DISSERTATION

NAME: SHUVECHCHHA SANYAL

DEPARTMENT: STATISTICS

ROLL NUMBER: 447

SUPERVISOR: DR. DURBA BHATTACHARYA

TITLE: PREDICTION OF DIABETES USING CLASSIFICATION ALGORITHMS

"I hereby declare that the dissertation work entitled "Prediction of diabetes using classification algorithms" is an original work done by me under the guidance of Dr Durba Bhattacharya. I have identified all my sources and no part of my dissertation work remains unacknowledged."

Signed-

Shuvechchha Sanyal

CONTENTS

ABSTRACT

Diabetes is one of the fastest-growing challenges of the 21st century. Therefore, it has become extremely crucial to predict whether a person is likely to have diabetes from the medical history of the person. If a correct prediction can be provided, then a person with borderline diabetes can be prevented from having diabetes beforehand by taking the appropriate measures. The main objective of this project is to make a classification model that predicts whether a person is likely to have diabetes or not from the medical history of the patients provided in the related dataset. The data has been modeled on various classification algorithms and a comparison of the accuracy of the models used has been done. Lastly, the most superior model for modeling the data has been selected.

INTRODUCTION

Diabetes is a chronic disease that occurs either when the pancreas doesn’t produce enough insulin or when the body cannot effectively use the insulin it produces. Hyperglycemia, or raised blood sugar, is a common effect of diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels. This makes it essential to classify whether a person has diabetes or not based on the attributes of the medical history provided.

Statistical classification is the problem of identifying to which category or class a new data will fall into based on the input values of the data on which the model has been trained. Classification is considered as an instance of supervised learning that is, learning where a training set of correctly identified observations is available. The individual observations are analyzed into a group of “explanatory variables” or “features”. The explanatory variables may be categorical, ordinal, integer-valued, or real-valued.

In statistics, there are various kinds of classification algorithms. Some of the different types of classification algorithms are *Logistic Regression, Decision Trees, Random Forest, Naïve Bayes classification algorithm, k-nearest neighbors, and many others.* Some of these classification algorithms are purely statistical in nature while some others are Machine learning (ML) based models. A comparison of the statistical and machine learning models will be drawn using the appropriate accuracy metric.

TERMINOLOGY

- Machine Learning (ML): Machine learning is an algorithm focused on building applications that learn from data and improve their accuracy over time without being programmed to do so.
- Supervised learning: Supervised learning is the process where one has both features and target variable and an algorithm is used to approximately learn the mapping function from the feature to the target variable to predict the target variables for the test set.
- Features (or predictor variables): The individual independent variables that usually act as the input in the model.
- Target variable: It is the variable whose values are to be modeled and predicted using the features. It is analogous to the dependent variable.
- Categorical variable: Categorical variables are variables that have a discrete set of possible outcomes. They can be both ordinal (variable where a pre-existing order exists) and nominal (variable where no pre-existing order exists).
- Classification model: The process of predicting a categorical output. Classification can be of two types:

  Binary classification: The process of predicting a categorical output when the output variable has only two possible outcomes.

  Multiclass classification: The process of predicting a categorical output when the output variable has multiple possible outcomes.

- Outlier: Outliers are observations that deviate significantly from the rest of the observations in the dataset.

- Normalization: Normalization is a technique of changing the numeric columns in the dataset to use a common scale, without distortion in the differences in the ranges of values or losing information.
- Training set: A training set is a subset of the original dataset used to train a model.
- Test set: A test set is a subset of the original dataset used to test the trained model.
- Classification threshold: The lowest probability value at which a positive classification can be asserted. The threshold value is generally different for different models. It is chosen in accordance with the requirements of the model.
- Confusion Matrix: It is a table that describes the performance of a classification model by grouping the predictions into four different categories.
  - True positive (TP): An outcome where the model correctly predicts the outcome to belong to the positive class.
  - True negative (TN): An outcome where the model correctly predicts the outcome to belong to the negative class.
  - False-positive (FP): An outcome where the model incorrectly predicts the outcome to belong to the positive class.
  - False-negative (FN): An outcome where the model incorrectly predicts the outcome to belong to the negative class.
- Accuracy: Accuracy is the percentage of correct predictions made by the model. For binary classification model, accuracy can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

- Sensitivity (True Positive Rate): Sensitivity is defined as the percentage of positives that are correctly identified. For the binary classification model, sensitivity can be calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}$$

Sensitivity is also known as recall. For most models in the medical field, a high recall value is needed.

- Specificity (True Negative Rate): Specificity is defined as the percentage of negatives that are correctly identified. For the binary classification model, specificity can be calculated as follows:

$$Specificity = \frac{TN}{TN + FP}$$

- False Positive Rate (FPR): False Positive Rate is defined as the percentage of a positive result being predicted when the true value is negative. For binary classification model, FPR can be calculated as follows:

$$FPR = 1 - Specificity = \frac{FP}{TN + FP}$$

The False Positive Rate also forms the x-axis of a ROC curve.

- ROC curve (Receiver Operating Characteristic curve): An ROC curve is a plot of the True Positive Rate against the False Positive Rate of the model at all classification thresholds. The ROC curve is used to evaluate the performance of a model at different classification thresholds.

- AUC (Area under the ROC curve): AUC provides a measure of performance for a classification model. It is the area under a ROC curve. If a classifier is 100% then its AUC score is 1.

METHODOLOGY

LOGISTIC MODEL:

In a binary logistic regression model, we want to model the probability P(Y=1| X=x) as a function of x where x is an explanatory (or independent) variable. The problem that we encounter is that how do we model the relationship between P(Y=1|X=x) and x? If we use a linear regression function to model the relationship as:

$$\pi(x)=\alpha+\beta x \qquad \ldots \text{(i)}$$

[We assume Y|X=x ~Bernoulli (π) where π=Probability of success given X=P(Y=1|X=x).

We denote P(Y=1|X=x) as π(x), as the value of Y depends on the given value of the variable X.]

Then if the values of x are very close to zero we might predict the value of π(x) as negative and if the values of x are very large then we might end up predicting the values of π(x) as higher than 1. These values of π(x) are not at all sensible as the values of P(Y=1|X=x) must fall between 0 and 1 irrespective of the values of x. This problem occurs whenever a straight line is made to fit a binary response and can predict π(x) <0 for some values of x and π(x)>1 for others. The problem to this solution can be provided by a function that always gives an output between 0 and 1 for all values of X.

The function we can use is as follows:

$$\pi(x)=\frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \qquad \ldots.. \text{(ii)}$$

[We assume Y|X=x ~Bernoulli (π) where π=Probability of success given X=P(Y=1|X=x).

We denote P(Y=1|X=x) as π(x), as the value of Y depends on the given value of the variable X.]

It is to be noticed that for the function in (ii), low values of X yield probability close to zero but never less than zero and high values of X yield probability close to one but never greater than one. Therefore, π(x) will always remain in the range (0, 1).

The above function in (ii) yields an S-shaped curve and therefore a sensible output is given for any value of X. We also notice that the logistic model can capture the range of values better than the linear regression model.

After manipulating the function in (ii), we obtain that,

$$\frac{\pi(x)}{1-\pi(x)} = e^{\alpha+\beta x} \qquad \text{……..(iii)}$$

The value $\frac{\pi(x)}{1-\pi(x)}$ is known as the 'Odds' and the value of 'Odds' can range anywhere between 0 and ∞. The value of 'Odds' close to 0 and ∞ indicate very low and very high probabilities.

By taking logarithm on both sides of the equation (iii), we obtain,

$$\log[\frac{\pi(x)}{1-\pi(x)}] = \alpha + \beta x \qquad \text{…….. (iv)}$$

The left-hand side of the equation (iv) is known as the log-odds or the logit. It is also to be noted that the logistic function in (iii) has a logit that is linear in X.

The quantity logit[π(x)] can also be interpreted as the logarithm of odds of success as,

$$\text{logit}[\pi(\text{x})]=\log[\frac{P(Y=1|\,X=x)}{P(Y=0\,|\,X=x)}]$$

as, P(Y=0| X=x) =1-P(Y=1|X=x) =1- π(x).

ESTIMATING THE REGRESSION COEFFICIENTS IN LOGISTIC REGRESSION MODEL:

The coefficients in equation (ii) are unknown and hence should be estimated. The method that we use to find the unknown regression coefficients is the Maximum Likelihood Method. We could also use non-linear least squares to fit the model but we use the Maximum Likelihood Method because of its better statistical properties.

We want to estimate the regression coefficients such that plugging the estimated values of α and β in the equation (ii), gives us a predicted probability for each individual which corresponds closely to the observed status of the individual.

The estimates α and β are estimated such that the maximum likelihood function is maximized.

For estimating the logistic regression coefficients using the Maximum Likelihood Method for any probability density function that belongs to the OPEF (One-parameter exponential family) we follow the method as follows.

The likelihood function of a PDF that belongs to the OPEF is as follows:

$$l(\theta, y) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)$$

$$\frac{\partial l}{\partial \beta} = \frac{\partial l}{\partial \theta} . \frac{\partial \theta}{\partial \mu} . \frac{\partial \mu}{\partial \eta} . \frac{\partial \eta}{\partial \beta} .....(1)$$

$$\frac{\partial l}{\partial \theta} = \frac{y - b^{'}(\theta)}{a(\phi)} ....(2)$$

$$\frac{\partial\theta}{\partial\mu}=\frac{1}{\frac{\partial\mu}{\partial\theta}}=\frac{1}{b''(\theta)}[\because \mu=b'(\theta)]......(3)$$

$$\frac{\partial\mu}{\partial\eta}=\frac{1}{\frac{\partial\eta}{\partial\mu}}=\frac{1}{g'(\mu)}[\because \eta=g(\mu)].....(4)$$

$$\frac{\partial\eta}{\partial\beta}=x[\because \eta=\alpha+\beta x].......(5)$$

Combining (1), (2),(3),(4), and (5)

$$\frac{\partial l}{\partial\beta}=\frac{y-b'(\theta)}{a(\phi)}.\frac{1}{b''(\theta)}.\frac{1}{g'(\mu)}.x$$

$$=\frac{(y-\mu)x}{a(\phi)}.\frac{a(\phi)}{v(y)}.\frac{1}{g'(\mu)}$$

$$=\frac{(y-\mu)x}{v(y)}.\frac{1}{g'(\mu)}$$

Score equations will be as follows:

$$S_1=\frac{\partial l}{\partial\alpha}=\sum_{i=1}^{n}\frac{(y_i-\mu_i)}{v(y_i)}.\frac{1}{g'(\mu_i)}$$

$$S_2=\frac{\partial l}{\partial\beta}=\sum_{i=1}^{n}\frac{(y_i-\mu_i)x_i}{v(y_i)}.\frac{1}{g'(\mu_i)}$$

For the logit link, the above method will translate into the following:

$$Y_i \sim Bernoulli(\pi_i)$$

$$\eta_i=g(\mu_i)=\ln(\frac{\mu_i}{1-\mu_i})\text{ [logit link]}$$

$$\eta_i = \alpha + \beta x$$

$$\mu_i = E(Y_i) = \pi_i$$

$$V(Y_i) = \pi_i(1-\pi_i)$$

We can write ηi in terms of πi as:

$$\eta_i = \ln(\frac{\pi_i}{1-\pi_i})$$

In accordance with the previous method, the score equations will then be obtained as:

$$S_1 : \frac{\partial l}{\partial \alpha} = \sum_{i=1}^{n}(y_i - \pi_i) = 0 \text{------(*)}$$

$$S_2 : \frac{\partial l}{\partial \beta} = \sum_{i=1}^{n}(y_i - \pi_i)x_i = 0 \text{-----(**)}$$

Now, for obtaining the Information Matrices,

$$S_1 = \frac{\partial l}{\partial \alpha} = \sum_{i=1}^{n}(y_i - \pi_i)$$

$$\therefore \frac{\partial^2 l}{\partial \alpha^2} = \frac{\partial S_1}{\partial \pi_i}.\frac{\partial \pi_i}{\partial \eta_i}.\frac{\partial \eta_i}{\partial \alpha} = \sum_{i=1}^{n}(-1)\pi_i(1-\pi_i)(1) = \sum_{i=1}^{n} -\pi_i(1-\pi_i)$$

$$\therefore -E(\frac{\partial^2 l}{\partial \alpha^2}) = \sum_{i=1}^{n}\pi_i(1-\pi_i) = I_{11}$$

$$\frac{\partial^2 l}{\partial \alpha \partial \beta} = \frac{\partial S_1}{\partial \pi_i}.\frac{\partial \pi_i}{\partial \eta_i}.\frac{\partial \eta_i}{\partial \beta} = \sum_{i=1}^{n} -\pi_i(1-\pi_i)x_i$$

$$\therefore -E(\frac{\partial^2 l}{\partial \alpha \partial \beta}) = \sum_{i=1}^{n}\pi_i(1-\pi_i)x_i = I_{12}$$

$$S_2 = \frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} (y_i - \pi_i) x_i$$

$$\therefore \frac{\partial^2 l}{\partial \beta^2} = \frac{\partial S_2}{\partial \pi_i} . \frac{\partial \pi_i}{\partial \eta_i} . \frac{\partial \eta_i}{\partial \beta} = \sum_{i=1}^{n} (-x_i) \pi_i (1-\pi_i)(x_i) = \sum_{i=1}^{n} -\pi_i (1-\pi_i) x_i^2$$

$$\therefore -E(\frac{\partial^2 l}{\partial \beta^2}) = \sum_{i=1}^{n} \pi_i (1-\pi_i) x_i^2 = I_{22}$$

$$\frac{\partial^2 l}{\partial \beta \partial \alpha} = \frac{\partial S_2}{\partial \pi_i} . \frac{\partial \pi_i}{\partial \eta_i} . \frac{\partial \eta_i}{\partial \alpha} = \sum_{i=1}^{n} -\pi_i (1-\pi_i) x_i$$

$$\therefore -E(\frac{\partial^2 l}{\partial \beta \partial \alpha}) = \sum_{i=1}^{n} \pi_i (1-\pi_i) x_i \ = I_{21}$$

With the above information we can obtain the Information matrix, which, in this case, will be:

$$I(\theta) = \begin{pmatrix} -E(\frac{\partial^2 l}{\partial \alpha^2}) & -E(\frac{\partial^2 l}{\partial \alpha \partial \beta}) \\ -E(\frac{\partial^2 l}{\partial \beta \partial \alpha}) & -E(\frac{\partial^2 l}{\partial \beta^2}) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} \pi_i (1-\pi_i) & \sum_{i=1}^{n} \pi_i (1-\pi_i) x_i \\ \sum_{i=1}^{n} \pi_i (1-\pi_i) x_i & \sum_{i=1}^{n} \pi_i (1-\pi_i) x_i^2 \end{pmatrix}$$

LOGISTIC REGRESSION IN A MULTIVARIATE SETUP:

The problem of the prediction of a binary response using several variables is analogous to the problem of the prediction of a binary response using a single variable.

By analogy, we can say that

$$\log[\frac{\pi(x_1, x_2, ..., x_n)}{1-\pi(x_1, x_2, ..., x_n)}] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \quad \text{………} \ (\text{v})$$

Where X_1, X_2, ………., X_n are predictor variables.

Also, by transforming the above equation, we can write

$$\pi(x_1,x_2,....,x_n)=\frac{e^{\beta_0+\beta_1x_1+\beta_2x_2+....+\beta_nx_n}}{1+e^{\beta_0+\beta_1x_1+\beta_2x_2+....+\beta_nx_n}}$$ …………(vi)

The quantity logit[π(x)] can be interpreted as the logarithm of odds of success as,

logit[$\pi(x_1,x_2,....,x_n)$]= $\log[\frac{P(Y=1|X_1=x_1,X_2=x_2,.....,X_n=x_n)}{P(Y=0|X_1=x_1,X_2=x_2,.....,X_n=x_n)}]$

In logistic regression model in a multivariate setup also, the logit of success probability has a linear relationship with the explanatory(or independent) variables as:

∴ logit[π(x$_1$,x$_2$,…..,x$_n$)]= $\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+....+\beta_nx_n$ ………(vii)

where logit[$\pi(x_1,x_2,....,x_n)$] = $\log[\frac{\pi(x_1,x_2,....,x_n)}{1-\pi(x_1,x_2,....,x_n)}]$

Therefore, the logistic regression model in a multivariate setup has:

Odds of success= $e^{\beta_0+\beta_1x_1+\beta_2x_2+....+\beta_nx_n}$

log(Odds of success)= $\beta_0+\beta_1x_1+\beta_2x_2+....+\beta_nx_n$

ASSUMPTIONS OF A LOGISTIC REGRESSION MODEL:

- Logistic regression assumes that there is minimal to no multicollinearity present in the data.
- A large sample has to be present for good prediction using logistic regression.
- Logistic regression assumes that the observations are independent of each other's influence.
- Logistic regression also assumes that there are no outliers present in the data.

DECISION TREE ALGORITHM:

A decision tree can be included under a supervised machine learning technique. A decision tree is a tree-like diagram or model of decisions that show the possible consequences or a statistical probability. Branches of the decision tree stand for a possible decision. Decision trees can be used for solving ‘classification’ as well as ‘regression’. A classification tree is when a decision tree is used to model a categorical variable.

For a classification tree, we predict that each observation belongs to the most frequently occurring class of training observations in the area to which it belongs. While interpreting the results of a classification tree, we are often interested not only in the class predictions corresponding to a particular terminal node region but also in the class proportions among the training observations that belong to that area.

Some terminologies that are needed to be known while dealing with Decision Trees are as follows:

- Root Node: The root node represents the beginning of a decision tree. The root node represents the entire data and gets further divided into two or more homogenous divisions. A root node can have multiple subdivisions.
- Branch: A branch in a decision tree represents a possible decision.
- Decision node: The process of a sub-node splitting into more sub-nodes is called a decision node.
- Leaf Node: Leaf nodes represent the end of a decision tree. After the occurrence of a leaf node, the decision tree cannot have further subdivisions.

- Splitting: Splitting is the process by which a root node is divided into further sub-nodes according to some previously laid out conditions. Different decision tree algorithms use different conditions of splitting.
- Branch/Sub Tree: A sub-tree is a subsection of the decision tree.
- Pruning: Pruning is the removal of sub-nodes from a decision node.
- Parent node: The node which gets divided into further sub-nodes is known as a parent node.
- Child node: The sub-nodes which are formed from the division of a node are known as child nodes.

Generally, any split in a decision tree is made such that it gives rise to the highest predictive accuracy. This is generally measured with some node impurity measure, which demonstrates the relative homogeneity of the resulting sub-nodes. In a classification setting, the following criteria used for making binary splits are as follows:

Classification Error Rate: Classification error rate is the proportion of training observations that do not belong to the most common class. It is given by:

$$E = 1 - \max_k(\hat{p}_{mk})$$

The above equation $\hat{p}_{mk}$ represents the fraction of training observations belongs to the mth region that is from the kth class.

Gini index: Gini index is defined by:

$$G = \sum_{i=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Gini index is a measure of total variance across the K classes. It is not very difficult to see that the Gini index is a very small value if all the $\hat{p}_{mk}$'s are close to zero or one. This is the very reason the Gini Index is used as a measure of the purity of a node.

Cross entropy: Cross entropy is very similar to the Gini index and can also be provided as an alternative to the Gini index. Cross entropy is defined as:

$$D = -\sum_{i=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

Since $0 \leq \hat{p}_{mk} \leq 1$it follows that $-\sum_{i=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \geq 0$ .One can show that the cross-entropy will take on a value near zero if the $\hat{p}_{mk}$'s are all near zero or near one. Therefore, like the Gini index, the cross-entropy will take on a small value if the mth node is pure.

Different decision tree algorithms use different measures of homogeneity to select the best split. Some decision trees use Entropy, while some others use the Gini Index measure. The CART algorithm makes use of the "Gini Index" measure in selecting the best split. The formation of sub-nodes from the root node increases the homogeneity of the consequent sub-nodes. In other words, it can be said that the purity of the resultant node with respect to the response variable increases. Broadly, a decision tree splits using all the variables present and then the split which results in the formation of the most homogenous sub-nodes is selected.

ADVANTAGES OF USING A DECISION TREE:

- The decision tree algorithm requires much less effort for preprocessing the data in comparison to other algorithms.

- The decision tree also does not require the data to be normalized.
- Outliers do not affect decision trees to a considerable extent.
- The decision tree model is very easy to read, interpret and make decisions.

FITTING THE CLASSIFICATION ALGORITHMS ON A REAL-LIFE DATASET

DATASET DESCRIPTION

The data used was collected and made available by the National Institute of Diabetes and Digestive and Kidney Diseases and later collected from kaggle.com. All patients here are females of at least 21 years old of Pima Indian Heritage (which is a subgroup of the Native American Community). Therefore, the data in usage is secondary data.

The objective of this dataset is to predict whether a patient has diabetes or not, based on certain diagnostic characteristics included in the dataset which are as follows:

- Pregnancies: This discrete integer-valued variable denotes the number of times a person was pregnant.
- Glucose: This discrete integer-valued variable denotes the plasma glucose concentration over 2 hours in an oral glucose tolerance test (rounded off to the nearest integer) (in mg/dL).
- Blood pressure: This discrete integer-valued variable denotes the diastolic blood pressure (in mm Hg) (rounded off to the nearest integer).
- SkinThickness: This discrete integer-valued variable denotes the triceps skin fold thickness (in mm) (rounded off to the nearest integer).
- Insulin: This discrete integer-valued variable denotes the 2-hour serum insulin (in mu U/ml) (rounded off to the nearest integer).

- BMI: This continuous variable denotes the Body mass index(BMI)[calculated as $\frac{(weight\ in\ kg)}{(height\ in\ m)^2}$]
- DiabetesPedigreeFunction: This continuous variable denotes the Diabetes pedigree function (a function that scores the likelihood of diabetes based on family history).
- Age: This discrete integer-valued variable denotes the age (in years) of a person (rounded off to the nearest integer).
- Outcome: This binary variable gives the status of an individual as 0 if non-diabetic and 1 if diabetic.

Therefore, the dataset contains 9 attributes in total out of which *"Outcome"* is the dependent variable and the rest of the variables (that is: *"Pregnancies", "Glucose", "blood pressure", "SkinThickness", "Insulin", "BMI", "DiabetesPedigreeFunction", "age"*) are the independent variables.

DATA EXPLORATION AND PREPROCESSING

EXPLORING THE RELATION OF THE TARGET VARIABLE TO THE VARIOUS PREDICTOR VARIABLES:

This project aims to predict the "Outcome" variable that is the 9th variable in the dataset (whether a person has diabetes or not) based on the values of the remaining 8 attributes.

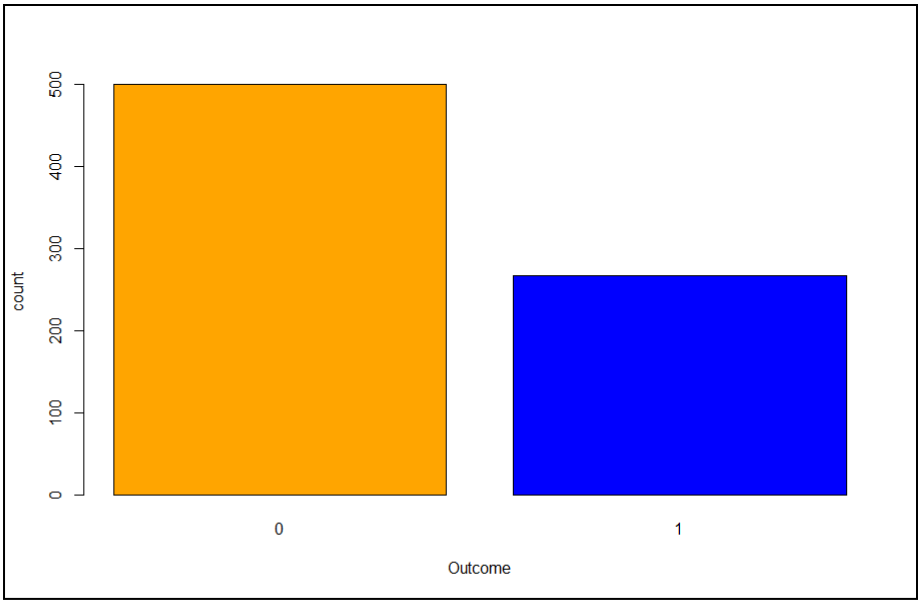

Figure 1: Count of non-diabetic and diabetic patients

From Figure 1, it is evident that of the 768 instances present in the dataset the number of instances with "0" values (indicating a non-diabetic person) is much more than the instances with "1" values (indicating a diabetic person).

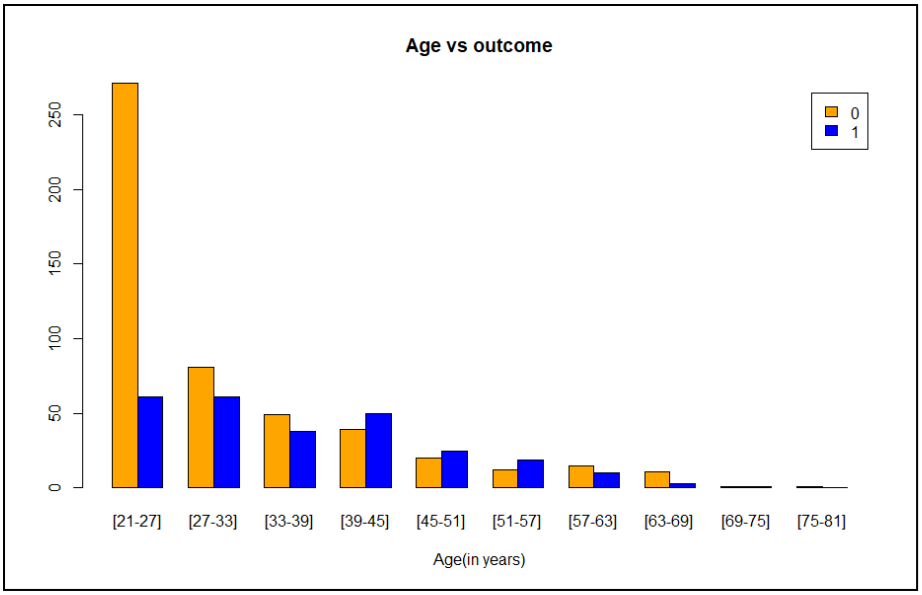

Figure 2: Age vs. Outcome

Figure 2 clearly shows that the number of diabetic patients increases as age increases. The number of diabetic patients is higher than the number of non-diabetic patients in the age group 39 to 57 years in this dataset. Though the number of non-diabetic patients in the lower age groups is higher than the diabetic patients, the number of diabetic patients is also quite pronounced.

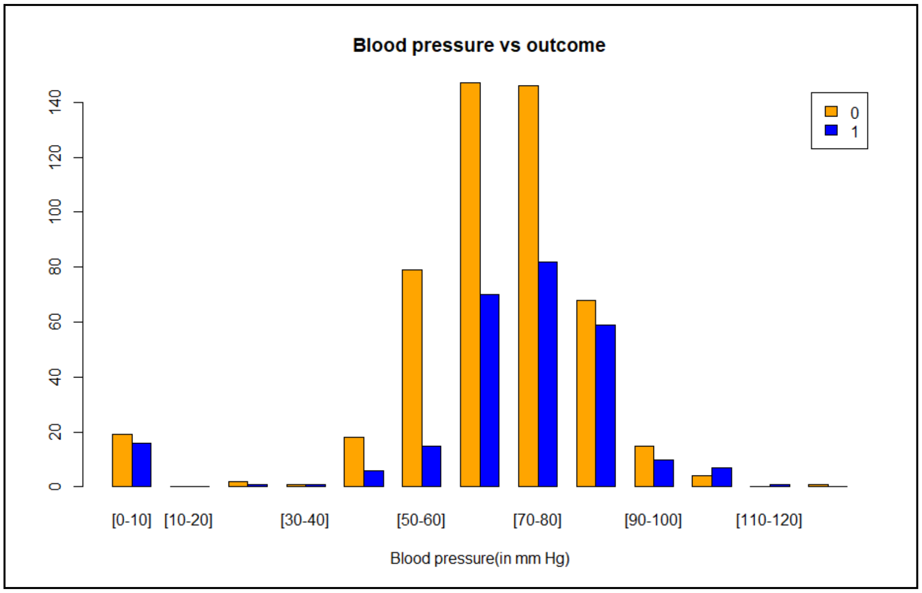

Figure 3: Blood pressure vs. Outcome

Figure 3 clearly shows that for normal diastolic blood pressure (that is when diastolic blood pressure is less than 80) the number of diabetic patients is much less than the number of non-diabetic patients. As the diastolic blood pressure exceeds 110 mm Hg, the number of diabetic patients exceeds the number of non-diabetic patients. The above graph also indicates that the values between 0 to 40 mm Hg are most likely outliers as most of the values are in the range of 50 to 100 mm Hg and the values 0 to 40 mm Hg are highly unlikely.

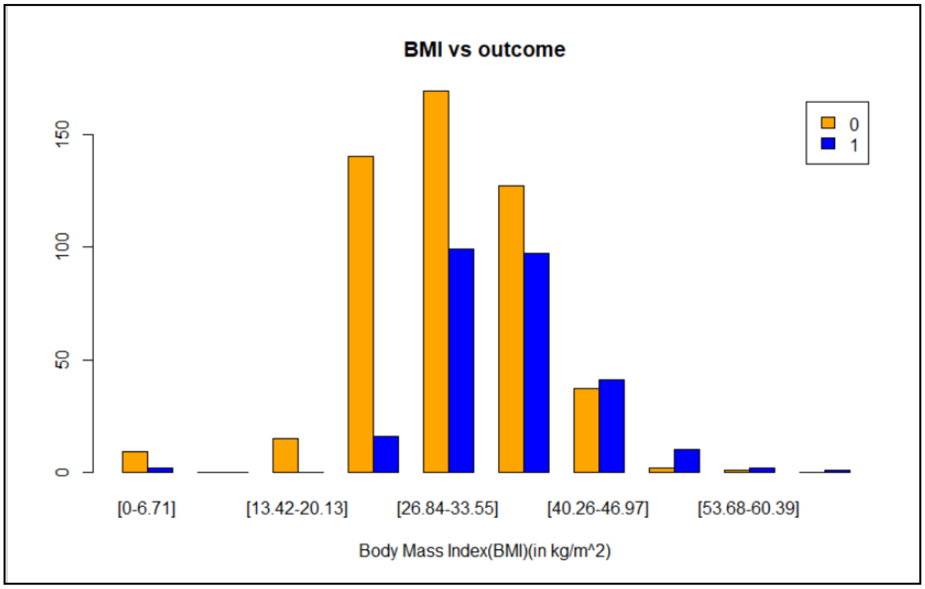

Figure 4: BMI vs. Outcome

Figure 4 clearly shows that as the BMI increases, the number of diabetic patient increases. From the above graph, we see that the number of diabetic patients exceeds the number of non-diabetic ones in the BMI group [40.26-46.97] which is categorized as ‘obese’. The groups [20.13-26.84] have a much less number of diabetic patients as they are categorized as the ‘healthy’ group. The values of BMI in the range of 0 to 13.42 kg/m^2 are highly unlikely and hence can be categorized as outliers.

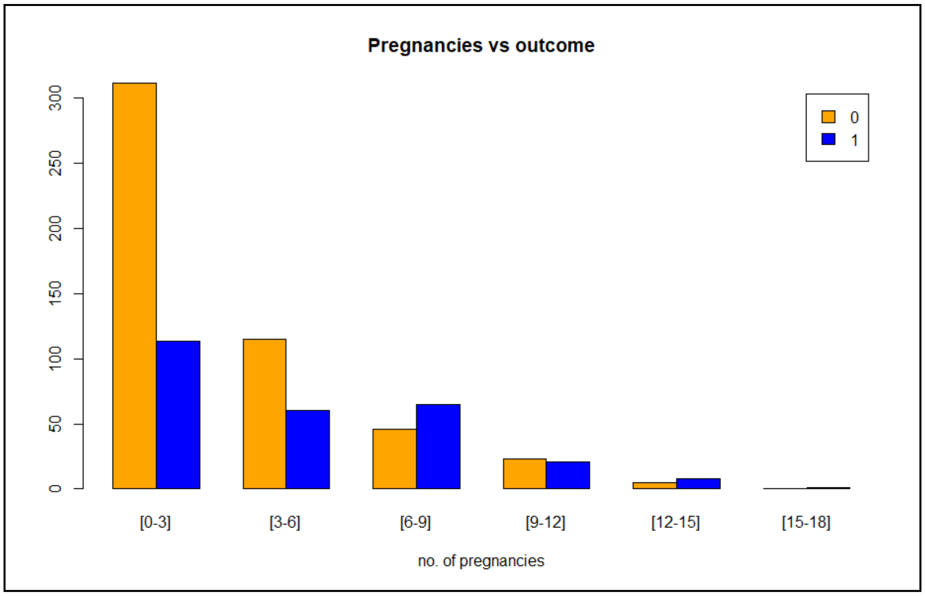

Figure 5: Pregnancies vs. Outcome

Figure 5 clearly shows that there is hardly any relation between pregnancies to being diabetic as the graph does not reveal any trend.

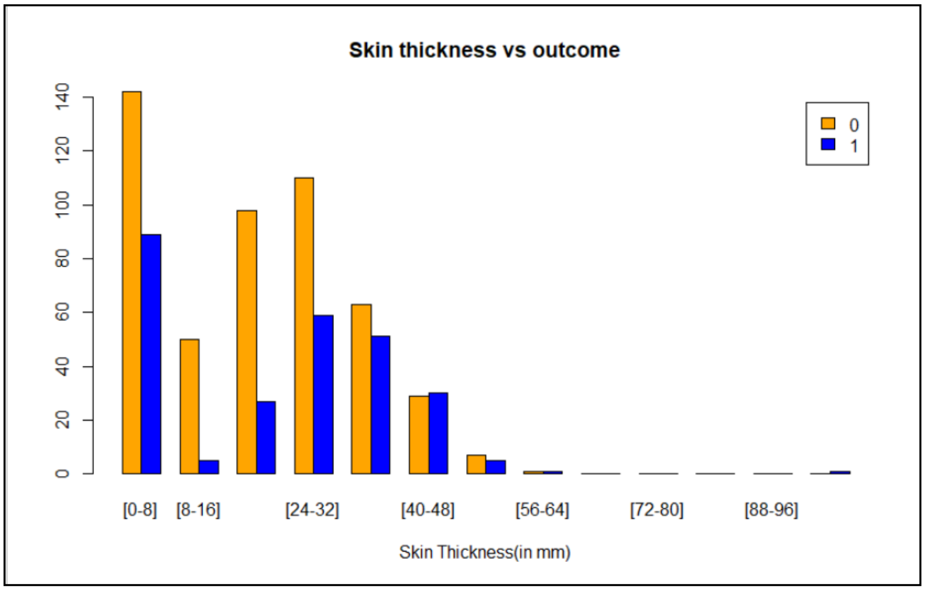

Figure 6: Skin thickness vs. Outcome

Figure 6 clearly shows that the most number of diabetic patients as compared to the non-diabetic ones are in the 40 to 48 mm skin thickness group. There are also large numbers of values with skin thickness between 0 to 8 mm which we suspect to be outliers due to its highly unlikely nature.

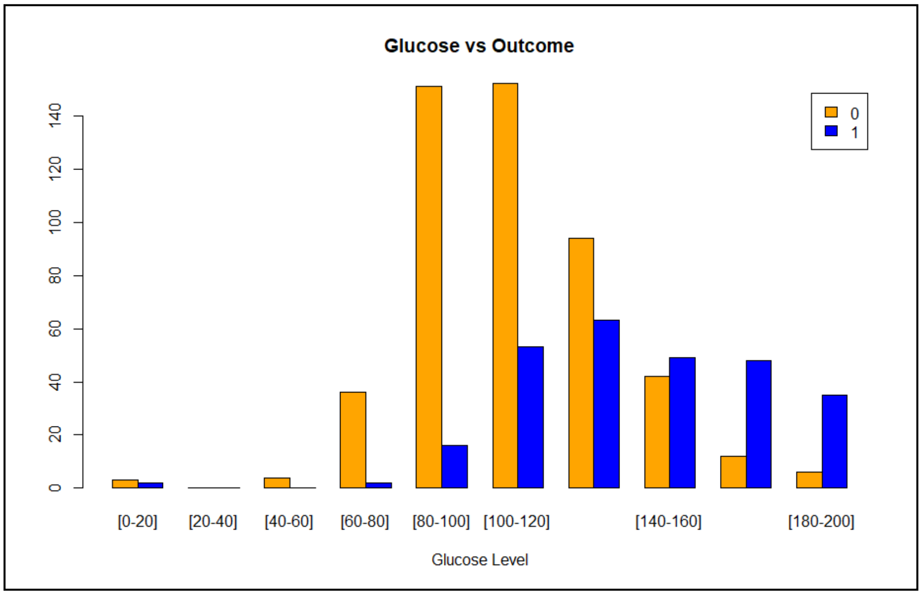

Figure 7: Glucose vs. Outcome

Figure 7 clearly shows that for normal blood glucose range (that is when blood glucose level is less than 140 mg/dL) the number of diabetic patients is much less than the number of non-diabetic patients. As the blood glucose level exceeds 140 mg/dL, the number of diabetic patients exceeds the number of non-diabetic patients. The above graph also indicates that the values between 0 to 60 mg/dL are most likely outliers as most of the values are in the range of 60 to 100 mg/dL and the values 0 to 40 mg/dL are highly unlikely.

CHECKING IF MULTICOLLINEARITY IS PRESENT IN THE DATASET:

As it was given in that Logistic Regression assumes that there is no multicollinearity present in the dataset, therefore we checked if multicollinearity was present among the variables of our dataset. For the above purpose, we plotted a correlation matrix for our data. The correlation matrix that we obtained is given below:

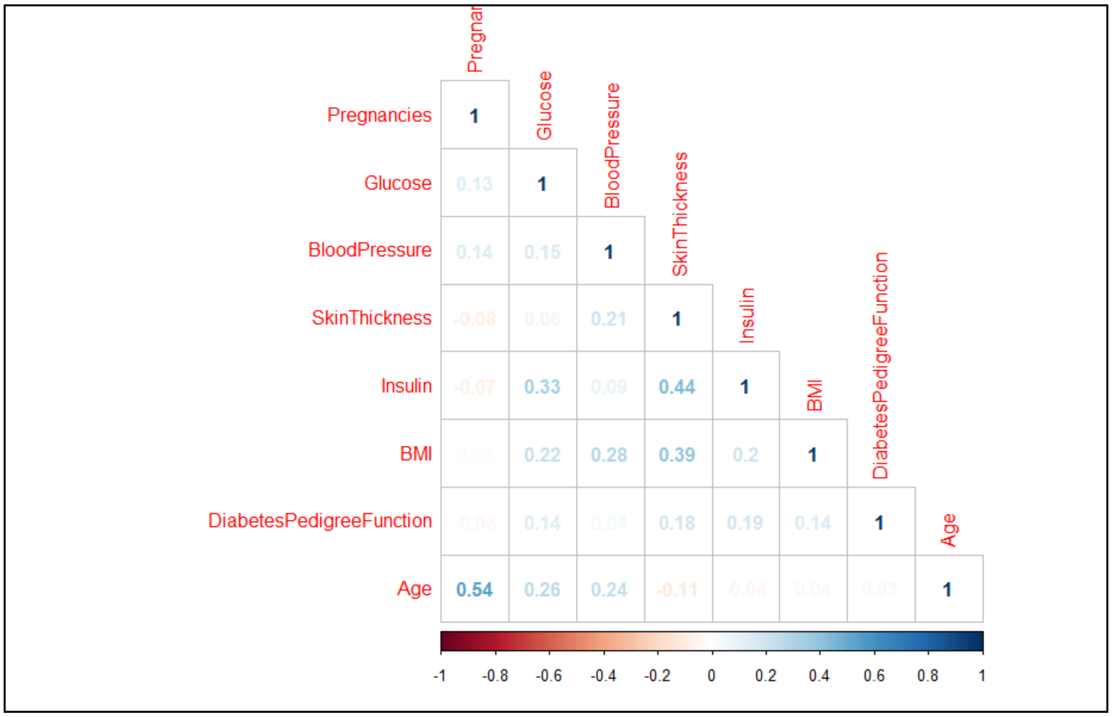

Figure 8: Correlation among predictor variables

From Figure 8, it can be seen that the correlation among the various predictor variables is low. The only variables which have a moderately high positive correlation among the features are “Age” and “Pregnancies”.

MISSING VALUE IMPUTATION

Although on checking for missing values in the dataset, we don’t find any missing values. On further introspection after seeing the graphs, it was found that there are 5 “0” values in the “Glucose” column, 11 “0” values in the “Body Mass Index (BMI)” column, 227 “0” values in the “Skin thickness” column, 374 “0” values in the “Insulin” column, 35 “0” values in the “Blood pressure” column, which is understandably erroneous as the above columns cannot have “0” values. All the 652 “0” values in the dataset were replaced by NA.

When missing values occur in a dataset, we can handle it in one of the following two ways:

- Removing the missing data
- Imputation of the missing data

We have used imputation of the missing data as for this dataset, the total number of individuals is small, and removing the missing observations will further decrease the number of observations in the dataset and will in turn compromise the accuracy of the model. Imputation of the data can be done using several methods:

- Imputation using mean or median values
- Imputation using the most frequent, zero, or constant values
- Imputation using kNN (k-nearest neighbors) algorithm

and many others.

Imputation using the kNN algorithm was used here with the aggregation method being mean. The *k* nearest neighbors is an algorithm that is used for simple classification. This algorithm makes use of ***‘feature similarity’*** to predict the missing data points. This means that the missing data point is assigned a value based on how closely it resembles the k-nearest points in the neighborhood of the missing data point in the dataset. This method was preferred over the others because of its universality. This method can be used for any kind of data-be it categorical, nominal, ordinal, or continuous and makes it particularly useful for imputation for all kinds of data.

After imputing the missing values, we label the “Output” variable as a factor variable and normalize the data and bring all the features to a similar scale and later use this data for fitting “Binary Logistic Regression”. For decision trees, we use the non-normalized data for fitting as decision trees do not require the features to be normalized and brought to the same scale.

FITTING OF CLASSIFICATION ALGORITHMS:

- **BINARY LOGISTIC REGRESSION:**

The final dataset was split into 75 percent training and 15 percent testing data. The training dataset consists of 576 observations and the testing dataset consists of 192 observations. Then the training dataset was fitted using binary logistic regression. The model summary is as follows:

SUMMARY TABLE OF BINARY LOGISTIC REGRESSION MODEL ON TRAINING DATASET:

Coefficients	Estimates	Standard error	z value	p-value	Odds ratio
Intercept	-5.3768	0.5961	-9.020	$<2x10^{-16}$	0.0046
Pregnancies	1.6727	0.6383	2.620	0.00878	5.3265
Glucose	5.7801	0.7594	7.612	$2.7x10^{-14}$	323.79
Blood pressure	-1.4861	0.9608	-1.547	0.12193	0.2262

Skin thickness	0.5426	1.4430	0.376	0.83070	1.7204
Insulin	-0.2593	1.2129	-0.214	0.83070	0.7715
BMI	4.5474	1.0504	4.329	1.5×10^{-5}	94.386
Diabetes pedigree function	1.9183	0.8235	2.329	0.001983	6.8093
Age	1.2403	0.6984	1.776	0.07574	3.4566

Table 1: Model Summary

From Table 1, it can be said that:

- The variables “Pregnancies”, “Glucose”, “BMI”, “Diabetes Pedigree Function” are significant as their p-values are less than 0.05.
- The amount by which the log odds of being diabetic changes due to one pregnancy keeping all other covariates fixed is 1.6727.The effect of an increase in the number of pregnancies by one increases the odds of being diabetic multiplicatively by a factor of 5.3265 keeping all other covariates fixed.
- The amount by which the log odds of being diabetic changes due to one mg/dL change in glucose plasma concentration keeping all other covariates fixed is 5.7801.The effect of an increase in glucose plasma concentration by one mg/dL increases the odds of being diabetic multiplicatively by a factor of 323.79 keeping all other covariates fixed.
- The amount by which the log odds of being diabetic changes due to one kg/m^2 change in Body Mass Index(BMI) keeping all other covariates fixed is 4.5474.The effect of an increase in BMI by one kg/m^2 increases the odds of being diabetic multiplicatively by a factor of 94.386 keeping all other covariates fixed.

- The amount by which the log odds of being diabetic changes due to one unit change in Diabetes Pedigree Function keeping all other covariates fixed is 1.9183.The effect of an increase in Diabetes Pedigree Function by one unit increases the odds of being diabetic multiplicatively by a factor of 6.8903 keeping all other covariates fixed.

SELECTING THE CLASSIFICATION THRESHOLD FROM ROC CURVE

As a logistic regression model returns a probability, in order to map a logistic regression model to a binary outcome a "classification threshold" must be decided in accordance with the model. To determine the threshold, we plotted a Receiver Operating Characteristic Curve (or ROC curve). A ROC curve plots TPR (True Positive Rate) versus FPR (False Positive Rate) at different classification thresholds. Lowering the classification threshold classifies more number of instances as positive, thus raising the number of both False Positives and True Positives.

Since the type of our study is clinical, therefore, we have to choose a model with high recall (or sensitivity).

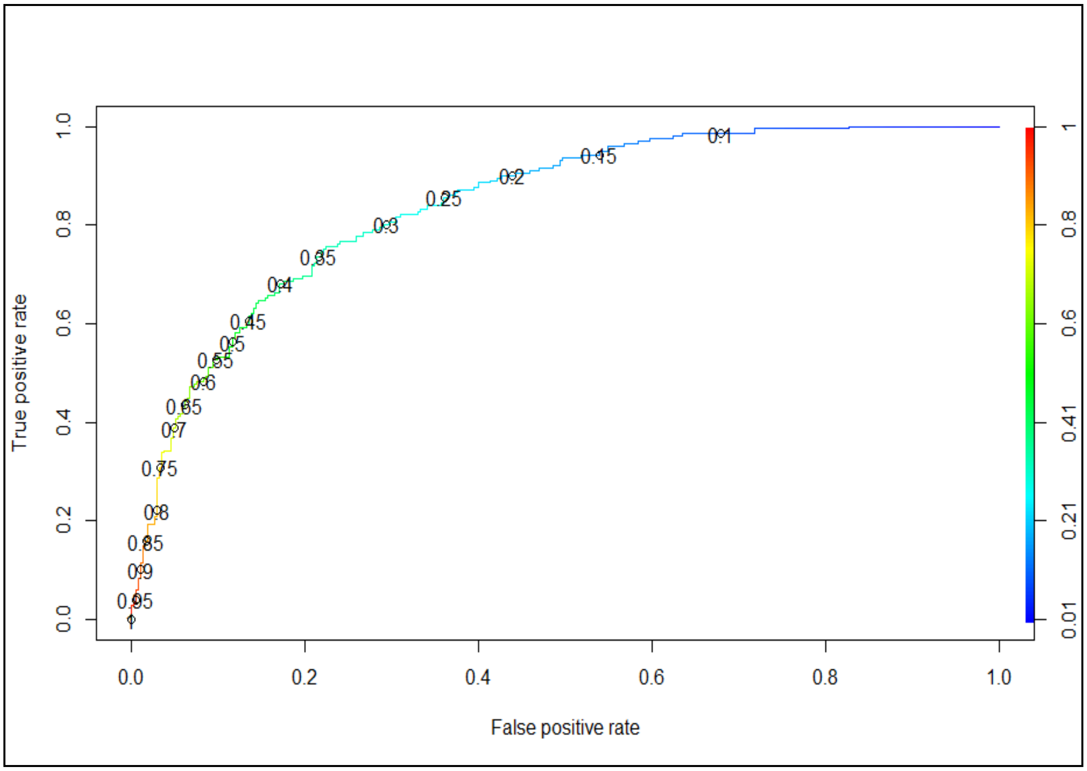

Figure 9: ROC curve of logistic regression model

We obtain an eye estimate from Figure 9 and select the value of the classification threshold as 0.5.

The model when fitted on the training data had an accuracy of 77.08%, a sensitivity of 79%, and a specificity of 71.97%.

The confusion matrix obtained when the model was fitted to training data was:

	Actual 'non-diabetic'(0)	Actual 'diabetic'(1)
Predicted-'non-diabetic'	331(57.46%)	44(7.64%)
Predicted-'diabetic'	88(15.27%)	113(19.62%)

From the Confusion matrix it is observed that in the training data (created by randomly selecting 75% data that is 576 observations), 331 observations with label '0' are predicted correctly while

88 observations are misclassified as ‘1’. 44 observations with label ‘1’ are predicted incorrectly as ‘0’ while 113 observations with label ‘1’ are predicted correctly. The fit seems to be moderately good as 77.08% of the training data has been classified correctly.

The model when fitted on the test data had an accuracy of 77.6%, a sensitivity of 86%, and a specificity of 61.19%.

The confusion matrix obtained when the model was fitted to test data was:

	Actual ‘non-diabetic’(0)	Actual ‘diabetic’(1)
Predicted-‘non-diabetic’	108(56.25%)	26(13.54%)
Predicted-‘diabetic’	17(8.85%)	41(21.35%)

From the Confusion matrix, it is observed that in the test data (created by randomly selecting 15% data that is 192 observations), 108 observations with label ‘0’ are predicted correctly while 17 observations are misclassified as ‘1’. 26 observations with label ‘1’ are predicted incorrectly as ‘0’ while 41 observations with label ‘1’ are predicted correctly. The fit seems to be moderately good as 77.6% of the training data has been classified correctly.

- **DECISION TREE:**

Many decision tree algorithms are present but the one that we used in this program is CART. CART algorithm in general uses Gini Index to determine the split that results in the most homogenous nodes. Using CART has several advantages such as:

- CART is non-parametric and does not make any assumptions about the probability distribution.
- CART automatically performs variable selection.

- CART does not require the normalization of data.
- CART handles outliers and missing values by itself effectively.

Since the implementation of CART does not require normalization of data, we use non-normalized data to make the decision tree. However, the missing values were handled as mentioned before. The non-normalized data was split into 75% training data (576 data points) and 15%testing data(192 data points).

Then a decision tree was build using the training data. The decision tree obtained is as follows:

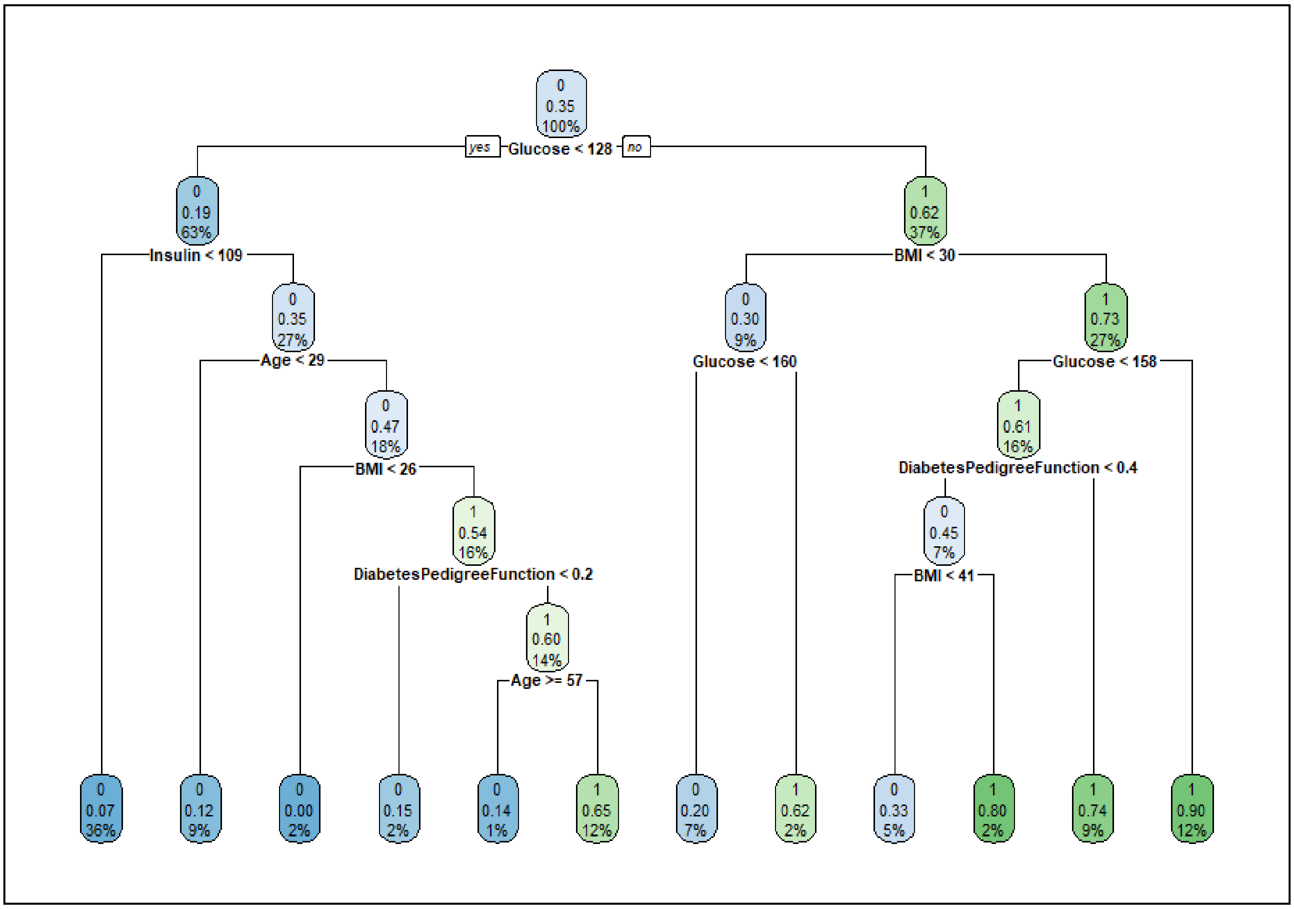

Figure 10: Decision Tree showing classification of diabetes

The above decision tree in Figure 10 can be interpreted as follows:

- Starting from the root node, it is the overall probability of being non-diabetic.It shows that 35% of people in the training dataset are diabetic.
- The node just below the root node asks whether the glucose plasma concentration is less than 128 units or not. If yes, then we go to the left child node. From there, we see that 63% are patients with glucose plasma concentration less than 128 units and a probability of being diabetic of 19%.
- The node just below the first left child node asks whether the insulin level of a person is 109 mu U/ml. If yes, then the chance of being non-diabetic is 7%.

This way we can keep interpreting the decision tree and understanding which features are important in a person being diabetic.

The decision tree model when fitted on the test data had an accuracy of 79.69%, a sensitivity of 86.40%, and a specificity of 67.16%.

The confusion matrix obtained when the model was fitted to test data was:

	Actual ‘non-diabetic’(0)	Actual ‘diabetic’(1)
Predicted-‘non-diabetic’	108(56.25%)	22(11.45%)
Predicted-‘diabetic’	17(8.85%)	45(23.43%)

From the Confusion matrix it is observed that in the test data (created by randomly selecting 15% data that is 192 observations), 108 observations with label ‘0’ are predicted correctly while 17 observations are misclassified as ‘1’. 22 observations with label ‘1’ are predicted incorrectly as ‘0’ while 45 observations with label ‘1’ are predicted correctly. The fit seems to be moderately good as 79.69% of the training data has been classified correctly.

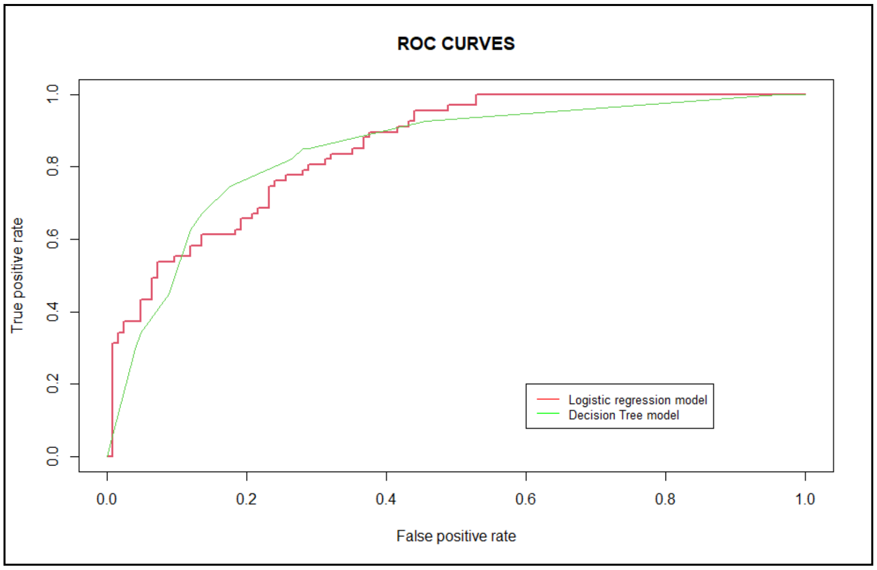

Figure 11: Comparison of ROC curves of both the models

<u>INFERENCE:</u> In terms of accuracy, sensitivity and specificity for the given dataset the Decision Tree model seem to be better fitting to the data, than the logistic regression model. From Figure 11, however, we see that for some lower and higher values, the Logistic Regression model fits better than the decision tree model whereas in the values of the middle the decision tree performs better.

CONCLUSION

In machine learning, classification problems are the most interesting to delve into and yet challenging to carry out. Each classification problem that we look into, we discover many new ways of looking at it. The work done above has made use of preliminary classification algorithms like decision tree and logistic regression model. Since, the AUC scores for both models are nearly the same and the sensitivity and specificity of the Decision tree model is better we can say that decision model performs better for this dataset. The implications of a competent classification model are enormous as classification models are used in many areas. Many types of research are being conducted on improving the classification model algorithms and making them more accurate with a smaller training dataset at hand.

REFERENCES

The following references helped me in the completion of my project:

- https://www.datacamp.com/community/tutorials/decision-trees-R
- https://link.medium.com/Qldrm1IfHfb
- https://www.reneshbedre.com/blog/logistic-regression.html
- https://link.medium.com/KDQa183tUfb
- https://www.kaggle.com/uciml/pima-indians-diabetes-database
- https://www.wikipedia.org
- https://www.who.int
- An Introduction to Statistical Learning with Applications in R-Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- The Hundred-Page Machine Learning Book-Andriy Burkov
- The Elements of Statistical Learning- Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Abdullahi A. Ibrahim, Raheem L. Ridwan, Muhammed M. Muhammed, Rabiat O. Abdulaziz, Ganiyu A. Saheed (2020)- Comparison of the CatBoost Classifier with other Machine Learning Methods
- Meraj Nabi, Abdul Wahid, Pradeep Kumar (2017) - Performance Analysis of Classification Algorithms in Predicting Diabetes.

ACKNOWLEDGEMENTS

This project is the outcome of a series of research carried out by me under the able guidance of my esteemed supervisor, who gave me the oppurtunity to gain valuable imformation practical insights into the chosen area of study.Without her help, it would not have been possible for me to frame the ideas and observations in a concrete form in my dissertation. Moving on I appreciate all constructive suggestions received from my friends and the constant inspiration and motivation received from my parents.I am hopeful that the knowledge, skills and insights gained through this project will go a long way in shaping my future academic endeavours.

SUPPLEMENTARY MATERIAL

```
rm(list=ls())


#installing required packages
library(Hmisc)
library(dplyr)
library(MASS)
library(ggplot2)
library(dplyr)
library(VIM)
library(gridExtra)
library(caret)
library(ROCR)
library(tree)
library(e1071)


#reading the dataset
df=read.csv("diabetes2.csv")


head(df)
#df contains 768 rows and 9 columns



#EDA(Exploratory Data Analysis)
breaks=seq(21,81,6)
tags=c("[21-27]","[27-33]","[33-39]","[39-45]","[45-51]","[51-57]","[57-63]","[63-69]","[69-75]","[75-81]")
bin_age=cut(df$Age,breaks=breaks,labels=tags,include.lowest=TRUE)
summary(bin_age)
```

```
tbl_age=table(df$Outcome,bin_age)
barplot(tbl_age,main="Age vs outcome",xlab="Age(in years)",col=c("orange","blue"),legend=rownames(tbl_age),beside=TRUE)
#the grouped vertical barplot for Age vs Outcome
#the rest of the barplots were also obtained in a similar manner

corrplot::corrplot(cor(df[,-9]),type="lower",method="number")
#correlation plot for predictor variables

is.na(df)
#no missing values

df1=df[,2:6]
head(df1)
#Further introspection shows "0"s in Glucose, BloodPressure, Skinthickness, BMI, Insulin which
#seem to be erroneous.Replacing them by NA.

#replacing all "0" values in these columns to NA
df1[df1=="0"]=NA

summary(df1)

#so that NA's are incorporated in the df
df$Glucose=df1$Glucose
df$BloodPressure=df1$BloodPressure
df$SkinThickness=df1$SkinThickness
df$Insulin=df1$Insulin
df$BMI=df1$BMI
summary(df)
```

```
df1=df[,-9]


#Imputation by kNN of the missing values
df1=kNN(df1,k=sqrt(nrow(df1)))
summary(df1)
df1=df1[,1:8]

#Replacing with imputed values
df$Pregnancies=df1$Pregnancies
df$Glucose=df1$Glucose
df$BloodPressure=df1$BloodPressure
df$SkinThickness=df1$SkinThickness
df$Insulin=df1$Insulin
df$BMI=df1$BMI
df$DiabetesPedigreeFunction=df1$DiabetesPedigreeFunction
df$Age=df1$Age
summary(df)



#converting "Outcome" to a dependent variable
df$Outcome=as.factor(df$Outcome)

df_tree=df

#normalization of the explanatory variables
preproc2=preProcess(df[,1:8],method=c("range"))
norm2=predict(preproc2,df[,1:8])
summary(norm2)
```

```
df[,1:8]=norm2[,1:8]
str(df)

#Partitioning dataset into training and testing set



set.seed(123)
index=createDataPartition(df$Outcome,p=0.75,list=F)
train_df=df[index,]
test_df=df[-index,]



#BINARY LOGISTIC REGRESSION
log_model=glm(Outcome~.,data=train_df,family="binomial")

summary(log_model)
train_df$pred_prob_outcome=fitted(log_model)

#plotting ROC curve for training dataset
pred=prediction(train_df$pred_prob_outcome,train_df$Outcome)
perf=performance(pred,"tpr","fpr")
plot(perf,colorize=T,print.cutoffs.at=seq(0.1,by=0.05))



train_df$pred_outcome=ifelse(train_df$pred_prob_outcome>0.5,1,0)
head(train_df)

#obtaining the confusion matrix for train data in logistic regression
confusionMatrix(table(train_df$Outcome,train_df$pred_outcome))
```

```
#testing the model
glm_probs=predict(log_model,newdata = test_df,type="response")
glm_pred=as.factor(ifelse(glm_probs>0.5,1,0))
#obtaining the confusion matrix for test data in logistic regression
confusionMatrix(glm_pred,test_df$Outcome)

acc_glm_fit=confusionMatrix(glm_pred,test_df$Outcome)$overal['Accuracy']



#DECISION TREE

set.seed(123)
index=createDataPartition(df_tree$Outcome,p=0.75,list=F)
train_df_1=df_tree[index,]
test_df_1=df_tree[-index,]

library(rpart)
library(rpart.plot)
dectree_model=rpart(Outcome~.,data=train_df_1)
summary(dectree_model)
par(mfrow=c(1,1))
#plotting the decision tree
rpart.plot(dectree_model,cex=0.7)

tree_pred=predict(dectree_model,newdata = test_df_1,type="class")
confusionMatrix(tree_pred,test_df_1$Outcome)
```

```
acc_treemod=confusionMatrix(tree_pred,test_df_1$Outcome)$overall['Accuracy']

#plotting ROC curves for both models

par(mfrow=c(1,1))

ROCRpred=prediction(glm_probs,test_df$Outcome)

perf1=performance(ROCRpred,"tpr","fpr")

plot(perf1,col=2,lwd=2,main="ROC CURVES")

pred2=predict(dectree_model,type="prob",newdata=test_df_1)

pred3=prediction(pred2[,2],test_df_1$Outcome)

perf2=performance(pred3,"tpr","fpr")

plot(perf2,add=TRUE,col=3)

legend(0.6,0.2,legend=c("Logistic regression model","Decision Tree model"),col=c("red","green"),lty=1:1,cex=0.8)
```