The applications of data mining and machine learning in Bangladesh for disease pattern analysis and prediction

by

Mahmudul Hasan Mahmud
15141010

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfilment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University

Declaration

I hereby declare that

1. The thesis submitted is my own original work while completing degree at BRAC University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Mahmudul Hasan Mahmud
15141010

Approval

The thesis/project titled "The applications of data mining and machine learning in Bangladesh for disease pattern analysis and prediction" submitted by

1. Mahmudul Hasan Mahmud (15141010)

of Spring, 2020 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in Computer Science on April, 2020.

Examining Committee:

Supervisor:
(Member)

Hossain Arif

Hossain Arif
Associate Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Rabiul

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Dr. Mahbub Alam Majumdar
Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Over the years, data mining and machine learning have proved to be very convenient in numerous fields of science and technology and their applications in the medical sector is an emerging one. With the world population rate increasing by the year, the medical sector is generating immense amount of data every day. By storing this data and analyzing it for disease patterns, using numerous data mining and machine learning techniques, predictive models can be built to assess future risk to potential patients. These models may have a very important role in a developing country like Bangladesh, where Non-Communicable Diseases (NCD) like diabetes and heart diseases have affected a large portion of its population. Clinical diagnosis of these diseases requires a lot of tests which complicates the prediction process and proves to be expensive for most patients as well. Predictive models based on data mining and machine learning techniques provides a much more efficient system of predicting future risks for patients, saving lives and a lot of money. This project looks at several data mining and machine learning techniques for analyzing medical data in order to recognize disease patterns, compare their performances and eventually produces a model with the highest accuracy in disease prediction.

Keywords: Diabetes prediction, Naïve Bayes, Decision Tree, Random Forest, Logistic Regression, SVC, Linear SVC, KNN, LassoCV, GridsearchCV, KFold, StratifiedKFold.

Acknowledgement

To start off, I would like to convey my deepest sense of heartfelt gratitude towards the Almighty Allah. Secondly, I would love to share my sincerest gratitude to my advisor Hossain Arif sir for his constant support, patience, motivation, and immense knowledge in my research. His guidance and assistance helped me in all parts of the progress.

Finally, I must express my cordial gratefulness to my beloved parents, brothers and sisters for ensuring an environment full of peace and harmony around me. And last but not the least; I am grateful to all of my friends who helped me sincerely to complete my thesis.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document.

NCD	Non-Communicable Disease
NB	Naïve Bayes
KNN	K-Nearest Neighbor
GNB	Gaussian Naïve Bayes
MNB	Multinomial Naïve Bayes
BNB	Bernoulli Naïve Bayes
SVM	Support Vector Machine
SVC	Support Vector Classifier

Chapter 1

Introduction

With the drastic increase of global population in the last century, the medical sector faces a huge challenge in diagnosis and treatment of diseases. As the human civilization expands, it not only diversifies their lifestyle, but also the diseases that affect them. One possible solution may already exist in the mountains of data, in the form of patient records, kept at hospitals and various other health care facilities. The advancement of technology in recent decades means that these clinical records could be analyzed for detecting patterns and certain traits of common yet dangerous diseases. Eventually, these patterns could help construct a support system to be used by the doctors and other medical professionals to make risk assessments of patients at a very early stage.

In terms of most well-known diseases, such as diabetes, the early detection and diagnosis can be very crucial. Diabetes, a non-communicable disease (NCD), can have severe effects on a patient if unchecked and can also cause various other health complications. Therefore, a support system to aid health care professionals in early and accurate diagnosis of such diseases is of paramount importance. Data mining and Machine learning techniques could play a huge role in this regard. These techniques can make predictions by modeling and training on datasets with patient's records available at various medical facilities and data repositories. In recent years, such predictive systems have already been used in health care industry for data analysis and have contributed immensely to numerous medical research works.

This paper illustrates a comparative study in order to assemble a prediction model for diabetes with a higher rate for accuracy than the existing models. A number of techniques would be examined with this goal in mind. At the end, after evaluating their performances, the best classification technique for disease pattern analysis and prediction would be recommended. The ultimate hope is for the recommended model to aid the doctors and medical professionals in early diagnosis and possible prevention of said diseases.

1.1 Thesis Orientation

The rest of this research is organized in the following order:

Chapter 1 – Introduction

A brief discussion about the project in hand and its scope, followed by thesis orientation.

Chapter 2 – Literature Review

This chapter discusses and reviews some of the previous related works on the topic.

Chapter 3 – Proposed Approach

In this section the algorithms used in this research work would be explained in short.

Chapter 4 – Dataset Analysis

In this chapter we discussed about the data we collected for implementation and how we processed it.

Chapter 5 - Methodology

In the methodology section we discussed about the algorithms that we will use for our research.

Chapter 6 – Experimental Result Analysis

All the results and findings of this project are presented in this chapter.

Chapter 7 Conclusion and Future Works

In conclusion, a summary about the work done till now, is given and a discussion about the scopes for future improvements.

Bibliography

Chapter 2

Literature Review

Significant amount of work has been done by numerous researches all over the world on disease prediction using data mining tools and machine learning. Mostly, they have used one of the two, either data mining or machine learning, to build a prediction model, often times by using several techniques of their choice. The following is a brief description of some of those works.

Chen et al.[1] have proposed a hybrid prediction model for diabetes using k-means and decision tree. Their data set, collected from the Pima Indian Diabetes Data (PIDD), contains records of 768 females, out of which 268 are in class "tested positive for diabetes" and 500 are for "tested negative for diabetes", and 376 records containing missing values. This work was done at four stages; data preprocessing, data reducing, classification and performance evaluation. In data preprocessing, they replaced all the missing or impossible values with the mean of the data. Using k-means algorithm, they reduced the dataset by removing incorrectly classified data, in order to cluster the data set. At classification stage, Decision Trees were built using the reduced data set. Finally, the performance of their model was evaluated by using several methods, namely k-fold cross validation and confusion-matrix. The model yielded a 90.04% accuracy, which was more than all the reference models they had used to compare with.

In Bangladesh, Sultana et al.[2], had worked on analyzing 5 different data mining techniques for disease prediction, specifically for heart disease. Their data set, collected from UCI Machine Learning Repository and a local diagnostic centre, contained 370 records in total, with various attributes such as age, sex, blood pressure etc. The 5 data mining techniques used here were KStar, Decision Trees, Sequential Minimal Optimization (SMO), Bayesian network and Multilayer Perceptron (MLP). They evaluated the performances using confusion-matrix based on accuracy, true positive rate (TPR) and false positive rate (FPR) and finally using the receiver operating characteristics (ROC) curve. The results indicate that the Bayesian network works with much more accuracy than the rest of the classifiers.

Naïve Bayes classifiers and J48 Decision Tree were used by Kunjir et al.[3] to build a prediction model, along with various data visualization techniques. Their datasets, collected from various repositories, contained data on diabetes (768 instances), breast cancer (286 instances), heart disease (270 instances) and arthritis (429 instances). More than 20 medical attributes relating to these diseases were considered. The data was presented using line graphs, bar graphs and pie charts for simplification. After evaluation, the Naïve Bayes classifier was observed to have outperformed the J48 Decision Tree in both accuracy and latency analysis i.e the time taken for prediction of class labels. They concluded that their work was efficient enough to build an expert decision support system which could be used by medical practitioners for enhanced diagnosis and could also be used by patient users for prediction purposes.

A different approach was taken by Vijayan and Anjali [4], who proposed a decision support system using AdaBoost algorithm. Decision Tree, Support Vector Machine (SVM), Naïve Bayes and Decision stump were used as base classifiers for the algorithm. A global dataset was collected to be used as the training set for this project and a local dataset was used for validation purposes. Their work found out that Decision stump worked with much more efficiency and accuracy, with the AdaBoost algorithm, than the rest of the classifiers.

In another place, Vinitha et al.[5] have combined Decision Tree algorithm and Map reduce algorithm for their work. Both structured and unstructured data were collected from various hospitals to be used in their dataset. In their proposed model, the Decision Tree algorithm predicted not only the main diseases, but also their sub diseases. The Map Reduce algorithm was implemented for operational efficiency. Their model showed a 94.8% accuracy and worked faster than CNN-based unimodal disease risk prediction (CNN-UDRP) algorithms.

Additionally, Kalyankar et al.[6] designed a predictive analysis system for diabetic patients. They used datasets from Pima Indian Database for their work and implemented various machine learning algorithms in Hadoop MapReduce environment to find patterns from the data. For missing values in the dataset, missing value imputation (MVI) algorithm was used, while Decision Trees derived using C4.5 algorithm were generated for pattern recognition. They concluded by proposing to use pattern matching in future on the discovered patterns for prediction.

With the sole goal of assembling Intelligent Diabetes Disease Prediction System Shetty et al.[7] proposed the usage of Naïve Bayes and KNN algorithms to analyze diabetic patients. They argued that most of the previous work done had a major flaw in them, i.e. the datasets used being too small. Consequently, according to them, the prediction models that were trained and tested on those small datasets couldn't possibly predict diabetes with a higher level of accuracy and precision. As such, they suggested building a model based on datasets that included the records of more than 2000 diabetic patients. Furthermore, their plan includes a recommendation system that would help patients with minor symptoms to control or avoid the disease.

Berina et al.[8] presented a comparative analysis of using Artificial Neural Networks (ANNs) and Bayesian Networks (BNs) for diabetes and cardiovascular disease classification. Their study was based on research papers issued within the time frame of 2008 and 2017. Multilayer feed-forward neural network and Naïve Bayesian network were noted to be the most commonly used algorithms in terms of ANNs and BNs, respectively. Their study revealed that implementation of ANNs offered the greater chance of obtaining most accurate results in classification of diabetes and CVDs.

Similar approach was taken by Theresa and Thomas [9] in predicting risk levels of heart disease in patients. Although, their main focus was on KNN, a number of other algorithms were used in this project for various reasons. For example, Decision Tree was used to provide classified reports for the heart disease, Naïve Bayes for the prediction through probability and Neural Network to minimize the margin of error in the prediction system. Their objective was to create a system of assistance for the doctors by monitoring the patients and sending out alerts to the doctors when risk levels rise in any particular patients. They concluded by suggesting the use of more attributes in the prediction models for increased accuracy.

A comprehensive survey was conducted by Tikotikar and Kodabagi [10] on various data mining techniques used for disease prediction. The main focus of the survey was to discuss about decision parameter, attribute, and features used for predicting the disease. Studies carried out on models predicting various types of heart disease and breast cancer were included in this survey. They concluded that the task of classification and prediction of these diseases using these techniques becomes difficult due to the complexity of interdependencies on a number of factors and suggested that the usage of new feature selection techniques and experimentations of the algorithms could improve the quality of the models, as well as their reliability.

A customizable clinical diagnosis data capturing system, termed Doctor's Desk, was projected by Byju et al.[11], that could be integrated with any hospital information system package. The idea was to store clinical data of patients and manipulate the data in order to search for similar disease patterns or classifications using data mining techniques. Decision Tree and KNN algorithms were selected as said techniques. At the end of their research, they were able to determine that Decision Tree algorithm yielded a higher accuracy in both classification of diseases and similar pattern searches.

Abhishek et al.[12] introduced a different method that utilized efficient genetic algorithm with the back propagation technique approach for heart disease prediction. Three different techniques were evaluated in this method, namely Decision Tree, Naïve Bayes and KNN. They presented a systematic approach for fragmenting and extracting substantial forms from the heart attack data warehouses for achieving accurate predictions. Selecting 13 different attributes from the clinical data, they developed a structure that may assist medical professionals in evaluating a patient's cardiopathy. The results of their work, however, showed KNN to be the best technique to use with more accurate and much faster predictions than the other two.

Hybrid models for rule-based classification of diabetic patients have also been looked at. Ibrahim et al.[13] experimented with a new hybrid model by exploring Agglomerative Hierarchical Clustering and Decision Tree Classifier. Comparing the performance accuracy of the Decision Tree Classifier against the same classifier augmented with Hierarchical Clustering, they proved the accuracy of their hybrid model to have been much higher than that of the standard model. They argued that the evidence called for adaptation of hierarchical clustering in rule-based classification and suggested the use of alternate datasets in the future.

In order to classify patients with risk of Cerebrovascular Accident Attack, Artificial Neural Networks were applied, by Olatubosun and Bola, in the construction of a prediction model [14]. Their reasoned that the odds of a successful treatment in such events relied heavily on the early diagnosis and often times the scope for detecting and preventing these attacks are very limited. Hence, they proposed a model that consisted of a three-layer feed forward artificial neural network with back-propagation error method. Although the model achieved a reasonable forecasting accuracy with the limited amount of data at their disposal, further work was encouraged on a much broader scale.

Neural Networks had also been used in another place in order to study cancer survival predictions. Lundin et al.[15] organized a study of breast cancer survival using said tools. Records of a number of breast cancer patients were analyzed and a predictive model built with neural networks for 5-, 10- and 15-year breast-cancer-specific survival. Eight different variables were used as input to the network. The consistent high accuracy and good performance of the network over a period of time indicated that neural networks could be an essential tool in this regard.

In other works, Naïve Bayes classifiers and Support Vector Machine were applied for the prediction of diabetic retinopathy. According to Ramalaniya [16], despite being a treatable disease, diabetic retinopathy is still one of the most common eye diseases in the world with numerous patients going blind due to lack of correct diagnosis. To fill the void, he made use of Naïve Bayes and Support Vector Machine, along with image enhancing techniques, to assemble an accurate diagnosis method of proliferative diabetic retinopathy. His results highlights better performance of SVM with respect to detecting micro aneurysm in the enhanced images. He concluded with the proposition of combining the method with an exudates detecting system for clinical applications.

Chapter 3

Proposed Approach

For this comparative study, a number of machine learning and data mining techniques would be analysed for disease prediction. At the end, using cross validation and various classification and performance metrics, these classifiers would be evaluated for recommendation. The ultimate goal would be to assemble a mixture of different techniques to produce a prediction model that could be of assistance to doctors and other medical professionals in early diagnosis and treatment of NCDs.

The different techniques used in this project and their brief description are as follows -

3.1 Logistic Regression

Logistic Regression [17] is a parametric classification model that has a certain number of parameters. These parameters are depended on the input features and their output is generally a binary categorical prediction, even though the model is referred to as regression model. Although Logistic Regression shares certain similarities with Linear Regression, it is unique in the way it perceives the data. While Linear Regression fits a straight line through the data, a S shaped curve, called the Sigmoid function, is fitted to the observations in Logistic Regression.

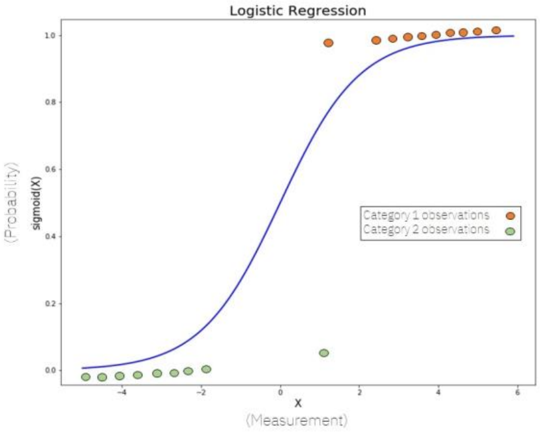


Figure 3.1: Sigmoid function fitted to some data in Logistic Regression

As the figure above demonstrates, the Y-axis goes from 0 to 1, due to the fact that *sigmoid* function always takes as maximum and minimum these two values. This characteristic makes Logistic Regression suitable for classification of data in to two different categories. In the x-axis of the curve is the weighted sum of the input features, denoted by X, which is used in calculations of the sigmoid function. At the end of the calculations, we obtain a probability between 0 and 1 that indicates which category a particular observation belongs to.

The formula for the *sigmoid* function is the following:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Weighted sum of the input features is calculated using the following equation:

$$x = \text{ɵ}.input\ feature + b$$

where, ɵ is a model parameter and *b* is some coefficient. In order to calculate the parameters of the model (the weights), an iterative optimization algorithm like *Gradient Descent* or a probabilistic method like *Maximum likelihood* is used. Thus, the model is trained and using the equation found through calculation of the parameters, the model makes predictions in the testing phase.

3.2 K – Nearest Neighbor (KNN)

It is a type of supervised Machine Learning algorithm that is usually used in predictive classification problems in industries [18]. However, this algorithm could also be used as a predictive regression model. The main idea of it is to assign a value to a new data-point based on how close it is to other data-points, already existing in the dataset. By assigning the value, KNN therefore classifies that certain data-point to a particular group nearest to it. As, instead of having a specialized training phase, this algorithm uses the data for training while classifying new data-points, it is termed as a Lazy Learner algorithm. It is also known as a Non-Parametric algorithm since it does not make any assumptions about the underlying data during classification.

The following pseudo code for the algorithm explains the entire process in details:

```
a. The training and test datasets are loaded.
b. A value of K is assigned.
c. For each point in test data:
        - The Euclidean distance to all training data points is measured
        - the distances are then stored in a list and sorted
        - the first k points are then chosen
        - based on the majority of classes present in the chosen points, the
   test point is assigned a class.
d.  End
```

3.3 Naïve Bayes

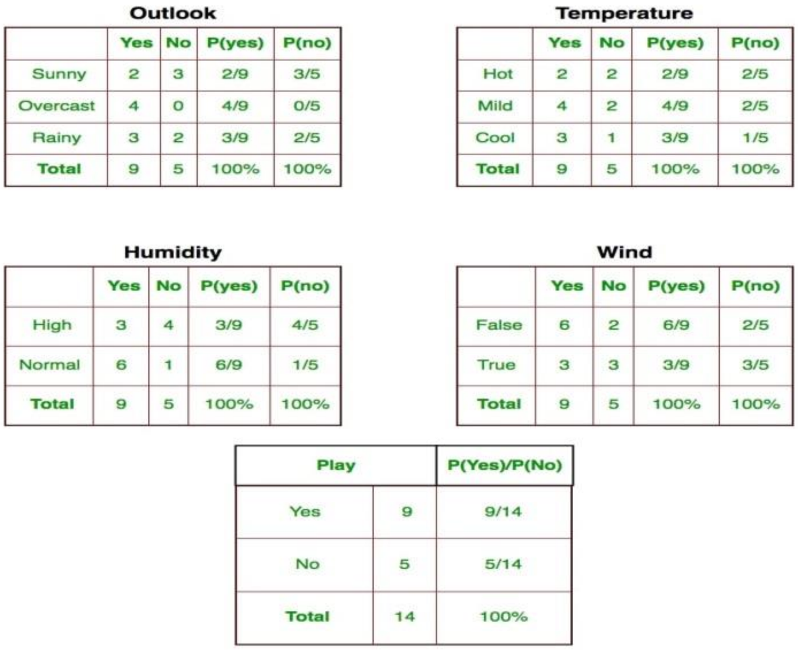
The basic idea behind Naïve Bayes is to implement Bayes' theorem with a strong assumption that all the features in a predictor class are completely independent of each other [19]. Hence, this classification technique implies that all the predictors are also independent of each other. Since, this assumption is not always correct, it earned this classification technique the term "Naïve". The other assumption by this technique is that all the features are equal as well. Therefore, each feature is given the same weight or importance during predictive analysis.

In this classification model, the goal is to find the conditional probability of an output or a label given some input or observed feature. In order to find this probability, several probability tables are derived using the features from a dataset. The values from these tables are then utilized in Bayes' theorem to obtain the desired probability as shown below –

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

Here, $P(A|B)$ is the conditional probability of class.

$P(A)$ is the prior probability of class.
$P(B|A)$ is the likelihood which is the probability of predictor given class.
$P(B)$ is the prior probability of predictor.



Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

Figure 3.2: Example probability tables in Naive Bayes

There are three major types of Naïve Bayes classifiers. Although the concept behind all three are the same, they differ in the assumption of the distribution of $\boldsymbol{P(B_i|A)}$.

3.3.1 *Gaussian NB* – This is the simplest of the three classifiers that assumes that all the continuous values associated with each feature is distributed in a gaussian distribution. It it also referred to as a Normal distribution since it forms a bell shaped curve, when plotted, with the mean of the features dividing it in two equal halves.

3.3.2 *Multinomial NB* – As the name suggests, this classifier assumes that the feature vectors are represented in a multinomial distribution. This is why this model is well suited for document classification and for features representing discrete counts.

3.3.3 *Bernoulli NB* – The features are assumed to be independent booleans or binary variables. Like the Multinomial NB, this classifier is also a good method to use in document classification.

3.4 Decision Tree

One of the most widely used and efficient method for classification and prediction is the Decision Tree [20]. It is a tree like structure but inverted, i.e. the root of it is at the top, made up of numerous nodes and branches. Each internal node is considered as an input and the branches are the outcome of selected inputs. The leaf nodes represent the final output or a class label. Using the target variables "success" and "failure", the entropy for each attribute is calculated, as well as that of the target variable. Using these entropies, Information Gain (IG) for each attribute is then determined. The attribute with the highest IG is selected as the root node, and the dataset is split into subsets with respect to it. This process is repeated in a recursive manner known as recursive partitioning. The recursion stops once the leaf nodes represent the target variables or in the event that further splitting is rendered useless. Since no domain knowledge or parameter setting is required in the process, decision tree is a very powerful tool in exploratory analysis and classification. Its ability to handle high dimensional data and higher accuray are further evidences to that fact.

The entire process of the Decision Tree could be described through the following pseudo code :

a. The best attribute is set as the root of the tree.
b. The training set is then split into subsets. Subsets are made in such a way that each attribute in each subset contains the same value.
c. The previous two steps are repeated on each subset until the lead nodes are reached.

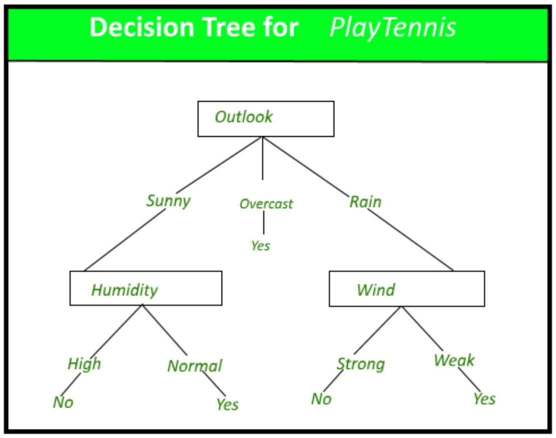


Figure 3.3: Example of a Decision Tree

3.5 Random Forest

A Supervised learning algorithm that is capable of performing both regression and classification tasks [21]. It is an ensemble technique that generates multiple Decision Trees on given data samples. Rather than choosing any output from a single Decision Tree, Random Forest evaluates the results from each individual tree and then selects the best outcome. The pseudo code for the algorithm is as follows –

a. K data points from the training set are picked at random.

b. Decision Trees associated with those K data points are built.

c. Predictions from individual trees are collected and evaluated.

d. The best predictive outcome is produced.

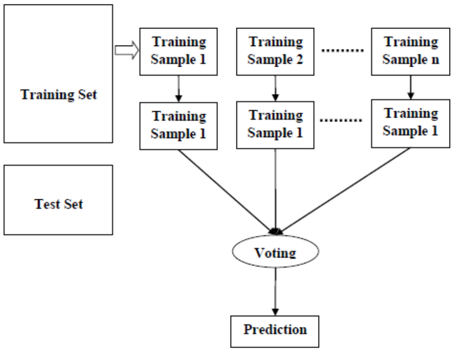


Figure 3.4: Implementation of Random Forest

3.6 Support Vector Machines (SVMs)

These are a set of supervised Machine Learning algorithms applicable to regression, classification and outlier detection models [22]. They are very effective with high dimensional spaces, even when the number of dimensions is higher than the number of samples. A variety kernel functions are provided by SVMs for specific decision functions, with the possibility of specifying custom kernels, making them very versatile. Also, they are very memory efficient as they use subsets of training points in the decision functions.

3.6.1 Support Vector Classifier (SVC) - It is capable of multi-class classification on a given dataset by implementing the "one-against-one" approach. If there are n number of classes, SVC would build {n*(n-1)}/2 classes. Each class would then be trained on data from two classes.

3.6.2 LinearSVC – This is similar to SVC with the exception of a kernel that is linear. The other difference is its implementation of "one-vs-the-rest" strategy. As a result only n number of models are trained.

Chapter 4

Dataset Analysis

4.1 Dataset Selection

Several datasets were collected from online data repositories for the purpose of this project. PIMA India Diabetes database from the UCI repository was the most widely used in previous works. It had records of 768 women on a range of 8 attributes. Similar to it, kaggle provided a dataset of about 2000 patients, all of whom were women. The attributes recorded in this database were identical to the one from the previous dataset. Another database, collected from data.world, had the records of over 100000 patients who visited various hospitals in United States of America over the period of almost 15 years, starting from 1999. It contained over 40 attributes including numerous several chemical compounds found in the human body.

However, none of the above mentioned datasets were selected due to three major reasons.

a. Although the dataset from the PIMA Indian database was very simple and easy to understand, it had been used excessively in past years. Therefore, the chances of obtaining new and useful results from it were very low.
b. In order to achieve better performance from the algorithms, the goal was to use datasets containing records of at least 5000 patients.
c. Despite having quite a huge amount of data of over 100000 patients, the dataset from US hospitals was not selected due to poor readability and complicated nature of its numerous attributes.

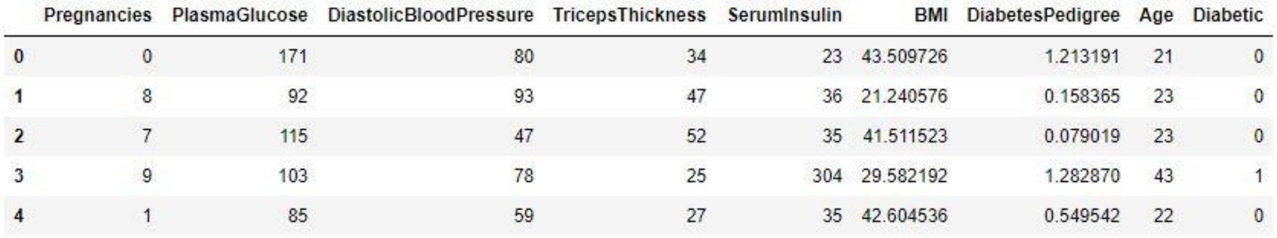
Finally, the dataset that was selected was collected from kaggle [23]. It contained medical records of 15000 women. Similar to the PIMA Database, its attributes were simple and very easy to understand.

Efforts were made to collect datasets from local hospitals or medical data repositories, in Bangladesh. But in most cases, authorities have rejected the request for sharing their database citing violation of confidentiality and security reasons.

4.2 Exploratory Analysis

As mentioned above, the dataset contains 15000 observations with 8 input variables and 1 output or target variable. The variables are as follows:

- Pregnancies: Number of times pregnant
- PlasmaGlucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- DiastolicBloodPressure: Diastolic blood pressure (mm Hg)
- TricepsThickness: Triceps skin fold thickness (mm)
- SerumInsulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigree.
- Age (years).
- Diabetic: Class variable (0 or 1). With 1 indicating diabetic and 0 indicating not diabetic.



	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
0	0	171	80	34	23	43.509726	1.213191	21	0
1	8	92	93	47	36	21.240576	0.158365	23	0
2	7	115	47	52	35	41.511523	0.079019	23	0
3	9	103	78	25	304	29.582192	1.282870	43	1
4	1	85	59	27	35	42.604536	0.549542	22	0

Table 4.1: First 5 observations from the top.

A value count revealed that there were 10000 observations with 0 as their class variable, and the rest had 1. In other words, 10000 of these patients were not diabetic and the rest of 5000 were. The proportion or ratio of diabetic patient to non-diabetic patient was calculated to 0.333 or $\frac{1}{3}$. In order to illustrate the ratio, a bar chart of the two outcomes was plotted.

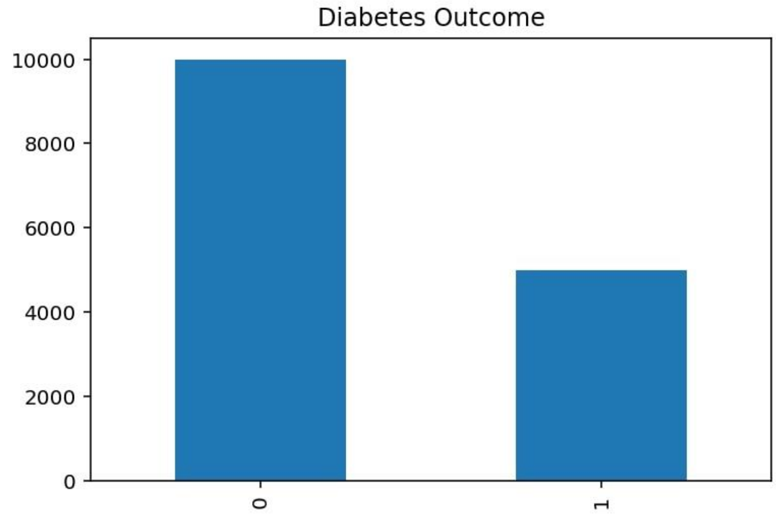
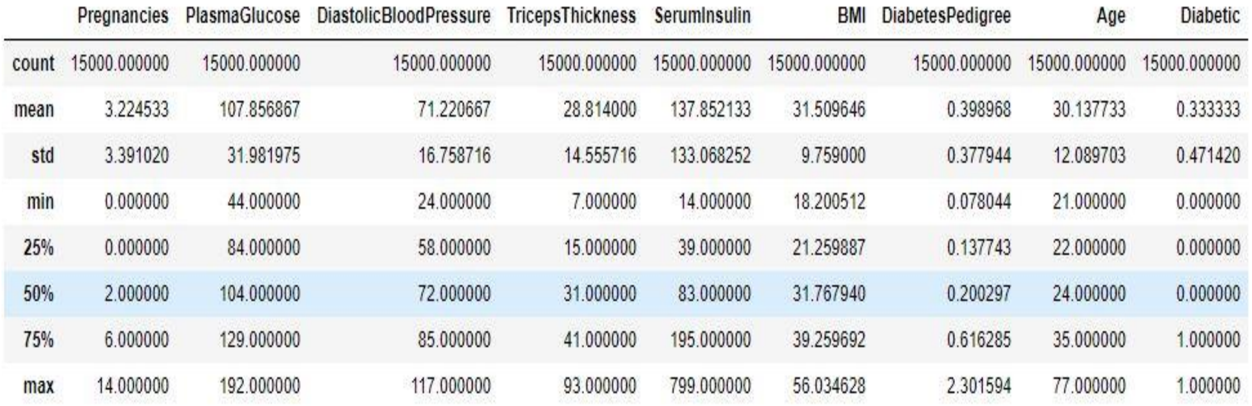


Figure 4.1: Ratio of Diabetic to non-diabetic

Calling the .describe method, the statistical data of all the attributes was tabulated.



	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
count	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000
mean	3.224533	107.856867	71.220667	28.814000	137.852133	31.509646	0.398968	30.137733	0.333333
std	3.391020	31.981975	16.758716	14.555716	133.068252	9.759000	0.377944	12.089703	0.471420
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200512	0.078044	21.000000	0.000000
25%	0.000000	84.000000	58.000000	15.000000	39.000000	21.259887	0.137743	22.000000	0.000000
50%	2.000000	104.000000	72.000000	31.000000	83.000000	31.767940	0.200297	24.000000	0.000000
75%	6.000000	129.000000	85.000000	41.000000	195.000000	39.259692	0.616285	35.000000	1.000000
max	14.000000	192.000000	117.000000	93.000000	799.000000	56.034628	2.301594	77.000000	1.000000

Table 4.2: Statistical interpretations of each attribute

The correlation heatmap was also generated.

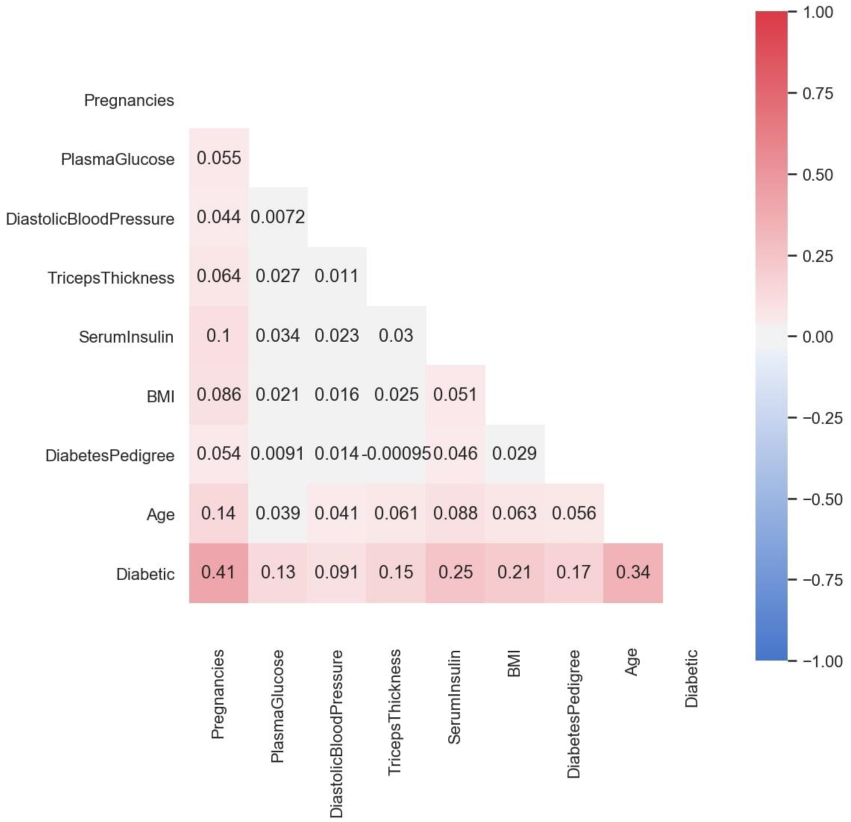


Figure 4.2: Correlation matrix heatmap visualization

To investigate the correlation between attributes a little further, seaborn pairplot were generated.

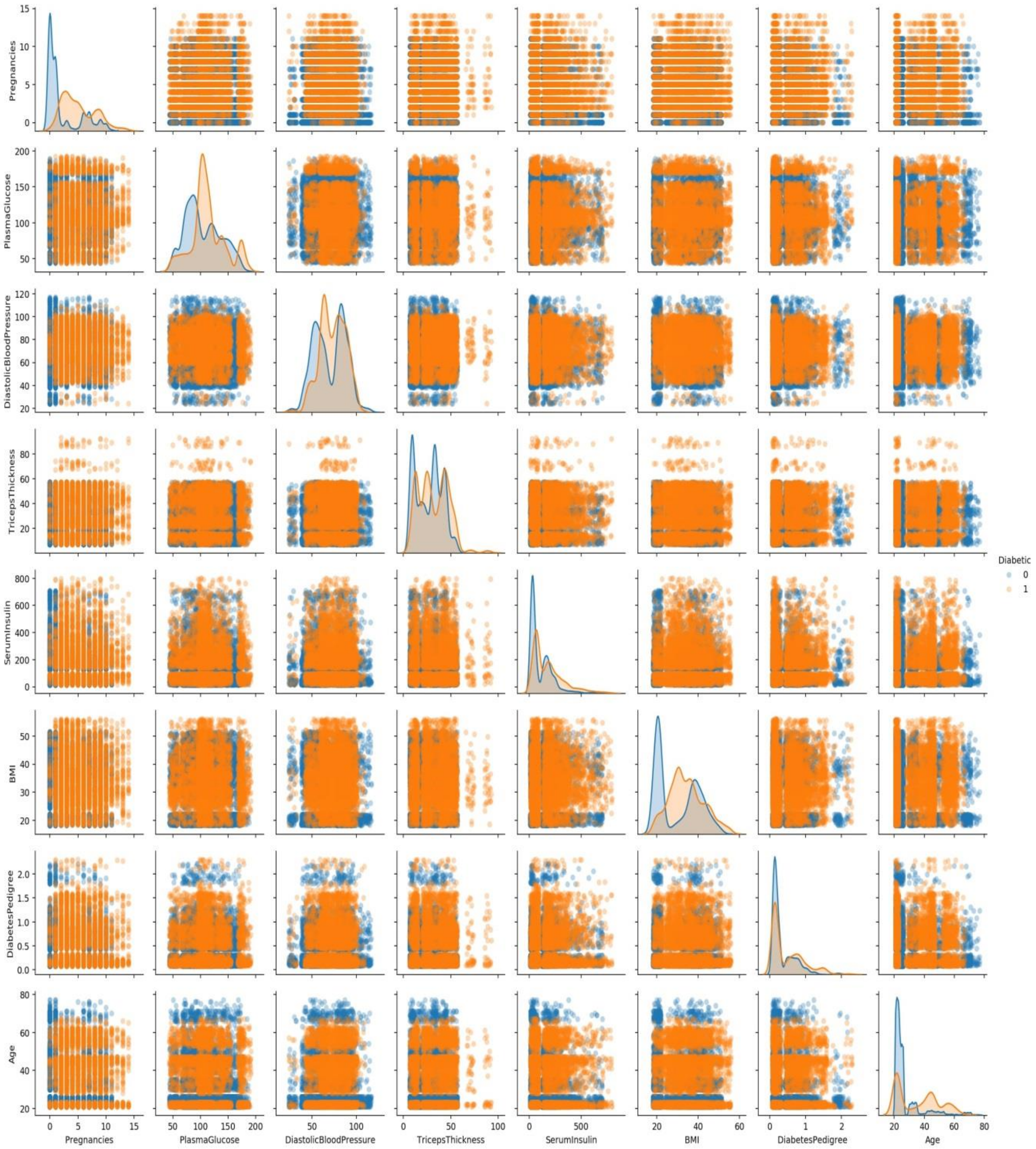


Figure 4.3: Visualizing pair plots

To analyse feature-outcome distribution in visualisation, histograms for each attributes were produced.

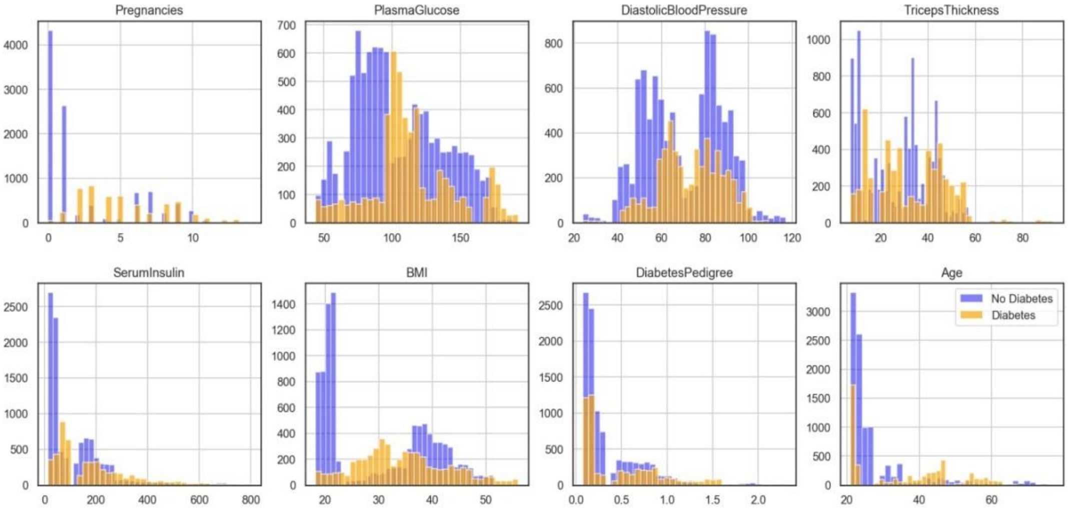


Figure 4.4: Feature-outcome distribution.

A few of the attributes had missing or zero values for some observations. The missing or zero values were replaced by the median value of the respective attribute, in two steps. First, the missing or zero value was replaced with NaN and then the NaN values were replaced by the median. To check if this had any effect on the feature-outcome distribution, the set of histograms above were generated once again.

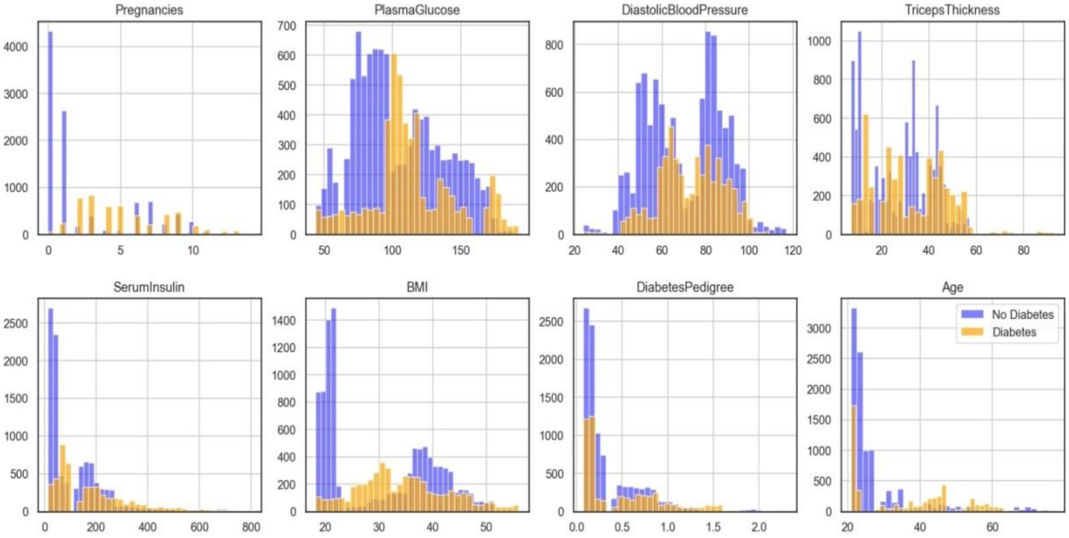


Figure 4.5: Feature-outcome distribution after median input

The top feature among the 8 features was derived by LassoCV, as well as the best alpha value. The optimal alpha values for each individual algorithm were calculated too. The results were plotted on a barplot.

```
LASSO best alpha: 0.001

[('Pregnancies', 0.1507284407625645),
('Age', 0.11962198027486799),
('SerumInsulin', 0.07970548291708356),
('BMI', 0.06948221683345782),
('DiabetesPedigree', 0.05806696317629435),
('TricepsThickness', 0.048549954329668805),
('PlasmaGlucose', 0.0403400230193551),
('DiastolicBloodPressure', 0.025980373940494655)]
```

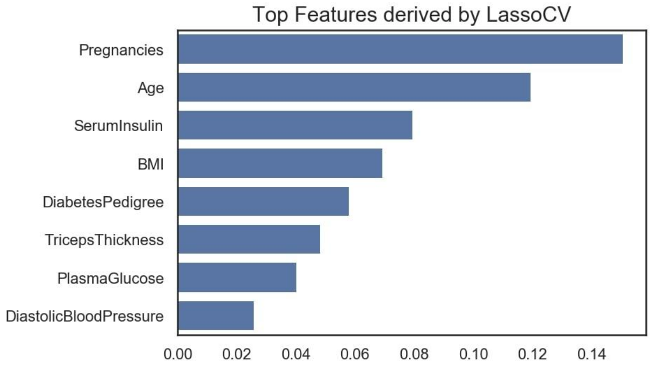


Figure 4.6: Top feature derived by LassoCV

Similarly, Random forest was used to determine the top feature.

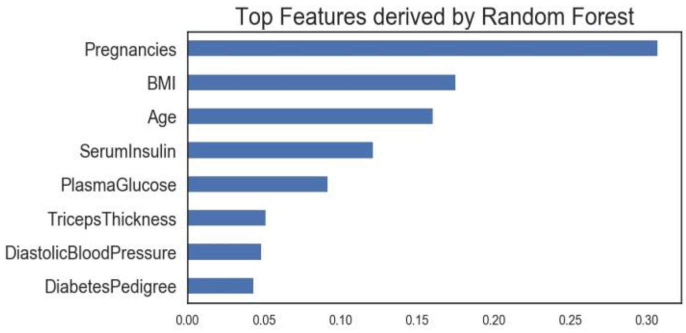
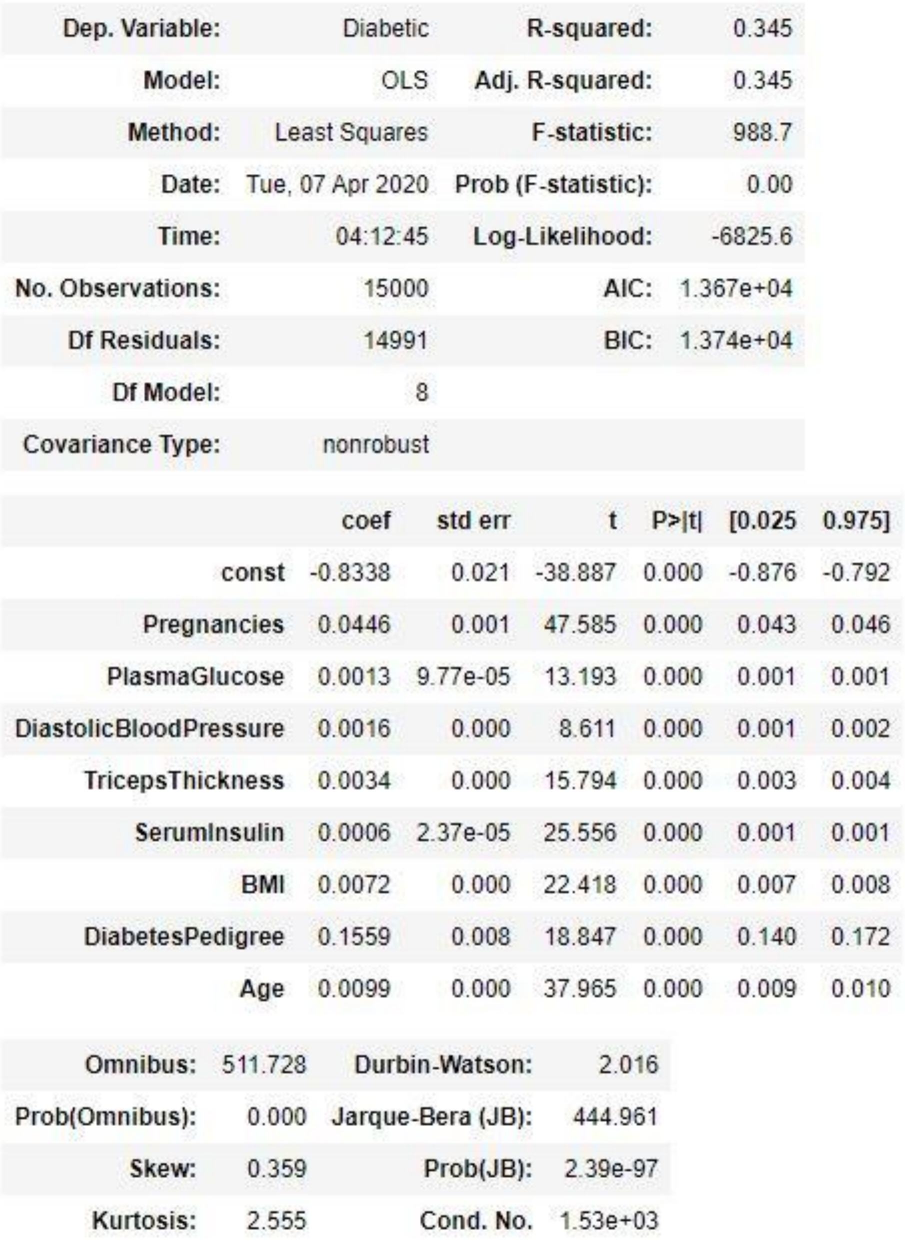


Figure 4.7: Top feature derived by Random Forest

To analyze the relationships between the features and the outcome a little further, Ordinary Least Square regression was implemented on the dataset. The results are as follows.



Dep. Variable:	Diabetic	R-squared:	0.345
Model:	OLS	Adj. R-squared:	0.345
Method:	Least Squares	F-statistic:	988.7
Date:	Tue, 07 Apr 2020	Prob (F-statistic):	0.00
Time:	04:12:45	Log-Likelihood:	-6825.6
No. Observations:	15000	AIC:	1.367e+04
Df Residuals:	14991	BIC:	1.374e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.8338	0.021	-38.887	0.000	-0.876	-0.792
Pregnancies	0.0446	0.001	47.585	0.000	0.043	0.046
PlasmaGlucose	0.0013	9.77e-05	13.193	0.000	0.001	0.001
DiastolicBloodPressure	0.0016	0.000	8.611	0.000	0.001	0.002
TricepsThickness	0.0034	0.000	15.794	0.000	0.003	0.004
SerumInsulin	0.0006	2.37e-05	25.556	0.000	0.001	0.001
BMI	0.0072	0.000	22.418	0.000	0.007	0.008
DiabetesPedigree	0.1559	0.008	18.847	0.000	0.140	0.172
Age	0.0099	0.000	37.965	0.000	0.009	0.010

Omnibus:	511.728	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	444.961
Skew:	0.359	Prob(JB):	2.39e-97
Kurtosis:	2.555	Cond. No.	1.53e+03

Table 4.3: OLS Regression results.

Chapter 5

Methodology

The entire project was implemented in Python (version 3.7.6), with the help several libraries and packages, namely pandas, matplotlib, numpy and scikit-learn.

After the preliminary data analysis and data pre-processing was completed, the features and the label were extracted. The first seven attributes in the dataset were declared as feature X and the last attribute, “Diabetic”, was specified as target y. The predictors were then scaled on both training and validation sets. After scaling, the top feature was derived, using LassoCV and Random Forest. The dataset was then split in 80:20 ratio, 8o% for training phase and 20% for testing phase. Once the splitting was completed, the model selection was performed using cross validation methods.

After cross validation, the baseline models were evaluated using several metrics. They were accuracy_score, precision_score, recall_score, f1score, rocauc and logloss. The models were then optimized for hyper-parameter tuning using GridSearchCV. Classification reports of the optimized models were generated and compared based on precision, recall and f1score. The confusion matrix, optimal threshold, logloss and auc score for each model were also calculated. The results of the tuned models were then concatenated in a tabular form alongside the results from the baseline models for careful evaluation. Four of the metrices used were plotted on separate histograms to demonstrate the differences between the performances of the algorithms used. Finally, ROC curves for all the models were plotted on the same axis for understating their performances.

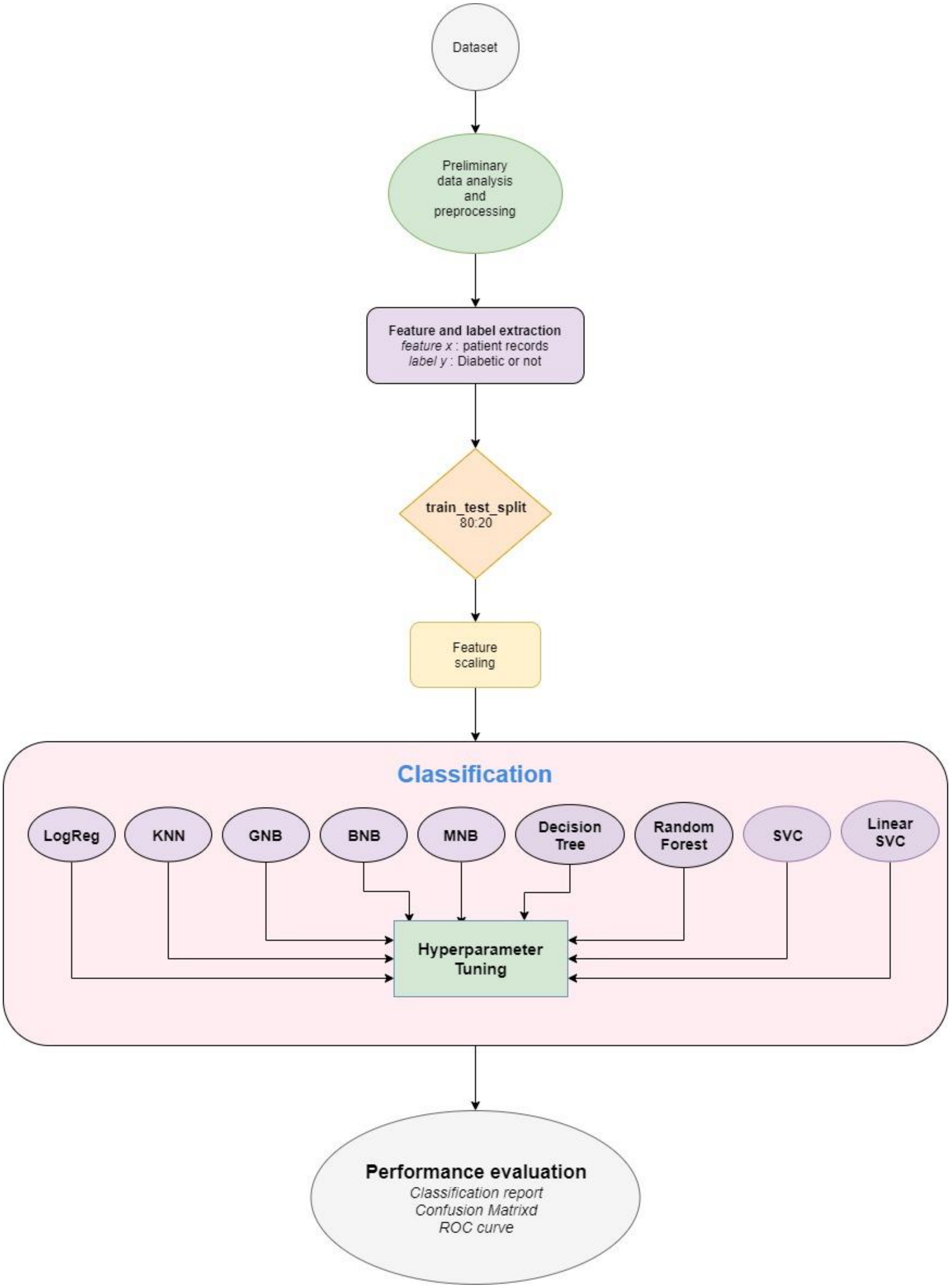


Figure 5.1: Flowchart of the proposed model

5.1 Cross validation

Cross validation is a great way to evaluate performances estimators or machine learning techniques. The dataset is split in to two parts, one for training the estimators and the other to test its performance. After the model is trained, its cross_val_score is calculated, which is the measurement of how good of a learner that model is. Scikit-learn library has several cross validation utilities that were used in this project for the evaluation and enhancement of performances of the various algorithms used here. Descriptions of some of them are as follows –

i. K-Fold - It is a resampling procedure with only one parameter, K, that assesses machine learning models on a finite data sample [24]. The data sample is to be divided in to K number of groups, hence the name K-Fold cross validation. The idea is to train a model using a little from the data sample, and approximate the performance of the model on unseen data, i.e. data not used in the training phase. The general procedure is as follows –
    1. The dataset is randomly shuffled.
    2. It is then split into k groups.
    3. For each unique group:
        a. That group is considered as test data
        b. The remaining groups are considered training data
        c. A model is fitted on the training data and assessed on the test data
        d. The score after the assessment is stored and the model discarded
    4. Using the model assessment scores, the expertise of each model is then outlined.

ii. Stratified K-fold – It is another variation of the K-Fold procedure that returns *stratified* folds; meaning the data is rearranged in order to make sure that each fold is a good representation of the entire set [25]. The folds are made by preserving the percentage of samples for each class.

5.2 GridSearchCV

It is a specialized and efficient parameter search strategy, provided by scikit-learn library in python, that thoroughly examines all possible parameter combinations [26]. It produces all possible prospects from a grid of parameter values specified with the param-grid parameter. Implementing all the usual estimator API, it judges all parameter combinations, froma given dataset to obtain the best parameter.

5.3 Metics [27]

5.3.1 Classification Report

It illustrates a number of classification metrics for each class. It provides a more comprehensive approach in the interpretation of the classifier over global accuracy which, in a multiclass problem, may fail to reveal the functional weaknesses in one particular class. These reports are utilized to differentiate between classification models in order to select models that have stronger metrics or that are relatively more unbiased. The terminologies used while computing these reports are true and false positives, and true and false negatives. Positive and negative in this case are generic names for the classes of a binary classification problem.

True positive is when an instance is labeled positive, by a classifier that is actually positive. A false positive is when the classifier labels an instance positive when in reality it was negative.

5.3.2 Precision

It is the ability of a classifier to correctly label an instance as positive which is actually positive. In other words, it’s the ratio of actual positive labels to all positive labels. Precision is computed using the following equation -

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

5.3.3 Recall

It is the ability of a classifier to find all positive instances. In short, it is the percentage of positive instances that were classified accurately. The following is the equation to calculate recall –

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

5.3.4 F1 score

It is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. As precision and recall scores are also used in its calculation, f1 scores are usually lower than accuracy measures. It is best practice to use the weighted average of F1 instead of global accuracy while differentiating between classifier models. The equation is as follows –

$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}\right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

5.3.5 ROC curve [28]

Receiver Operating Characteristics curve or roc curve is a probability curve that shows how well a classification model has performed. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis. The calculations are as follows –

- True Positive Rate (TRP)/ Recall/ Sensitivity = $\frac{True\ Positive}{True\ Positive + False\ Negative}$
- Specificity = $\frac{True\ Negative}{True\ Negative\ +\ False\ Positive}$
- False Positive Rate = 1 – Specificity

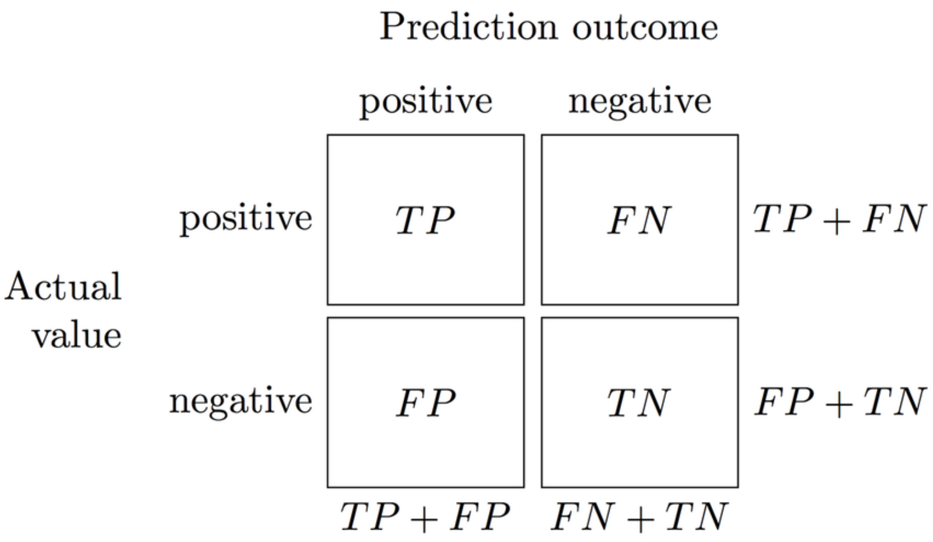
  $= \frac{False\ Positive}{True\ Negative + False\ Positive}$

5.3.6 AUC

Area under the Curve or auc is a useful tool to visualize the performance of classification models. It uses trapezoid rule to calculate the area to indicate the degree or separability. The higher the auc, the better the performance of the model.

5.3.7 Confusion Matrix

Confusion Matrix, which is also known as error matrix is used in Machine Learning especially in supervised learning, provides a specific table layout. In this layout, we can visualize the performance of an algorithm used in the model. It is apparently the least demanding approach to control the presentation of a classification model by looking at what number of positive occasions is effectively/mistakenly ordered and what number of negative occurrences is accurately/inaccurately grouped. Here, as appeared, the lines speak to the real label while the sections speak to the anticipated label.



		Prediction outcome		
		positive	negative	
Actual value	positive	TP	FN	$TP + FN$
	negative	FP	TN	$FP + TN$
		$TP + FP$	$FN + TN$	

Figure 5.2: Understanding Confusion Matrix

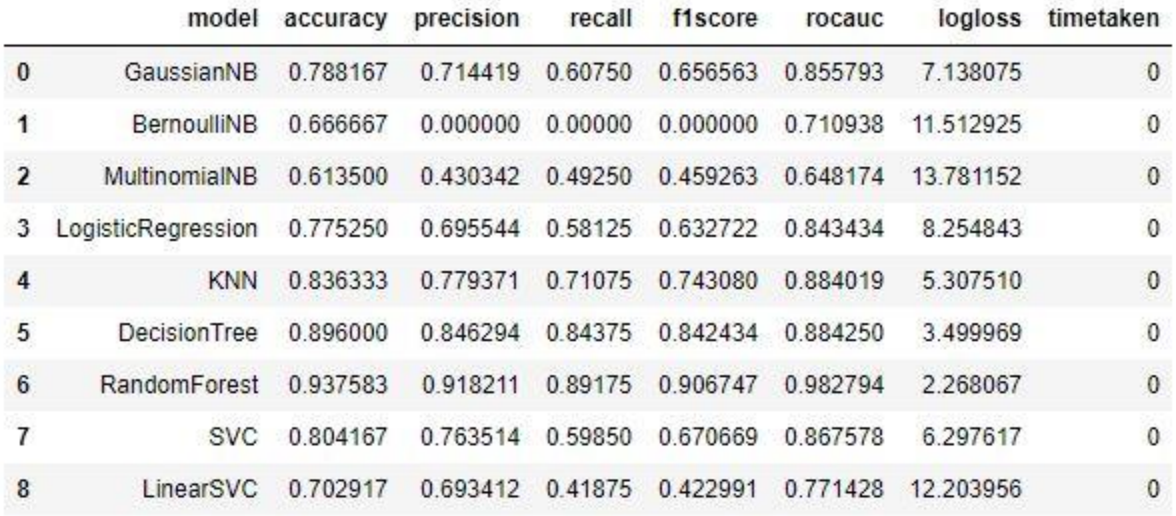
Chapter 6

Experimental Result Analysis

After obtaining the OLS Regression results, at the end of Dataset Analysis, the dataset was split using train_test_split utility from sklearn.model_selection package. The split was on a 80:20 ratio for train/test. A quick check of the shape revealed the dimensions as follows –

X_train (12000, 8)
y_train (12000,)
X_test (3000, 8)
y_test (3000,)

The baseline models were then executed and the performances evaluated. Stratified KFold was utilized for cross-validation purposes as to not result in imbalance classes in each fold.



	model	accuracy	precision	recall	f1score	rocauc	logloss	timetaken
0	GaussianNB	0.788167	0.714419	0.60750	0.656563	0.855793	7.138075	0
1	BernoulliNB	0.666667	0.000000	0.00000	0.000000	0.710938	11.512925	0
2	MultinomialNB	0.613500	0.430342	0.49250	0.459263	0.648174	13.781152	0
3	LogisticRegression	0.775250	0.695544	0.58125	0.632722	0.843434	8.254843	0
4	KNN	0.836333	0.779371	0.71075	0.743080	0.884019	5.307510	0
5	DecisionTree	0.896000	0.846294	0.84375	0.842434	0.884250	3.499969	0
6	RandomForest	0.937583	0.918211	0.89175	0.906747	0.982794	2.268067	0
7	SVC	0.804167	0.763514	0.59850	0.670669	0.867578	6.297617	0
8	LinearSVC	0.702917	0.693412	0.41875	0.422991	0.771428	12.203956	0

Table 6.1: Performances of baseline models.

After optimising the models by tuning, the optimal threshold for each classifier were found and consequent metrics were generated.

Gaussian NB:

Optimal threshold 0.247

```
Precision: 0.6222,   Recall: 0.7970,   F1 Score: 0.6988
GaussianNB confusion matrix:
[[1516  484]
 [ 203  797]]
GaussianNB AUC: 0.8545
GaussianNB Log-loss: 0.4995
```

Bernoulli NB:

```
Optimal threshold 0.013
Precision: 0.4656,   Recall: 0.9890,   F1 Score: 0.6332
BernoulliNB confusion matrix:
[[ 865 1135]
 [  11  989]]
BernoulliNB AUC: 0.7107
BernoulliNB Log-loss: 0.5088
```

Multinomial NB:

```
Optimal threshold 0.000
Precision: 0.4045,   Recall: 0.5590,   F1 Score: 0.4694
MultinomialNB confusion matrix:
[[1177  823]
 [ 441  559]]
MultinomialNB AUC: 0.6515
MultinomialNB Log-loss: 7.0741
```

Logistic Regression:

```
Optimal threshold 0.314
Precision: 0.6293,   Recall: 0.7810,   F1 Score: 0.6970
LogisticRegression confusion matrix:
[[1540  460]
 [ 219  781]]
LogisticRegression AUC: 0.8557
LogisticRegression Log-loss: 0.4411
```

K-Nearest Neighbour:

```
Optimal threshold 0.200
Precision: 0.6932,   Recall: 0.8520,   F1 Score: 0.7645
KNN confusion matrix:
[[1623  377]
 [ 148  852]]
KNN AUC: 0.8962
KNN Log-loss: 1.1485

Compare with KNN classification_report (same as default threshold 0.50)
              precision    recall  f1-score   support

           0       0.87      0.90      0.89      2000
           1       0.79      0.73      0.76      1000

    accuracy                           0.85      3000
   macro avg       0.83      0.82      0.82      3000
weighted avg       0.84      0.85      0.84      3000
```

Decision Tree:

```
Optimal threshold 0.429
Precision: 0.8535,   Recall: 0.8800,   F1 Score: 0.8666
DecisionTree confusion matrix:
[[1849  151]
 [ 120  880]]
DecisionTree AUC: 0.9351
DecisionTree Log-loss: 0.8405
```

Random Forest:

```
Optimal threshold 0.497
Precision: 0.9178,   Recall: 0.8930,   F1 Score: 0.9052
RandomForest confusion matrix:
[[1920   80]
 [ 107  893]]
RandomForest AUC: 0.9820
RandomForest Log-loss: 0.1700
```

Support Vector Classifier (SVC):

```
Optimal threshold 0.309
Precision: 0.6653,   Recall: 0.7950,   F1 Score: 0.7244
SVC confusion matrix:
[[1600  400]
 [ 205  795]]
SVC AUC: 0.8756
SVC Log-loss: 0.4136
```

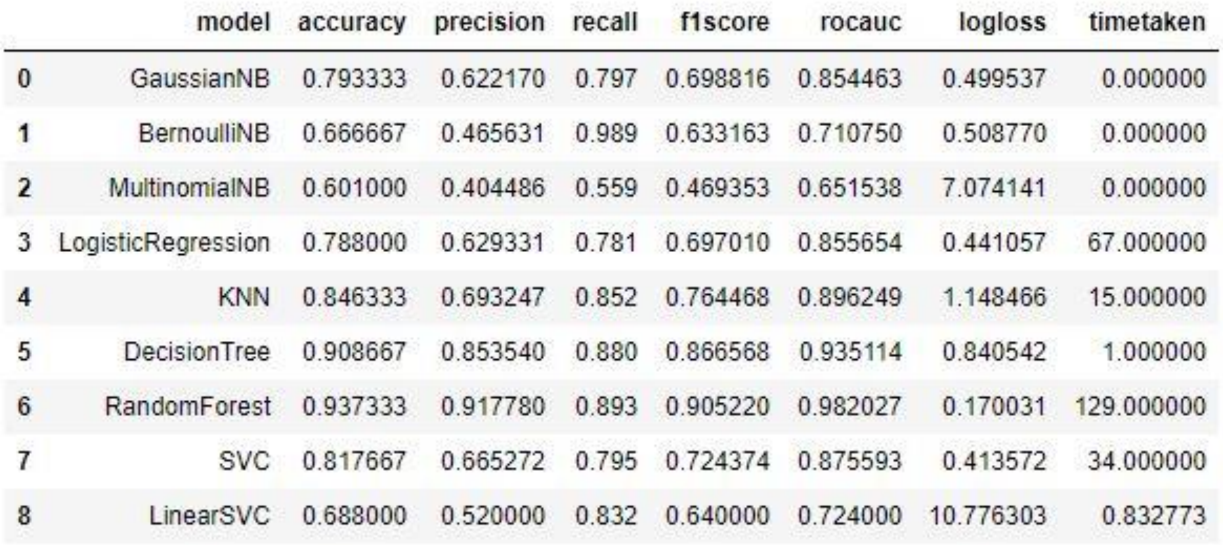
Linear SVC:

```
                LinearSVC accuracy score is
                     Training: 68.94%
                     Test set: 68.80%
          precision    recall  f1-score   support

        0       0.88      0.62      0.72      2000
        1       0.52      0.83      0.64      1000
  accuracy                           0.69      3000
 macro avg       0.70      0.72      0.68      3000
weighted avg       0.76      0.69      0.70      3000

                LinearSVC confusion matrix:
                       [[1232  768]
                       [ 168  832]]
                   LinearSVC AUC: 0.7240
                LinearSVC Log-loss: 10.7763
```

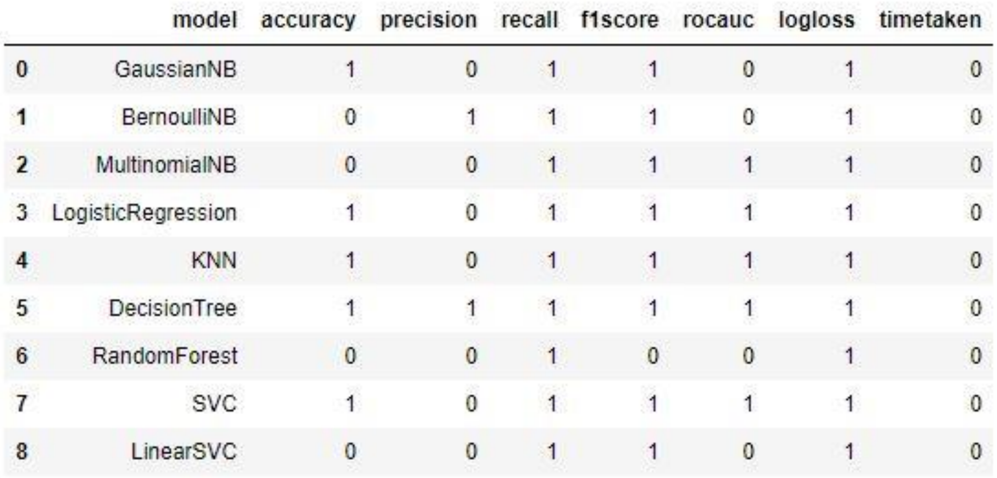
After concatenating the tuned models –



	model	accuracy	precision	recall	f1score	rocauc	logloss	timetaken
0	GaussianNB	0.793333	0.622170	0.797	0.698816	0.854463	0.499537	0.000000
1	BernoulliNB	0.666667	0.465631	0.989	0.633163	0.710750	0.508770	0.000000
2	MultinomialNB	0.601000	0.404486	0.559	0.469353	0.651538	7.074141	0.000000
3	LogisticRegression	0.788000	0.629331	0.781	0.697010	0.855654	0.441057	67.000000
4	KNN	0.846333	0.693247	0.852	0.764468	0.896249	1.148466	15.000000
5	DecisionTree	0.908667	0.853540	0.880	0.866568	0.935114	0.840542	1.000000
6	RandomForest	0.937333	0.917780	0.893	0.905220	0.982027	0.170031	129.000000
7	SVC	0.817667	0.665272	0.795	0.724374	0.875593	0.413572	34.000000
8	LinearSVC	0.688000	0.520000	0.832	0.640000	0.724000	10.776303	0.832773

Table 6.2: Performances of the tuned models.

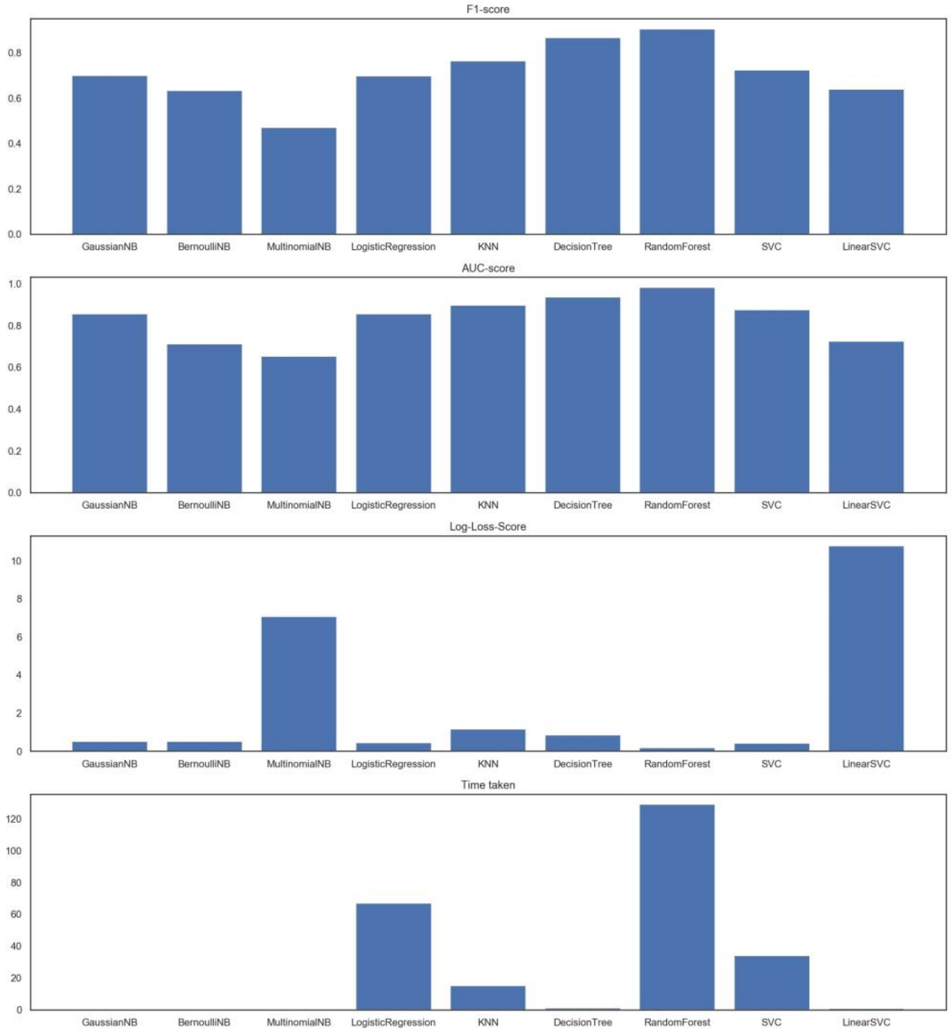
A comparison between the baseline model performances and the tuned model performances were evaluated and the improvements illustrated in a tabular form. In this table, 0 means no improvem ent and 1 means tuned models have improved.



	model	accuracy	precision	recall	f1score	rocauc	logloss	timetaken
0	GaussianNB	1	0	1	1	0	1	0
1	BernoulliNB	0	1	1	1	0	1	0
2	MultinomialNB	0	0	1	1	1	1	0
3	LogisticRegression	1	0	1	1	1	1	0
4	KNN	1	0	1	1	1	1	0
5	DecisionTree	1	1	1	1	1	1	0
6	RandomForest	0	0	1	0	0	1	0
7	SVC	1	0	1	1	1	1	0
8	LinearSVC	0	0	1	1	0	1	0

Table 6.3: Demonstrating Improvement

The performance metric scores, namely F1-score, AUC-score, Log0Loss-Score and the Time taken, were plot.



F1-score

AUC-score

Log-Loss-Score

Time taken

Figure 6.1: Performance Metrics

Separate roc curves for each classifier were plotted on the same axis and the area under each curve analyzed. The curve with the most area under the curve is the best classifier among the group.

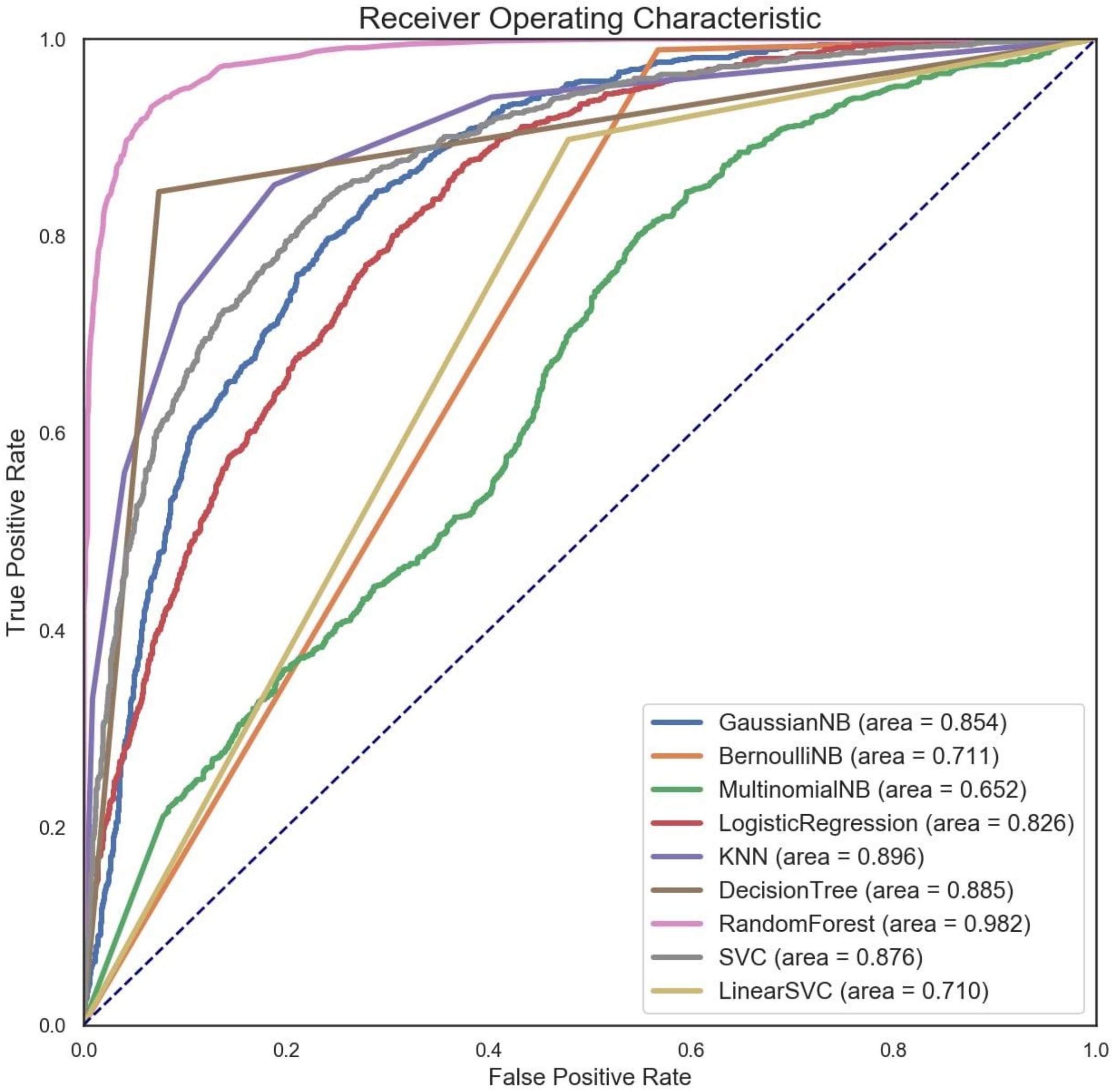
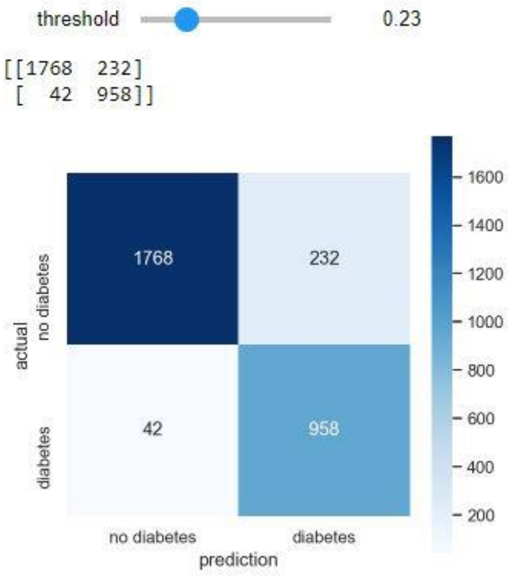


Figure 5: ROC curve

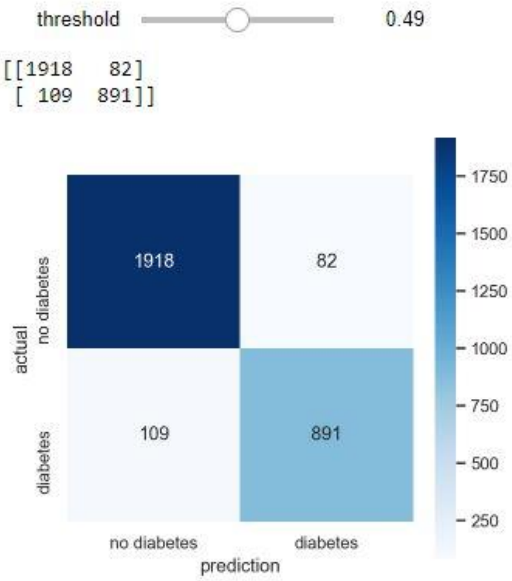
Since the diagram clearly depicts that the area under the curve for Random Forest is almost equal to 1, it is evident that the Random Forest classifier has outperformed the rest of the classifiers on the given dataset.

In order to investigate the effect of threshold on the confusion matrix, we selected the best model, that is the Random Forest classifier, and created an interactive plot of the confusion matrix that varied with threshold.



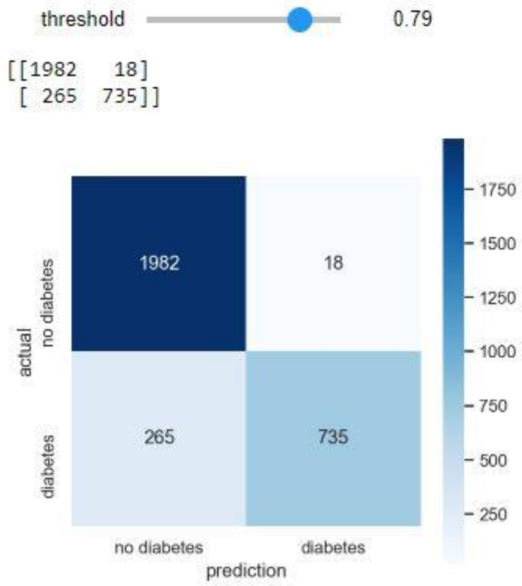
```
[[1768  232]
 [  42  958]]
```

Figure 6: Less than optimal threshold



```
[[1918   82]
 [ 109  891]]
```

Figure 6.4: Optimal threshold



threshold 0.79

```
[[1982   18]
 [ 265  735]]
```

Figure 6.5: Greater than optimal threshold

As we can observe that with increasing threshold, the values for True Negative and False Negative are increasing, whereas, values for True Positive and False Positive are decreasing.

Finally, the auc_scores of Random Forest and roc_auc_score, before and after tuning, were calculated to determine if there was any noticeable difference. The final values show the difference to be of a significant amount.

```
randomforest roc_auc_score: 0.9237
randomforest AUC before tuning: 0.9237
randomforest AUC after tuning: 0.9819
```

Chapter 7

Conclusion and Future works

7.1 Conclusion

The main purpose of this project was to choose an efficient diabetes predictions model. A total of 9 different classifiers were utilized in this study. The dataset containing 15000 observations had 8 different attributes. The results indicate that Random Forest classifier produced the best performance among all these classifiers with optimal threshold of 0.497. The precision, recall and f1score were 0.9178, 0.8930 and 0.9052 respectively and auc was 0.982. Top feature selected by Random Forest was Pregnancies. However, when the same procedure was implemented PIMA database, as an explorative analysis, the results were very different. The top feature selected both by LassoCV and Random Forest for that database was Glucose and the best classifier on that database proved to be the Gaussian NB. The precision, recall, f1score and auc for GNB were 0.5467, 0.7593, 0.6357 and 0.7646 respectively.

7.2 Future Works

Since the same model produced different results on two different datasets, the model could be applied on several other datasets with varying number of observations. As this study was intended to be applied on local health care systems, local datasets could be obtained for future use. Different fields of research could be chosen for evaluating the performances of these classifiers. The use of Boosting or Neural Networks could result in much more improved models. Even though hybrid models exist at present, a more sophisticated combination of techniques could be assembled for building the ultimate prediction model.

Bibliography

[1] W. Chen, S. Chen, H. Zhang and T. Wu, "A hybrid prediction model for type 2 diabetes using K-means and decision tree," *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, 2017, pp. 386-390.

[2] M. Sultana, A. Haider and M. S. Uddin, "Analysis of data mining techniques for heart disease prediction," *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, 2016, pp. 1-5.

[3] A. Kunjir, H. Sawant and N. F. Shaikh, "Data mining and visualization for prediction of multiple diseases in healthcare," *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, Chirala, 2017, pp. 329-334.

[4] V. V. Vijayan and C. Anjali, "Prediction and diagnosis of diabetes mellitus — A machine learning approach," *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Trivandrum, 2015, pp. 122-127.

[5] S. Vinitha, S. Sweetlin, H. Vinusha and S. Sajini, "Disease Prediction Using Machine Learning Over Big Data", *Computer Science & Engineering: An International Journal (CSEIJ)*, Vol.8, No.1, February, 2018.

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

[7] D. Shetty, K. Rit, S. Shaikh and N. Patil, "Diabetes Disease Prediction Using Data Mining", *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017.

[8] Berina A., L. Gurbeta and A. Badnjevic, "Machine learning techniques for classification of diabetes and cardiovascular diseases", *2017 6th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO*), 11-15 JUNE 2017, BAR, MONTENEGRO.

[9] Theresa P. R. and J. Thoams, " Human Heart Disease Prediction System using Data Mining Techniques", *2016 International Conference on Circuit, Power and Computing Technologies [ICCPCT]*, 2016.

[10] A. Tikotikar and M. Kodabagi, “ A Survey on Techniques for Prediction of Disease in Medical Data”, *2017 International Conference On Smart Technology for Smart Nation*, 2017.

[11] Byju N. B., Alexander G., Devanand P. and Ravindra K. R., “A Knowledge Framework to Search Similar Disease Patterns using Data Mining”, *2009 IEEE International Advance Computing Conference (IACC 2009)*, Patiala, India, 6-7 March 2009.

[12] Abhishek R., V. Kulkarni, V. Sabale, H. Kale and A. Lagunde, “Heart Disease Prediction using Data Mining Techniques”, *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017.

[13] N. H. Ibrahim, A. Mustapha, R. Rosli and N. H. Helmee, “A Hybrid Model of Hierarchical Clustering and Decision Tree for Rule-Based Classification of Diabetic Patients, *International Journal of Engineering and Technology (IJET)*, Vol. 5 No 5 Oct-Nov 2013.

[14] O. Olabode and B. T. Olabode, “Cerebrovascular Accident Attack Classification using Multilayer Feed Forward Artificial Neural Network with Back Propagation Error”, *Journal of Computer Science*, Vol. 8 (1): Page 18-25, 2012.

[15] Lundin M, Lundin J, Burke H, B, Toikkanen S, Pylkkänen L, Joensuu H, “Artificial Neural Networks Applied to Survival Prediction in Breast Cancer”, *Oncology*, Vol. 57 Page 281-286, 1999.

[16] A. Ramalaniya, “Prediction of Chances Diabetic Retinopathy using Data Mining Classification Techniques”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 6, Issue 8, August 2017.

[17] J. Zornoza, “Logistic Regression Explained”, Feb 2020. [Online]. Available: https://towardsdatascience.com/logistic-regression-explained-9ee73cede081

[18] “KNN Algorithm – Finding Nearest Neighbors”, [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm

[19] "Naïve Bayes Classifiers", [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[20] "Decision Tree", [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[21] "Classification Algorithms – Random Forest", [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm

[22] "Support Vector Machines", [Online]. Available: https://scikit-learn.org/stable/modules/svm.html#

[23] F. Fernandez, "Diabetes from DAT263x Lab01", [Online]. Available: https://www.kaggle.com/fmendes/diabetes-from-dat263x-lab01

[24] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation", May 2018, [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/

[25] XZZ, "StratifiedKFold v.s KFold v.s StratifiedShuffleSplit", Jan 2020, [Online], Available: https://medium.com/@xzz201920/stratifiedkfold-v-s-kfold-v-s-stratifiedshufflesplit-ffcae5bfdf

[26] "Exhaustive Grid Search", [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html#grid-search

[27] "Classification Report", [Online]. Available: https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html

[28] S. Narkhede, "Understanding AUC – ROC Curve", June 2018, [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5