A Comparative Study of Classification Algorithms on Multivariate Data

(Project of MTH514A)

Submitted by,

Rajdeep Saha (201380)
Sagnik Dey (201397)
Saumyadip Bhowmick (201408)
Shuvam Gupta (201421)
Soumik Karmakar (201428)

Supervised by,

Dr. Minerva Mukhopadhyay

Submitted on,

21st April, 2022

Contents

1 Introduction

In real life scenarios, the Classification problem arises more frequently than the basic regression one.Suppose we have to classify if patient is diabetic or not , a patient will or will not respond to the drug , a tumour indicates breast cancer or not based on that dataset.Now, there are several classfiers available for our use but not all of them will provide good fit and thus accurate prediction to the datasets. In our project, we are going to study the accuracy of several classifiers like Logistic Regression ,Linear Discriminant Analysis,Quadratic Discriminant Analysis,Decision , and Random Forest on 4 separate datasets i.e. PIMA Indian Diabetes Dataset, Caravan Insurance Dataset, Banknote Authentication Dataset, Weekly Dataset .Then we will provide a comparative study based on our results for all 4 classifiers on the 4 datasets.

2 Different Classification Algorithms

2.1 Logistic Regression

Logistic Regression is one of the most popular classification algorithms.In logistic regression, rather than modeling the response variable directly, we model the probability that response belongs to a specific class.

Suppose a population has two classes π_1 & π_2 and $\mathbf{x}$ is a p-variate observation. According to **Bayes' Rule**, we would classify $\mathbf{x}$ to π_1 if,

$$P(\pi_1|\mathbf{x}) \geq P(\pi_2|\mathbf{x})$$

Logistic Regression assumes that the **class conditional densities** if the two class satisfies,

$$\log\left(\frac{P(\mathbf{x}|\pi_1)}{P(\mathbf{x}|\pi_2)}\right) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1'}\mathrm{x}$$

$$\text{or, } \log\left(\frac{f(\mathbf{x})P(\pi_1|\mathbf{x})/P(\pi_1)}{f(\mathbf{x})P(\pi_2|\mathbf{x})/P(\pi_2)}\right) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1'}\mathbf{x}$$

$$\text{or, } \frac{P(\pi_2)}{P(\pi_1)}\frac{P(\pi_1|\mathbf{x})}{P(\pi_2|\mathbf{x})} = \exp(\boldsymbol{\beta_0} + \boldsymbol{\beta_1'}\mathbf{x})$$

$$\text{or, } \frac{P(\pi_1|\mathbf{x})}{P(\pi_2|\mathbf{x})} = \exp\left(\log\frac{P(\pi_2)}{P(\pi_1)} + \boldsymbol{\beta_0} + \boldsymbol{\beta_1'}\mathbf{x}\right)$$

$$\text{or, } \frac{1 - P(\pi_2|\mathbf{x})}{P(\pi_2|\mathbf{x})} = \exp(\boldsymbol{\beta_0^*} + \boldsymbol{\beta_1'}\mathbf{x}), \left[\boldsymbol{\beta_0^*} = \log\frac{P(\pi_2)}{P(\pi_1) + \beta_\mathbf{0}}\right]$$

$$\boxed{\text{or, } P(\pi_2|\mathbf{x}) = \frac{1}{1 + \exp(\boldsymbol{\beta_0^*} + \boldsymbol{\beta_1'}\mathbf{x})}}$$

and,

$$\boxed{\text{or, } P(\pi_1|\mathbf{x}) = \frac{\exp(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_1^{'}\mathbf{x})}{1+\exp(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_1^{'}\mathbf{x})}}$$

We assign $\mathbf{x}$ to π_1 if, $P(\pi_1|\mathbf{x}) \geq P(\pi_2|\mathbf{x})$
We assign $\mathbf{x}$ to π_2 if, $P(\pi_1|\mathbf{x}) < P(\pi_2|\mathbf{x})$

We can also use **Threshold Probability** to determine the class to which $\mathbf{x}$ belongs to. If $P(\pi_1|\mathbf{x})$ is greater than some threshold probability (by default, it is taken to be 0.5, but we can vary this threshold value in (0, 1)), we assign it to π_1, else we assign it to π_2.

2.2 Discriminant Analysis

Discriminant Analysis is the optimal way to seperate heterogeneous populations. Here we classify new observation vector in one of the possible populations using discriminant function. Suppose a population has two classes π_1 & π_2. $\mathbf{X}|\pi_1$ has support $\mathfrak{X}_1$ and $\mathbf{X}|\pi_2$ has support $\mathfrak{X}_2$, $\mathfrak{X} = \mathfrak{X}_1 \cup \mathfrak{X}_2$
Let, c(i|j) be the cost of misclassification of an observation from π_j to π_i, i.e., c(1|1)= c(2|2)= 0.
Let, the class conditional density of class 'π_i' is $f_i(\mathbf{x})$, i= 1, 2.
Let, the prior probabilities of two population be p_1, p_2 and p_1+ p_2= 1.
Let, the classification partition be R_1 and R_2, where, $R_1 \cap R_2 = \phi$ and $R_1 \cup R_2 = \mathfrak{X}$.
If $\mathbf{x}\in R_1$, we classify it to π_1.
If $\mathbf{x}\in R_2$, we classify it to π_2.
Define, $P(i|j)$= Probability of misclassifying an observation from π_j to π_i. For a two class problem, we define,
Total Probability of Misclassification (TPM)= $p_1P(2|1) + p_2P(1|2)$
Expected Cost of Misclassification (ECM)= $c(2|1)p_1P(2|1)$+$c(1|2)p_2P(1|2)$

2.2.1 Linear Discriminant Analysis

TPM Minimizing classifier can be used to obtain Linear Discriminant Function. Now, TPM would be minimized if,

$$R_1^* = \{\mathbf{x} : p_2 f_2(\mathbf{x}) \leq p_1 f_1(\mathbf{x})\}$$

or,

$$R_1^* = \left\{\mathbf{x} : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}\right\}$$

$$R_2^* = \left\{\mathbf{x} : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}\right\}$$

We classify $\mathbf{x}$ to π_1 if,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}$$

and to π_2 if,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}$$

Now, to obtain **Linear Discriminant Function**, we assume,

$$\pi_1 \equiv N_p(\mu_1, \Sigma)$$
$$\pi_2 \equiv N_p(\mu_2, \Sigma)$$

We have TPM minimizing classifier,

$$R_1^* = \{\mathbf{x} : p_2 f_2(\mathbf{x}) \leq p_1 f_1(\mathbf{x})\}$$

or,

$$R_1^* = \left\{\mathbf{x} : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}\right\}$$

now,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} = \frac{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{1}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{1}}))}{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{2}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{2}}))}$$

Now,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}$$

$$\text{or, } \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{1}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{1}}) + -\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{2}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{2}})\right\} \geq \frac{p_2}{p_1}$$

$$\text{or, } \left\{-\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{1}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{1}}) + -\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{2}})^T\Sigma^{-1}(\mathbf{x}-\mu_{\mathbf{2}})\right\} \geq \log\left(\frac{p_2}{p_1}\right)$$

$$\text{or, } -(\mathbf{x}^T\Sigma^{-1}\mathbf{x}+{\mu_{\mathbf{1}}}^T\Sigma^{-1}\mu_{\mathbf{1}}-2{\mu_{\mathbf{1}}}^T\Sigma^{-1}\mathbf{x})+(\mathbf{x}^T\Sigma^{-1}\mathbf{x}+{\mu_{\mathbf{2}}}^T\Sigma^{-1}\mu_{\mathbf{2}}-2{\mu_{\mathbf{2}}}^T\Sigma^{-1}\mathbf{x}) \geq 2\log\left(\frac{p_2}{p_1}\right)$$

$$\text{or, } 2{\mu_{\mathbf{1}}}^T\Sigma^{-1}\mathbf{x} - 2{\mu_{\mathbf{2}}}^T\Sigma^{-1}\mathbf{x} \geq {\mu_{\mathbf{1}}}^T\Sigma^{-1}\mu_{\mathbf{1}} - {\mu_{\mathbf{2}}}^T\Sigma^{-1}\mu_{\mathbf{2}} + 2\log\left(\frac{p_2}{p_1}\right)$$

$$\text{or, } 2(\mu_1-\mu_2)^T\Sigma^{-1}\mathbf{x} \geq {\mu_{\mathbf{1}}}^T\Sigma^{-1}\mu_{\mathbf{1}}-{\mu_{\mathbf{2}}}^T\Sigma^{-1}\mu_{\mathbf{2}}+{\mu_{\mathbf{1}}}^T\Sigma^{-1}\mu_{\mathbf{2}}-{\mu_{\mathbf{2}}}^T\Sigma^{-1}\mu_{\mathbf{1}}+2\log\left(\frac{p_2}{p_1}\right)$$

$$\text{or, } (\mu_1 - \mu_2)^T \Sigma^{-1} \mathbf{x} \geq \frac{1}{2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 + \mu_2) + \log\left(\frac{p_2}{p_1}\right)$$

Hence,

$$\boxed{R_1^* = \left\{ \mathbf{x} : (\mu_1 - \mu_2)^T \Sigma^{-1} \mathbf{x} \geq \tfrac{1}{2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 + \mu_2) + \log\left(\tfrac{p_2}{p_1}\right) \right\}}$$

and,

$$\boxed{R_2^* = \left\{ \mathbf{x} : (\mu_1 - \mu_2)^T \Sigma^{-1} \mathbf{x} < \tfrac{1}{2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 + \mu_2) + \log\left(\tfrac{p_2}{p_1}\right) \right\}}$$

2.2.2 Quadratic Discriminant Analysis (QDA)

ECM minimizing classifier can be used to obtain Quadratic Discriminant Function. Now, ECM would be minimized if,

$$\tilde{R_1}^* = \{\mathbf{x} : p_2 c(1|2) f_2(\mathbf{x}) \leq p_1 c(2|1) f_1(\mathbf{x})\}$$

or,

$$\tilde{R_1}^* = \left\{ \mathbf{x} : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2 c(1|2)}{p_1 c(2|1)} \right\}$$

$$\tilde{R_2}^* = \left\{ \mathbf{x} : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2 c(1|2)}{p_1 c(2|1)} \right\}$$

We classify $\mathbf{x}$ to π_1 if,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2 c(1|2)}{p_1 c(2|1)}$$

and to π_2 if,

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2 c(1|2)}{p_1 c(2|1)}$$

Now, here,

$$\pi_1 \equiv N_P(\boldsymbol{\mu_1}, \Sigma_1), \qquad \Sigma_1 > 0$$
$$\pi_2 \equiv N_P(\boldsymbol{\mu_2}, \Sigma_2), \qquad \Sigma_2 > 0$$

$$
\begin{aligned}
&\frac{f_1(\boldsymbol{x})}{f_1(\boldsymbol{x})} \geq \frac{p_2c(1|2)}{p_1c(2|1)}\\
&\Rightarrow \frac{(2\pi)^{-\frac{p}{2}}|\Sigma_1|exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})'\Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right)}{(2\pi)^{-\frac{p}{2}}|\Sigma_1|exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})'\Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2})\right)} \geq \frac{p_2c(1|2)}{p_1c(2|1)}\\
&\Rightarrow -\frac{1}{2}log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) - \frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{\mu_1})'\Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1}) - (\boldsymbol{x}-\boldsymbol{\mu_2})'\Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2})\right) \geq log\left(\frac{p_2c(1|2)}{p_1c(2|1)}\right)\\
&\Rightarrow -\frac{1}{2}log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) - \frac{1}{2}\left(\boldsymbol{x}'\Sigma_1^{-1}\boldsymbol{x} - \boldsymbol{x}'\Sigma_2^{-1}\boldsymbol{x} + \boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{\mu_1} - \boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{\mu_2} - 2\boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{x} + 2\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{x}\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \geq log\left(\frac{p_2c(1|2)}{p_1c(2|1)}\right)\\
&\Rightarrow -\frac{1}{2}\left(\boldsymbol{x}'\Sigma_1^{-1}\boldsymbol{x} - \boldsymbol{x}'\Sigma_2^{-1}\boldsymbol{x} - 2\boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{x} + 2\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{x}\right) \geq log\left(\frac{p_2c(1|2)}{p_1c(2|1)}\right) +\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad -\frac{1}{2}log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \left(\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{\mu_2} - \boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{\mu_1}\right)
\end{aligned}
$$

Function in the LHS is called **Quadratic Discriminant Function**.
Hence,

$$
\tilde{R_1}^* = \left\{\mathbf{x} : -\frac{1}{2}\left(\boldsymbol{x}'\Sigma_1^{-1}\boldsymbol{x} - \boldsymbol{x}'\Sigma_2^{-1}\boldsymbol{x} - 2\boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{x} + 2\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{x}\right) \geq log\left(\frac{p_2c(1|2)}{p_1c(2|1)}\right) + \right.
$$

$$
\left. \frac{1}{2}log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \left(\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{\mu_2} - \boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{\mu_1}\right)\right\}
$$

and,

$$
\tilde{R_2}^* = \left\{\mathbf{x} : -\frac{1}{2}\left(\boldsymbol{x}'\Sigma_1^{-1}\boldsymbol{x} - \boldsymbol{x}'\Sigma_2^{-1}\boldsymbol{x} - 2\boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{x} + 2\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{x}\right) < log\left(\frac{p_2c(1|2)}{p_1c(2|1)}\right) + \right.
$$

$$
\left. \frac{1}{2}log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \left(\boldsymbol{\mu_2'}\Sigma_2^{-1}\boldsymbol{\mu_2} - \boldsymbol{\mu_1'}\Sigma_1^{-1}\boldsymbol{\mu_1}\right)\right\}
$$

2.3 Decision Tree

Decision Tree is an instance of a multistage decision process. Rather than using the complete set of features jointly to make an output decision, different subsets of features are used at different levels of tree. Decision trees are drawn upside down. The points throughout the tree at which the predictor space is split is called **internal nodes**. The topmost internal node is called **root of the tree**. The nodes, with which a class label is associated, is called **terminal nodes** or **leaf** of the tree.For a decision tree, we predict that each observation belongs to the most frequently occuring class of training observations in the particular region to which it belongs.

Now, the most important question in building a tree is that when should we stop splitting. We may grow the tree till all the terminal nodes are pure (all patterns belonging to the partitions have same class label). But it may cause the problem of **overfitting**.

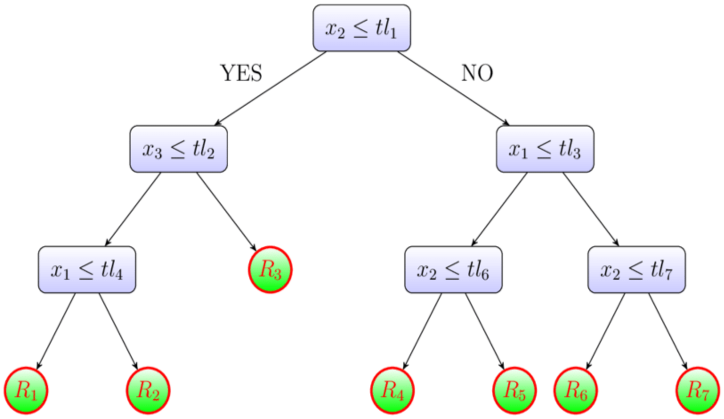

Figure 1: Random Forest Model

To address this problem, we bring the concept of **pruning** of a tree. Pruning of a grown tree after applying splitting of nodes basically means cutting branches of the tree to get subtree of the original tree.

For pruning, we use **Cost Complexity Pruning** or **Weakest Link Pruning**.

(i)<u>Cost Complexity Pruning</u>:

Let, r(t)=1-max P(y=i|t), i=0,1
Define, R(t)=p(t)r(t)
If α denotes the cost of complexity per terminal node, then, define,

$$R_\alpha(T) = \sum_{t \in \tilde{T}} R(t) + \alpha|\tilde{T}|$$

where, $|\tilde{T}|$ is the cardinality of trtminal node set $\tilde{T}$.
In Cost Complexity Pruning, we start from a pure tree $T_0$and for a fixed α we find a subtree T_α such that $R_\alpha(T)$ is minimum.

(ii)<u>Weakest Link Pruning</u>:

Let, T_t denote the subtree at node 't'. Then we define **"Strength of Link"** at node 't' as,-

$$g(t) = \frac{R(T) - R(T_t)}{|\tilde{T}| - 1}$$

We prune the tree at the node t^* for which $g(t^*)$ is minimum.

2.4 Random Forest

Decision trees suffer from high variance problem. To get rid of this high variance problem, we use random forest. In this method, we use the concept of **"Averaging independent observations reduce variance"**. In random forest, we take a bootstrapped sample of size N from the training data and grow a decision tree with these N samples. The only difference is that, in this case, in each split, we take a random sample of total number of predictors, preferable sample size is the square root of the number of predictors. We repeat this process, say, M number of times. For classification setting, we note down the class predicted by each of the 'M' trees and for final prediction, we take the majority of these 'M' outputs.

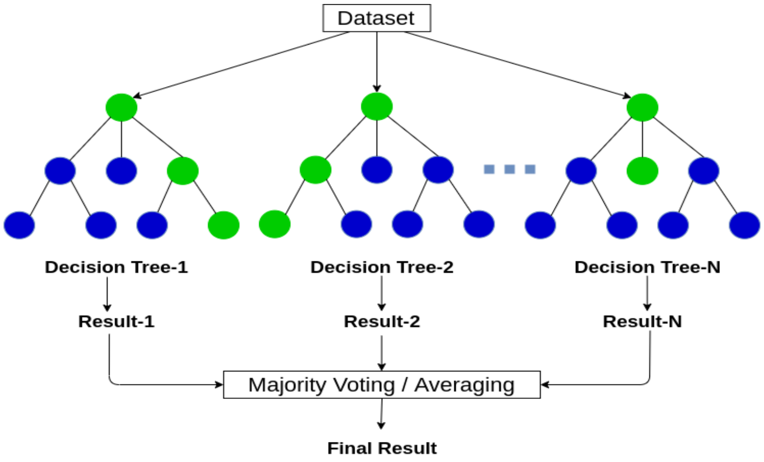

Figure 2: Random Forest Model

The advantage of Random Forest lies in the fact that it **decorrelates** the trees. Suppose there is a strong predictor in our dataset. Then there will be a tendency to use this predictor in each top split. So the trees may look similar. In Random Forest, we are using a random sample of predictors and in some cases, the most important variables may even not be considered. We can think this as decorrelating the trees, thereby making predictions less variable.

3 Application

3.1 PIMA Indian Diabetes Dataset

This dataset is taken from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective here is to predict if a patient has diabetes or not based on diagnostic measurements .

a large number of constraints were placed to select these instances from a larger database. Here, all patients here are females such that their age is at least 21 years .

- Pregnancies: Number of times pregnant.
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
- Blood Pressure: Diastolic blood pressure with unit(mm Hg)
- Skin Thickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)2)
- DiabetesPedigreeFunction: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1)

Here our goal is to predict whether or not a patient has diabetes. First we have checked for **missing values** in the dataset since missing values can cause problems in modeling purpose. In this dataset, there are no missing values. So, we proceed to check whether our dataset is **imbalanced** or not, since imbalanced dataset may often results in a erroneous model. We have seen in our dataset 65% observations have no diabetes and 35% observations have diabetes. So, we have implemented **Synthetic Minority Oversampling Technique(SMOTE)** to make our dataset balanced.

After all the preprocessing with our data, we have split our dataset into 3 parts, namely, **train set, cross validation set, test set**. We have trained logistic regression, decision tree, random forest using train set and checked whether our dataset validates the assumption of lda and qda.

After training the models with our training set, we have applied them on cross validation set and to evaluate the performances of the model, we have looked for **F-Score** and **AUC** of **ROC** curve. We have chosen that model to be the best for which we have obtained the highest F-Score and the highest AUC.

After obtaining the best model, we have applied it on a completely new test dataset and checked for how it performs on test set and for evaluating the performance of the chosen model, we have taken the same evaluation metric F-Score and AUC.

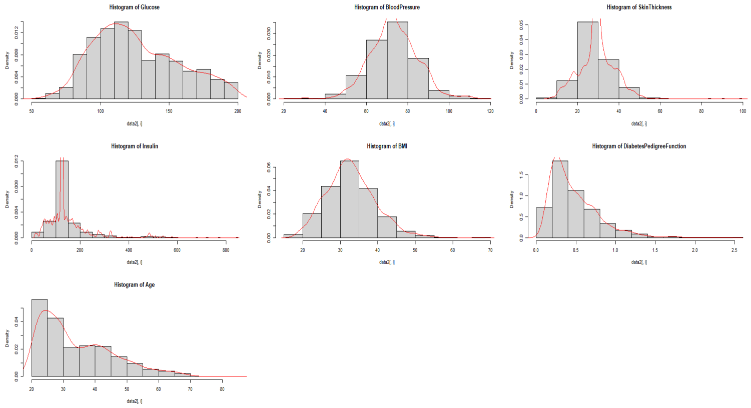

Figure 3: Histogram of Predictors

It is evident from the above figure that the assumption of LDA and QDA has been violated and for this reason, we cannot apply Discriminant Analysis method to this multivariate dataset. However, we applied **Box-Cox Transformation** for applying LDA and QDA. After applying Box-Cox Transformation, we have obtained the following histograms,

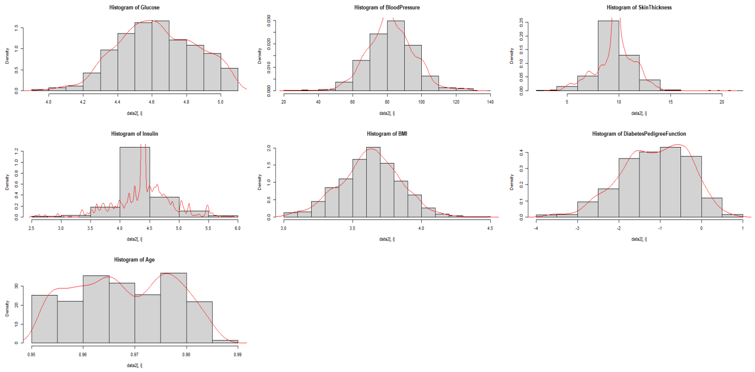

Figure 4: Histogram of Predictors After Box-Cox Transformation

3.1.1 LDA

After training LDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

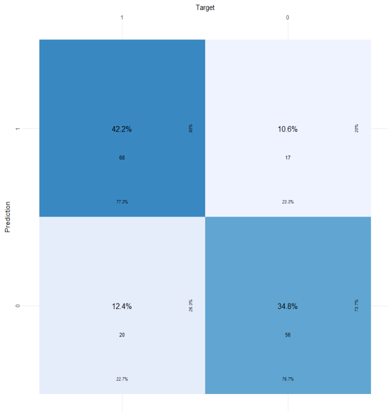

Figure 5: Confusion Matrix of LDA

3.1.2 QDA

After training QDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

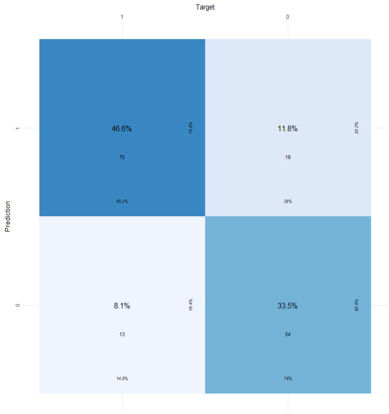

Figure 6: Confusion Matrix of QDA

3.1.3 Logistic Regression

After training logistic regression using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

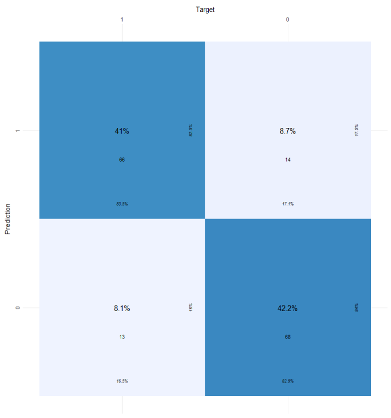

Figure 7: Confusion Matrix of Logistic Regression

3.1.4 Decision Tree

After training decision tree model using training set, we have pruned the tree and obtained the **minimum misclassification error rate** for the tree with 8 terminal nodes. we have applied it on the cross validation dataset and obtained the following confusion matrix,

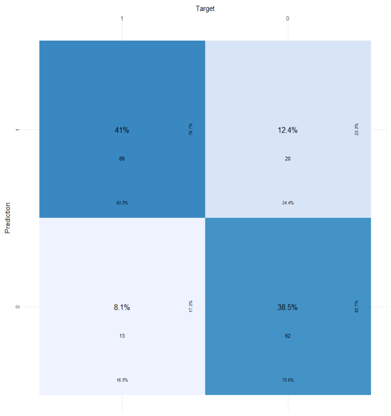

Figure 8: Confusion Matrix of Decision Tree

3.1.5 Random Forest

We have 8 predictors in our dataset and to train random forest model, we have chosen $\sqrt{8} \approx 3$ predictors in each split. After training Random Forest Model using training set, we have applied it on cross validation set and obtained the following confusion matrix,

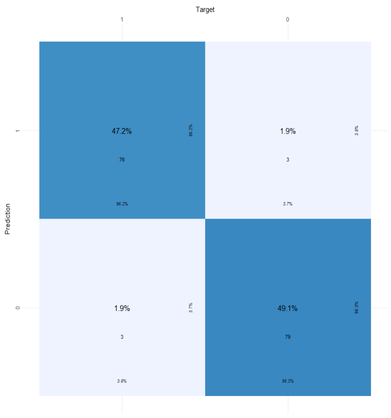

Figure 9: Confusion Matrix of Random Forest

3.1.6 Evaluating Performances of the Models

To evaluate and compare the performances of our models, we have used the evaluation metric **F-Score** and **AUC of the ROC Curve**. The AUC curves are as follows,

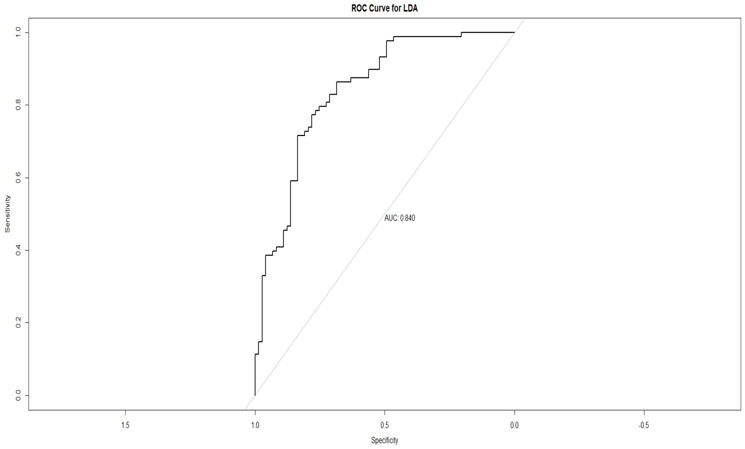

Figure 10: ROC Curve for LDA

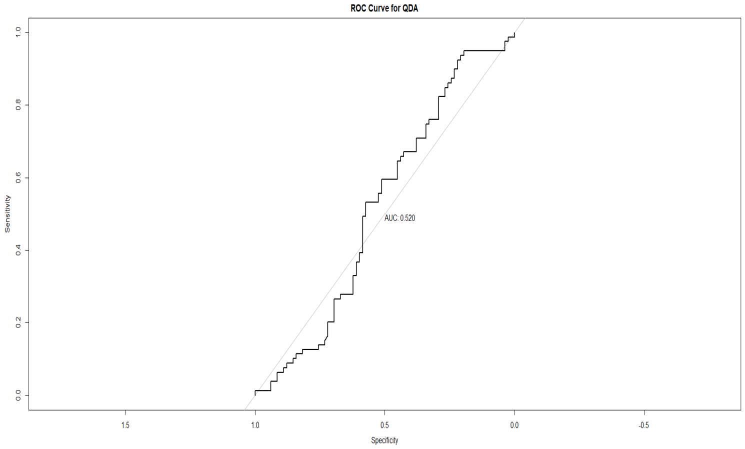

Figure 11: ROC Curve for QDA

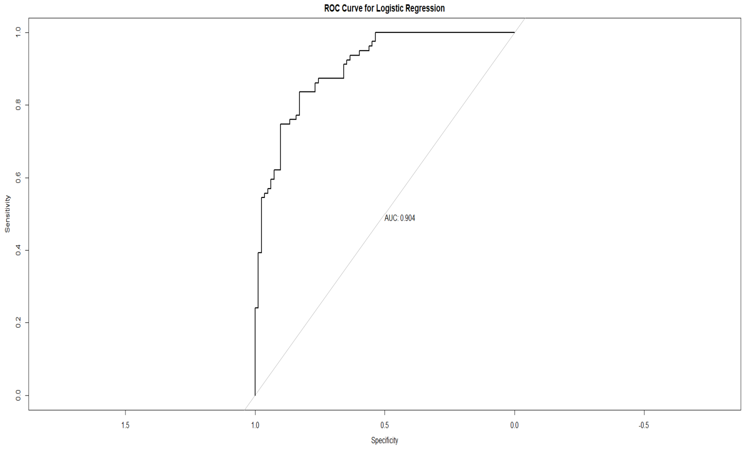

Figure 12: ROC Curve for Logistic Regression

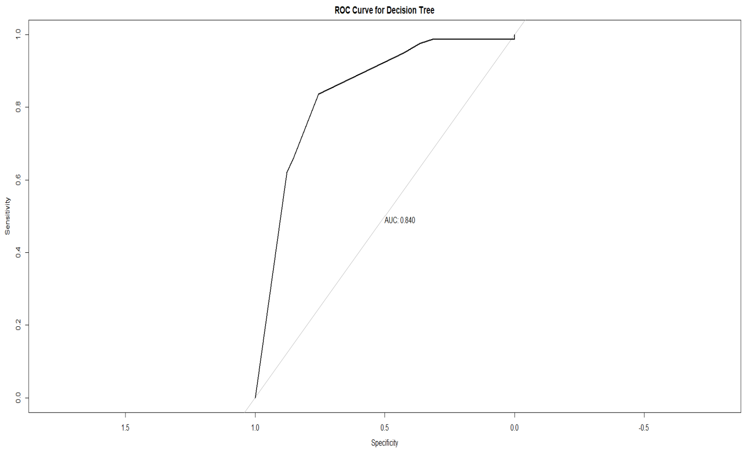

Figure 13: ROC Curve for Decision Tree

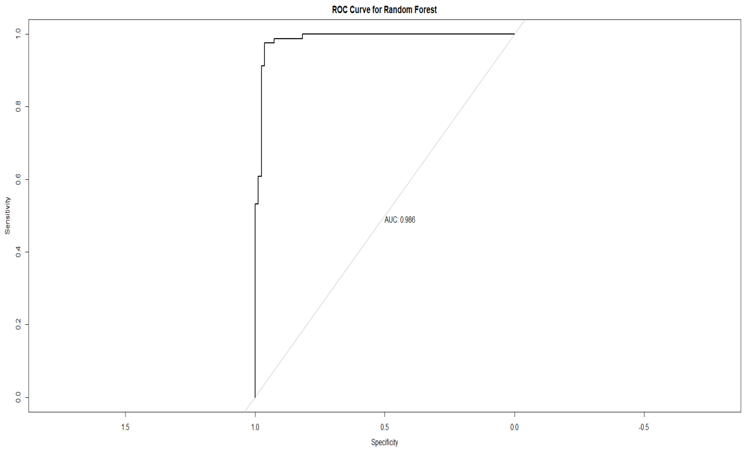

Figure 14: ROC Curve for Random Forest

From the graphs, it is evident that **Random Forest outperforms all the other models in that case**. After that, we have also compared F-Score and AUC to choose the final model.

From the above table, also, we can see that **Random Forest has the highest F-Score and AUC**. So, we have chosen random forest model to predict if a patient has diabetes or not.

	F-Score	AUC	Time (in sec)
Logistic Regression	0.83	0.90	0.20
Decison Tree	0.80	0.84	0.07
Random Forest	0.96	0.99	1.16
LDA	0.79	0.84	0.03
QDA	0.82	0.52	0.03

Table 1: F-Score and AUC for Various Models

3.1.7 Final Performance

After having chosen Random Forest to be the best model, we have applied it on test set and obtained the confusion matrix to visualize the correct classification.

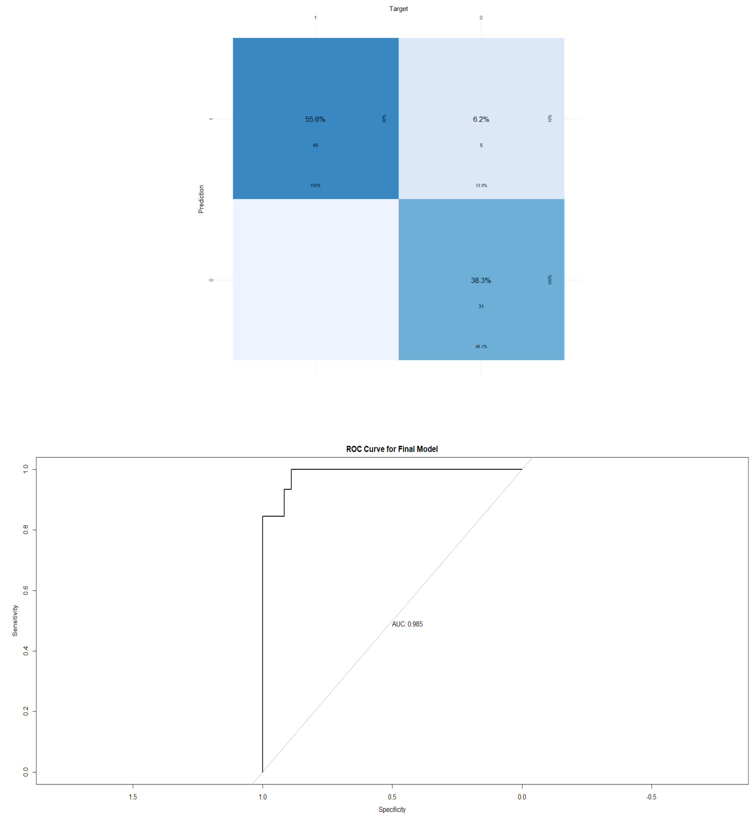

Figure 15: Confusion Matrix and ROC Curve for Final Model

For the final model, we have obtained **F-Score of 0.95** and **AUC of 0.99**. So, we can conclude **it is satisfactory to use random forest model to predict whether a patient has diabetes or not.**

3.2 Caravan Insurance Dataset

This dataset is owned and supplied by the Dutch datamining company Sentient Machine Research, and is based on real world business data. This data set includes 85 predictors that measure demographic characteristics for 5,822 individuals. The response variable is Purchase, which indicates whether or not a given individual purchases a caravan insurance policy. In this data set, only 6 % of people purchased caravan insurance. The variables are as follows,-

- MOSTYPE: Customer Subtype
- MAANTHUI: Number of houses 1 - 10
- MGEMOMV: Avg size household 1 - 6
- MGEMLEEF: Avg age; see L1
- MOSHOOFD: Customer main type; see L2
- MGODRK: Roman catholic
- MGODPR: Protestant ...
- MGODOV: Other religion
- MGODGE: No religion
- MRELGE: Married
- MRELSA: Living together
- MRELOV: Other relation
- MFALLEEN: Singles
- MFGEKIND: Household without children
- MFWEKIND: Household with children
- MOPLHOOG: High level education
- MOPLMIDD: Medium level education
- MOPLLAAG: Lower level education
- MBERHOOG: High status
- MBERZELF: Entrepreneur
- MBERBOER: Farmer
- MBERMIDD: Middle management
- MBERARBG: Skilled labourers

- MBERARBO: Unskilled labourers
- MSKA: Social class A
- MSKB1: Social class B1
- MSKB2: Social class B2
- MSKC: Social class C
- MSKD: Social class D
- MHHUUR: Rented house
- MHKOOP: Home owners
- MAUT1: 1 car
- MAUT2: 2 cars
- MAUT0: No car
- MZFONDS: National Health Service
- MZPART: Private health insurance
- MINKM30: Income <30.000
- MINK3045: Income 30-45.000
- MINK4575: Income 45-75.000
- MINK7512: Income 75-122.000
- MINK123M: Income >123.000
- MINKGEM: Average income
- MKOOPKLA: Purchasing power class
- PWAPART: Contribution private third party insurance
- PWABEDR: Contribution third party insurance (firms) ...
- PWALAND: Contribution third party insurane (agriculture)
- PPERSAUT: Contribution car policies
- PBESAUT: Contribution delivery van policies
- PMOTSCO: Contribution motorcycle/scooter policies
- PVRAAUT: Contribution lorry policies
- PAANHANG: Contribution trailer policies

- PTRACTOR: Contribution tractor policies
- PWERKT: Contribution agricultural machines policies
- PBROM: Contribution moped policies
- PLEVEN: Contribution life insurances
- PPERSONG: Contribution private accident insurance policies
- PGEZONG: Contribution family accidents insurance policies
- PWAOREG: Contribution disability insurance policies
- PBRAND: Contribution fire policies
- PZEILPL: Contribution surfboard policies
- PPLEZIER: Contribution boat policies
- PFIETS: Contribution bicycle policies
- PINBOED: Contribution property insurance policies
- PBYSTAND: Contribution social security insurance policies
- AWAPART: Number of private third party insurance 1 - 12
- AWABEDR: Number of third party insurance (firms) ...
- AWALAND: Number of third party insurance (agriculture)
- APERSAUT: Number of car policies
- ABESAUT: Number of delivery van policies
- AMOTSCO: Number of motorcycle/scooter policies
- AVRAAUT: Number of lorry policies
- AAANHANG: Number of trailer policies
- ATRACTOR: Number of tractor policies
- AWERKT: Number of agricultural machines policies
- ABROM: Number of moped policies
- ALEVEN: Number of life insurances
- APERSONG: Number of private accident insurance policies
- AGEZONG: Number of family accidents insurance policies
- AWAOREG: Number of disability insurance policies

- ABRAND: Number of fire policies
- AZEILPL: Number of surfboard policies
- APLEZIER: Number of boat policies
- AFIETS: Number of bicycle policies
- AINBOED: Number of property insurance policies
- ABYSTAND: Number of social security insurance policies
- CARAVAN: Number of mobile home policies 0 - 1

Here our goal is to predict whether or not a given individual purchases Caravan Insurance Policy. First we have checked for **missing values** in the dataset since missing values can cause problems in modeling purpose. In this dataset, there are no missing values. So, we proceed to check whether our dataset is **imbalanced** or not. Since imbalanced dataset may often results in a erroneous model. Since our dataset is imbalanced, we have implemented **Synthetic Minority Oversampling Technique(SMOTE)** to make our dataset balanced.

After all the preprocessing with our data, we have split our dataset into 3 parts, namely, **train set, cross validation set, test set**. We have trained logistic regression, decision tree, random forest using train set and checked whether our dataset validates the assumption of lda and qda.

After training the models with our training set, we have applied them on cross validation set and to evaluate the performances of the model, we have looked for **F-Score** and **AUC** of **ROC** curve. We have chosen that model to be the best for which we have obtained the highest F-Score and the highest AUC.

After obtaining the best model, we have applied it on a completely new test dataset and checked for how it performs on test set and for evaluating the performance of the chosen model, we have taken the same evaluation metric F-Score and AUC.

To see, whether we can apply Discriminant Analysis method to this dataset, we have plotted histogram of the predictor variables. In each of the plots, we have taken 17 variables and after obtaining the plots, we have checked whether the assumption of normality holds or not.

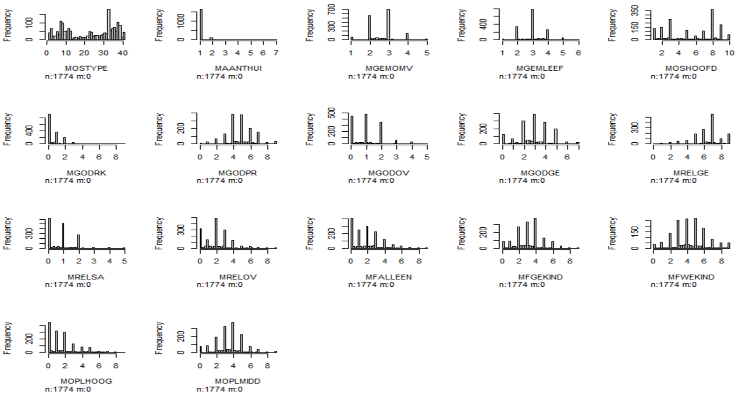

Figure 16: Histogram of Predictors

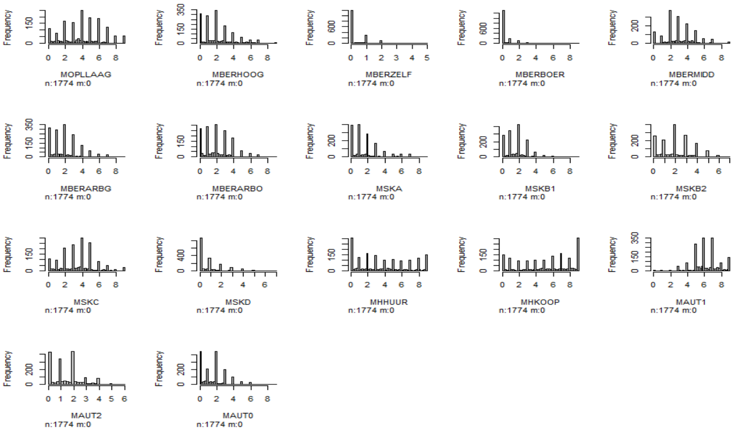

Figure 17: Histogram of Predictors

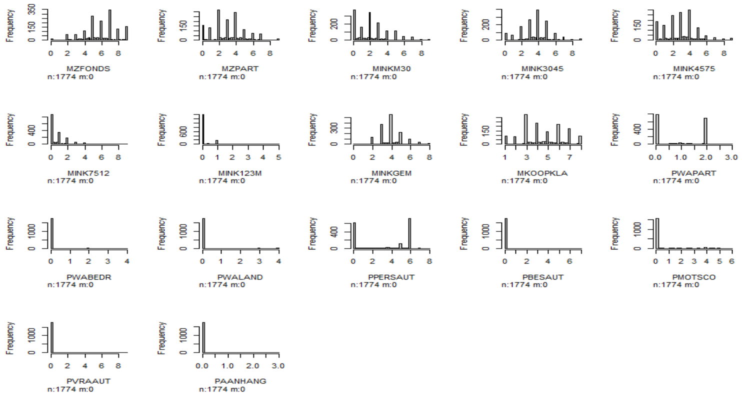

Figure 18: Histogram of Predictors

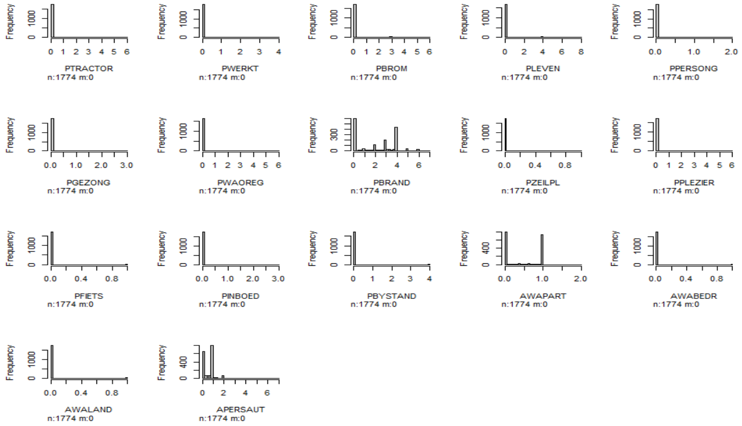

Figure 19: Histogram of Predictors

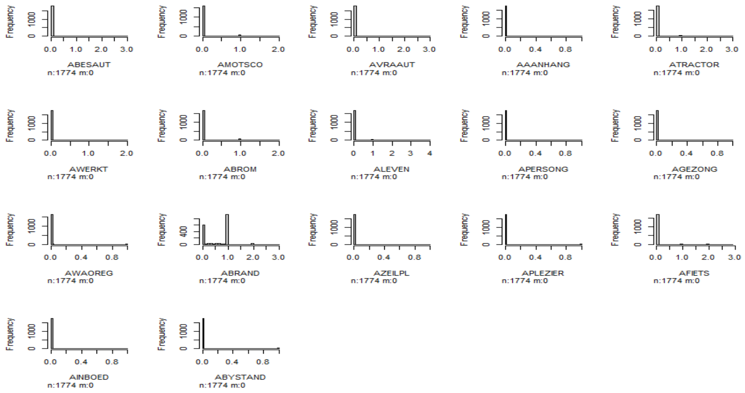

Figure 20: Histogram of Predictors

The condition for applying discriminant analysis method is that all the variables should be continuous and jointly follow a multivariate normal distribution. However, from the plots, we can say normality assumptions have been violated and also the variables are of numeric type, but they take discrete values rather than continuous ones. Moreover, the dataset has discrete features, so, we cannot apply box-cox transformation too. So, we proceed for applying all other methods.

3.2.1 Logistic Regression

After training logistic regression using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

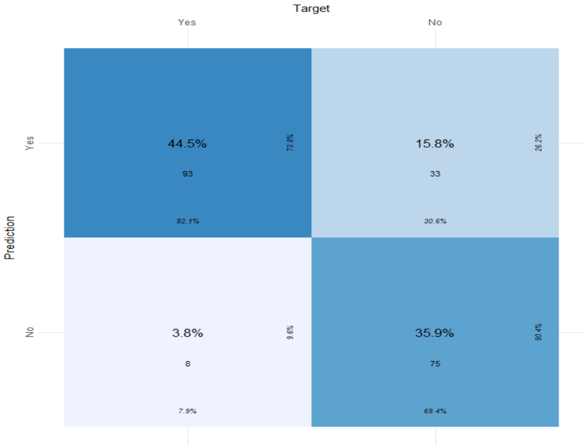

Figure 21: Confusion Matrix of Logistic Regression

3.2.2 Decision Tree

After training decision tree model using training set, we have pruned the tree and obtained the **minimum misclassification error rate** for the tree with 11 terminal nodes. we have applied it on the cross validation dataset and obtained the following confusion matrix,

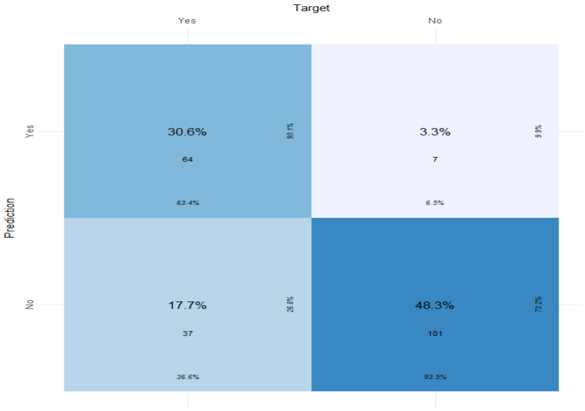

Figure 22: Confusion Matrix of Decision Tree

3.2.3 Random Forest

We have 8 predictors in our dataset and to train random forest model, we have chosen $\sqrt{85} \approx 9$ predictors in each split. After training Random Forest Model using training set, we have applied it on cross validation set and obtained the following confusion matrix,

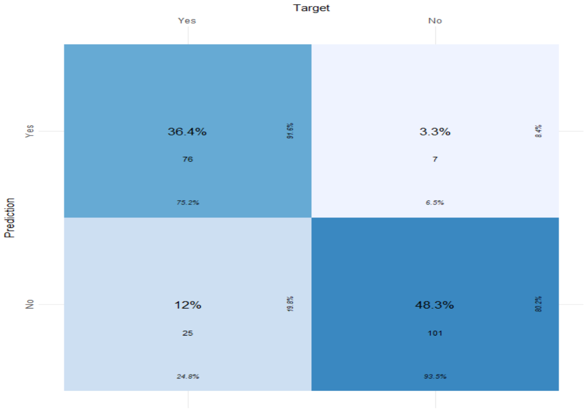

Figure 23: Confusion Matrix of Random Forest

3.2.4 Evaluating Performances of the Models

To evaluate and compare the performances of our models, we have used the evaluation metric **F-Score** and **AUC of the ROC Curve**. The AUC curves are as follows,

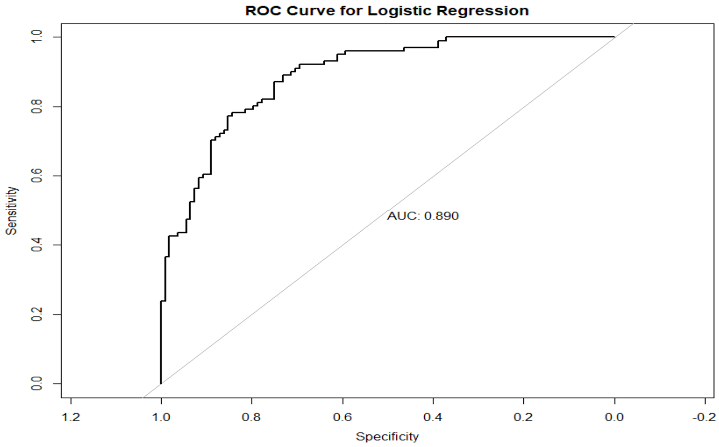

Figure 24: ROC Curve for Logistic Regression

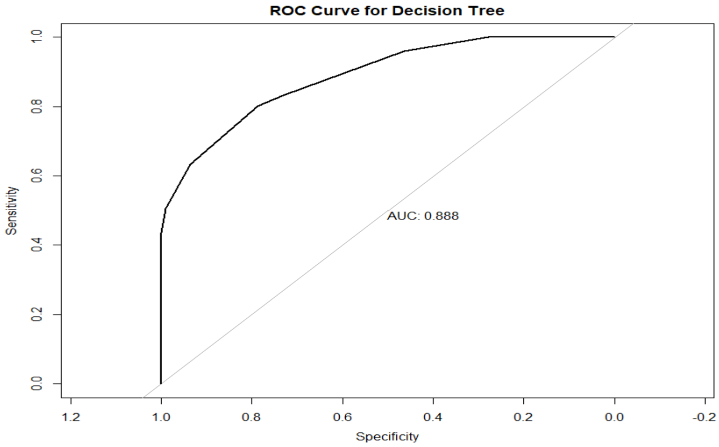

Figure 25: ROC Curve for Decision Tree

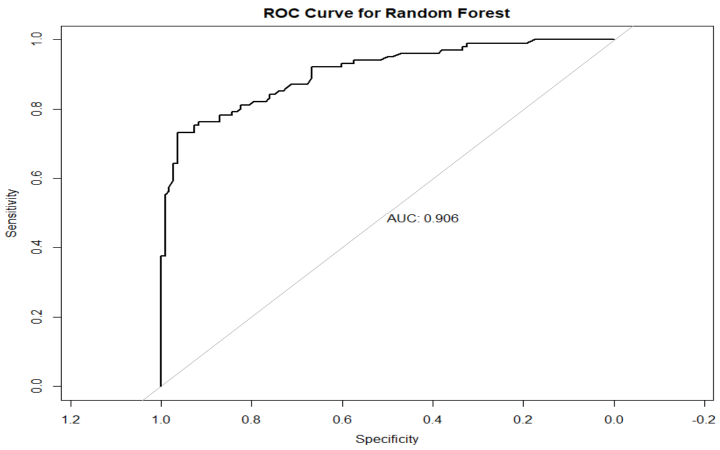

Figure 26: ROC Curve for Random Forest

From the graphs, it is evident that **Random Forest outperforms all the other models in that case**. After that, we have also compared F-Score and AUC to choose the final model.

	F-Score	AUC	Time (in sec)
Logistic Regression	0.81	0.89	0.36
Decison Tree	0.74	0.88	0.14
Random Forest	0.81	0.90	9.28

Table 2: F-Score and AUC for Various Models

From the above table, also, we can see that **Logistic Regression and Random Forest has almost same F-Score and AUC**. However, time taken for Random Forest is much more than that of Logistic Regression. So, we have chosen Logistic Regression model to predict whether a given individual purchases a caravan insurance policy or not.

3.2.5 Final Performance

After having chosen Logistic Regression to be the best model, we have applied it on test set and obtained the confusion matrix to visualize the correct classification. After that, to assess the performance of the model on this completely new dataset, we have obtained F-Score and AUC respectively.

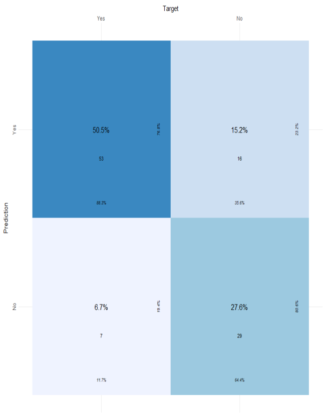

Figure 27: Confusion Matrix of Final Model

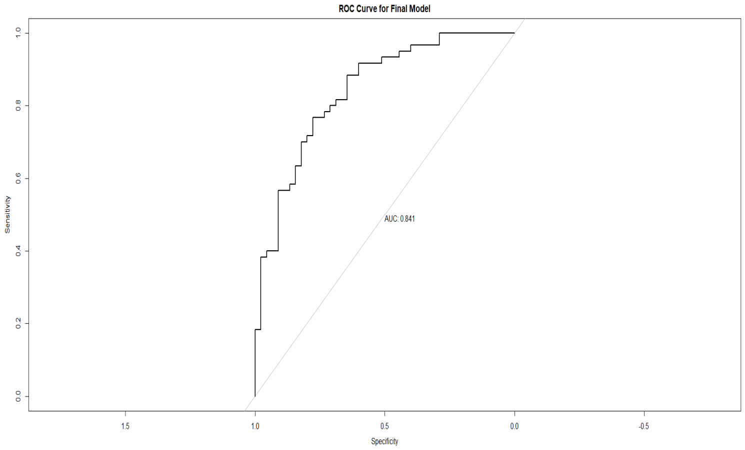

Figure 28: ROC Curve for Final Model

For the final model, we have obtained **F-Score of 0.82** and **AUC of 0.84**. So, we can conclude **it is satisfactory to use Logistic Regression model to predict whether a given individual purchases a caravan insurance policy or not.**

3.3 Banknote Authentication Dataset

In this dataset, data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400× 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images. The variables are as follows,-

- variance: variance of Wavelet Transformed image.
- skewness: skewness of Wavelet Transformed image.
- curtosis: kurtosis of Wavelet Transformed image.
- entropy: entropy of image.
- class: 0 represents real and 1 represents fake banknote.

Here our goal is to predict whether a banknote is fake or real. First we have checked for **missing values** in the dataset since missing values can cause problems in modeling purpose. In this dataset, there are no missing values. So, we proceed to check whether our dataset is **imbalanced** or not. In our dataset, we have 56 % real banknote and 44 % fake banknote. So, in this case, our dataset is balanced. After all the preprocessing with our data, we have split our dataset into 3 parts, namely, **train set, cross validation set, test set**. We have trained logistic regression, decision tree, random forest using train set and checked whether our dataset validates the assumption of lda and qda.

After training the models with our training set, we have applied them on cross validation set and to evaluate the performances of the model, we have looked for **F-Score** and **AUC** of **ROC** curve. We have chosen that model to be the best for which we have obtained the highest F-Score and the highest AUC.

After obtaining the best model, we have applied it on a completely new test dataset and checked for how it performs on test set and for evaluating the performance of the chosen model, we have taken the same evaluation metric F-Score and AUC.

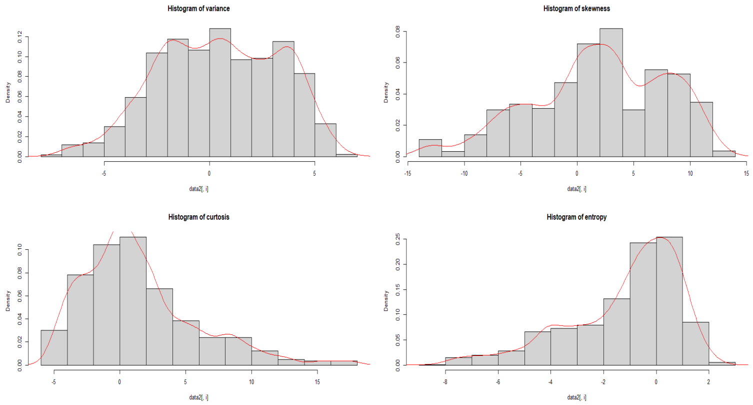

Figure 29: Histogram of Predictors

It is evident from the above figure that the assumption of LDA and QDA has been violated and for this reason, we cannot apply Discriminant Analysis method to this multivariate dataset. However, we applied **Box-Cox Transformation** for applying LDA and QDA. After applying Box-Cox Transformation, we have obtained the following histograms,

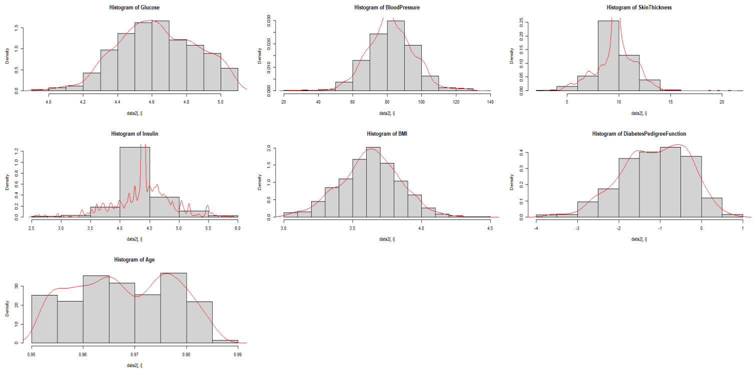

Figure 30: Histogram of Predictors After Box-Cox Transformation

3.3.1 LDA

After training LDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

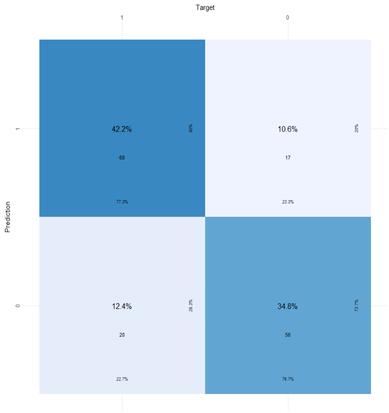

Figure 31: Confusion Matrix of LDA

3.3.2 QDA

After training QDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

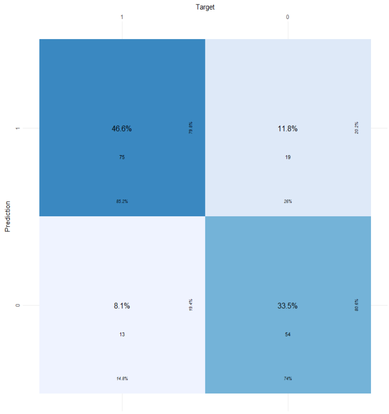

Figure 32: Confusion Matrix of QDA

3.3.3 Logistic Regression

After training logistic regression using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

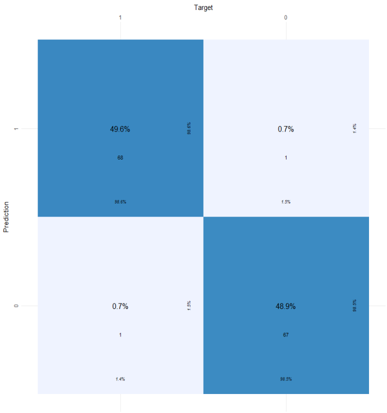

Figure 33: Confusion Matrix of Logistic Regression

3.3.4 Decision Tree

After training decision tree model using training set, we have pruned the tree and obtained the **minimum misclassification error rate** for the tree with 11 terminal nodes. we have applied it on the cross validation dataset and obtained the following confusion matrix,

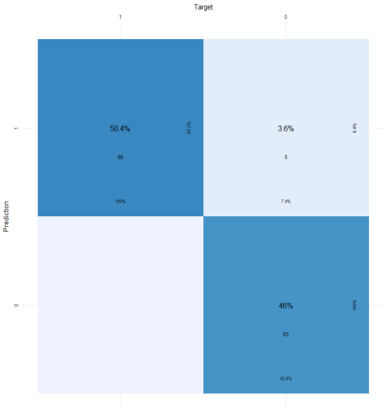

Figure 34: Confusion Matrix of Decision Tree

3.3.5 Random Forest

We have 8 predictors in our dataset and to train random forest model, we have chosen $\sqrt{4} = 2$ predictors in each split. After training Random Forest Model using training set, we have applied it on cross validation set and obtained the following confusion matrix,

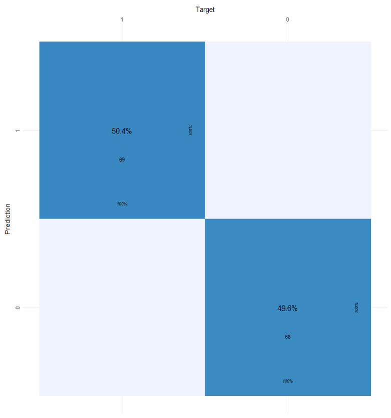

Figure 35: Confusion Matrix of Random Forest

3.3.6 Evaluating Performances of the Models

To evaluate and compare the performances of our models, we have used the evaluation metric **F-Score** and **AUC of the ROC Curve**. The AUC curves are as follows,

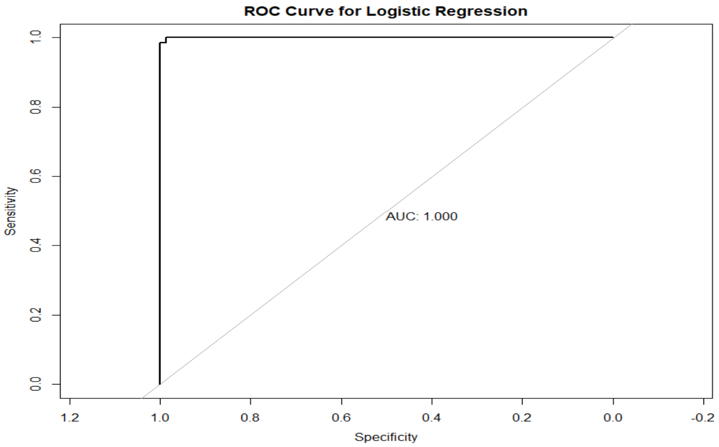

Figure 36: ROC Curve for Logistic Regression

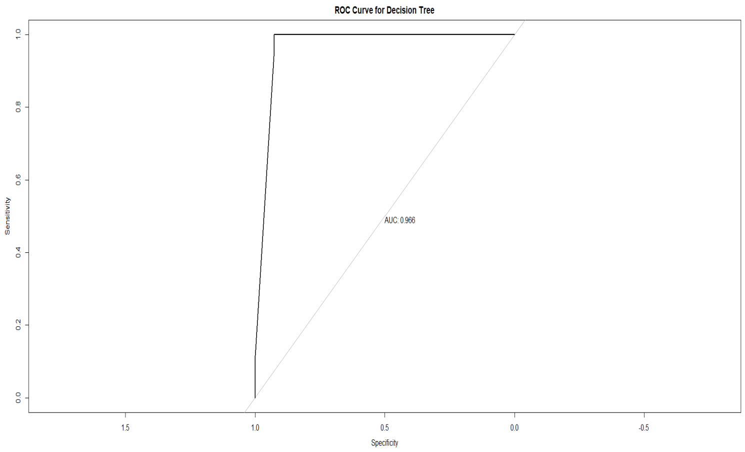

Figure 37: ROC Curve for Decision Tree

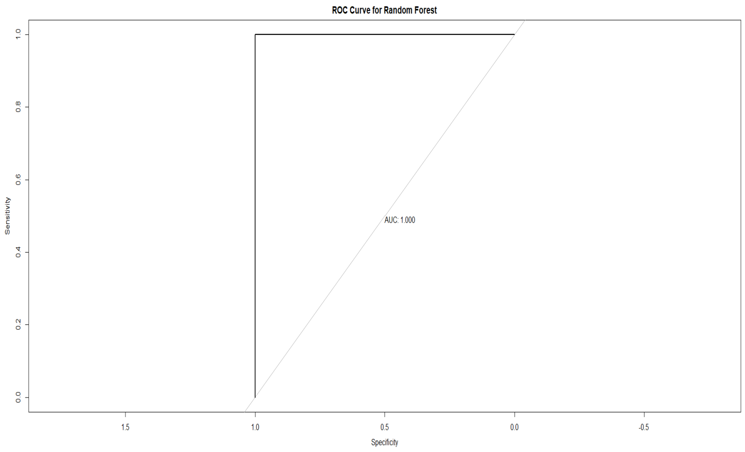

Figure 38: ROC Curve for Random Forest

From the graphs, it is evident that **Random Forest outperforms all the other models in that case**. After that, we have also compared F-Score and AUC to choose the final model.

	F-Score	AUC	Time(in sec)
Logistic Regression	0.9855	0.99979	0.04
Decison Tree	0.9650	0.96590	0.08
Random Forest	1.0000	1.00000	0.44
LDA	0.9700	1.0000	0.01
QDA	0.9900	0.5100	0.09

Table 3: F-Score and AUC for Various Models

From the above table, also, we can see that **Random Forest has the highest F-Score and AUC**. So, we have chosen random forest model to predict whether a given banknote is fake or real.

3.3.7 Final Performance

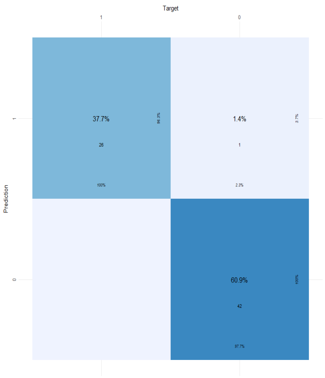

Figure 39: Confusion Matrix of Final Model

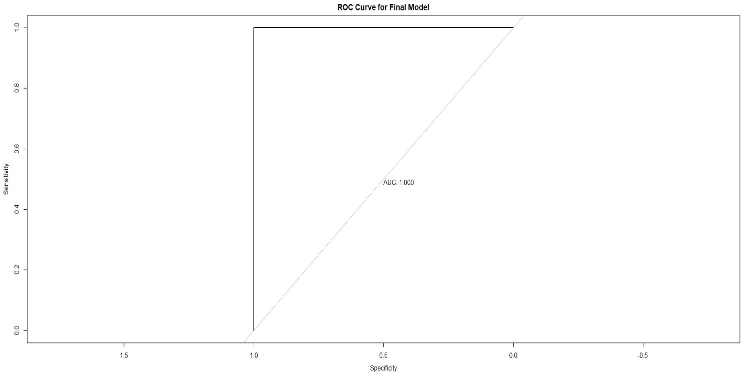

Figure 40: ROC Curve for Final Model

For the final model, we have obtained **F-Score of 0.98** and **AUC of 1**. So, we can conclude **it is satisfactory to use random forest model to predict whether a given banknote is fake or real.**

3.4 Weekly Dataset

In this dataset, weekly percentage returns for the S and P 500 stock index between 1990 and 2010 are given. The variables are as follows,

- Year: the year that the observation was recorded.
- Lag1: percentage return for previous week.
- Lag2: percentage return for two weeks previous.
- Lag3: percentage return for three weeks previous.
- Lag4: percentage return for four weeks previous.
- Lag5: percentage return for five weeks previous.
- Volume: Volume of shares traded (average number of daily shares traded in billions).
- Today: Percentage return for this week.
- Direction: A factor with levels Down and Up indicating whether the market had a positive or negative return on a given week.

Here our goal is to predict whether a market will have a positive or negative return on a given week. First we have checked for **missing values** in the dataset since missing values can cause problems in modeling purpose. In this dataset, there are no missing values. So, we proceed to check whether our dataset is **imbalanced** or not. In our dataset, we have 56 % positive return and 44 % fake negative return. So, in this case, our dataset is balanced. After all the preprocessing with our data, we have split our dataset into 3 parts, namely, **train set, cross validation set, test set**. We have trained logistic regression, decision tree, random forest using train set and checked whether our dataset validates the assumption of lda and qda.

After training the models with our training set, we have applied them on cross validation set and to evaluate the performances of the model, we have looked for **F-Score** and **AUC** of **ROC** curve. We have chosen that model to be the best for which we have obtained the highest F-Score and the highest AUC.

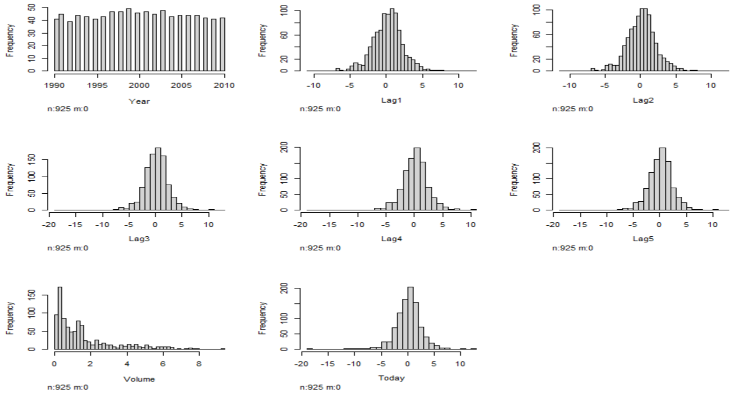

Figure 41: Histogram of Predictors

We will take Lag1, Lag2, Lag3, Lag4, Lag5, Today for applying LDA and QDA, since the histograms of these predictors seems symmetric, so we asume normality in these cases.

3.4.1 LDA

After training LDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

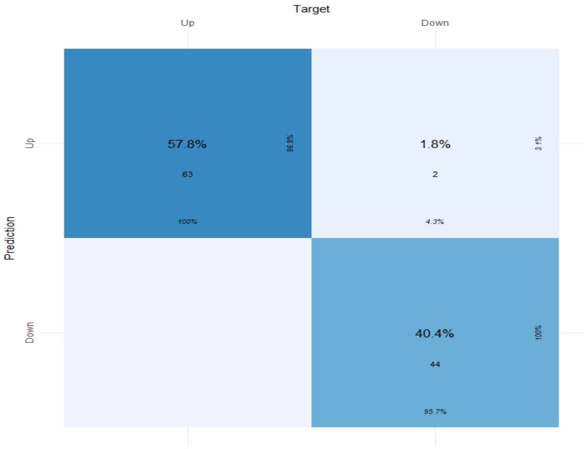

Figure 42: Confusion Matrix of LDA

3.4.2 QDA

After training QDA using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

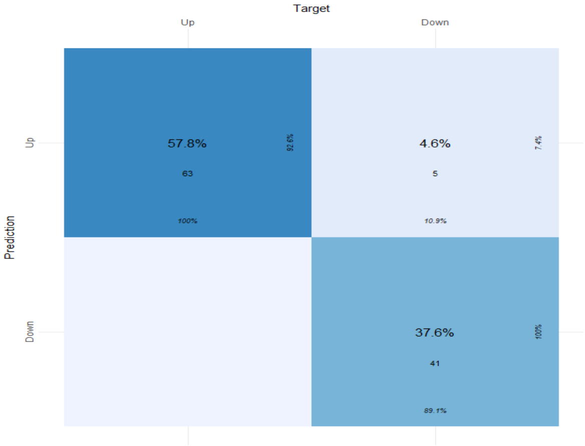

Figure 43: Confusion Matrix of QDA

3.4.3 Logistic Regression

After training logistic regression using training dataset, we have applied it on the cross validation dataset and obtained the following confusion matrix,

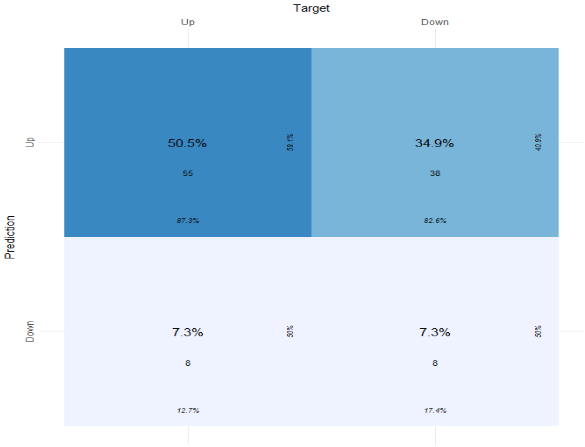

Figure 44: Confusion Matrix of Logistic Regression

3.4.4 Decision Tree

After training decision tree model using training set, we have pruned the tree and obtained the **minimum misclassification error rate** for the tree with 2 terminal nodes. we have applied it on the cross validation dataset and obtained the following confusion matrix,

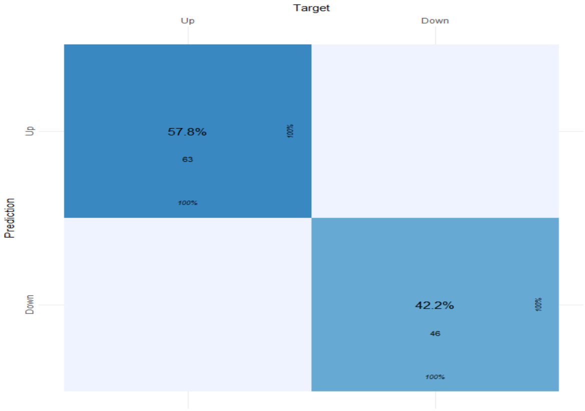

Figure 45: Confusion Matrix of Decision Tree

3.4.5 Random Forest

We have 7 predictors in our dataset and to train random forest model, we have chosen $\sqrt{7} \approx 2$ predictors in each split. After training Random Forest Model using training set, we have applied it on cross validation set and obtained the following confusion matrix,

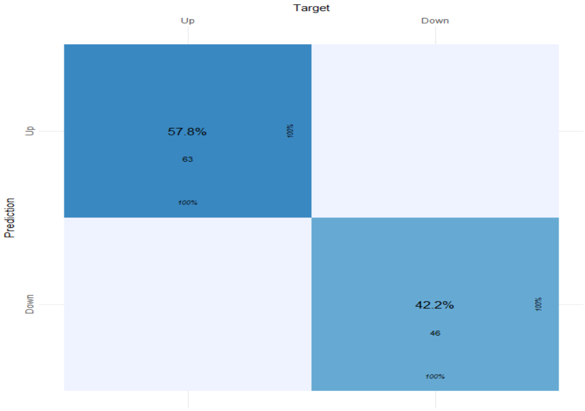

Figure 46: Confusion Matrix of Random Forest

3.4.6 Evaluating Performances of the Models

To evaluate and compare the performances of our models, we have used the evaluation metric **F-Score** and **AUC of the ROC Curve**. The AUC curves are as follows,

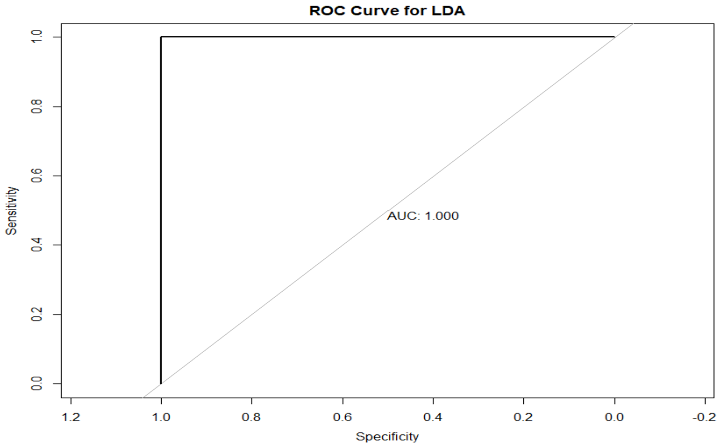

Figure 47: ROC Curve for LDA

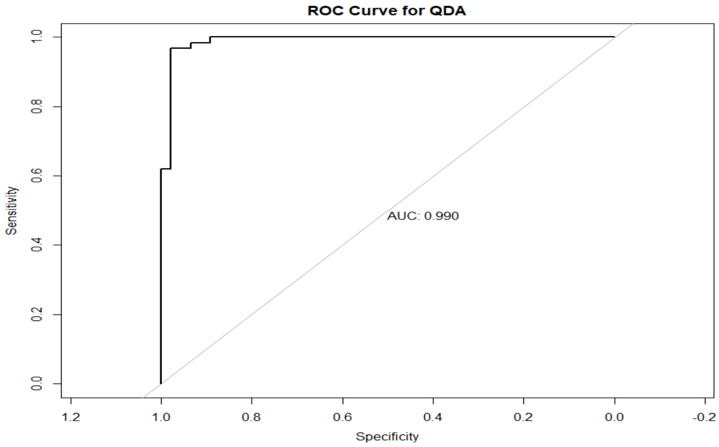

Figure 48: ROC Curve for QDA

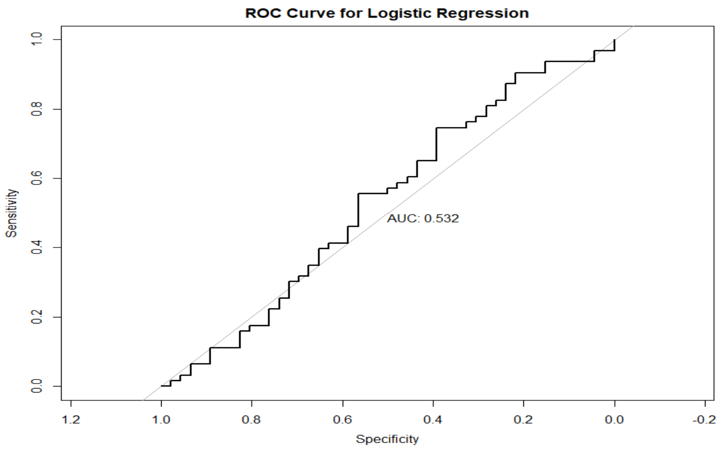

Figure 49: ROC Curve for Logistic Regression

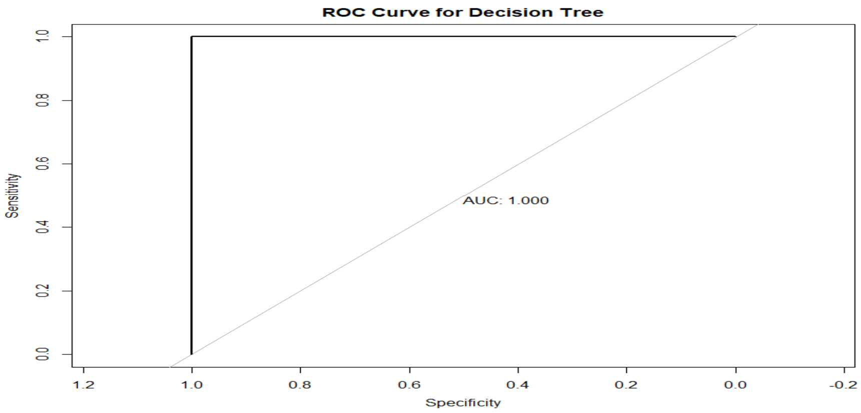

Figure 50: ROC Curve for Decision Tree

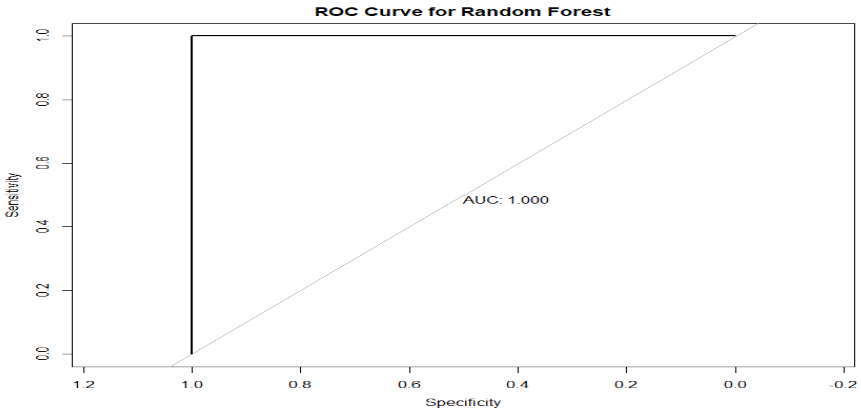

Figure 51: ROC Curve for Random Forest

From the graphs we can say, we can use either Decision Tree or Random Forest as our final model. But to choose only one model, we have seen the time complexity of each model also.

	F-Score	AUC	Time (in Sec)
Logistic Regression	0.71	0.53	0.02
LDA	0.98	1.00	0.04
QDA	0.96	0.99	0.05
Decison Tree	1.00	1.00	0.05
Random Forest	1.00	1.00	0.46

Table 4: F-Score and AUC for Various Models

From the above table, we can choose either Decision Tree or Random Forest either. But, decision trees are more time efficient than random forest. So, we choose Decision Tree to predict whether a market will have a positive or negative return on a given week.

3.4.7 Final Performance

After having chosen Decision Tree to be the best model, we have applied it on test set and obtained the confusion matrix to visualize the correct classification. After that, to assess the performance of the model on this completely new dataset, we have obtained F-Score and AUC respectively.

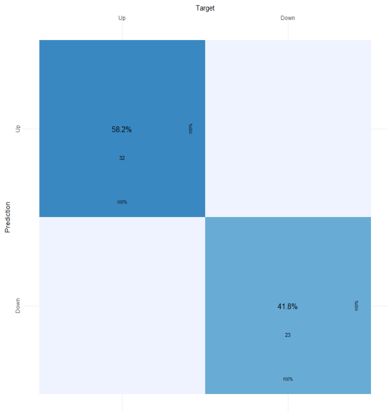

Figure 52: Confusion Matrix of Final Model

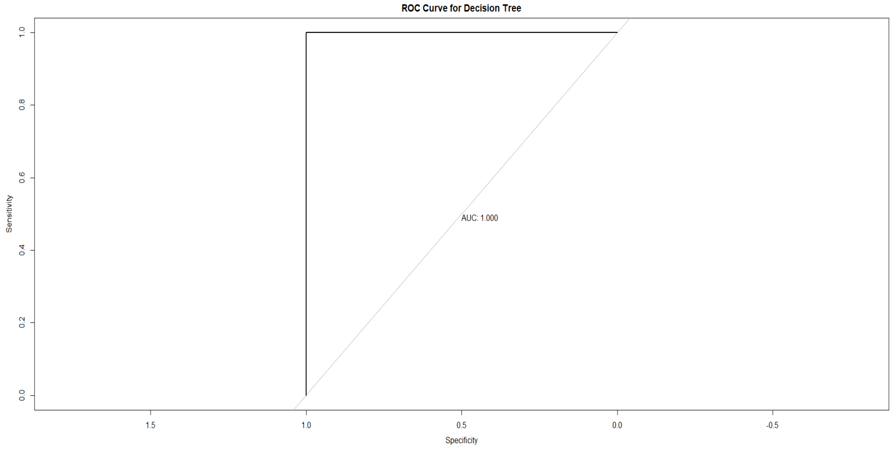

Figure 53: ROC Curve for Final Model

For the final model, we have obtained **F-Score of 1** and **AUC of 1**. So, we can conclude **it is satisfactory to use Decision Tree model to predict whether a market will have a positive or negative return on a given week.**

4 Conclusion

The main findings of the project are as follows,

1. PIMA Indian Diabetes Dataset: For this dataset, Random Forest outperforms all the other models. F-Score and AUC for this model is close to 1 and the model is fitting the test dataset also very well. So, we can be assured that it is not overfitting.

2. Caravan Insurance Dataset: For this dataset, F-Score for Random Forest and Logistic Regression are coming out to be same, but, time efficiency for Logistic Regression is slightly better than that of Random Forest and for that reason we have chosen Logistic to be the best model for this dataset.

3. Banknote Authentication Dataset: For this dataset, Random Forest outperforms all the other models. F-Score and AUC for this model is 1 and the model is fitting the test dataset also very well. So, we can be assured that it is not overfitting.

4. Weekly Dataset: For this dataset, both the Decision Tree and Random Forest models have F-Score and AUC of 1. But, since, Random Forest is time inefficient than Decision Tree, so we have chosen Decision Tree as our final model.

5 Appendix

5.1 R Codes

The R Codes for respective datasets can be found here.

5.2 Synthetic Minority Oversampling Technique (SMOTE)

A problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary.

One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.

An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space. This procedure can be used to create as many synthetic examples for the minority class as are required.

5.3 F-Score

To define **F-Score**, we have to define **Precision** and **Recall** first.

$$\text{Precision (P)} = \frac{\text{True Positive}}{\text{Total Number of Predicted Positive}}$$

$$\text{Recall (R)} = \frac{\text{True Positive}}{\text{Total Number of Actual Positive}}$$

Then, F-Score is given by,-

$$\text{F-Score} = \frac{\text{2PR}}{\text{P+R}}$$

The higher the F-Score of a model, the greater the accuracy is.

5.4 ROC-AUC Curve

ROC (Receiver Operating Characteristics) Curve is a curve of **Sensitivity vs Specificity**, where,

$$\text{Specificity} = \frac{\text{True Negative}}{\text{Total Number of Actual Negative}}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{Total Number of Actual Positive}}$$

The area under this curve is known as AUC (Area Under the Curve). The higher the AUC of an ROC curve, the better the model is.

5.5 Box-Cox Transformation

A Box Cox transformation is used to transform non-normal dependent variables into a normal ones. Since, Normality is one of the important assumptions for many statistical techniques; if our data isn't normal, applying a Box-Cox means that we are able to run a broader number of learning algorithms. Consider the transformation as follows,-

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda \dot{y}^{\lambda}-1} & ,\lambda \neq 0 \\ \text{ln y} & ,\lambda = 0 \end{cases}$$

where, $\dot{y} = \frac{1}{\ln\left(\frac{1}{n}\sum_{i=1}^{n}\ln y_i\right)}$

6 Data Source

1. **PIMA Indian Diabetes Dataset:** We have collected this dataset from Kaggle and the data can be found here.

2. **Caravan Insurance Dataset:** This dataset is a part of the **ISLR** library in R.

3. **Banknote Authentication Dataset:** We have collected this dataset from Kaggle and the data can be found here.

4. **Weekly Dataset:** This dataset is a part of the **ISLR** library in R.

7 References

1. James, G., Witten, D., Hastie, T., Tibshirani, R., An Introduction to Statistical Learning- with Applications in R, Springer.

2. `https://towardsdatascience.com/`

3. `https://www.analyticsvidhya.com/`

4. `https://stackoverflow.com/`

8 Acknowledgement

We would like to express our deeply felt gratitude towards **Dr. Minerva Mukhopadhyay**, whose continuous support and exemplary guidance has helped us in the fulfillment of this project and her valuable suggestions helped us to make the project fruitful.