after Pre-Processing and Classification Results after Outlier Removal

After outlier removal, the classification performances of the KNN, NB, and SVM have been improved. The classification accuracy of KNN and NB have been increased by 2% and 0.2% respectively. The SVM has achieved 97.82% classification accuracy which is 1.62% greater than the accuracy gained on the pre-processed data. Figure 4.2 represents a line chart that compares the results achieved by the SVM on the pre-processed data and outlier removed data.

From Figure 4.2, it is clear that the performance increases in all cases of the evaluation matrices.

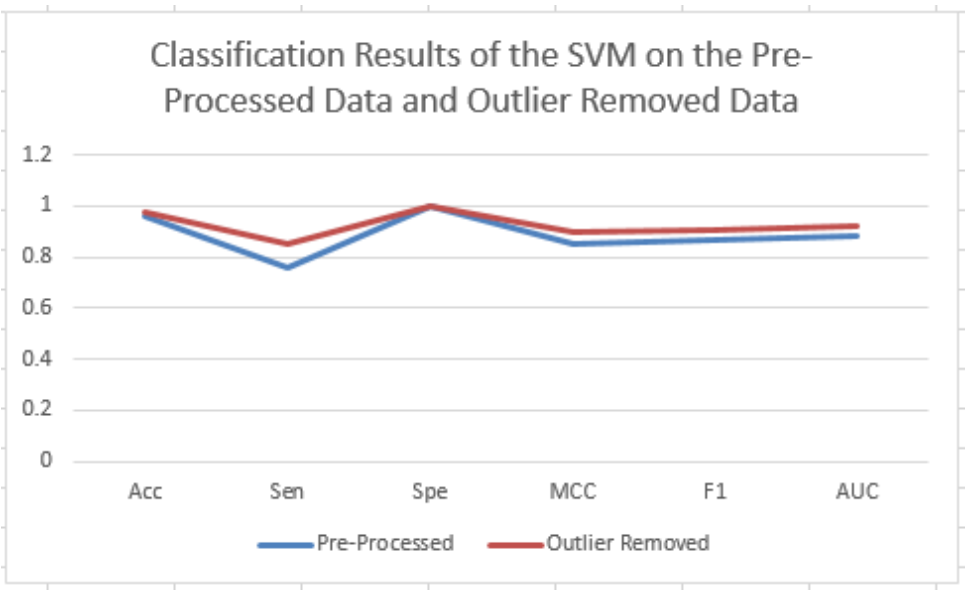

Figure 4.2: Comparison between the Results Achieved by the SVM on Pre-Processed Data and Outlier Removed Data.

4.4 Results of Feature Selection

The results of feature selection methods are described in section 3.6. The selected features for the χ^2 test are listed in Table 4.1.

Feature Name
Glycosylated hemoglobin
Stabilized glucose
Age
Cholesterol/hdl ratio
Waist

Table 4.1: Name of the Selected Feature from the χ^2 Test.

From the Table 3.6, the 'Glycosylated hemoglobin' has scored highest. 'Stabilized glucose' is the second highest scorer feature.

The selected features from the mRMR test are listed in Table 4.2

According to Table 3.7, 'Glycosylated hemoglobin' and 'Stabilized glucose' are also the top two selected features from the mRMR test.

Table 4.3 listed the selected features from the RFE-RF test. From Table 3.8, it is seen that

Feature Name
Glycosylated hemoglobin
Stabilized glucose
Age
Cholesterol/hdl ratio
First diastolic blood pressure

Table 4.2: Name of the Selected Feature from the mRMR Test.

'Glycosylated hemoglobin' and 'Stabilized glucose' are also the top two selected features from the RFE-RF test.

Feature Name
Glycosylated hemoglobin
Stabilized glucose
Age
Waist
Frame

Table 4.3: Name of the Selected Feature from the RFE-RF Test.

Analyzing the Table 4.1, 4.2, and 4.3, we can say that 'Glycosylated hemoglobin', 'Stabilized glucose', and 'Age' are common in all the feature selection tests. So these three features have a great impact on diabetes predictions. Besides these, 'Cholesterol/hdl ratio' is common in the χ^2 test and mRMR test. 'Waist' is common in the χ^2 test and RFE-RF test.

4.5 Classification Results Using the Selected Features

After feature selection, the datasets have been complied with the selected feature from different feature selection methods. The classifier models have been applied to the new training datasets. The classification test results for different feature selection methods have been recorded in Table 3.9.

For the χ^2 test, the SVM has achieved the highest classification performance. The SVM has achieved the accuracy, sensitivity, specificity, MCC, F1 Score, and AUC as 99.51%, 0.97.78%,

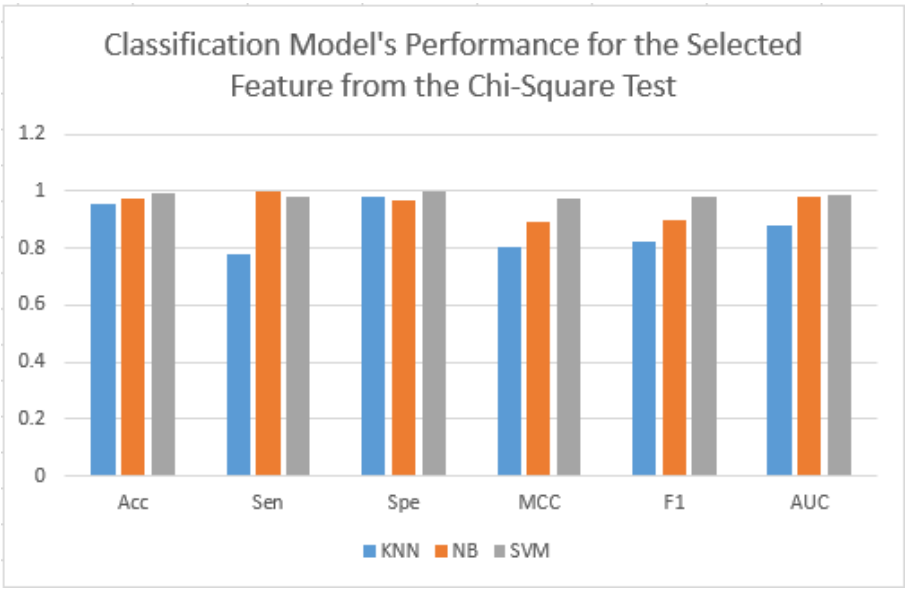

Figure 4.3: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the χ^2 Test.

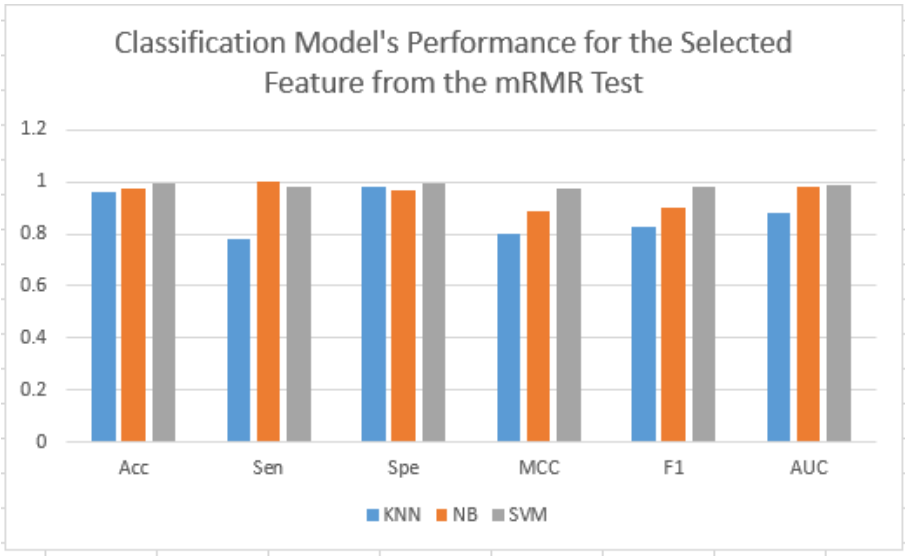

Figure 4.4: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the mRMR Test.

99.76%, 0.9776788, 0.9803922, and 0.9876792. Figure 4.3 illustrates the comparison among the results achieved by the classifiers for the selected feature from the Chi-Square test.

The SVM outperforms other classifiers for the selected feature from the mRMR test. The SVM has achieved the accuracy, sensitivity, specificity, MCC, F1 Score, and AUC as 99.44%, 97.78%, 99.68%, 0.974552, 0.9777778, and 0.987276. Figure 4.4 illustrates the comparison among the results achieved by the classifiers for the selected feature from the mRMR test.

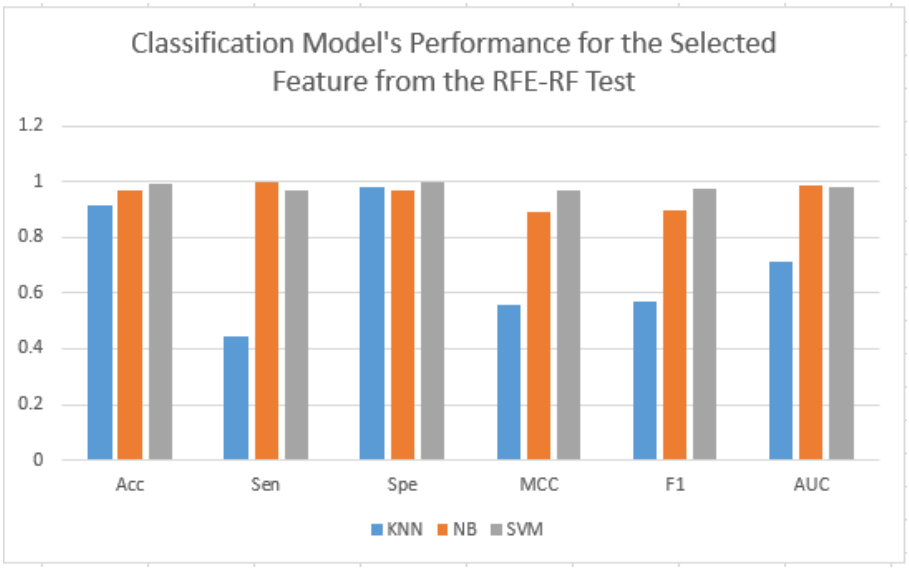

Figure 4.5: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the RFE-RF Test.

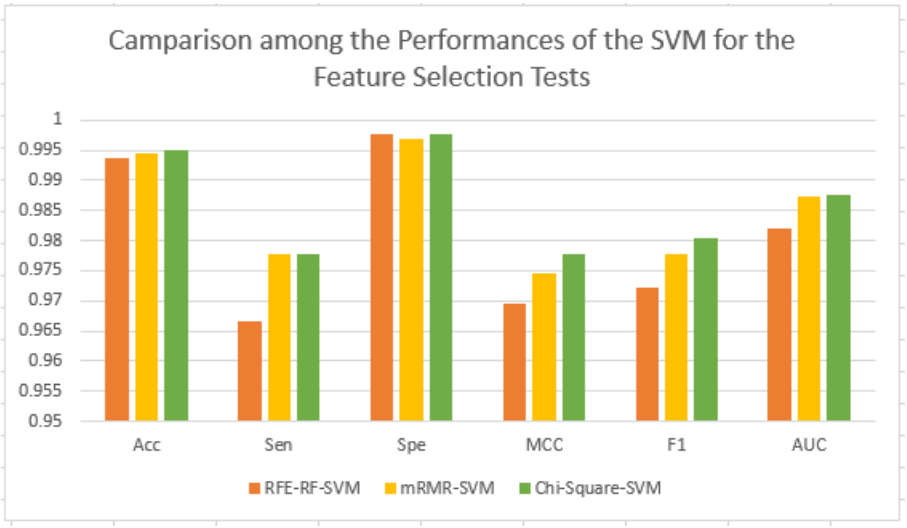

Figure 4.6: Comparison among the Results Achieved by the SVM for the Selected Feature from Different Feature Selection Tests.

From Figure 4.5, it is seen that the SVM outperforms other classifiers for the selected feature from the RFE-RF test.

For all the feature selection cases, the SVM outperforms other classifiers. We have also compared the results achieved by the SVM for different feature selection cases. From Figure 4.6, we can say the overall result achieved by the SVM for the selected features from the Chi-Square test is better than others. From Figure 4.1, we can say that the SVM has also performed better for all features. In the next section, we will compare these two results.

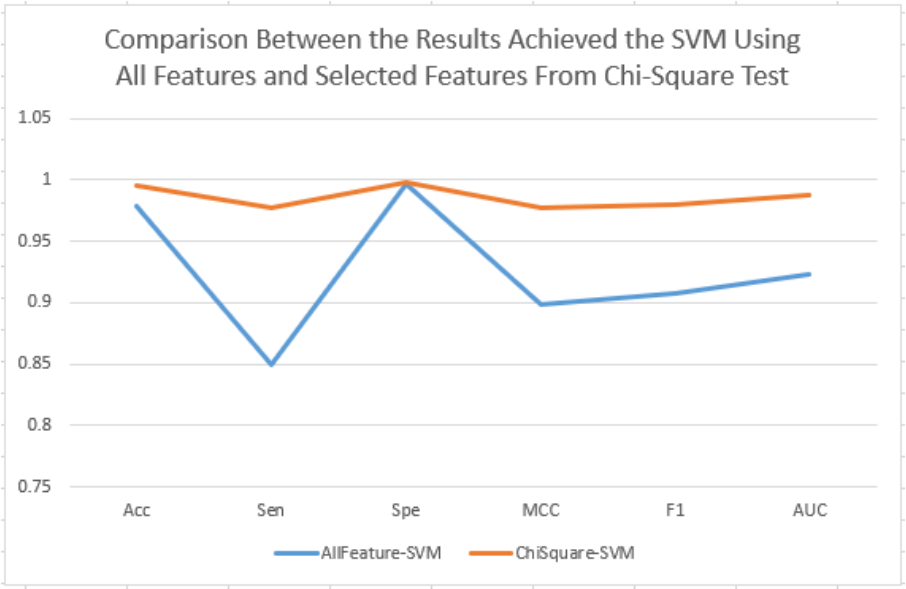

Figure 4.7: Comparison Between the Results Achieved the SVM Using All Features and Selected Features from the Chi-Square Test.

4.5.1 Comparison between the Classification Results Achieved by the SVM Using All Features and Selected Features from the Chi-Square Test

In this section, we have compared the results achieved by the SVM using all features with the results achieved by the SVM using the selected features from the Chi-Square test. Figure 4.7 illustrates this comparison. From Figure 4.7, we can decide that the classification performance of the SVM has been increased after feature selection. After feature selection, the accuracy, sensitivity, specificity, MCC, F1 Score, and AUC are increased by 1.69014%, 12.77778%, 0.08064%, 0.0794156, 0.0731128, and 0.0642921.

4.6 Classification Results after Over-Sampling

As the dataset is class imbalanced, we have to balance the dataset using over-sampling. For over-sampling, we have used the SMOTE. Section 3.7 describes the experiments related to over-sampling using the SMOTE.

After over-sampling, we have applied the classifiers on the training dataset with the selected features, and results are recorded in Table 3.11. After over-sampling, for the training dataset with the selected features from the χ^2 test, the SVM outperforms others. The SVM has achieved the accuracy, sensitivity, specificity, MCC, F1 Score, and AUC as 99.58% 97.78%

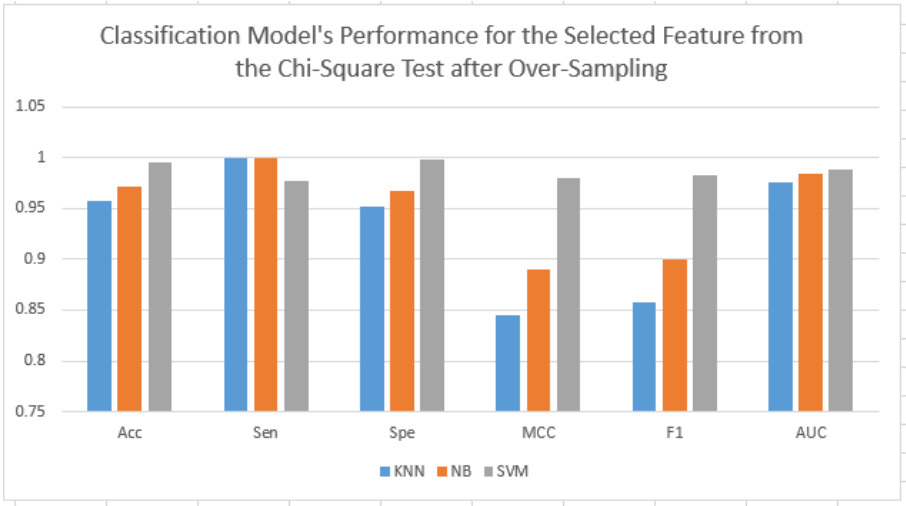

Figure 4.8: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the χ^2 Test after Over-Sampling.

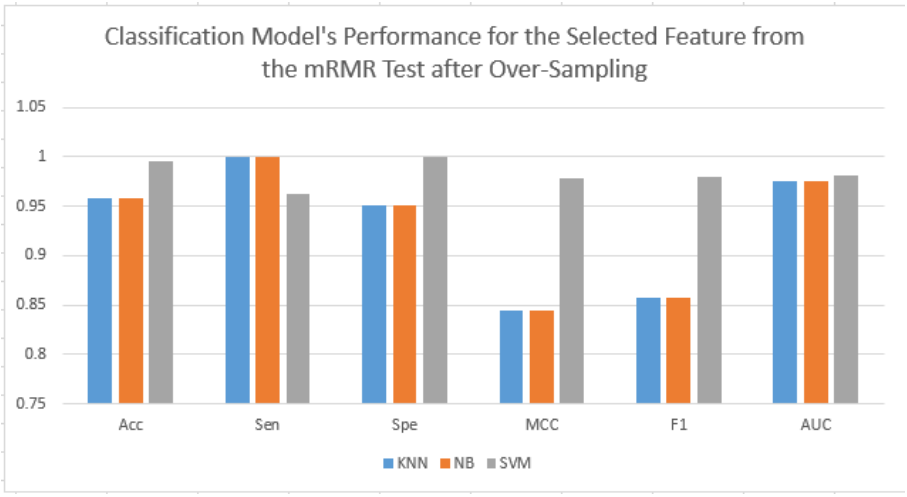

Figure 4.9: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the mRMR Test after Over-Sampling.

99.84%, 0.9801413, 0.9823529, and 0.9880824. Figure 4.8 illustrates the comparison among the classifiers for the dataset with the selected features from the Chi-Square test after oversampling.

Figure 4.9 illustrates the comparison among the classifiers for the dataset with the selected features from the mRMR test after oversampling. The SVM also outperforms other classifiers.

From Figure 4.10, it is seen that the SVM has achieved better classification performance than the KNN and NB.

Analysing Figures 4.8, 4.9, and 4.10, we can say that the SVM performs better after over-

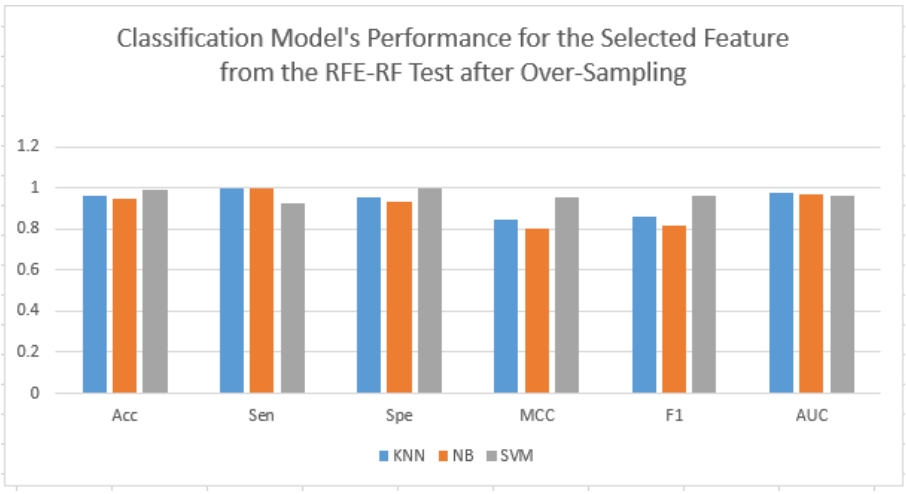

Figure 4.10: Comparison among the Results Achieved by the Classifiers for the Selected Feature from the RFE-RF Test after Over-Sampling.

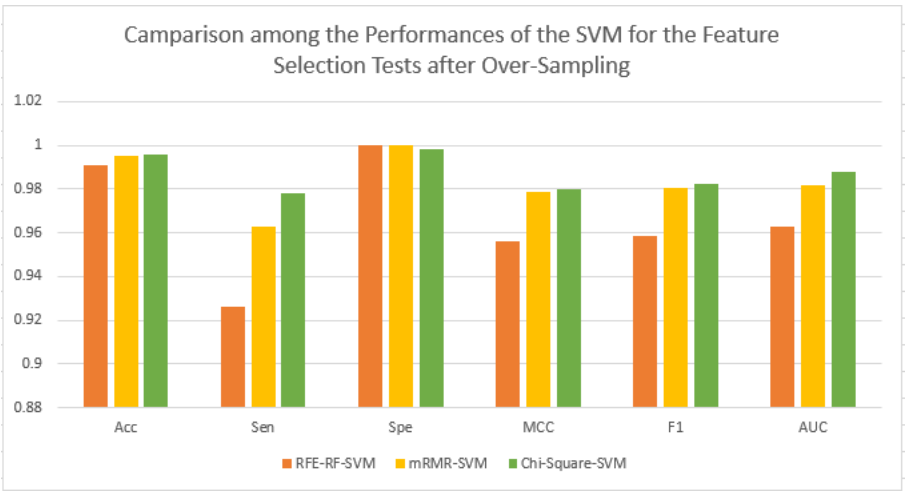

Figure 4.11: Comparison among the Results Achieved by the SVM after Over-Sampling.

sampling in all cases.

From Figure 4.11, we can decide that the Chi-Square-SVM performs better than the mRMR-SVM and RFE-RF-SVM.

Figure 4.12 illustrates the comparison between the results achieved by the SVM after feature selection and the results achieved by the SVM after over-sampling. We can say that the performance of the SVM has been improved slightly after over-sampling and it is also the best result achieved in this research.

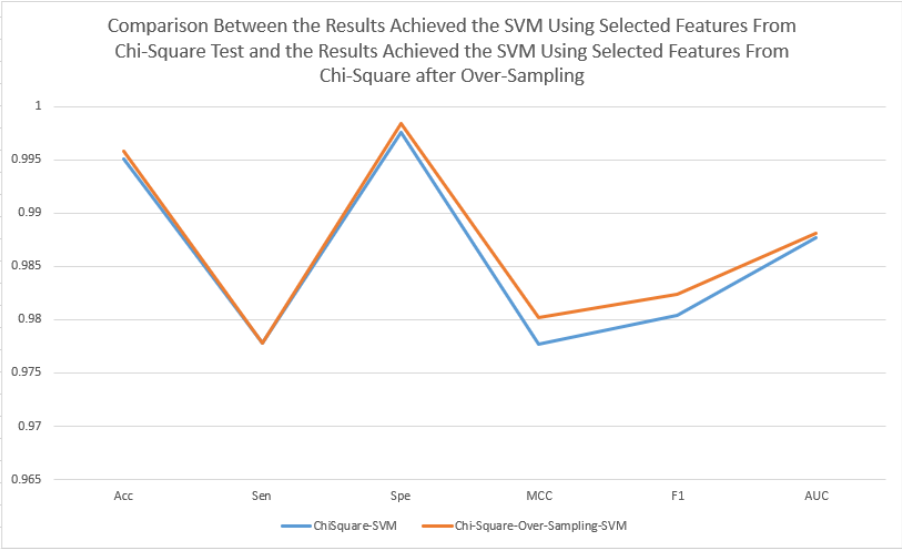

Figure 4.12: Comparison between the Results Achieved by the SVM after Feature Selection and the Results Achieved by the SVM after Over-Sampling.

4.7 Comparison with Previous Research

We have compared our best results with the previous studies. During the comparison, we have considered both the number of features and the corresponding best classification accuracy. Table 4.4 tabulated the comparison between the proposed and previous studies.

Figure 4.13 represents the comparison between the proposed and previous studies. From Figure 4.13, we can easily find out that no. of features considered in the proposed model is lower than the previous studies. But, the accuracy of the proposed models is better than the previous studies. We have made improvements in both cases.

Table 4.4: Performance Comparison between the Proposed Model and Previous Studies Considering the Number of Features and Classification Accuracy

Study	Number of Features	Accuracy
RF [48]	9	92.55%
RF [33]	9	96.74%
Proposed Model [iForest-χ^2-SMOTE-SVM(RBF)]	5	99.58%

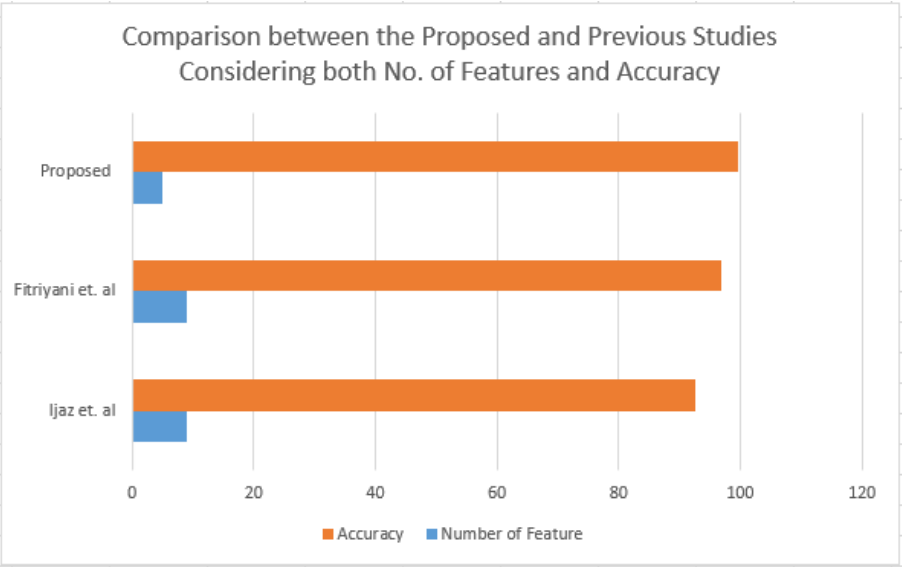

Figure 4.13: Comparison between the Proposed Model and Previous Studies.

4.8 Model Explanation Using the SHAP

The SHapley Additive exPlanations (SHAP) is a game theory-based AI model for model explanation. It actually discloses the contribution of each feature to the classification or target variable. The methodology of the SHAP has been described in section 2.6.7.1. To explain our best classification model using SHAP, we have considered some measurements from SHAP tools like **swarm plot**, **waterfall plot**, and **bar plot**.

4.8.1 Swarm Plot

The swarm plot is used to explain the contribution of a feature for classification considering the whole test dataset. Figure 4.14 represents the swarm plot of the test dataset.

Low values of the 'glyhb' feature have a high negative contribution to classification. While high values have a high positive contribution to classification. Similarly, low values of 'stab.glu' have a great negative impact on classification. And, high values have a positive impact on classification. 'ratio', 'Age', 'waist' have impact on classification like 'stab.glu', but, not as like as 'gluhb'.

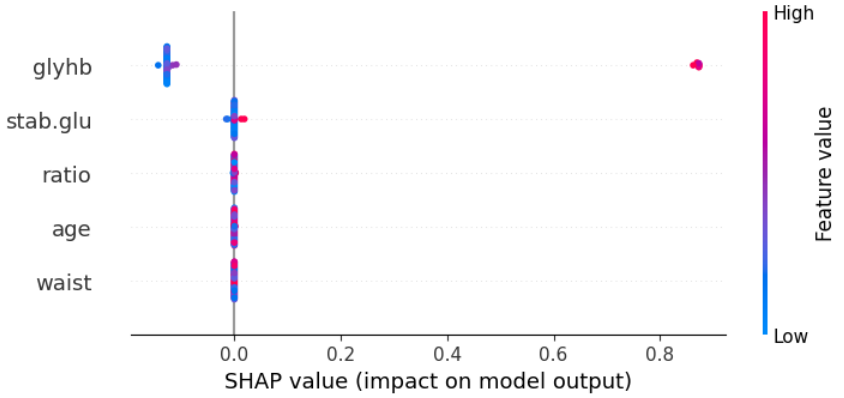

Figure 4.14: Swarm Plot on the Test Dataset.

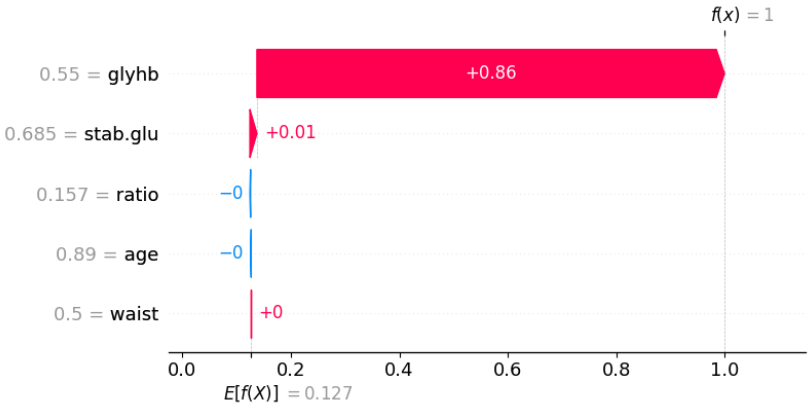

Figure 4.15: Waterfall Plot for a Positive Test Case.

4.8.2 Waterfall Plot

The waterfall plot of the SHAP is used to plot the importance of features for a specific test case. In the waterfall plot, the sum of the SHAP values for a test case is equal to $E(f(x)) - f(x)$. Here, we have attached the waterfall plot for a positive case and a negative case.

Figure 4.15 represents the waterfall plot for a positive test case. 'glyhb' has the highest SHAP value and it is 0.86. And, the SHAP for 'stab.glu'. is 0.01. These two features have much impact on positive classification.

On the other hand, Figure 4.16 represents the waterfall plot for a negative test case. The SHAP for 'glyhb' and 'stab.glu'. are 0.12 and 0.01 respectively. These two features have the highest impact on negative classification.

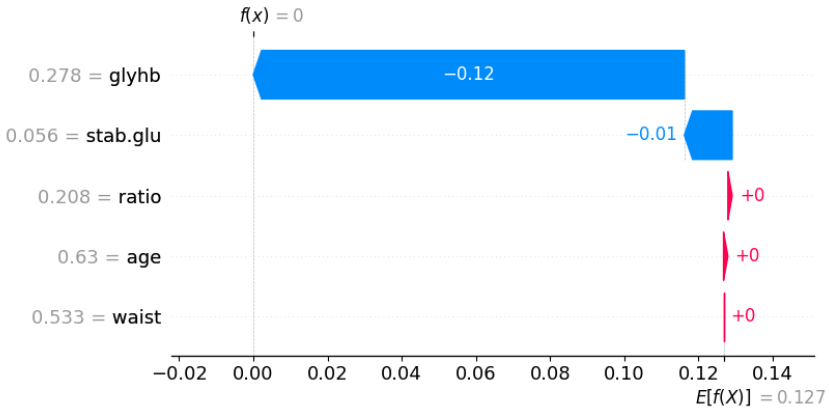

Figure 4.16: Waterfall Plot for a Negative Test Case.

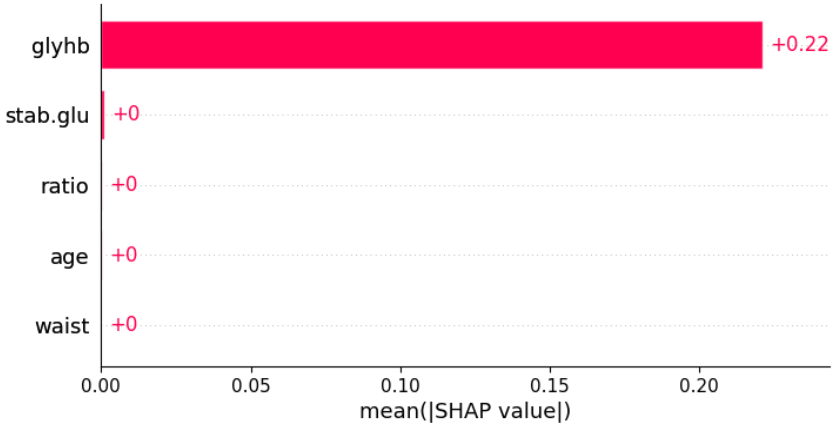

Figure 4.17: Bar Plot of the SHAP Values for the Features

4.8.3 Bar Plot

The Bar plot of the SHAP is used to represent the sorted impact of the features in decreasing order. The feature having the highest impact on classification is listed at the top of the bar plot. From Figure 4.17, we can see that 'glyhb' has the highest impact on the classification. 'stab.glu' has the second highest impact. 'Waist' has the lowest impact on the classification.

4.9 Conclusion

This chapter described the results of the experiments done in this research. We have also presented a comparative study of the results of the classifiers on different steps of the prediction model. The model explanation is also added in this chapter. The impacts of the individual features are analyzed and described in this chapter.

Chapter 5

Conclusion and Future Scopes

5.1 Introduction

The research findings will be summarized in this chapter, and it will also look at how the recommended methods could influence early-stage diabetes prediction studies. Additionally, we will attempt to examine the shortcomings of the current research and recommend alternative directions for future developments. We have explored two well-known diabetes datasets. We have applied our proposed diabetes prediction model to both datasets. The outcomes of the research have been compared with previous studies. We will discuss the application of this research in both academic and industry. Finally, we will summarize the limitations of this study and future scopes in the field of early-stage diabetes prediction.

5.2 Thesis Summarization

This research focuses on the usage of Machine Learning methods in early-stage diabetes prediction. The term early-stage prediction of diabetes means finding out such features which will be easily diagnosed, just like observing the physical condition without any complex hormonal test. To address this research problem, we needed to select a diabetes dataset that can fulfill our criteria. We have selected the Jhon Schorling diabetes dataset, a type 2 diabetes dataset, as our primary dataset. Then, we have completed the operations for data pre-processing. We have handled the missing values and encoded the categorical variables. We have also normalized the numerical data using the Min-Max normalization. We have removed the outlier samples from the pre-processed dataset. Then dataset was divided into training and testing datasets. After that, the training dataset has been used for feature selection. We have applied three well-known

feature selection approaches from three different feature selection method domains. We have compiled the testing dataset with the selected feature. Then, we have applied the classification models on the training dataset for building the classification models. We have followed the ten-fold cross-validation during the model tuning. The testing datasets are tested and results are compared among themselves, also with the previous studies. To validate the model, we have also applied the prediction model on another diabetes dataset. The results for the second dataset are also up to the mark. Finally, we have explained the model using the SHAP, a game-theory-based AI model for classification model explanation.

5.3 Impact of Thesis

Machine learning has been used in disease prediction for many years. Many researchers have applied machine learning for diabetes prediction also. Sometimes the feature selection for diabetes prediction was not touched properly. Sometimes, the accuracy of the prediction model was not up to the mark. That's why we have tried to address both steps in our research.

5.3.1 Academic Impact

Feature selection for early-stage diabetes prediction has a great impact on the academic sector. We have built up a complete research methodology for diabetes prediction. We have addressed every type of machine learning methodology, starting from data pre-processing to model explanation. We have proven the validity of our selected features using the model explanation. We have mentioned the impact of each feature for classification which also supports our results from the feature selection algorithm. Our proposed model can be used in other disease prediction models like heart disease, kidney disease, cancer, etc. This early-stage prediction of these critical diseases can be helpful to save millions of lives. The patients will be aware of his/her health condition as well as taking preventive measures. It can be a new era of early disease prediction studies. The explanation of the classification model has also enriched the acceptability of the prediction model. This type of study can also be applied to the market prediction.

5.3.2 Industrial Impact

Early-stage diabetes prediction has a great impact on the medical industry. It can be a valuable part of the smart health monitoring system. AI-based health monitory system is getting popularity day by day. Nowadays, some complex surgical operations are done by robotic hands and computer programs. But, AI-based health monitory system is a new concept. Where people can check up on their health condition and get proper suggestions at any time without a long waited appointment with a doctor. Some mobile-based application has been developed for monitoring the health condition of the users. Actually, these applications are using machine learning models in the back-end. For example, Samsung Health is an AI-based health-monitoring Android application that monitors the user's weight, height, daily walking steps, pulse rate, etc. By monitoring these, Samsung Health suggests the users about calorie burning, taking physical exercise, etc. Recently, some web applications are found on the Internet for diabetes prediction. Our proposed model can be used to develop such a type of application.

5.4 Limitations

This study has some shortcomings despite making some noteworthy advances. Firstly, the dataset used in this research contains a small number of samples. The dataset contained outlier samples. After removing the outliers, we size of the dataset become smaller. For outlier removal, we consider only a decision tree-based model. To handle the class imbalance, we have generated synthetic data. These synthetic data are used during model training.

5.5 Future Scopes

In this area, there are many potential future applications. As noted in section 5.3, the model can be used in smart health monitoring systems. Web Based applications can be developed where people can easily access and get suggestions. To study more about outlier removal, cluster-based methods can be applied in this research. As the dataset is small, we needed to generate synthetic data to solve the class imbalance issue. Other diabetes datasets containing a large number of samples can be studied. These prediction models can also be studied for other disease prediction models.

5.6 Conclusion

This study has given a near-perfect model for diabetes prediction at an early stage. An optimal set of features has been selected for diabetes prediction. These selected features are also validated using an AI-based model explanation technique. The outcome of the research can be used in sustainable academic research and industry application. This research can be helpful to save the lives of people and minimize the cost of diabetes treatment.

REFERENCES

[1] R. Thomas, S. Halim, S. Gurudas, S. Sivaprasad, and D. Owens, "Idf diabetes atlas: A review of studies utilising retinal photography on the global prevalence of diabetes related retinopathy between 2015 and 2018," *Diabetes research and clinical practice*, vol. 157, p. 107840, 2019.

[2] G. D. Kalyankar, S. R. Poojara, and N. V. Dharwadkar, "Predictive analysis of diabetic patient data using machine learning and hadoop," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC)*, (Palladam, India), pp. 619–624, 2017.

[3] S. Wild, G. Roglic, A. Green, R. Sicree, and H. King, "Global prevalence of diabetes: Estimates for the year 2000 and projections for 2030," *Diabetes Care*, vol. 27, no. 5, pp. 104–1053, 2004.

[4] *Diabetes Newsletter. Issue 100 February 2023, [Online]. Available: https://www.dab-bd.org/diabetes_newsletter.php. Accessed on: May 13, 2023*.

[5] W. H. Organization, "Diabetes fact sheet n 312. october 2013," *Archived from the original on*, vol. 26, 2013.

[6] E. Saed, M. R. Gheini, F. Faiz, and M. A. Arami, "Diabetes mellitus and cognitive impairments," *World journal of diabetes*, vol. 7, no. 17, p. 412, 2016.

[7] W. H. Organization, "Projections of mortality and causes of death, 2016 to 2060," *[Online]. Available: https://www.who. int/healthinfo/global_burden_disease/projections/en/*, 2016.

[8] A. D. Association, "Economic costs of diabetes in the us in 2017," *Diabetes care*, vol. 41, no. 5, pp. 917–928, 2018.

[9] B. Patil, R. Joshi, and D. Toshniwal, "Hybrid prediction model for type-2 diabetic patients," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8102–8108, 2010.

[10] H. Wu, S. Yang, Z. Huang, J. He, and X. Wang, "Type 2 diabetes mellitus prediction model based on data mining," *Informatics in Medicine Unlocked*, vol. 10, pp. 100–107, 2018.

[11] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *008 Eighth IEEE International Conference on Data Mining*, (Pisa, Italy), pp. 413–422, 2008.

[12] *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–57, 2002.

[14] A. Ugoni and B. F. Walker, "The chi-square test: an introduction," *COMSIG review*, vol. 4, no. 3, pp. 61–64, 1995.

[15] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[16] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products.," *Chemometrics and intelligent laboratory systems*, vol. 83, no. 2, pp. 83–90, 2006.

[17] K. P. Murphy, "Machine learning: A probabilistic perspective (adaptive computation and machine learning series)," 2018.

[18] W. R. [Qiu, "iphos-pseen: identifying phosphorylation sites in proteins by fusing different pseudo components into an ensemble classifier," *Oncotarget 7.32 (2016): 51270*.

[19] M. A. M. [Hasan, "predcar-site: Carbonylation sites prediction in proteins using support vector machine with resolving data imbalanced issue," *Analytical biochemistry*, vol. 525, pp. 107–113, 2017.

[20] J. Brownlee, "A gentle introduction to k-fold cross-validation," in *Machine Learning Mastery*.

[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[22] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," *Aaai*, vol. 90, pp. 223–228, 1992.

[23] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 86–106, 1986.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[25] C. Corinna and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] S. M. Lundberg and S.-L. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, 2017.

[27] Ribeiro, M. Tulio, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, 2016.

[28] A. Alloubani, A. Saleh, and I. Abdelhafiz, "Hypertension and diabetes mellitus as a predictive risk factors for stroke," *Diabetes Metabolic Syndrome, Clin. Res. Rev.*, vol. 12, no. 4, pp. 577–584, 2018.

[29] B. Patil, R. Joshi, and D. Toshniwal, "Hybrid prediction model for type-2 diabetic patients," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8102–8108, 2010.

[30] B. Farran, A. Channanath, K. Behbehani, and T. Thanaraj, "Predictive models to assess risk of type 2 diabetes, hypertension and comorbidity: machine-learning algorithms and validation using national health data from kuwait–a cohort study," *BMJ open*, vol. 3, 2013.

[31] H. Wu, S. Yang, Z. Huang, J. He, and X. Wang, "Type 2 diabetes mellitus prediction model based on data mining," *Informatics in Medicine Unlocked*, vol. 10, pp. 100–107, 2018.

[32] T. M. Alam, M. A. Iqbal, Y. Ali, A. Wahab, S. Ijaz, T. I. Baig, A. Hussain, M. A. Malik, M. M. Raza, S. Ibrar, and Z. Abbas, "A model for early prediction of diabetes," *Informatics in Medicine Unlocked*, p. 100204, 2019.

[33] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, "Development of disease prediction model based on ensemble learning approach for diabetes and hypertension," *IEEE Access*, p. 144777, 2019.

[34] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," *Computer Vision and Machine Intelligence in Medical Image Analysis*, pp. 113–125, 2020.

[35] M. M. Islam, M. J. Rahman, M. M. Abedin, B. Ahammed, M. Ali, N. F. Ahmed, and M. Maniruzzaman, "Identification of the risk factors of type 2 diabetes and its prediction using machine learning techniques," *Health Systems, Taylor & Francis*, pp. 1–12, 2022.

[36] H. M. Deberneh and I. Kim, "Prediction of type 2 diabetes based on machine learning algorithm," *International journal of environmental research and public health, MDPI*, no. 6, p. 3317, 2021.

[37] A. H. Syed and T. Khan, "Machine learning-based application for predicting risk of type 2 diabetes mellitus (t2dm) in saudi arabia: a retrospective cross-sectional study," *IEEE Access*, vol. 8, pp. 199539–199561, 2022.

[38] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.

[39] U. Ahmed, G. F. Issa, M. A. Khan, S. Aftab, M. F. Khan, R. A. T. Said, T. M. Ghazal, and M. Ahmad, "Prediction of diabetes empowered with fused machine learning," *IEEE Access*, vol. 10, pp. 8529–8538, 2022.

[40] G. Tripathi and R. Kumar, "Early prediction of diabetes mellitus using machine learning," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1009–1014, 2020.

[41] S. S. Bhat, V. Selvam, G. A. Ansari, M. D. Ansari, and M. H. Rahman, "Prevalence and early prediction of diabetes using machine learning in north kashmir: A case study of district bandipora," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[42] D. Dutta, D. Paul, and P. Ghosh, "Analysing feature importances for diabetes prediction using machine learning," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 924–928, 2018.

[43] S. Sivaranjani, S. Ananya, J. Aravinth, and R. Karthika, "Diabetes prediction using machine learning algorithms with feature selection and dimensionality reduction," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 141–146, 2021.

[44] U. e. Laila, K. Mahboob, A. W. Khan, F. Khan, and W. Taekeun, "An ensemble approach to predict early-stage diabetes risk using machine learning: An empirical study," *Sensors*, vol. 22, no. 14, 2022.

[45] M. F. Ijaz, M. Attique, and Y. Son, "Data-driven cervical cancer prediction model with outlier detection and over-sampling method," *Sensors*, vol. 20, no. 10, 2020.

[46] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[47] M. J. Rezaei, J. R. Woodward, J. Ramirez, and P. Munroe, "Combination of isolation forest, smote and ensemble learning for the classification of atrial fibrillation and ventricular arrhythmia," (Varna, Bulgaria), pp. 45–50, 2022.

[48] M. F. Ijaz, G. Alan, M. Syafrudin, and J. Rhee, "Hybrid prediction model for type 2 diabetes and hypertension using dbscan-based outlier detection, synthetic minority over-sampling technique (smote), and random forest," *Applied Sciences*, vol. 8, no. 8, 2018.

[49] S. Wang, S. Liu, J. Zhang, X. Che, Y. Yuan, Z. Wang, and D. Kong, "A new method of diesel fuel brands identification: Smote oversampling combined with xgboost ensemble learning," *Fuel*, vol. 282, p. 118848, 2020.

[50] S. Sridhar and S. Sanagavarapu, "Handling data imbalance in predictive maintenance for machines using smote-based oversampling," (Lima, Peru), pp. 44–49, 2021.

[51] J. P. Willems, J. T. Saunders, D. E. Hunt, and J. B. Schorling, "Prevalence of coronary heart disease risk factors among rural blacks: A community based study," *Southern Medical Journal*, vol. 90, no. 8, pp. 814–820, 1997.

[52] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Computer Vision and Machine Intelligence in Medical Image Analysis*, pp. 113–125, Springer, 2020.