A Supervised Learning Approach by Machine Learning and Deep Learning Algorithms to Predict Type II DM Risk

by

Abdullah Al Farabe
15101081
Tarin Sultana Sharika
15301131
Nahian Raonak
15301109
Ghalib Ashraf
19141023

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

Department of Computer Science and Engineering
BRAC University
September 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our 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. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Abdullah Al Farabe
15101081

Nahian Raonak
15301109

Tarin Sultana Sharika
15301131

Ghalib Ashraf
19141023

Approval

The thesis/project titled “Indoor Positioning Techniques using RSSI from Wireless Networks” submitted by

1. Abdullah Al Farabe (15101081)
2. Tarin Sultana Sharika (15301131)
3. Nahian Raonak (15301109)
4. Ghalib Ashraf (19141023)

Of Summer, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 28, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Jia Uddin, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor and Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

The application of Artificial intelligence (AI) has become a valuable part of medical research. These days diabetes is one of the top maladies on the planet. Nowadays it has become a common disease and alarming as people are living in polluted areas and eating unhygienic foods. People with diabetes are probably going to pass on at a more youthful age than individuals who don't have diabetes. We hope this study could be very helpful in medical science to predict the risk score of type II Diabetes Mellitus (DM). Our model consists of four machine learning algorithms which are-K-Nearest Neighbor, Random forest, Decision tree and Logistic Regression. These algorithms have been applied on a dataset containing 15000 type 2 diabetes patients along with eight features that describe the state of patients such as glucose, BMI, age, pregnancy, blood pressure (BP), Diabetes Pedigree Function, Skin thickness and insulin. Moreover, one deep learning algorithm called CNN has been used. All of the five algorithms have been used on the dataset and the Random forest gives the best accuracy of almost 92.60 percent where other algorithms give less accuracy.

Keywords: Linear Discriminant Analysis (LDA), Logistic Regression, Random Forest, Decision Tree, KNN and CNN.

Dedication

We would like to dedicate this thesis to our loving parents.

Acknowledgement

We want to dedicate our acknowledgement of gratitude to our thesis supervisor Amitabha Chakrabarty, PhD, Associate Professor, Department of Computer Science and Engineering of BRAC University for his guidance for the completion of our thesis. We are thankful to CSE department, BRAC University for providing us the necessary equipment for the completion of this project.

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

β	An association between set of input and output values are termed as beta function
δ	Shows difference between two numbers
ϵ	The permitivity of free space
γ	A simply closed curve on a complex plane
α	representing quantities such as angles
ADAM	Adaptive Moment Estimation
AHA	American Heart Association
AI	Artificial Intelligence
AUC	Area Under Curve
BP	Blood Pressure
CART	Classification and Regression Trees
CNN	Convolutional Neural Network
IMID	Immune-mediated Inflammatory Disease
KNN	K- Nearest Neighbor Algorithm
LDA	Linear Discriminant Analysis
ROC	Receiver Operating Characteristic
T1DM	Type 1 Diabetes Mellitus
T2DM	Type 2 Diabetes Mellitus

Chapter 1

Introduction

1.1 Overview

In our research we have worked for medical data consists of 15 thousands patient information which has helped us to predict Type II DM (Diabetes Mellitus) risk. Our main purpose here is to predict diabetes early so that medical sites can take early steps to prevent it. We have used five algorithms to predict, have shown data analysis to show the correlation among the attributes and also have shown the performance of our model. In the medical field after analyzing medical data we use AI to estimate human cognition. Without direct human input AI has the ability to get valuable results which can bring to great innovation. Diabetes is one of the most widely recognized maladies for which many people die without doing any kind of surgery. Diabetes does not let patients have surgery and The American Heart Association (AHA) has found that almost 68 percent of people die because of diabetes. These are the criteria by which a diabetic patient could predict himself as starting of diabetes of having thirst unnecessarily, troubling to focus, suddenly losing weight, losing energy, consistent urination, tiredness, constant hunger.

Diabetes type I basically occurs for the destruction of beta-cell due to a shortage of insulin [1]. In the exact same way in diabetes type II increasing of glucotoxicity, lipotoxicity, endoplasmic reticulum set on stress and death cells occurs which leads to the dynamical loss of beta cells. Diabetes type I depends on immune-mediated and idiopathic [2]. Diabetes type II consists of defecting by genetic of beta-cell function, insulin action, pancreatic disease, drug-induced and many more. Diabetes Type I is an unusual form of phenotypic with almost complete lack of a shortage of insulin. Another unrecognized for diabetes type II is Pancreatic disease. It is largely related to alcohol excess. Alcohol can cause the destruction of the pancreas. Moreover, defects in insulin secretion can cause monogenic diabetes and those usually effect under 25 age. It mostly commonly affected by diabetes type II. As we can see there are so many syndromes of diabetes that we can have some are recognized and some are still unrecognized.

In our thesis, we have analyzed on diabetes type II. We have a dataset that includes seven input attributes and one outcome attribute. The eight inputs are -Pregnancies, Glucose, BP, Skin, Insulin, BMI, Pedigree, Age and outcome (if they have diabetes then 1 and if they do not have diabetes it is 0). Machine learning, artificial intelligence and deep learning have become a noble approach to detect diseases or to prevent all kinds of diseases at an early state. Lots of algorithms have been made which are contributing to medical science a lot and making doctors work more comfortable. Many kinds of machine learning algorithms can give us accuracy and predictions with different ways of output. The output could be as variables even as various diagrams in which we want to see them. We use many kinds of parameters to get the outputs. At first, we have used Linear Discriminant Analysis (LDA) for data preprocessing [3]. For our research purpose, we are using four machine learning algos like- Logistic Regression, Random Forest, KNN and Decision tree to get the best prediction algorithm and to classify it. Moreover, we want to use one deep learning algorithm like – CNN. Four algorithms from machine learning and one algorithm from deep learning so a total of five algorithms. Before using these algorithms first we have to analyze the dataset attributes. We have checked co-relationship among the attributes. That is why we have used heatmap and bar charts to show relationships among attributes. Moreover, it shows on which attribute the output relies on most. The correlation and relying on attribute varies the good quality of the dataset. We take the most two correlated input attributes with the output to plot a scatter diagram. The diabetes dataset consists of fifteen thousand patient information. Next, we have used AUC-ROC curves to show how much model is able to characterize among classes [4]. Moreover, we have discussed about sensitivity, specificity, confusion Matrix, performance metrics, precision, negative predictive value, false-positive rate, false discovery rate and false-negative rate. In addition, we have also focused on accuracy, F1 score, and Matthews Correlation Coefficient to show the performance of our model.

1.2 Motivation

Diabetes is now occurring in tremendously day by day. It is causing problem, especially in urban areas than in rural areas. At urban area people use to eat fast food, oily food, unhygienic food which consists of unhealthy. Maximum people are not health conscious and for this they suffer with diabetes so easily. Even children at very early age fell into diabetes type 1 which could be declared as how much our society is evolving with health danger. Every year millions of people die because of having diabetes. The biggest problem for having diabetes as this disease is connected with other diseases too such as – sight loss, kidney disease, heart disease, brain stoke amputations. Moreover, before having an operation doctor have to check patient diabetes and if it is high then it is highly risky to have operation. That is why doctor asks the patients to control diabetes before having an operation. Diabetes can lead to early death. Nowadays researchers are finding and researching protocols to solve this issue and these protocols can prevent human distinction.

1.3 Objectives

1. Our paper defines a model utilizing Machine Learning algorithms.

2. It can effectively anticipate the likelihood of various infections beginning due to diabetes type II (T2DM).

3. It can successfully predict the probability of T2DM due to the use of machine learning algorithms.

4. Using those algorithms we are predicting accuracy and finding that which algorithm gives the best accuracy.

5. We have used LDA to avoid unnecessary values from the dataset and using correlations among attributes. Correlation refers to the connection among the attributes and it also helps which attributes diabetes is dependent on so that medical science can give more test on that particular attribute.

6. To predict the next fives years possibilities of having diabetes.

7. Predicting accuracy by reducing errors and overfitting as much as possible.

1.4 Thesis Outline

The following segment in our research tells the comparative past works which have done by different researchers. After that, there is a detailed discussion of the dataset and features used in our model. The next section of this paper explains an insight into the dataset including numeric features and categorical features. In this section 4 has discussed missing values imputation and train-test split. Next, in section 5, we have discussed our used algorithms. Then in section six of this paper explains the system implementation and results from analysis including the performance metrics, confusion false discovery rate, Matrix, sensitivity, specificity, precision, negative predictive value, false-negative rate and false-positive rate. Also have focused on F1 score, accuracy and Matthews Correlation Coefficient to show the performance of our model. Finally, the paper closes with a summary of the whole model along with the subtleties of the difficulties looked in this task and few remarks.

Chapter 2

Literature Review

In engineering Machine Learning is a part of Artificial Intelligence (AI) which works as a framework and without being explicitly modified to improve for a fact. Machine Learning is showing machines how to get the hang of utilizing Statistics and Probability. Machine Learning divides the significant classes by verifying data. [5] Supervised dataset contains input and output. The training dataset contains inputs data and the values which we want to predict as numerical or categorical value. Training or testing will help to learn about a link between inputs and outputs. Linking between input and output our algorithms will train and test to identify the accuracy. Unsupervised dataset contains only input. There is no output like supervised data. Algorithms mainly train and test the inputs not using outputs. Reinforcement learning is a technique which finds the best way to get the destination result for the betterment of training. This strategy enables the machine to locate the most ideal approaches to procure the best reward. Prizes can be winning a game, procuring more cash or beating different rivals.

Nowadays, several researchers have been working on it to predict the best accuracy algorithm with various ways of results. The medical field has become an important field as for many reasons people are getting weaker for food habits, sleepless nights, stress, etc. The biomedical science field is becoming stronger as the whole world is relying on their health concerns. Each hospital has to get through the big dataset for their research purpose to prevent those diseases or to find out for which main reasons diseases are spreading. From Kaggle we have found a dataset of diabetes which consists of diabetes and non-diabetes [6]. They have used eight features such as- Pregnancies, glucose, BP, skin, insulin, BMI, pedigree and age. They have used these eight features to get the accuracy and diagrams to see the output.

Fernando Fernandez, et al [6], studied on PIMA Indians Diabetes dataset. They have implemented some machine learning calculation such as Logistic Regression, Gradient Boosting and Decision Tree. In their research Logistic Regression gives 80.73%, Decision Tree gives 75% and Gradient Boosting gives 80.73% accuracy.

P.Sujarani et al [7], this paper predicts Diabetes using Artificial Neural Networks algorithms. this paper predicts Diabetes using Artificial Neural Networks algorithms. They have used Probabilistic ANN (Artificial Neural Network), PNN (Neural Network), with GA and Generalized Regression Neural Network (GRNN) which give

73.4%, 89.56% and 80.21% accuracy. They have also added studied from other literature and used their dataset by using several techniques. They have used another two types of dataset with different attributes. One has studied by Logistic Regression and ANN which give 76.13% and 73.23% accuracy. Another one is used by PNN and Levenberg–Marquardt (LM) which give 78.05% and 79.62% accuracy.

Zuo, Q et al [8], this paper is about using machine learning techniques to predict diabetes Mellitus. They have used Neural network, Random Forest, and Decision Tree as algorithms. After analyzing all of the algorithms for their prediction Random Forest shows the best accuracy which is 80.84%.

Karegowda, A. G et al [9], this paper consists of a neural network algorithm known as Genetic Algorithm and it connects the parameters for medical diagnosis. Their proposal methods says that the technique for crossover model that coordinates Genetic Algorithm and Back Propatation organize (BPN) where GA is utilized to introduce and upgrade the conjunction loads of BPN. They have found accuracy from GA with BPN classification is 77.07-84% for various parameters and classifiers.

Rahman, T. et al [10] has predicted to incite diabetes complications by using machine learning. They have used Logistic Regression, SVM, Naïve Bayes, Decision tree, Decision Tree AdaBoost and RF (Random Forest). From the beginning model Logistic regression gives 79%, SVM gives 82%, Naive Bayes gives 79%, Decision Tree gives 89%, Decision Tree Adaboost gives 87% and Random Forest gives 89% accuracy. In these algorithms we are almost familiar to all as machine learning algorithms but except Decision Tree Adaboost. Decision Tree is boosted by Adaboost which contains 1D dataset. is boosted.

Alic, B. et al [11] has proposed about classifying diabetes by using Machine learning algorithms.especially for cardiovascular disease. Cardiovascular disease, kidney failure, fatty liver etc various diseases are dependent on diabetes. The literature consists of using Artificial Neural Network (ANN) which is another useful deep learning algorithm to an interconnected group of nodes. Using ANN or other deep learning algorithms is like solving problems exactly like a human brain. So, in this paper, ANN gives 80-90% accuracy. They have also used Bayesian Network (BN) which gives 78% and Naive Bayesian network which gives 71% accuracy. They have a highly good score but their results may fluctuate as they mentioned their accuracy distance is much too big.

Vincent Lugat [12] from online kaggle, the project is about EDA and prediction on PIMA Indian diabetes by using RandomSearch + LightGBM - Accuracy = 89.8% and GridSearch + LightGBM KNN- Accuracy = 90.6%. A model named hyperparameter which is a characteristic of a model and hard to estimate from data. Grid-search is used To solve hyperparameters Grid-search is used to get the most accurate result. LightGBM is a parameter to get an accuracy.

From the above discussion, we have studied that all the literatures have used machine learning and deep learning algorithm to predict early to prevent it. We have also worked with four machine learning selected algorithms such as - RF (Random

Forest), LR (Logistic Regression), KNN, Decision Tree and for deep learning - CNN algorithm. We have also clarified and described using those five algorithms. Moreover, we have used LDA for data preprocessing system so that we can reduce the chance of overfitting. In addition, we have showed correlation among the attributes and also discussed briefly about analyzing the dataset. Through our analyzing on our dataset it includes confusion Matrix, performance metrics, sensitivity, specificity, precision, false-positive rate, negative predictive value, false-negative rate, false discovery rate, accuracy, Matthews Correlation Coefficient and F1 score to show the performance of our model.

Chapter 3

Research Methodology

3.1 Data Dimensions

In our research we have worked with eight attributes and one output which consists of “yes=1” and “no= 0”. “Yes” means he/she has diabetes and “no” means he/she doesn’t have any diabetes. Database is supervised as it has output. Moreover, it has only numerical value inputs. And categorical output. Ut we have changed the output from categorical to numerical value. It does not consist of any string values. Our diabetes data consists of nine attributes such as -
BP = Blood Pressure
BMI = BMI means Body Mass Index. It is predict a person’s weight regarding his/her height which is used by the medical profession.
Pedigree= diabetes condition and status in family members
Pregnancies= number of pregnancy
Age= number of age
Skin= skin thickness of our body
Pedigree= In biology study a diagram that shows the relationship between an ancestor and his/her organism.
Glucose= Sugar level in our blood cell.
Output = has diabetes (1) or not (0).

3.2 Proposal Method

In our proposed model we are using around 15000 patient diabetes type II patient information. As we are using a big dataset so there could be lots of null or unnecessary values which could be a big problem to get actually accuracy. So that we have used Linear Discriminant Analysis (LDA) algorithm to remove all those unnecessary values. From the data -

1. Marking missing values from the dataset.
2. Where we perceive how an AI calculation can affect badly when it contains missing qualities.
3. Removing rows with missing Values.
4. Replacing missing values.

After those steps we have used five algorithms - Logistic Regression, Random Forest, Decision Tree, KNN and CNN. In this model, we have applied several machine learning and deep learning algorithms on a dataset containing 15000 type 2 DM patient data to determine the algorithms which provides us the best accuracy of diabetes prediction. Figure 3.1 below represents workflow of the model.

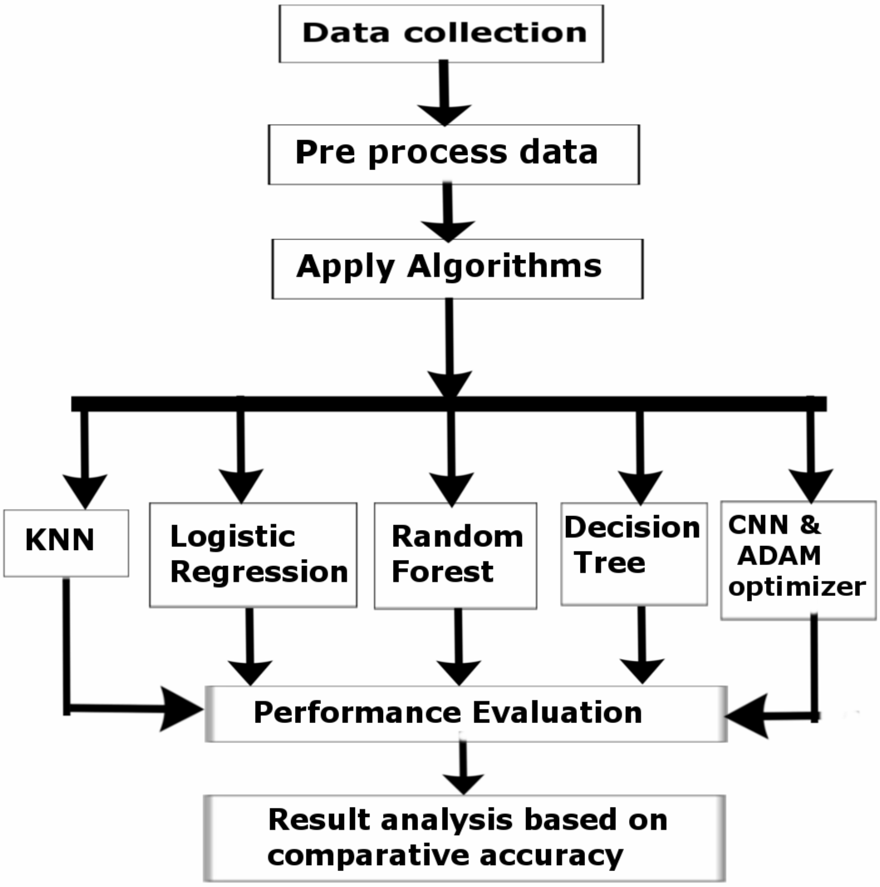


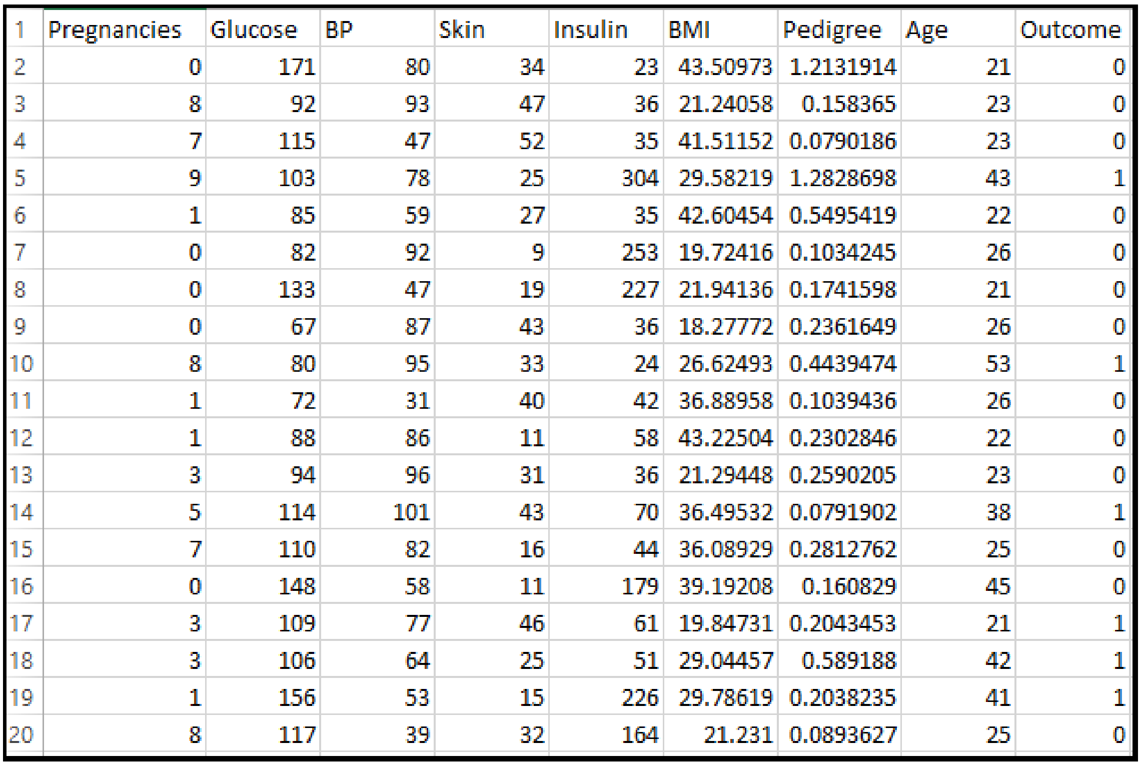
Figure 3.1: Proposal model of predicting accuracy by algorithms

After that we have selected features for our model and applied different classification algorithms such as KNN, Logistic Regression(LR),Random Forest,Decision Tree and a deep learning algorithm which is CNN. By applying these algorithms we have measured performance evaluation and picked the best algorithm which predicts diabetes most accurately. Moreover, we have analyzed our whole dataset by doing targeting variables. Analysing the whole diabetes type-2 dataset about giving inputs and outputs. Finding correlations among the attributes to show the relationship among them. Then we have used heatmap, boxplot, Scatter matrix and Histogram to analysis our dataset. Through all these plots are useful as it shows us the correlation among the attributes. The bonding among attributes which helps us to get a better quality method. In addition, we have used statistical analyses like - Confusion Matrix to show the quality of our models. Specificity, Precision, Sensitivity,Negative Predictive Value with also some false rates. Also have showed accuracy, Matthews Correlation Coefficient and F1 Score to get better performance through our models. Moreover, we have used ROC-AUC curve for showing the performances of our processes.

Chapter 4

Dataset

We have collected a supervised diabetes dataset from kaggle.com [6] which contains 15000 instances of attributes including pregnancy, Glucose level, blood pressure (BP), Skin Thickness, Insulin, BMI, Pedigree, Age and Output. Supervised dataset is a sort of AI calculation that utilizes a known dataset (called the preparation dataset) to make forecasts. Supervised dataset is given as input and output. It is basically examining both inputs and outputs features concluding with a result of betterment path to solve any kind of problem. The training dataset includes input data and response values. The whole dataset can be classified into two categories of attributes numerical and categorical.



1	Pregnancies	Glucose	BP	Skin	Insulin	BMI	Pedigree	Age	Outcome
2	0	171	80	34	23	43.50973	1.2131914	21	0
3	8	92	93	47	36	21.24058	0.158365	23	0
4	7	115	47	52	35	41.51152	0.0790186	23	0
5	9	103	78	25	304	29.58219	1.2828698	43	1
6	1	85	59	27	35	42.60454	0.5495419	22	0
7	0	82	92	9	253	19.72416	0.1034245	26	0
8	0	133	47	19	227	21.94136	0.1741598	21	0
9	0	67	87	43	36	18.27772	0.2361649	26	0
10	8	80	95	33	24	26.62493	0.4439474	53	1
11	1	72	31	40	42	36.88958	0.1039436	26	0
12	1	88	86	11	58	43.22504	0.2302846	22	0
13	3	94	96	31	36	21.29448	0.2590205	23	0
14	5	114	101	43	70	36.49532	0.0791902	38	1
15	7	110	82	16	44	36.08929	0.2812762	25	0
16	0	148	58	11	179	39.19208	0.160829	45	0
17	3	109	77	46	61	19.84731	0.2043453	21	1
18	3	106	64	25	51	29.04457	0.589188	42	1
19	1	156	53	15	226	29.78619	0.2038235	41	1
20	8	117	39	32	164	21.231	0.0893627	25	0

Figure 4.1: Pima Indians Diabetes Dataset

4.1 Numeric Features

Among the features of the dataset age, glucose level, pedigree, pregnancy, BMI, blood pressure, skin thickness and insulin are numeric features. We are using all these numeric features for our project. We might be aware of the fact that glucose has a direct connection to diabetes. It is also is also a component of many carbohydrates. If a person eats food including glucose daily basis there is a high chance to having diabetes for which the doctor says not to use too much with sweets. Blood pressure could be another reason to have diabetes.Skin thickness is another reason where diabetes can influence the little veins of the body. Diabetes can cause a skin condition called diabetic dermopathy. The patches are now and again called skin spots. Diabetes damages veins and can make it focus on atherosclerosis. Atherosclerosis can affect body into hypertension, harming vein, heart onfall, and kidney failure. Insulin is another reason where diabetes body without insulin is called an immune system response. The factors are making up the danger of creating diverse kinds of diabetes mellitus. Calculating BMI by age, height and weight. Another attribute diabetes pedigree which means diabetes mellitus is gathering of a metabolic issues where the glucose levels are higher than typical glucose levels type — diabetes is brought about by the undamaged framework shattering the cells in the pancreas.

4.2 Binary Categorical Features

In our dataset, there is only one feature which was categorical and that is ”Outcome”. It is a variable in our dataset which expresses if a person has diabetes or not. This feature contains binary values of either a 0 (no diabetes) or 1 (have diabetes). This feature is very important since it shows the result if a person is diabetic or not. For our easier purpose we have converted “Yes” to “1” and “no” to “0”. So, now our whole dataset is numerical value.

4.3 Target Variables

Target variable, in the AI setting is the variable that is or ought to be the yield. For instance, it could be paired 0 or 1 in the event that you are grouping or it could be a consistent variable in the event that you are completing a relapse. In insights you additionally allude to it as the reaction variable. In this project, we have selected outcome as our target variables since our model depends on the values of this variable.

Indicator factors in the AI setting the info information or the factors that is mapped to the objective variable through an exact connection dispatch generally decided through the information. In measurements you allude to them as indicators. Each arrangement of indicators might be called perception. In this case, our indicator variables are pregnancy, Glucose level, blood pressure (BP), Skin Thickness, Insulin, BMI, Pedigree, Age and Output. Based on the target variable we have implemented a system that will help us in finding the risk score of type II DM on any dataset.

4.4 HeatMap

A heat map is based on data analysis which uses colors by showing a bar graph uses width and height for a data visualization tool. Showing attribute to attribute relationship among the attributes which we have given as inputs. Validity of the process is 0 upto 1 which means partially good enough. If it crosses more than 1 or less than 0 then it should be rechecked the whole data if there is any wrong or not otherwise no relation is depending between them. We have used heatmap on our 15000 diabetes type-2 patient dataset.

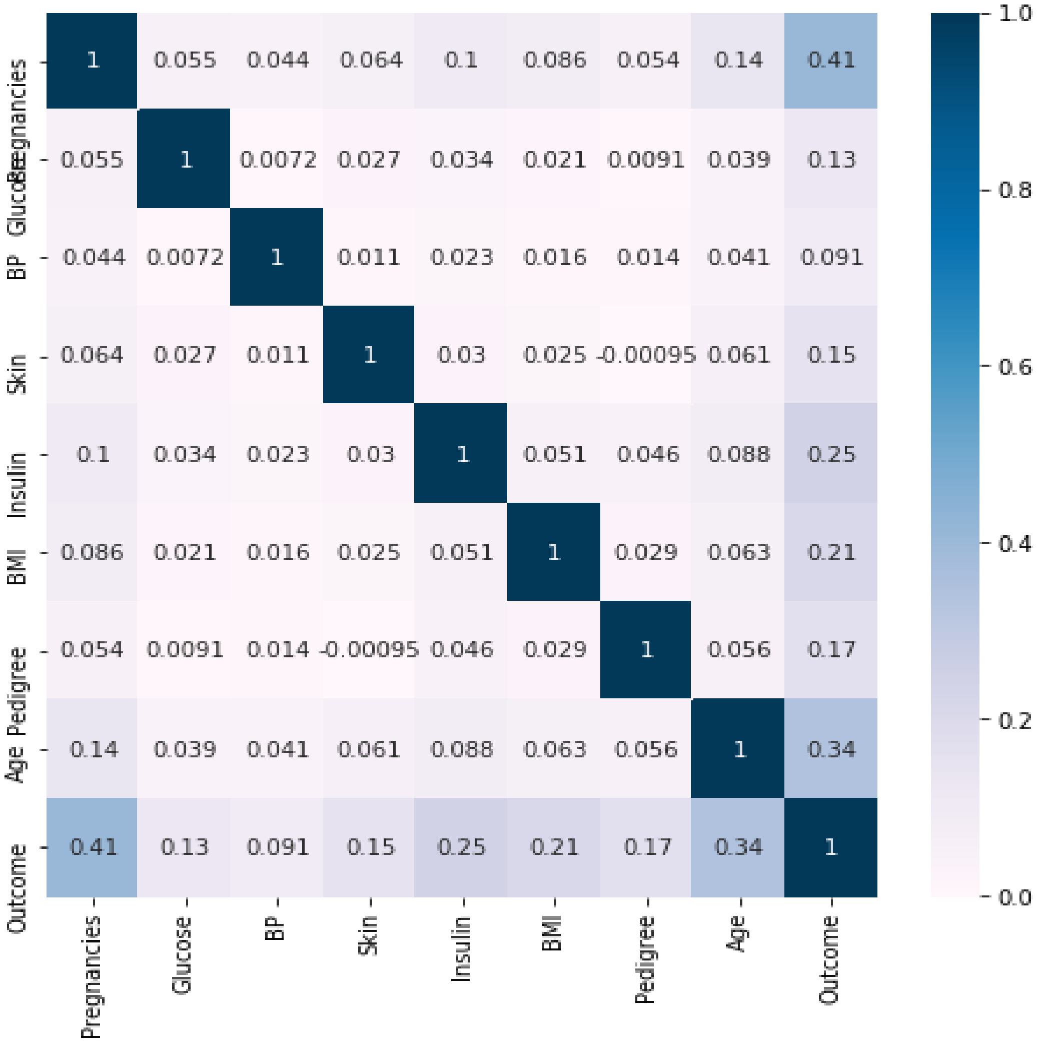


Figure 4.2: Showing Heatmap from the dataset

From Figure 4.2 we can see the correlation among the attributes. Value from 1 to 0 declares best correlation to worst correlation among the attributes which are indicated by blue color and top dark blue color.

4.5 Boxplot

A basic method for speaking to factual information on a plot where the square or rectangular shape is drowned to speak to the second and third quartiles with vertical lines inside to show the middle worth. Showing the shape of the distribution and its variability and also explicatory of data analysis are the main reasons.

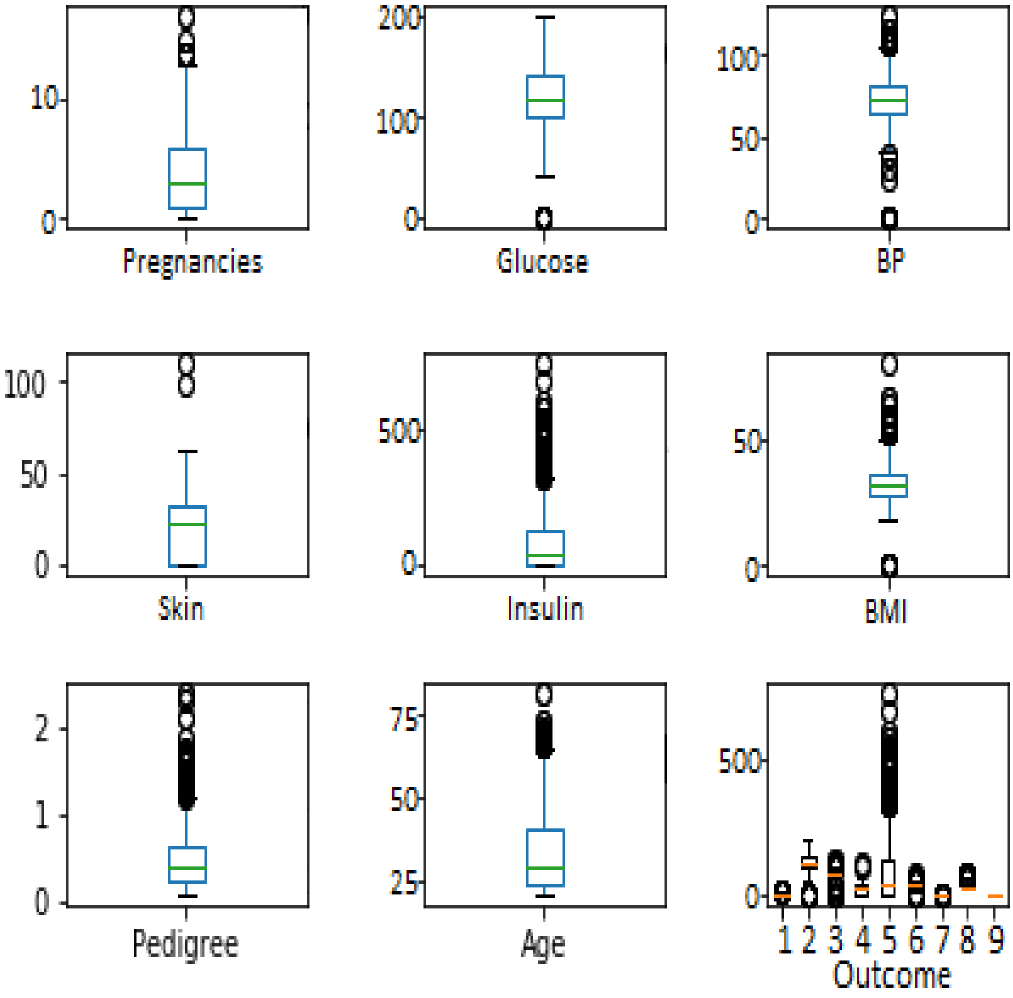


Figure 4.3: Showing Boxplot from the dataset

From Figure 4.3 we can see that Insulin, Pedigree, Age and BMI have upper of the 3rd quartile. BP has got in 1st and 3rd quartile.

4.6 Scatter Matrix

It is a multivariate statistical analysis with the scatter matrix that is used to evaluate the covariance matrix. It is used to detect whether the attributes are correlated and whether the correlation is positive or negative.

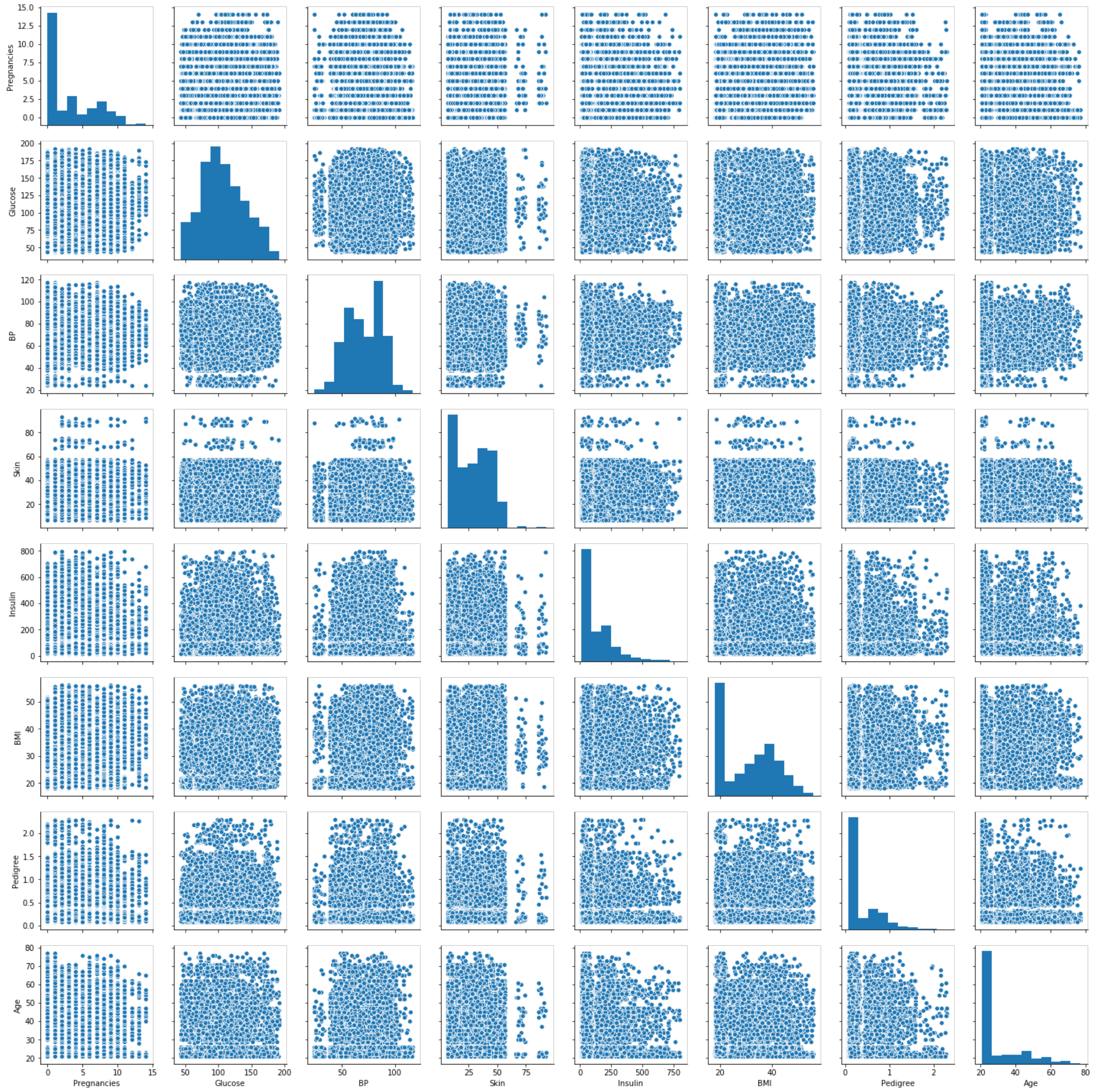


Figure 4.4: Showing Scatter Matrix from the dataset

The overview of how your different variables are correlated with each other. From the Figure 4.4, we can see that there have been shown correlations among the features. For example, between age and pregnancies, there is no correlation as this is not linear. Another one if we have to discuss for Pedigree and BMI there is also no correlation. Similarly, for all attributes in the picture for Pregnancies, Glucose, BP, Skin, Insulin, BMI, Pedigree and Age there is no perfect and strong correlation as there are no linear shape and all are distributed in high (x-axis and y-axis) and low (x-axis and y-axis).

4.7 Histogram

We use the histogram to plot the frequency of score occurrences from our continuous dataset. From the Figure 4.5, we can see the exact same thing in the histogram even more clearly. BMI and Glucose have got the Gaussian determination diagram so that they do not need any more test or train. But as the rest of the attributes do not have gaussian determination so they need proper train and test.

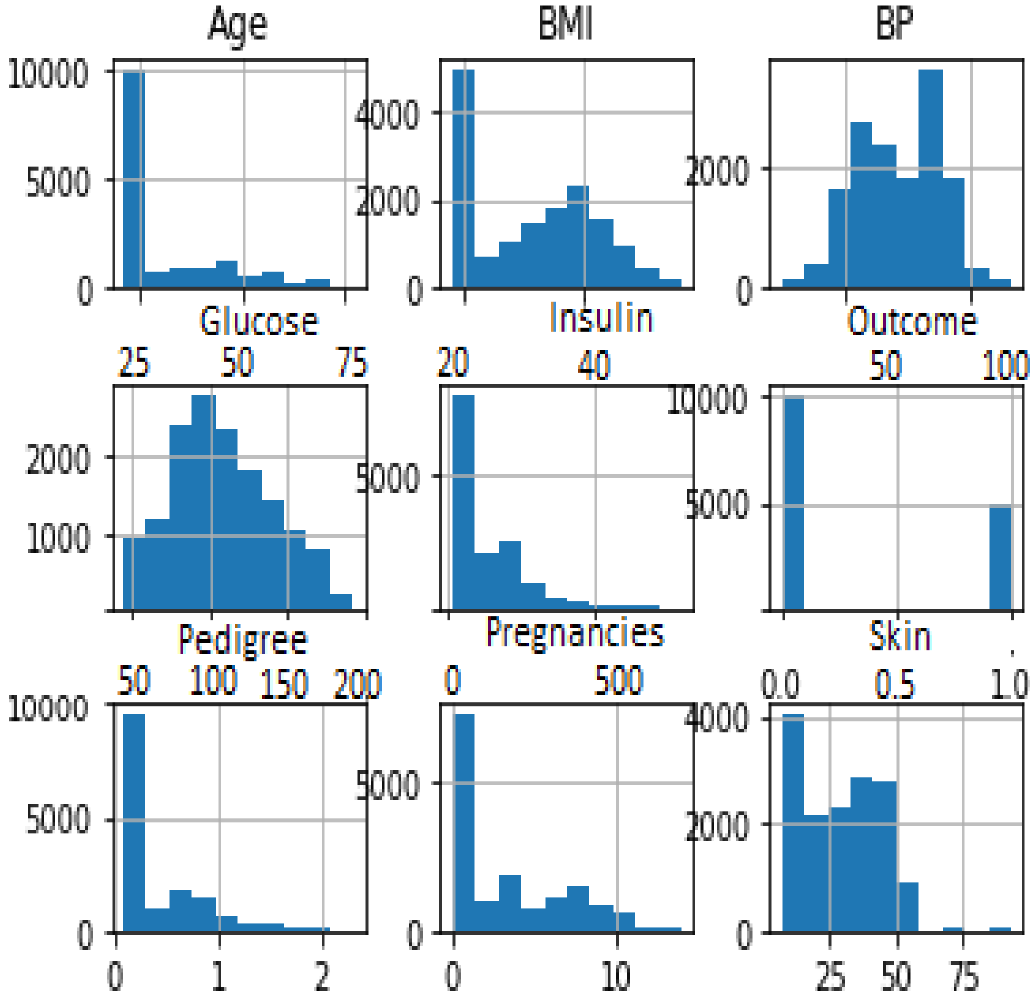


Figure 4.5: Showing histogram using on dataset

4.8 Feature Score

In the feature score in Figure 4.6, we have shown the relations among the attributes. As we can see from the picture that ”Age” and “Pregnancy” are the best attributes on which the output is related. From Figure 4.6, we can see that “Pregnancy” is the most correlated with the output and “Blood Pressure (BP)” is the less one.

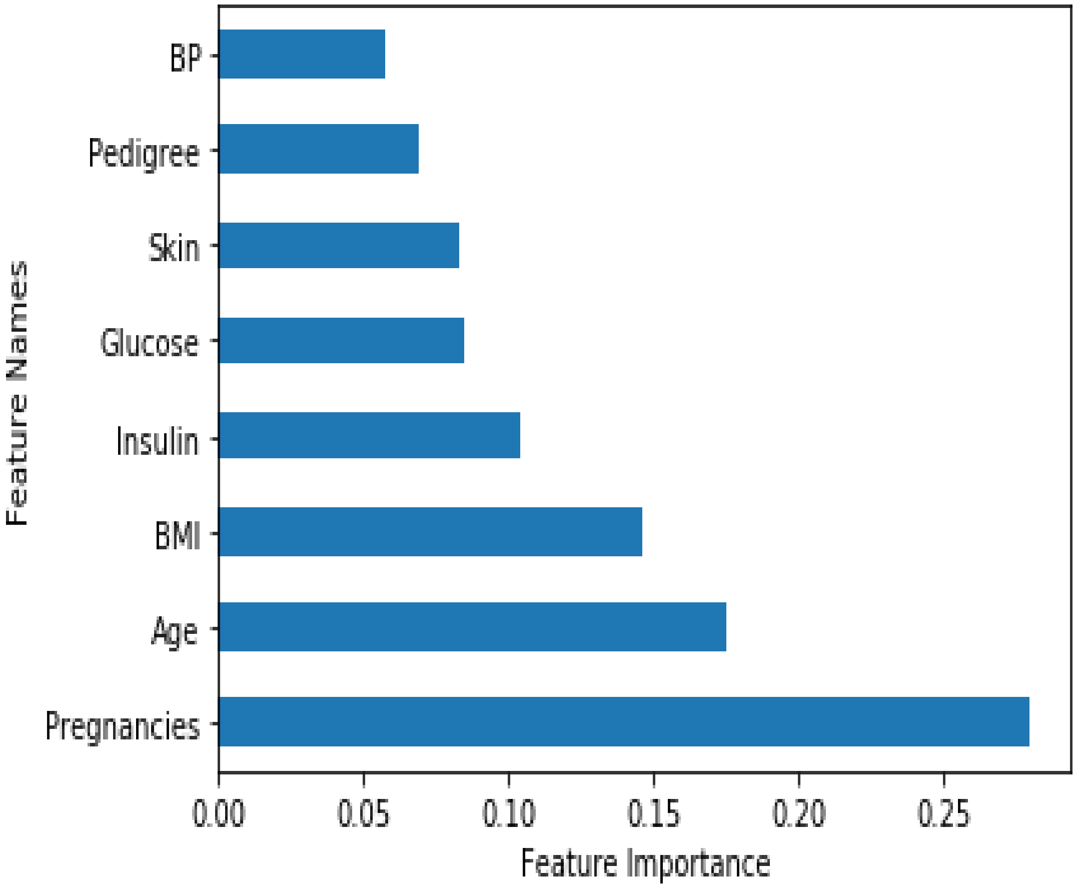


Figure 4.6: Feature Score

Chapter 5

System Implementation

5.1 Missing Values Imputation

First of all, we have figured out the missing values of our dataset and then applied the LDA [13] technique to impute missing values. To ignore the missing or Nun values we have used the LDA process on our dataset. After that, all these missing values are filled with mean column values using the imputer function. We have found that 4377 patients have 0 values which include "Pregnancy" attribute. From the rest of the attributes like- Glucose, BP, Skin, Insulin, BMI, Pedigree, and Age have totally zero values. Our main purpose to use LDA for ignoring unnecessary values from our dataset but without pregnancy attribute there are no null values in other attributes. So, we are directly using the dataset as our data is quite good. We are not rejecting pregnancy (contain with null value) because there is a chance that a woman have not become pregnant yet. But if we look at other attributes they could not be null value as if we do not have diabetes it does not mean that our age, skin thickness, insulin, glucose, BP and pedigree will be null or zero value. From this logical concept we have not use the new data after using LDA process. So, we can use the dataset directly as our thesis purpose to use our five algorithms on it.

Table 5.1: Imputing missing values

Features (Input)	**Count of zero values**
Pregnancies	4377
Glucose	0
BP	0
Skin	0
Insulin	0
BMI	0
Pedigree	0
Age	0

We have got the accuracy of our LDA model is 76.8%.

5.2 Categorical Variable Conversion

In our dataset, one variable is categorical which is outcome. If a patient has type II DM, the value of the outcome is “yes” and a patient not having type II DM contains a value “no”. To counter this problem, we have mapped the values of “outcome” variable with 0 or 1. So, if the patient has diabetes, the value is now ‘1’ and ‘0’ represents that the patient does not have diabetes.

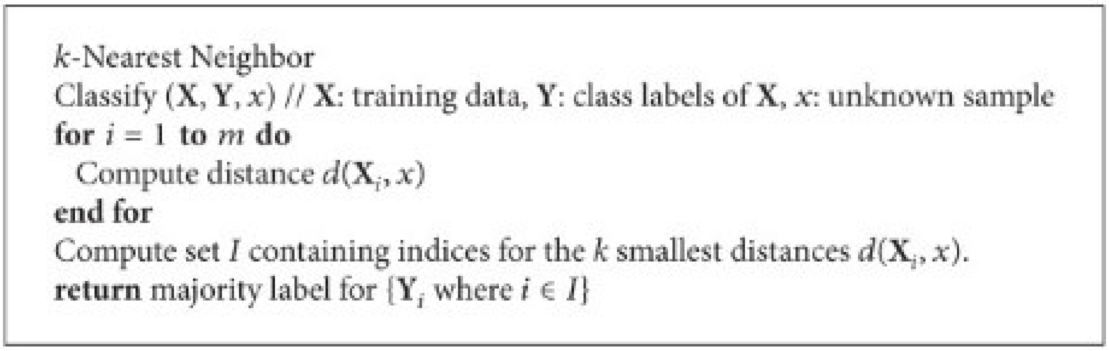
5.3 Train-test Split

In Machine Learning models, a typical issue confronted is with over fitting and under fitting [14]. Over fitting happens when the model gains from the information so well that it prompts no preparation mistake. It learns the commotions and deviations of the information as ideas and fits the model excessively well to the information. In any case, when the model is given new information, it neglects to make precise forecasts since it was over fit with the clamors and deviations. Under fitting is the inverse, it can neither fit the information accessible nor to new information. The train/test split has implemented by using packages like - scikit-learn‘s model. The data was split into a 80:20 ratio with 80% going to use as train and rest of the 20% are used for testing. In most cases this is the ideal ratio of splitting data. A 50:50 ratio was also considered but since the dataset did not contain a large number of instances, taking only 50% of the original data might have led to under fitting of the model.

5.4 Algorithms

5.4.1 KNN (K-Nearest Neighbor)

KNN [15] is a lazy algorithm with non-parametric. The total of neighbours, it takes the most common neighbor among them and assigning it as the classification. The accuracy value depends on neighbor where we have worked for n = 1, 2 and 3 and for n = 3 we have got the most accuracy. [16] Here is the pseudo-code of KNN algorithm in Figure 5.1



```
$k$-Nearest Neighbor
Classify ($\mathbf{X}, \mathbf{Y}, x$) // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, $x$: unknown sample
for $i = 1$ to $m$ do
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set $I$ containing indices for the $k$ smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$
```

Figure 5.1: Pseudo code of KNN

In our dataset, we have used the KNN algorithm to find the accuracy and also have used a parameter of scatter diagram to see [17] the output as a diagram.

By using KNN algorithm we have seen that it has given around 86.8 percent accuracy. For KNN algorithm we have done for 1, 2 and 3 neighbors to find the accuracy-

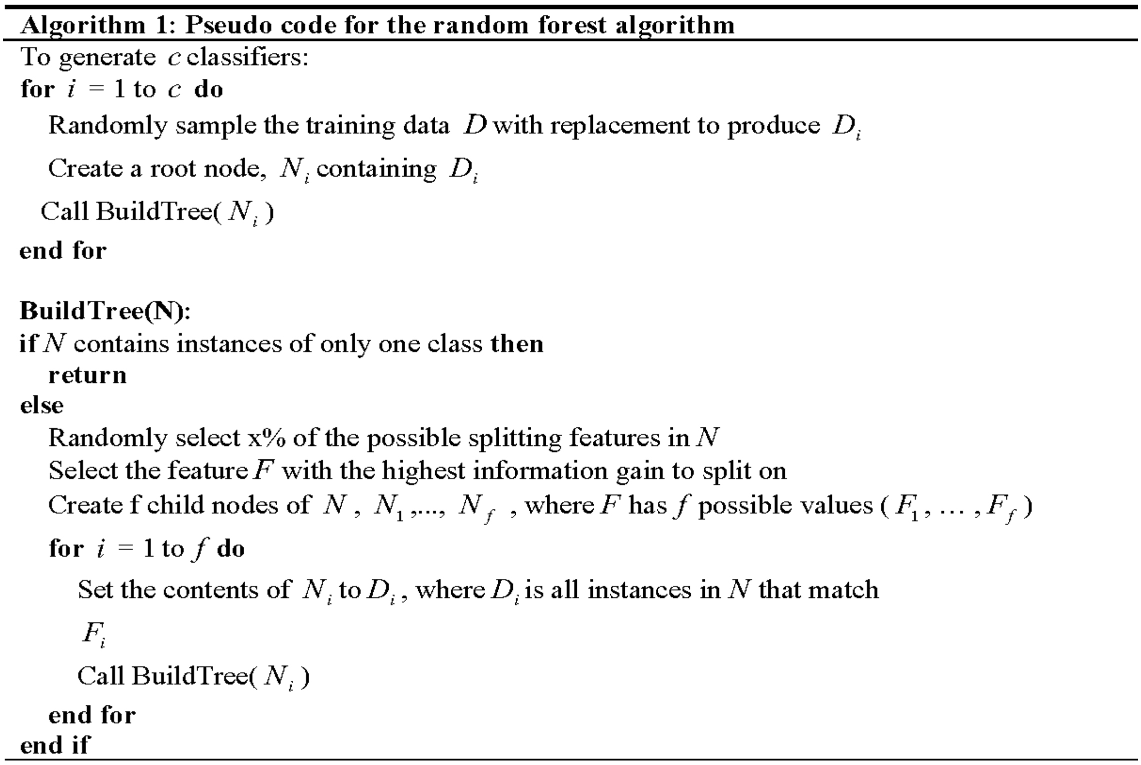
Table 5.2: Difference in accuracy with respect to neighbor values

K- Value	Accuracy (%)
1	84.6
2	83.4
3	86.8

From table 5.2, we can see that we have got the best accuracy 86.8% for K=3 neighbor. So, in this case, we are considering neighbor 3 for KNN.

5.4.2 Random Forest

At training it averages the multiple decision trees. It even gives more accurate results than decision tree since it average multiple decision trees to get the result. [18] In figure 5.2 we have given the pseudo-code of the Random forest-



```
Algorithm 1: Pseudo code for the random forest algorithm
To generate $c$ classifiers:
for $i = 1$ to $c$ do
    Randomly sample the training data $D$ with replacement to produce $D_i$
    Create a root node, $N_i$ containing $D_i$
   Call BuildTree( $N_i$ )
end for

BuildTree(N):
if $N$ contains instances of only one class then
    return
else
    Randomly select x% of the possible splitting features in $N$
    Select the feature $F$ with the highest information gain to split on
    Create f child nodes of $N$ , $N_1$,..., $N_f$ , where $F$ has $f$ possible values ( $F_1, \dots, F_f$ )
    for $i = 1$ to $f$ do
        Set the contents of $N_i$ to $D_i$ , where $D_i$ is all instances in $N$ that match
        $F_i$
        Call BuildTree( $N_i$ )
    end for
end if
```

Figure 5.2: Pseudo code of Random Forest

We have used accuracy parameters as RandomForestClassifier and we have got around 92.6% accuracy which is the maximum accuracy of our model.
We have used the parameter-
Model = RandomForestClassifier(n_estimators = 10, max_depth=10, random_state=5)

5.4.3 Logistic Regression

Using for both classification and regression we have used a supervised algorithm. Moreover, it can be used to predict categorical outcomes. It is used for fitting model to detect outcome variables from independent variables. Third algorithm from machine learning we have used logistic regression [19] for our thesis purpose.

Logistic Regression Pseudo code

1. Calculate using logistic function.
2. Learn the coefficients for a logistic regression model.
3. Finally, make predictions using logistic regression model.

The logistic function is,

$$fx = L/(1 + e^{-}k(x - x0)) \tag{5.1}$$

Where,
e = Euler's number
X0 = Middle x value of sigmoid function
L = The maximum value of curve
k = Abruptness of the curve
This logistic regression equation is used for Input values (x) to estimate an output value (y).

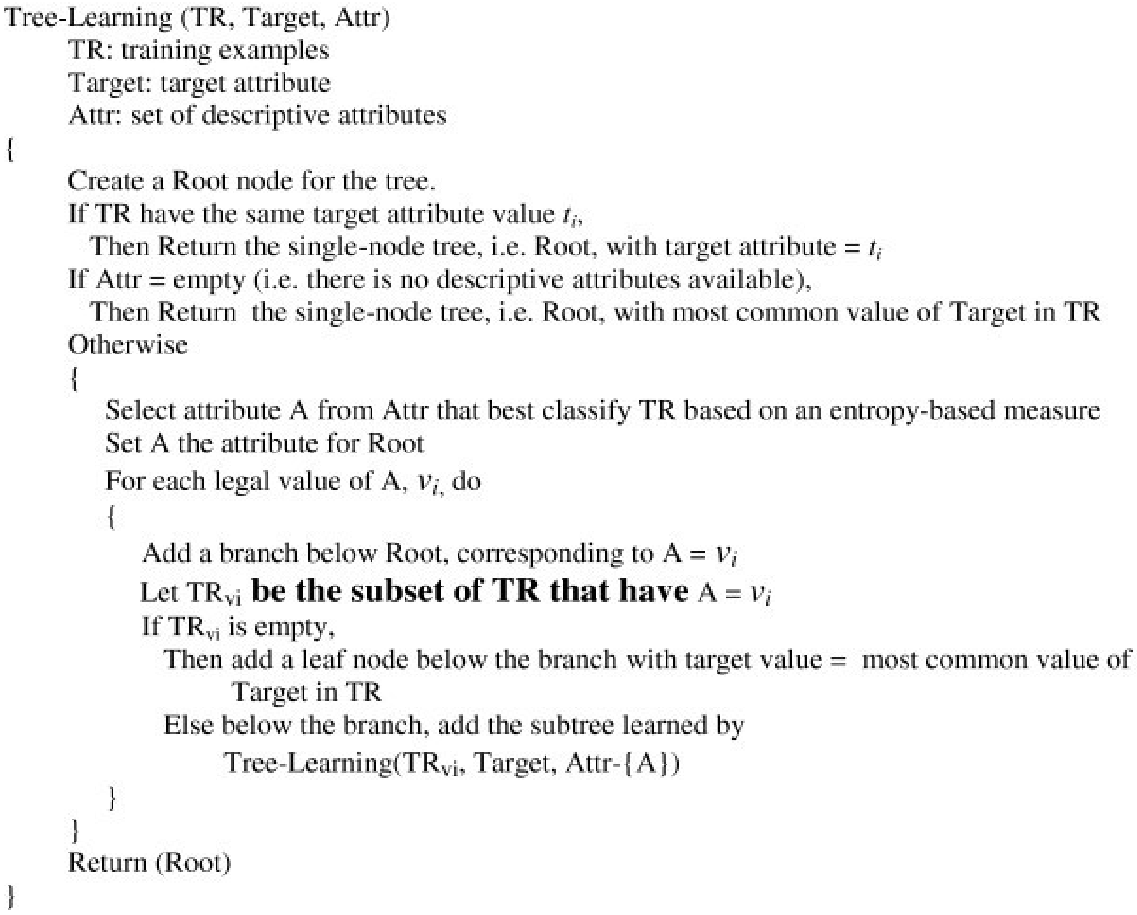
The logistic regression model is given in equation as:

$$y = (e^{b}0 + b1 * x)/(1 + e^{b}0 + b1 * x) \tag{5.2}$$

We have found accuracy of 78.6% for the logistic regression by using parameter x_train,x_
test,y_train,y_test = train_test_split(x, y, test_size=0.3,random_state=100). And also used classifier- classifier = LogisticRegression(random_state = 10).

5.4.4 Decision Tree

Decision Tree is an administered learning calculation that is utilized for order and relapse. It works by part the information into at least two subsets dependent on the estimation of the information factors [20]. A cost capacity or part basis is utilized to decide the best split among all the focusing splits. The information is part recursively into gatherings until the leaves contain. This model is [21] an enhanced adaptation of the CART (Classification and Regression Trees) calculation is utilized to actualize the Decision Tree classifier utilizing Scikit-Learn.Gini contamination is utilized as the part foundation to quantify the vulnerability. Choice Trees are anything but difficult to decipher and comprehend, contrasted with other grouping calculations. Besides, Decision Trees require small preprocessing and doesn't influence the presentation. They are not founded on the Euclidean separation, consequently and not requiring of highlight scaling. It can deal with both categorical and numerical factors as the dataset contains both info so it is suitable for this model. In this model, the connection between the component variable and the target variable is mind-boggling and exceedingly non-straight. So a Decision Tree has a more noteworthy shot of beating straight models like Logistic Regression [22].



```
Tree-Learning (TR, Target, Attr)
      TR: training examples
      Target: target attribute
      Attr: set of descriptive attributes
{
      Create a Root node for the tree.
      If TR have the same target attribute value $t_i$,
        Then Return the single-node tree, i.e. Root, with target attribute = $t_i$
      If Attr = empty (i.e. there is no descriptive attributes available),
        Then Return  the single-node tree, i.e. Root, with most common value of Target in TR
      Otherwise
      {
         Select attribute A from Attr that best classify TR based on an entropy-based measure
         Set A the attribute for Root
         For each legal value of A, $v_i$, do
         {
            Add a branch below Root, corresponding to A = $v_i$
            Let $TR_{vi}$ be the subset of TR that have A = $v_i$
            If $TR_{vi}$ is empty,
              Then add a leaf node below the branch with target value =  most common value of
                    Target in TR
              Else below the branch, add the subtree learned by
                    Tree-Learning($TR_{vi}$, Target, Attr-{A})
         }
      }
      Return (Root)
}
```

Figure 5.3: Pseudocode of Decision tree

We have also used accuracy parameters as RandomForestClassifier and we have got around 88.6% accuracy. We have used classifier as-
treeclf = DecisionTreeClassifier(random_state=42)
tree = DecisionTreeClassifier(max_depth = 6, max_features = 4, min_samples_split = 5, random_state=42)

5.4.5 Convolutional Neural Network (CNN)

CNN is the utilization of counterfeit neural systems utilizing present-day equipment [23]. Convolutional Neural Networks were intended to guide picture information to a yield variable. These methods are so developed models that they are so useful methods for any type of prediction problem. CNN is a deep learning algorithm which contains few layers which takes inputs, perform some mathematical operation non-linear activation function using ReLU and Sigmoid for predicting accuracy from the output. In CNN layer is not connected to all other neurons. In the CNN model in first two or three layers where we have used an activation function Relu might detect the lowest level features. Relu takes input as binary (0 and 1) to train and test. It will ignore those negative values from the input. The next layers might detect middle-level features and at last in the last layer we could use another activation function sigmoid which could find the higher-level features. In deep learning this unique technique makes CNN the most valuable algorithm. The more layers we use the more complex it will get. After giving input in the first layer it tries to squish the data and send them to the next layer. Through the whole process we tried to figure out the comparison between our output and actual value. It helps how we adjust complex operation that we perform in each layer to get actual answer. We also find the cross function which means how much the network's answer missed compared to actual answer. In addition, we have used Ludwig, ADAM and Keras to overcome the complexity of the algorithm. Before doing any kind of deep learning methods we should be known to some of the features which are related to it.
Spatial feature: In spite of image it can also work for non- image data. CNN can be used on any data if the dataset consists of spatial features. Spatial data can be described as those which are directly or indirectly occupied to a specific location. If it is not related to a location then it is a non-spatial data. In our data set all the values are specifically told that which values are consist of which attribute. The spatial feature contains with specific location which could be described of their validity. Before feeding into deep learning algorithm preprocessing should be done which refers to all transformation of raw data. Preprocessing is helpful to get better testing and training results by the whole process.

Adaptive Moment Estimation (ADAM)

Adam is an optimization algorithm [24] used in neural networking. The Adaptive Moment Estimation or Adam enhancement calculation is one of those calculations that function admirably over a wide scope of profound learning designs. There are lots of optimizers like - SAGA, NADAM, SAG etc and ADAM is also one of them. We have used ADAM to compute individual learning rates for different parameters.

Before using CNN we need some libraries such as - Keras and TensorFlow. Next, we have to install an environment of Python Spyder (py35) environment. After going through the whole process and installing all the libraries it gives 90% accuracy score. But before using ADAM we have used a model called Keras. In computer science, Keras is an open-source deep learning library. Keras pursues best practices for decreasing subjective burden. It gives clear and significant input to the client. Doing machine learning is also an extra majority is it ignores unnecessary values all by itself.

Epoch: One Epoch is the point at which an entire dataset is passed forward and in reverse through the neural system just once. If it seems that one epoch is too big for the computer then it has to be divided into several smaller batches. As we need to work on the whole dataset several times in a neural network process but have to bear in mind that there is a limitation for all kinds of the dataset. If the numbers of epochs increases, it could cause a problem of overfitting dataset.

Number of batch: We cannot send the whole dataset in a neural network at once and so that we have to divide it as part which is called the number of batches. It is useful to reduce overfitting.

Dropout layer: As we have used CNN a deep learning method we cannot use the big dataset at once. We have to divide it as part as number of batches. There is another subject to know about dropout layer. Dropout layer helps to ignore all those unnecessary data from the dataset. It is a technique where features are selected randomly during training and ignore them.As we are using CNN the output or result we have got as three sections such as – train, validation and test.

Train: The dataset we have used to fit the whole process. Through this model we can watch and learn from the data from which we have got the accuracy.

Validation: Basically it works for tuning the unbiased dataset to fit the training dataset. It tunes the hyperparameters so that we have also got an accuracy of validation that how much our validation is working.

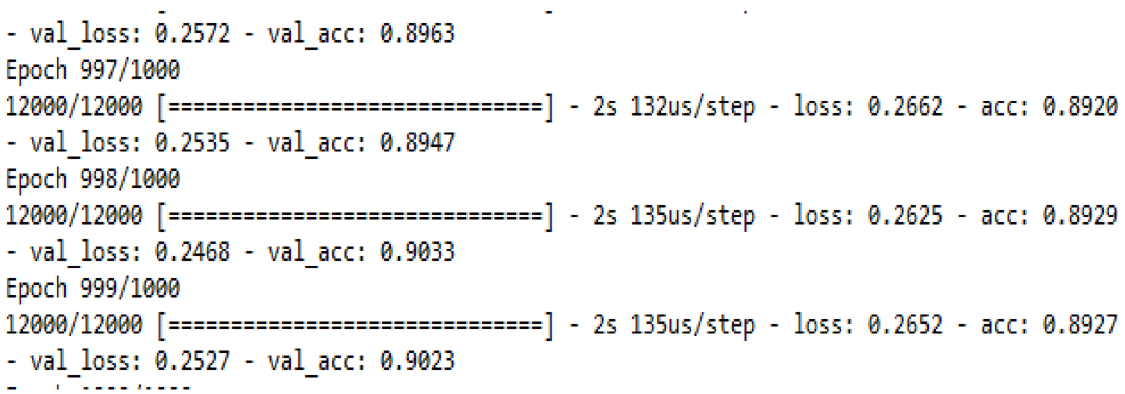
Testing: It works for tuning the unbiased dataset to evaluate the final model fit on the training dataset.

Activation Function: We have used Relu and Sigmoid to compile our model more accurately. We have made a total four layers in which the first three layers contain of Relu (takes 0 and 1) and last layer consists of Sigmoid (takes 0 and 1).

Loss: If the lower value is less then the model is better working. The loss is measured by on validation and training. Loss is not measured as a percentage. It is a summation of the errors made for each example in training or validation sets. The summation of errors which are made in every validation and training set.

We have used epochs randomly in our code to look back on finding missing values. As we have got accuracy for every epoch which depends on validation and loss. If there is no improvement validation then the epoch automatically stopped to work more. When the model works in every epoch at the end if it seems that there is no improvement of loss and validation then epoch stops it's working after showing that the answer is the same in every loop. Even if it goes on the model does not need to look at more epoch.

From Figure 5.4, we can see that in each epoch we can see that our dataset is training in each epoch. It is also showing validation accuracy which could called improvement of the process. If the improvement goes up the model will find more accuracy . In epoch number 998 it showed the highest accuracy of 89.29%. Then it shows no improvement in the model. We have used two activation functions called Relu and Sigmoid. These functions help to compile the dataset more accurately.



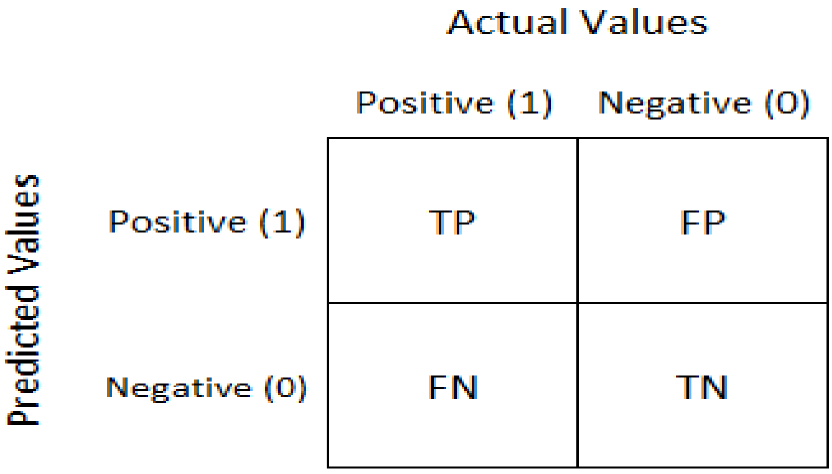
```
- val_loss: 0.2572 - val_acc: 0.8963
Epoch 997/1000
12000/12000 [==============================] - 2s 132us/step - loss: 0.2662 - acc: 0.8920
- val_loss: 0.2535 - val_acc: 0.8947
Epoch 998/1000
12000/12000 [==============================] - 2s 135us/step - loss: 0.2625 - acc: 0.8929
- val_loss: 0.2468 - val_acc: 0.9033
Epoch 999/1000
12000/12000 [==============================] - 2s 135us/step - loss: 0.2652 - acc: 0.8927
- val_loss: 0.2527 - val_acc: 0.9023
```

Figure 5.4: Testing and training in every epoch

Chapter 6

Result and Analysis

In the field of AI and explicitly the issue of the factual arrangement, a perplexity framework, otherwise called a confusion matrix.[19] It is a particular table format that permits the access of calculation the execution commonly an administered learning one. If it is unsupervised learning then it is typically called a coordinating network). In each of the algorithms we have used confusion matrix as a parameter as–
From sklearn.metrics import confusion_matrix cm = confusion_matrix(y_test,y_pred)
So, from the confusion matrix is a parameter of machine learning algorithms by which we can find out accuracy. By this matrix we will find that which algorithm give the best accuracy.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 6.1: Confusion Matrix

So, we have used four machine learning algorithms and a deep learning algorithm where we have used two parameters for now. One is for scatter plot and another one is for accuracy by which we can find out the best algorithm. We have used Confusion Matrix in every algorithm to find accuracy.

Table 6.1: Confusion Matrix of Algorithms

Algorithms	**Confusion Matrix**
K-Nearest Neighbor	$\begin{array}{ccc} & Yes & No \\ Yes & 1851 & 169 \\ No & 226 & 754 \end{array}$
Random Forest Tree	$\begin{array}{ccc} & Yes & No \\ Yes & 1923 & 64 \\ No & 157 & 856 \end{array}$
Logistic Regression	$\begin{array}{ccc} & Yes & No \\ Yes & 1769 & 251 \\ No & 390 & 590 \end{array}$
Decision Tree	$\begin{array}{ccc} & Yes & No \\ Yes & 2331 & 181 \\ No & 192 & 1046 \end{array}$

Table 6.1 represents confusion matrix of each algorithms where we have used our two outcomes having “Yes” and “No”. From the table we can see that KNN has the best accuracy among the machine learning algorithms.

Table 6.2: Accuracy of Algorithms

Algorithms	**Accuracy (%)**
Random Forest	92.60
CNN with ADAM Optimizer	90.00
Decision Tree	88.60
KNN	86.80 (for neighbor=3)
Logistic Regression	78.6

So, from table 6.2 we can see that Random Forest gives 92.60%, Adam 90%, Decision Tree gives 88.6%,KNN gives 86.80% and Logistic Regression gives 78.6% of accuracy. Now we can differentiate easily that Random Forest which gives 92.60% accuracy is the best machine algorithm for our predicted model.

6.1 Reasons Behind Using Algorithms

For Random Forest as we know that it takes the average value of many decision trees. Random forests consist of multiple single trees which are based on the training data of a random sample. Moreover, the Random tree reduces overfitting and gives a more accurate result than the decision tree. From here we can go to the conclusion that Random Forest is a better algorithm than decision tree and Random Forest accuracy is the highest one. Next, we are considering to use Logistic regression rather than using Linear Regression. In linear Regression, it consists of a single straight linear line. So, all values separately distributed in two areas of the straight line. But in the graph of the Logistic algorithm, we can see a curve line for which Logistic could take maximum values to it's position and gets better accuracy than the Linear algorithm. Moreover, as our dataset is supervised so logistic is the perfect one to use. Logistic Regression models is used for binary values as our output shows as binary (have diabetes=1 and no diabetes=0). Linear Regression is used for numerical values. In addition, linear regression depends on dependent and independent variables but for Logistic regression it is not mandatory.

We have used the KNN algorithm which is a non-linear and linear algorithm. Basically, as this is a non-linear algorithm it tries to take overall all types of data to train and test. But for Linear Algorithm such as SVM, Naive Bayes etc. are linearly distributed. In linear as it distributes randomly there is a high chance that it will not be distributed perfectly and there could be an imbalance. KNN can detect both linear or non-linear distributed data. So, it could be a very useful algorithm for any kind of project.

Moreover, we have used CNN algorithm which is a deep learning algorithm consists of layers. Each layer helps to squish the data to get an output. The best part of using CNN is that there are hidden layers where all datas are trained and tested. There are lots of algorithms that sometimes they miss value by default which could be very useful for the result. Deep learning tries to work as human and layers work as neuron. So, in CNN process the model tries to take almost every data from the dataset.

6.2 Performance Metrics

Performance Metrics is a set of metrics used in the designed system to evaluate the model. In a word, we measure the performance of machine learning algorithms using some metrics including confusion matrix,sensitivity,specificity, Precision, accuracy, F1 score, ROC curve, area under the ROC called AUC etc. In our research our destination variable is 'outcome' which has a binary value. If the value of outcome is '1' then it means the patient has diabetes. If the value is '0' then it describes that the patient has no diabetes.

6.2.1 Confusion Matrix

A Confusion Matrix is a popular example of the general performance of class fashions. The matrix suggests us the amount of efficaciously and incorrectly classified examples, compared to the real results (target charge) within the test facts. One main benefit of using confusion matrix as assessment tool is that it permits more precise analysis than percentage of correctly classified accuracy which can provide misleading consequences if the dataset is unbalanced.

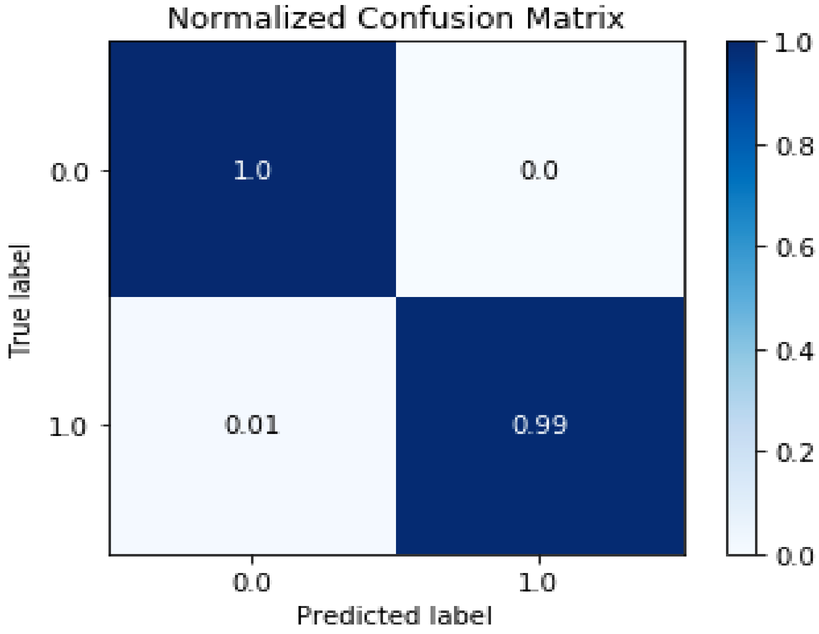


Figure 6.2: Confusion Matrix

This matrix is of dimensions n by n, where n denotes the number of classes. The excellent classifiers, referred to as binary classifiers, has only instructions: “yes”/”no”. [7] The normal performance of a binary classifier is summarized in a confusion matrix that shows cross-tabulated predictions and determined examples into four options: True Positive, True Negative, False-Positive, False-Negative.

True Positive(TP): This includes all the instances which were positive and true at the same time
True Negative(TN): This includes all the instances which were negative and true at the same time.
False Positive(FP): This includes all the instances which were marked positive but they were negative actually.
False Negative(FN): This includes all the instances which were marked negative but they were positive instead.

6.2.2 Sensitivity

It is also known as True Positive Rate (TPR) or recall. Sensitivity is calculated as the number of True Positive Rate (TPR) divided by the summation of True Positive (TP) and false-negative (FN). The best Sensitivity score is ‘1’ and the worst score is ‘0’.

$$SN = TP/(TP + FN) \tag{6.1}$$

6.2.3 Specificity

It is also known as True Negative Rate (TNR). Specificity is calculated as the number of True Negative Rate (TNR) divided by the summation of False Positive (FP) and True Negative (TN).The specificity score is better to have a higher value. The best score for specificity score is ‘1’ and the worst is ‘0’.

$$SP = TN/(TN + FP) \tag{6.2}$$

6.2.4 Precision

Precision denotes the ratio of the true positive perception to the total predicted perception which is the summation of True Positive (TP) and False Positive (FP).The more high precision score leads to less false positive score. The best score for precision is ‘1’ and the worst score is ‘0’.

$$PREC = TP/(TP + FP) \tag{6.3}$$

6.2.5 Negative Predictive Value

Negative Predictive value shows the probability of not having diabetes. This score is better if it is ‘0’ and and the worst is ‘1’. It is calculated by the ratio of the true negative perception of the summation of True Negative (TN) and False Negative (FN).

$$NPV = TN/(TN + FN) \tag{6.4}$$

6.2.6 False Positive Rate

False positive is calculated as the ratio of false-positive rate and summation of false positive rate and true negative rate. The best false positive rate is ’0’ and the most exceedingly awful false positive rate is ’1’.

$$FPR = FP/(FP + TN) \tag{6.5}$$

6.2.7 False Discovery Rate

It is a ratio which shows us the people identified for diabetes do not have diabetes. It is calculated as the number of False Positive Rate (FP) divided by the summation of False Positive (FP) and True Positive (TP). The best false positive rate is ’0’ and the most exceedingly awful false positive rate is ’1’.

$$FDR = FP/(FP + TP) \tag{6.6}$$

6.2.8 False Negative Rate

It is the rate of negative test scores for which an individual is being tested. It is the rate of negative test scores for which an individual is being tested. It is calculated by the ratio of false-negative to the summation of false-negative and true positive. The best false positive rate is '0' and the most exceedingly awful false positive rate is '1'.

$$FNR = FN/(FN + TP) \tag{6.7}$$

6.2.9 Accuracy

Accuracy is calculated by the ratio of all true predictions to the total dataset.The best score for accuracy is '1' and the worst score is '0'

$$ACC = (TP + TN)/(P + N) \tag{6.8}$$

6.2.10 F1 Score

F1 score is a mean score between precision and recall which is used to measure statistical performance rate.

$$F1 = 2TP/(2TP + FP + FN) \tag{6.9}$$

6.2.11 Matthews Correlation Coefficient (MCC)

It is a correlation coefficient that varies from -1 to +1. The best score for MCC is '1' which shows the perfect agreement between the actual value and the predicted value. If the value of MCC is '0', it means the random agreement between predicted and actual value. It is calculated as below:

$$TP * TN - FP * FN/sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)) \tag{6.10}$$

6.2.12 ROC Curve

AUC means the measure of divisible and ROC means a probability curve. AUC-ROC basically helps the measurement of the quality of a model. In Machine Learning, [18] execution estimation is a basic task. So with regards to an order issue, we can depend on an AUC - ROC Curve. When we have to check or envision the exhibition of the multi-class we use AUC and ROC to identify or picture the exhibition of the multi-class order issue. It says that the model is characterized by the attributions.AUC - ROC bend is a presentation estimation for order issues at different limits settings. AUC represents the measurement of separability and ROC is a probability curve. Higher the AUC, the better the model is at anticipating 0s as 0s and 1s as 1s. Depending on the rate the higher AUC, the better the model is.

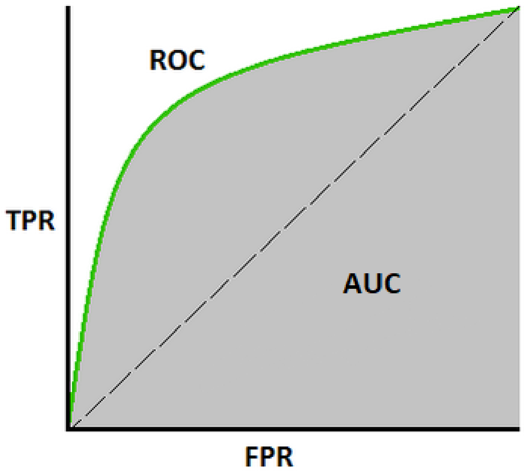


Figure 6.3: ROC-AUC curve [25]

The receiver operating characteristic curve an X-axis (TPR) and Y-axis (FPR) graph which is showing representing a classification model. The curve plots have two parameters such as-

1. True Positive Rate (TPR)
   Recall/ TPR/ Sensitivity= TP/(TP+FN)

2. False Positive Rate (FPR)
   FPR = FP/(FP+TN)

6.2.13 Area Under Curve (AUC)

AUC shows measuring of better performance of the processes. If AUC is near 1 then it has a good prediction of separability. If the model is poor then the AUC value will be near 0. Moreover, if the AUC is 0.5 then it means that the model has no class capacity to be seperated.

6.3 Model performance

In our model, we have used six algorithms- Random Forest, KNN, Decision Tree, Logistic Regression and CNN. For each of these algorithms. We have measured some performance metrics We have used an online calculator for getting the insight about our model using these performance metrics. For this, we have used confusion matrix values such as TP, TN, FP, FN and putting these values into several equations, we have calculated sensitivity, specificity, precision, negative predictive value,false-positive rate, false discovery rate,false-negative rate, accuracy, F1 score, Matthews Correlation Coefficient, ROC curve and AUC score for each of the algorithms of our model to predict type DM risk score. The table below shows the summary of performance metrics we have used for each of the algorithms of our model.

Table 6.3: DM Risc Score using performance Metrics

Algorithm	**Sensitivity**	**Specificity**	**Precision**	**Negative Predictive value**	**False positive rate**
Random Forest	0.9245	0.9304	0.9678	0.8450	0.0696
KNN	0.8912	0.8169	0.9163	0.7694	0.1831
Decision Tree	0.9239	0.8525	0.9279	0.8449	0.1475
Logistic Regression	0.8625	0.5780	0.8451	0.6117	0.4220

Table 6.4: DM Risc Score using performance Metrics

Algorithm	**False Discovery rate**	**False negative rate**	**Accuracy**	**F1 score**	**Matthews correlation coefficient**
Random Forest	0.0322	0.0755	0.9263	0.9457	0.8336
KNN	0.0837	0.1088	0.8683	0.9036	0.6968
Decision Tree	0.0721	0.0761	0.9005	0.9259	0.7746
Logistic Regression	0.1549	0.1375	0.7850	0.8537	0.4486

Down below [26] we have used ROC-AUC curve in four algorithms - KNN, Random Forest, Logistic Regression and Decision Tree

6.3.1 Random Forest

Table 6.4 below illustrates the ROC of the Random Forest algorithm. The result shows how Random Forest performs well for this problem with an accuracy of 0.9263 and AUC score of around 0.906. There is also a rise in F1 and recall score which is 0.9457 and 0.9245 respectively. For predicting any disease recall or sensitivity score has more impact than precision. Random forest is the best machine learning algorithm for our model among the four.

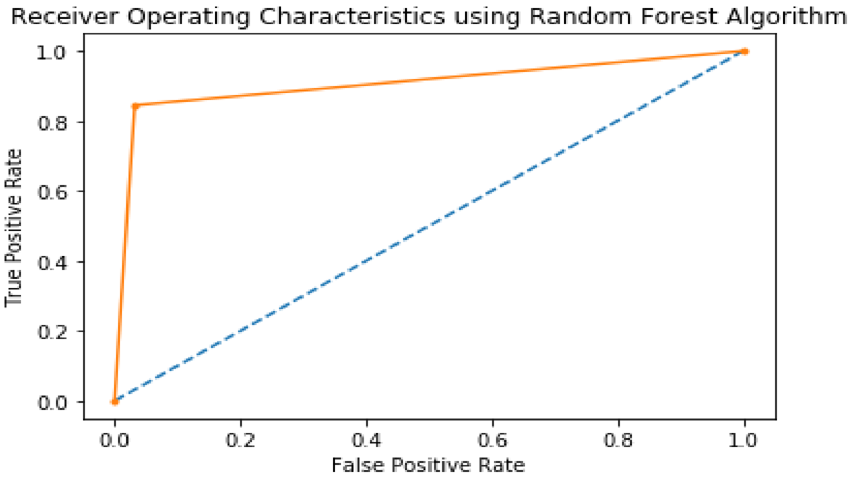


Figure 6.4: ROC of Random Forest

6.3.2 KNN

Figure 6.5 illustrates the ROC of the KNN algorithm. We can see that KNN has good accuracy of around 0.8683 but the AUC score is poor. The precision score is 0.9163 but there are higher values using Random Forest and Decision Tree. However, the recall score is poor than Random forest and Decision tree. The sensitivity score of KNN algorithm is 0.8912 which is quite good but we have better scores using the other two algorithms.

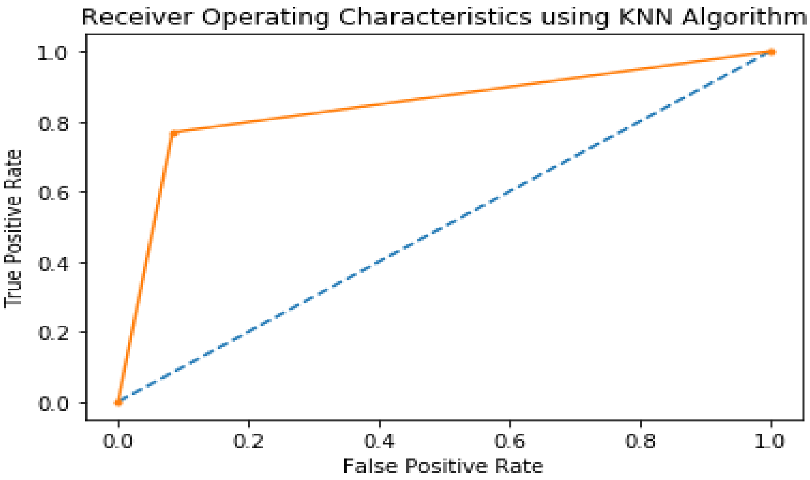


Figure 6.5: ROC of KNN

6.3.3 Decision Tree

Figure 6.6,below illustrates the ROC of the Decision Tree algorithm. The result shows than Decision Tree performs well for this problem with an accuracy better than KNN which is 0.90. There is also a rise in F1 and recall score which is 0.9259 and 0.9239 respectively. The AUC score of the Decision Tree algorithm is 0.886 which is better than KNN but poor than Random Forest. It will be a wise idea to choose the Decision Tree algorithm for this model over KNN but the Random forest will be the best choice.

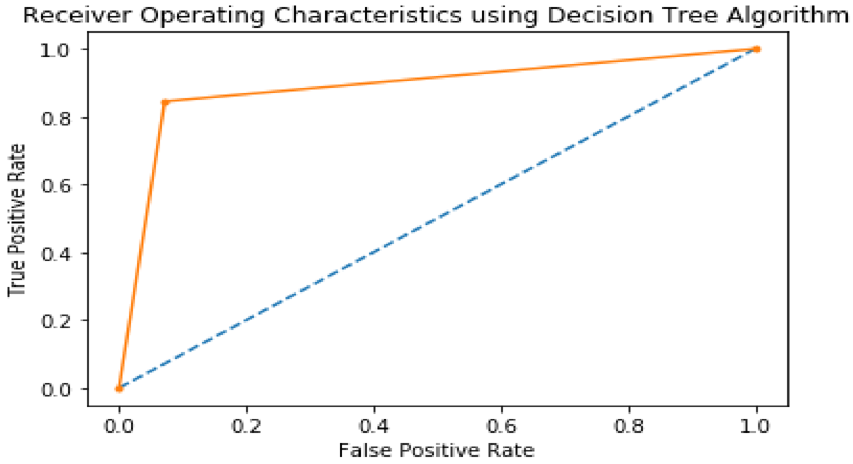


Figure 6.6: ROC of Decision tree

6.3.4 Logistic Regression

Figure 6.7 illustrates the ROC of the Logistic Regression algorithm. We can see that the accuracy of this algorithm is around 0.7850 which is very poor compared to other algorithms. Therefore, the AUC score is also poor which 0.739 is. The precision score is 0.8451 which is also poor .Moreover, the recall score is poorer than all three algorithms. So, it will be a wise decision not to choose Logistic Regression algorithm for our predicted model.

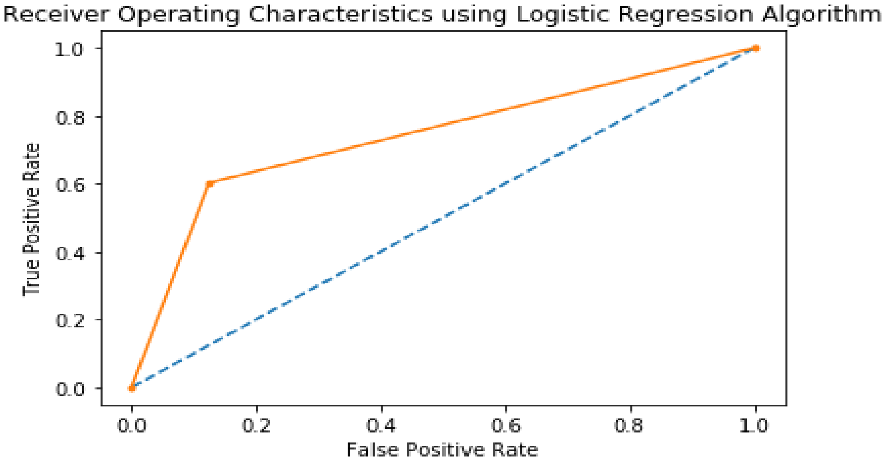


Figure 6.7: ROC of Logistic Regression

Table below shows the AUC score of the algorithms we have used for predicting type2 DM risk score.

Table 6.5: AUC score of machine learning algorithms

Algorithm	**AUC**
Random Forest	0.906
Decision Tree	0.886
KNN (for n=3)	0.843
Logistic regression	0.739

From the table 6.5 we can see that AUC score of Random Forest is 0.906, Decision Tree is 0.886, KNN is 0.843, and Logistic regression is 0.739 which are near 1 value. So, we can say that our models have given good performance.

6.4 Comparative Analysis

To predict type II DM risk score, we have analyzed performance metrics using confusion matrix. Among all the algorithm we have used, Random Forest gives accuracy of 0.9263 and specificity of 0.9304.Therefore, sensitivity score is 0.9245 which is quite good. We have also calculatedSensitivity, Specificity, Precision, NPV, FPR, FDR, FNR, F1 score and Matthews Correlation Coefficient(MCC) for measuring performance of Random Forest. All these values we have got for Random Forest are quite good. After this, we have also implemented ROC curve and calculated AUC score for random forest which is 0.906.

On the other hand, Decision Tree gives accuracy score of 0.9 and specificity of 0.8525 which is less than Random forest. All the scores of performance metrics are still good but they Random Forest gives better scores for all of them. The AUC score of Decision Tree is 0.886 which is less than Random Forest as well. Therefore, we have also used KNN which gives an accuracy score of 0.8683 (for n=3) and specificity score of 0.8169 which is less than Random Forest and Decision Tree. All other scores of performance metrics is less than Random Forest and Decision Tree. The AUC score for KNN is 0.843 which is also less than Random Forest and Decision Tree.

We have used another machine learning algorithm which is Logistic Regression. It gives an accuracy of 0.7850.The specificity score for this algorithm is 0.5780 which is not quite good. Besides, other scores of performance metrics is also not good. AUC score of Logistic Regression is 0.739 which is less than all three algorithms we have used.

Finally, after analyzing all these scores ,there is a clear indication that Random Forest algorithm gives the best accuracy which is 92.63%.It gives the best score forall of the data statistical analysis we have found in our project. The ROC curve and AUC score is also higher than all other algorithms which is 0.906.

Moreover, we have used CNN which is a deep learning algorithm. Using the algorithm we have trained our whole dataset 150000 patient information. Through the training we have used epoch = 1000. In every epoch the algorithm has trained the dataset. After training we have got loss and validation. Finally, using ADAM optimizer, we have received an accuracy of about 89.29%.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this project, we have used five algorithms – KNN, Random Forest Tree, Logistic Regression, Decision Tree and CNN on the diabetes dataset from kaggle. We have figured out the missing values from the data and replaced them with mean values of the column using lda algorithm. Then we have performed statistical analysis on the dataset using heatmap, boxplot, scatter matrix, histogram, and feature score. After that, we have applied five algorithms to find out the best accuracy. We have also applied several metrics and data statistical analysis to see the performances of our selected models. After analyzing all these results we have come to a conclusion that among all these algorithms we have used, Random Forest provides the best accuracy for our predicted model. Random forest is the best algorithm for regression or classification based problem and it handles binary or categorical features easily. Therefore, in random forest, prediction speed is also faster than training speed, so it will require less time as well. The sensitivity score of Random Forest is also good. Our proposed system can be used in hospitals. Doctors can use this model to predict risk scores of patient database and depending on the predictive result they can take steps accordingly. The result may vary depending on the dataset but this analysis will help to choose algorithms wisely that can predict the risk score of type II DM.

7.2 Future Works

Our main goal was to identify the best algorithm that shows us the risk score of diabetes based. We have applied for a dataset that is based on Bangladeshi diabetic patients and we are glad to have permission to collect data. Meanwhile, we made this model so that whenever we get our real data, we can apply that data set into this model. The dataset we will be collect in the future will have 24 different attributes of diabetes patients with more than five thousand instances. Our plan is to apply feature engineering on that data set to see which attributes have much impact on diabetes and identify that so that we can work on that particular part to be careful with diabetes patients. In the future, we will also try to put MRMR [27] feature selection to understand deeply about the dataset and analyze it. Also, we are planning to work on the possible diseases that a diabetes patient can have such as eyesight problems, cardiovascular diseases, kidney problems, etc. Moreover, the number of diabetes patients is alarmingly increasing in the world and the number is still on the rise. The doctor who referred us is also interested to work with us and have asked to do the project through raw coding rather than using build-in tools.

Bibliography

[1] J. F. Ndisang, A. Vannacci, and S. Rastogi, “Insulin resistance, type 1 and type 2 diabetes, and related complications”, 2017.

[2] J. Bronlee, “How to handle missing data with python”, Mar. 2017.

[3] D. E. Sharland, “Davidson’s principles and practice of medicine”, vol. 58(677), Mar. 1982.

[4] A. Pant, “Introduction to logistic regression”, Jun. 2019, [online] https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148. [Online]. Available: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

[5] P. Chaovalit and L. Zhou, “Movie review mining: A comparison between supervised and unsupervised classification approaches”, p. 112c, Jan. 2015.

[6] F. Fernandez, “Diabetes from dat263x lab01”, Aug. 2018, [online] https://www.kaggle.com/fmendes/diabetes-from-dat263x-lab01. [Online]. Available: https://www.kaggle.com/fmendes/diabetes-from-dat263x-lab01.

[7] P. Sujarani, “Prediction of diabetes using artificial neural networks: A review”, *Journal of Advanced Research in Dynamical Control Systems*, vol. 10, 2018. [Online]. Available: http://jardcs.org/papers/v10/20181451.pdf.

[8] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, “Predicting diabetes mellitus with machine learning techniques”, *Frontiers in Genetics*, vol. 9, Nov. 2018. DOI: 10.3389/fgene.2018.00515.

[9] A. G. Karegowda, A. Manjunath, and M. Jayaram, “Application of genetic algorithm optimized neural network connection weights for medical diagnosis of pima indians diabetes”, *International Journal on Soft Computing*, vol. 2, no. 2, pp. 15–23, 2011.

[10] T. Rahman, S. M. Farzana, A. Z. Khanom, *et al.*, “Prediction of diabetes induced complications using different machine learning algorithms”, PhD thesis, BRAC University, 2018.

[11] B. Alić, L. Gurbeta, and A. Badnjević, “Machine learning techniques for classification of diabetes and cardiovascular diseases”, in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, 2017, pp. 1–4.

[12] V. Lugat, *Pima indians diabetes - eda prediction (0.906)*, Jul. 2019. [Online]. Available: https://www.kaggle.com/vincentlugat/pima-indians-diabetes-eda-prediction-0-906.

[13] R. Senapti, K. Shaw, S. Mishra, and D. Mishra, “A novel approach for missing value imputation and classification of microarray dataset”, *Procedia engineering*, vol. 38, pp. 1067–1071, 2012.

[14] J. Brownlee, “Overfitting and underfitting with machine learning algorithms”, *Machine Learning Mastery*, Apr. 2019, [online] https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms. [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms.

[15] T. Tulgar, A. Haydar, and İ. Erşan, “A distributed k nearest neighbor classifier for big data”, *Balkan Journal of Electrical and Computer Engineering*, pp. 105–111, 2018.

[16] S. Shalev-Shwartz and S. Ben-David, “Introduction”, *Understanding Machine Learning*, pp. 1–10, DOI: 10.1017/cbo9781107298019.002.

[17] T. Srivastava, “Introduction to k-nearest neighbors: A powerful machine learning algorithm (with implementation in python r)”, Mar. 2018, [online] https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.

[18] N. Sirikulviriya and S. Sinthupinyo, “Integration of rules from a random forest”, 2018.

[19] G. Biau, “Analysis of a random forest model”, *Journal of Machine Learning Research*, pp. 1063–1095, 2012.

[20] P. Gupta, “Decision trees in machine learning”, *Medium*, Jul. 2019, [online] https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

[21] J. Brownlee, “Classification and regression trees for machine learning”, *Machine Learning Mastery*, Jul. 2019, [online] https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/. [Online]. Available: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/.

[22] Y. J. Hu, T. H. Ku, R. H. Jan, K. Wang, Y. C. Tseng, and S. F. Yang, “Decision tree-based learning to predict patient controlled analgesia consumption and readjustment”, *BMC medical informatics and decision making*, vol. 12(1), p. 131, 2012.

[23] J. Brownlee, “When to use mlp, cnn, and rnn neural networks”, Jul. 2018, [online] https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks. [Online]. Available: https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks.

[24] Kingma, P. Diederik, and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

[25] S. Narkhede, “Understanding auc - roc curve”, Jan. 2018, [online] https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5. [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[26] J. Brownlee, “How and when to use roc curves and precision-recall curves for classification in python”, *Machine Learning Mastery*, Aug. 2019. [Online]. Available: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.

[27] H. Peng, F. Long, and C. Ding, *Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy*, 2005.