**REVISITING THE FRAMINGHAM HEART STUDY: REGRESSION AND CLASSIFICATION BASED APPROACHES**

Name – Udvas Das

Roll No. - 457

Reg. No. - A01-1112-0809-18

St. Xavier’s College, Kolkata (Autonomous)

Supervisor – Prof. Surabhi Dasgupta

**DECLARATION**

“I affirm that I have identified all my sources and that no part of my dissertation paper uses unacknowledged materials.”

Signed,

C:\Users\dell\Desktop\OFFICIAL DOCUMENTS\Udvas Sign.jpg

**CONTENTS**

* Introduction
* Abstract
* Data description and visualization
* Objective and Methodology
* Analysis and Discussion
* Summary and Comparison of accuracy
* Conclusion
* References
* Acknowledgement
* Appendix (R-code)

**INTRODUCTION**

Suppose, we have a data in hand with the information outcome column corresponding to an experiment A and we are to make a statement about the outcome of some other experiment B. This is called ‘**Statistical prediction**’. For example, if I have the data on lung cancer status (As outcome) for a number of people with sufficient information about other corresponding covariates, then by statistical prediction we can predict a person’s possible outcome or the risk of having lung cancer beforehand, given that data on covariates is available for the concerned individual.

Statistical prediction is indeed a vast area if considered. There are many methods which can be applied based on the data for predictive purposes. The most simple and popular approach is the theory of regression. In regression, there is a dependent variable (outcome) and independent variables (covariates or predictors). We mainly focus on establishing a relationship, a mathematical equation which would describe the outcome variable with respect to the predictor variables. This relation can be used for predicting the outcome for any other experimental unit or individual of that sort in future.

There are various types of models based on the data and also the nature of the outcome or dependent variable. For example, we can consider Cox regression to estimate the survival time of a patient, there is ordinal logistic and binary logistic model if the outcome is respectively ordinal and binary in nature, we have Poisson regression model when the dependent variable is non-negative whole numbers and follows a Poisson distribution etc.

We can also use various predictive models or classification models based on neural networks and decision trees like Random Forest model, Naïve Bayes model, KNN etc. Based on the nature of the data, we can use these predictive models alongside trivial statistical models because they give high prediction accuracy and perform well for large datasets.

**ABSTRACT**

Life expectancy of a human being is decreasing day by day in this era of pollution and diseases. A majority of people are suffering from various diseases like heart problems, diabetes, hypertension etc and the numbers are growing incessantly. One cannot predict if he or she will suffer from one of these diseases because most of them do not have any prevalent symptoms. So, it has been an important aspect of clinical studies to predict a person’s risk of having a disease in future. Statisticians have developed many methods, models etc to serve the purpose and the prediction accuracies are getting better as the days are passing. This project work deals with the same idea that which model gives better accuracy in predicting risk of contracting a disease. Particularly, we will focus on a comparative analysis of binary logistic regression model and Random Forest predictive model. Now there are particular reasons for choosing these two models.

I have chosen binary logistic regression model for my data because,

* The dependent variable is binary in nature and the data is sufficiently large.
* Regression coefficients other results are easy to interpret and it is easy to implement and efficient to train.
* It does not require any assumptions on the distribution and nature of predictors.
* It provides extent of effectiveness along with direction of it for a predictor.

The reasons behind selecting Random Forest predictive model are,

* It classifies very well in case of large data.
* It introduces additional randomness in the model and it avoids over fitting using random bootstrap sampling and bagging method.
* It hardly needs any assumptions and gives accurate results using decision trees.

**DATA DESCRIPTION AND VISUALIZATION**

Coronary heart disease is a cardiovascular disease which happens if arteries in our body become narrower and harder than usual causing difficulty in blood and oxygen circulation to the heart. This may cause cardiac arrest and death to a patient. In USA, every year many people die from heart diseases like CHD.In this project work, the data on Framingham heart study is considered. It is a very popular data and many works have been done with it in the past. This is a secondary data. The study started in 1948 with an adult cohort of size 5209 individuals from the city of **Framingham, Massachusetts, USA**. This is a long-term ongoing cardiovascular study and I have taken the data on the 4th generation of participants.

There is a binary dependent or outcome variable and 15 predictors or independent variables or covariates. The descriptions of these variables or columns of the data are given by,

* **TenYearCHD**: This binary outcome variable gives the status of an individual is he or she develops coronary heart disease in 10 years time span or not (0 denotes the presence of CHD and 1 denotes the absence).

Framingham heart study datasets involves some covariates describing demographical risk factors like,

* **Sex**: This binary covariate gives information about the sex of the individuals. (0 denotes Female and 1 denotes Male).

From the diagram below, we can visualize that male subjects are far more prone to develop CHD in the 10 year follow up study than female subjects.

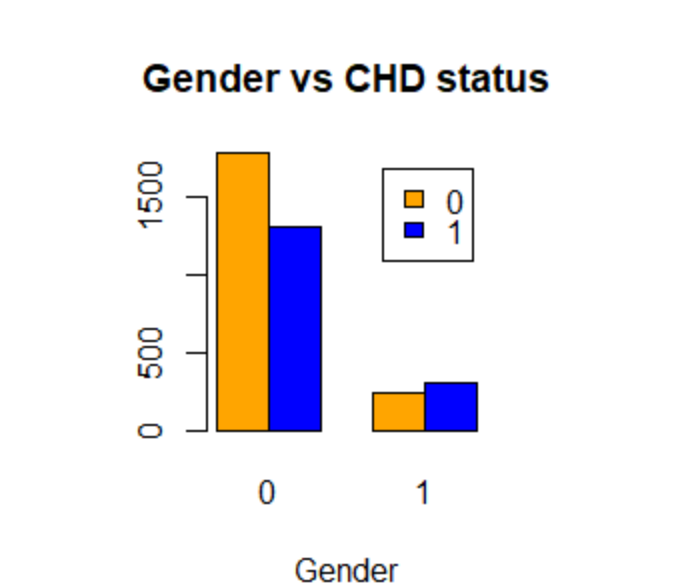


Fig.1

* **Age**: It is a continuous independent variable giving the age of an individual in years.

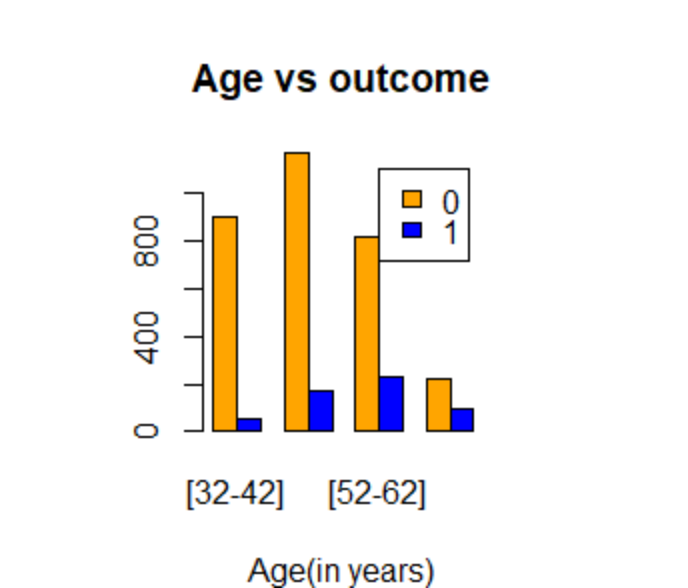


Fig.2

From the bar-plot above we can say that people in the age group 32-42 are less prone to develop CHD than the other age groups.

* **Education**: Gives us information about educational qualification of an individual. It is ordinal is nature. The levels are 1,2,3,4 respectively denoting high school, high school diploma or GED, college or vocational school and college degree.

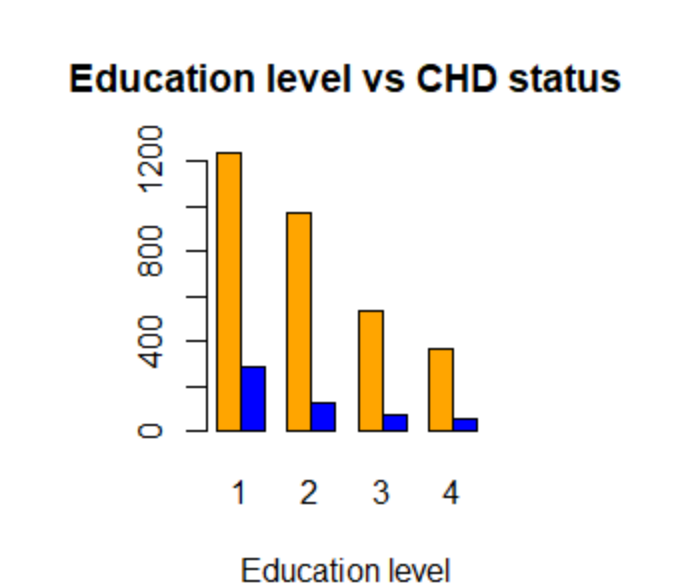


Fig.3

In the dataset, majority of the subjects have primary level of education. But from the diagram we cannot draw any conclusive intuition because in each case the proportion of CHD positive and negative case seems to be nearly equal.

Some behavioral risks are also involved in the data. They are,

* **currentSmoker**: Binary covariate which contains information whether an individual is a smoker or not ( 0 denotes that the individual is a non smoker and 1 means he or she is a smoker).

From the multiple bar-plot of smoking status and CHD status, we can visualize that those subjects who smoke, are more prone to develop CHD in 10 years than the ones who do not smoke.

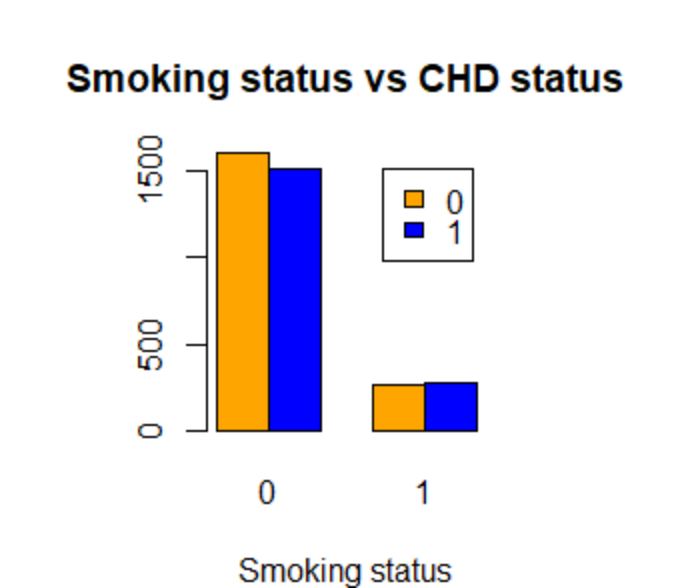


Fig.4

* **cigsPerDay**: Continuous independent variable giving us the number of cigarettes that the person smokes on a daily average.

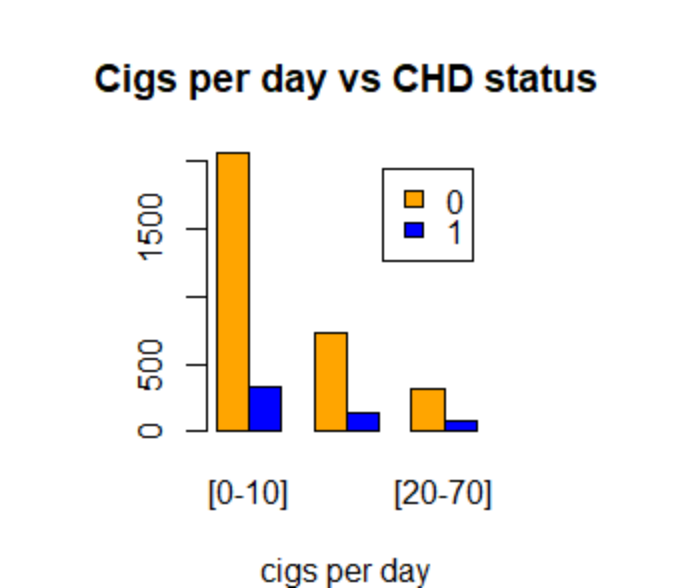


Fig.5

There are other dependent variables as well or we can also name them as medical history risk factors,

* **BPMeds**: It is a binary predictor variable giving information about BP medicine intake of an individual. It has two values 0 and 1. 0 is assigned to the individual who do not take medicines and 1 to those who take medicines for blood pressure.

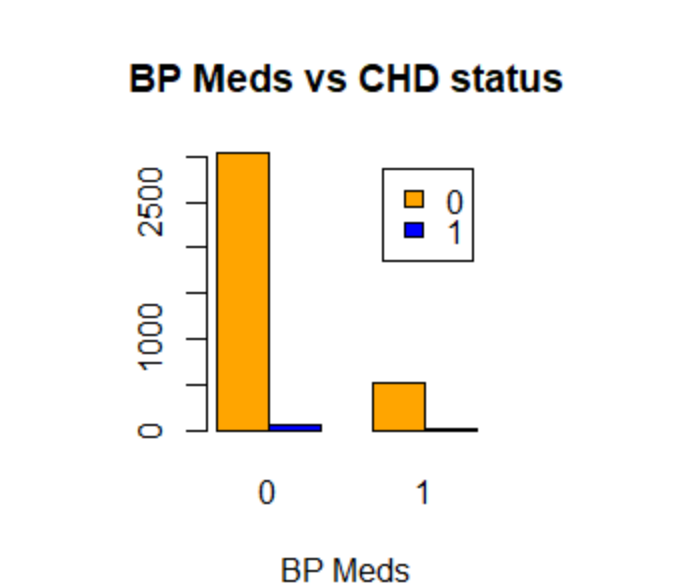


Fig.6

* **prevalentStroke**: Binary independent variable telling us whether or not the patient had previously had a stroke. The value 0 is assigned to those who don’t have any history to stroke and 1 to those who have suffered from a cardiac arrest in the past.

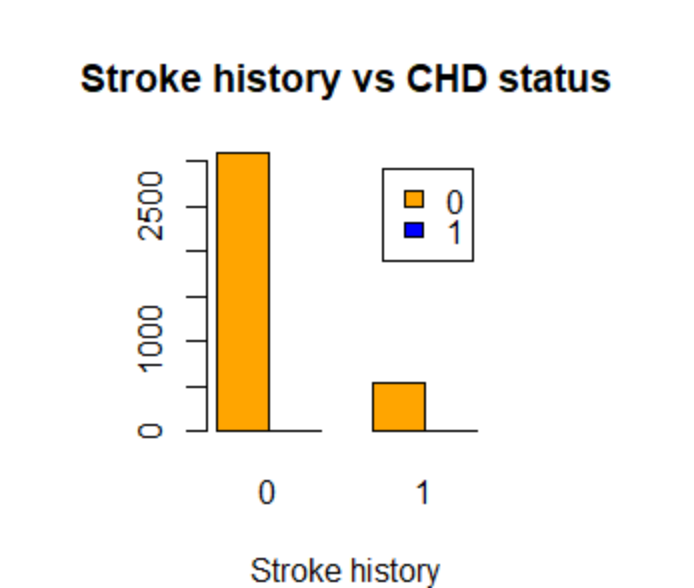


Fig.7

* **prevalentHyp**: It is also a binary variable having the values 0 and 1. It tells us if a person has a history of hypertension or not. 0 denotes that the individual doesn’t have any record of hypertension and 1 denotes the contrary.

We can notice in the diagram below that, people with a history of hypertension are far more prone to develop CHD in future than those who do not have any history with hypertension.

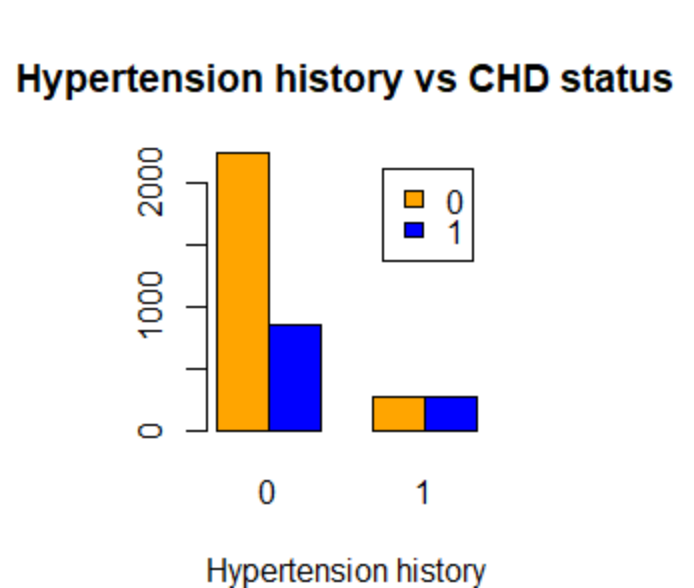
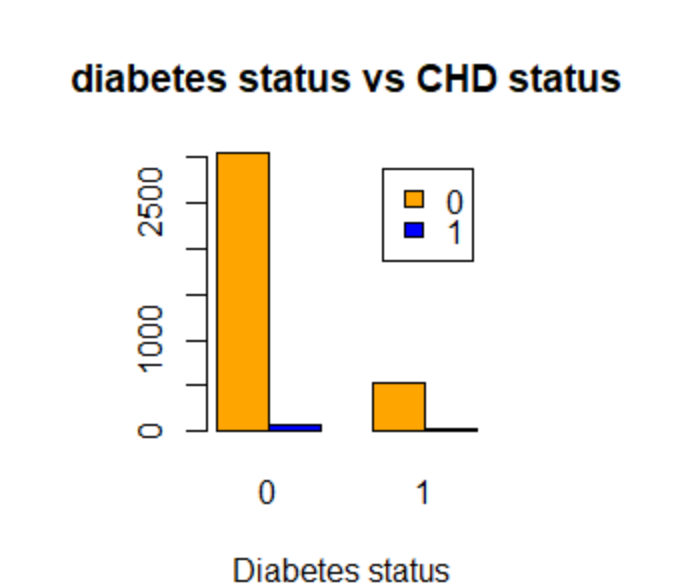


Fig.8

* **diabetes**: This is also binary in nature and has two indicators 0 and 1. The 0 value indicates that the patient is diabetic and 1 denotes that the patient does not have any history of diabetes.

Fig.9

All the participants or subjects in this study are physically examined at first when the data is being collected. So, Framingham heart disease data involves some risk factors based on this physical examination. These are expressed by some continuous predictor variables like,

* **totChol**: Gives the total cholesterol level of a patient in mg/dl.

Majority of the subjects in this data have cholesterol level greater than 200 mg/dl. But, we cannot surely say anything intuitively from bar plot about how the outcome depends on the cholesterol level of a patient.

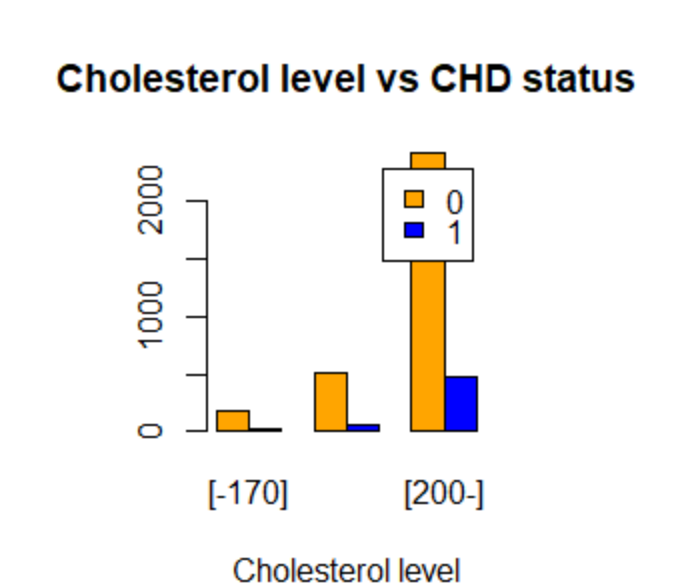


Fig.10

* **sysBP**: Gives the total systolic blood pressure of a individual in mmHg.

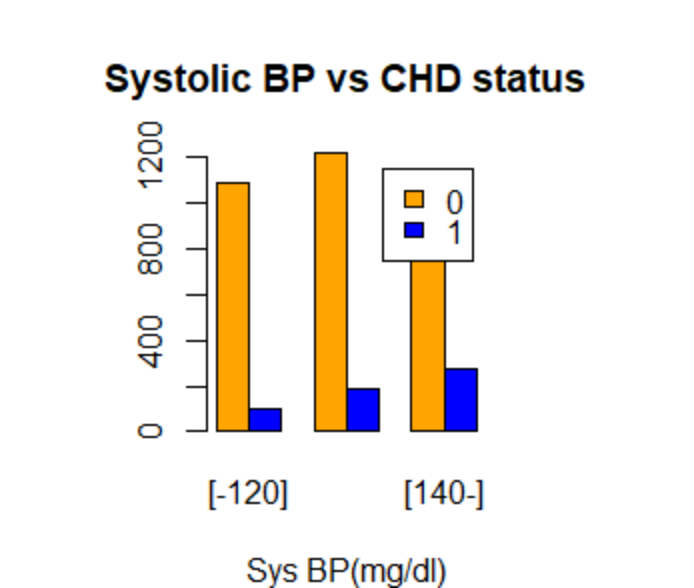


Fig.11

From the multiple bar plot we can suspect that hypertensive people are more prone to develop the disease than others.

* **diaBP**: Gives the total diastolic blood pressure of a individual in mmHg. From the multiple bar plot we can suspect that hypertensive people are more prone to the disease than others.

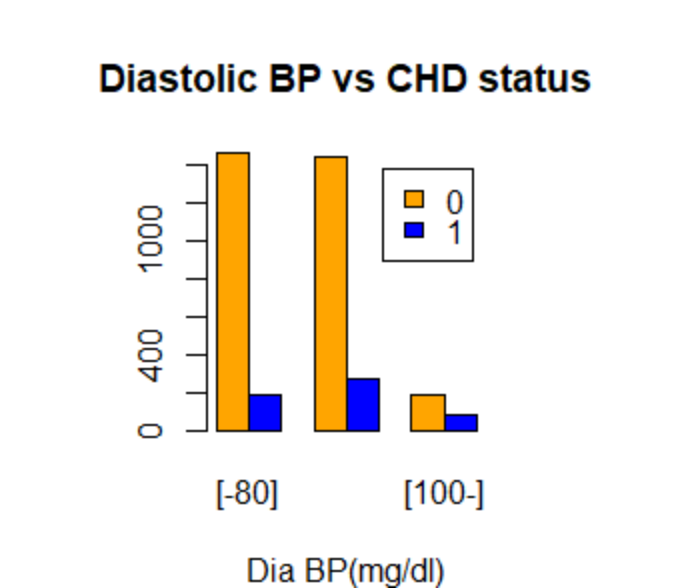
****

Fig.12

* **BMI**: It gives the body Mass Index of each subject in kg/meter squared.

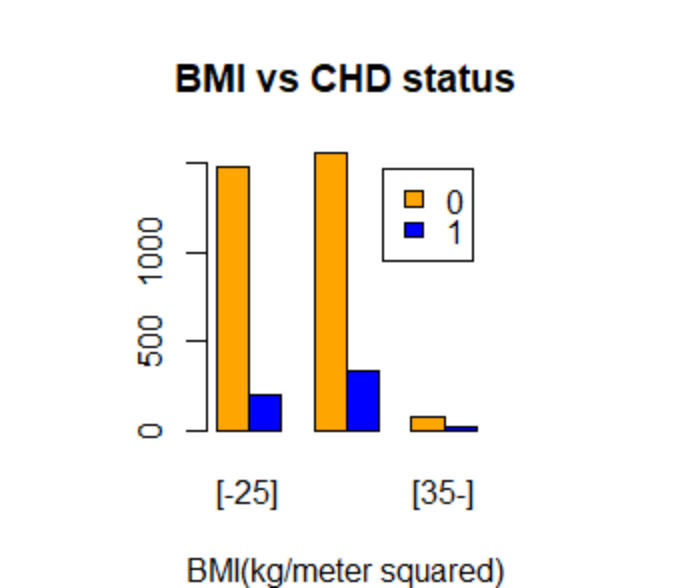
****

Fig.13

* **heartRate**: This column gives the heart rate of each person in beats/minute.

Majority of the people in the data set have normal heart rate or 65 to 75 beats per minute. We also can suspect that people with normal heart rate are less likely to develop CHD in the 10 years follow up time.

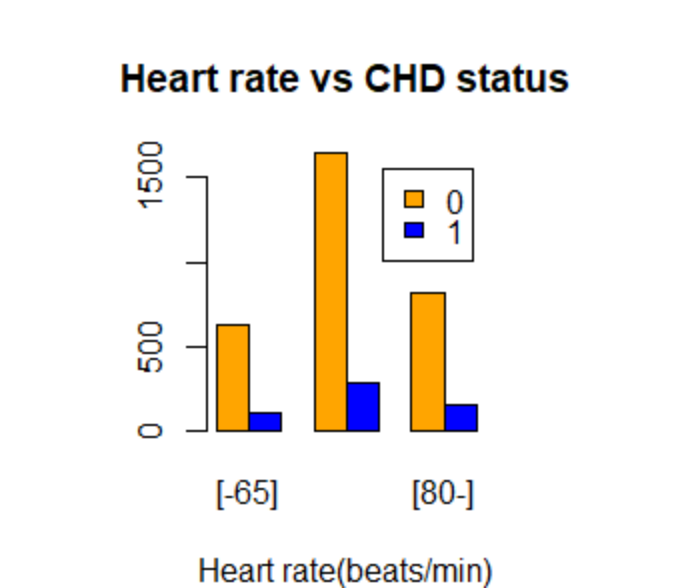
****

Fig. 14

* **glucose**: This column gives the glucose level of participants in mg/dl.

Majority of the subjects in this dataset have less than normal glucose level in their blood. Also, looking at the bar diagram we can intuitively say that, these people with hypoglycemia are observed to be far less likely to develop CHD in the next 10 years of collecting the data.

The multiple bar-plot representation is given below,

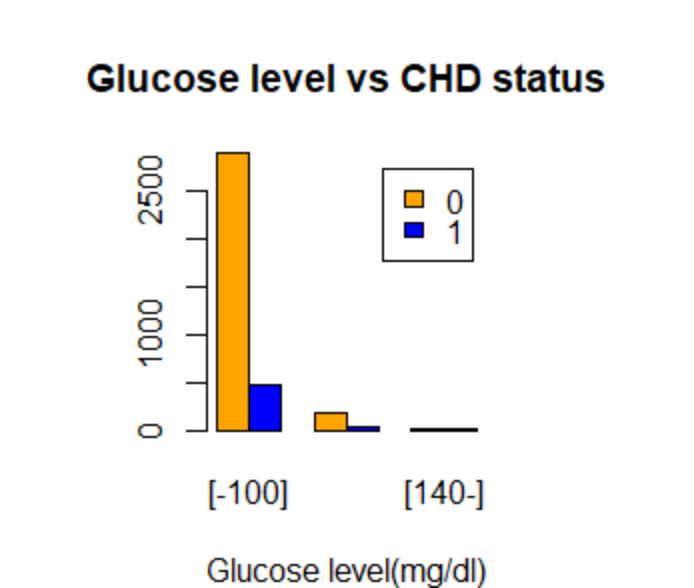


Fig.15

This fourth generation Framingham heart study data consists of information about a cohort of size 4240 participants in total. After omitting the rows which contained missing observations, the data happens to have 3658 rows of information. This is quite a big dataset. So, data imputation for missing values is not needed here.

**OBJECTIVE AND METHODOLOGY**

This data, which I have considered has 15 independent variables like smoking status, hypertension status, systolic BP, glucose level etc. After initial adjustments on the data, my objective is to fit two predictive models. One is binary logistic regression model as my usual statistical model, another is Random Forest predictive model which is a classification model based on decision trees of ML. After fitting these models, I will compare their prediction accuracies through some measures and optimize both the models by different methods. At last, I will conclude which model has better optimal prediction accuracy on this data.

Regression models are generally chosen on the basis of the nature of the response variable and both the above models are appropriate for binary response variable. But in this project work, I will mainly discuss how these two models perform when most of the covariates are ordinal in nature. For that we need to do some primary adjustments to our data which will be discussed in the analysis part later.

Let us now discuss the methodology in brief that is how to fit these two predictive models and compare their accuracies.

**FITTING OF THE MODELS**:

I have chosen two predictive models which I considered are fit for my data. Those are binary logistic and Random Forest predictive model.

* **BINARY LOGISTIC MODEL**:

We are to fit a binary logistic regression model on this data.

Let, Y be our binary response variable denoting the CHD status of a randomly selected individual in the 10 years follow up study.

Therefore,

Y=1, if we get a positive response or success,

Y=0, if we get a negative response or failure

In binary logistic regression, we model the success probability or P(Y=1) on the basis of given covariates. We will consider single explanatory variable case here.

Let, our explanatory variable is X.

I denote P(Y=1) as π(x) because the value of the dichotomous response variable Y depends on the given value of explanatory variable X.

So, we assume Y|X=x ~ Bernoulli(p), where p= π(x) =Probability of success given X= P(Y=1|X=x).

In logistic regression model, the logit of success probability has a linear relationship with the covariate or predictor,

logit[π(x)] = α + βx …(i)

where, logit [π(x)] = log [ ]

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

logit [π(x)] = log [ ]

From equation (i), we can calculate the probability of success or π(x) as,

π(x) =  . . . (ii)

Therefore, π(x) will remain in the interval (0, 1) always.

We can also calculate the odds from equation (ii),

 . . . (iii)

Equation (ii) shows that, π(x) changes as an S-shaped function of x.

The logistic model involves two parameters α and β, which are to be estimated. The parameter α is the intercept or it is called **log of background odds (Baseline)** and β can be interpreted as the change in log odds of success for unit change in x or the value of the covariate. It can be also interpreted as the rate of increment or decrement of the curve and the sign of β indicates the nature of the curve, whether it is increasing or decreasing. If we write the pmf of Bernoulli distribution in OPEF form, we get,



Comparing both sides of the above equation above, we get,

, =1 …(\*)

Therefore, η = g(µ) = g(p) = =, so we can say the logit link function in case of Bernoulli variable is a **canonical link function** and there exists sufficient statistic for the distribution whose dimension is equal to β and sufficient statistic is given by XTY, where X is the design matrix and the generalized linear model is given by,

 …(iv)

Where, is the random error vector associated with the model.

We need to fit the data to the model or in other words, we need to estimate the parameters α and β. In recent days, we have access to modern statistical software like R, SAS etc. But, previously the method used to estimate the parameters was **Discriminant function analysis,** which is essentially a form of least square estimation. But it had some drawbacks like,

* The independent variables have a restriction of being normally distributed to infer about model parameters.
* It gives too high odds ratios.

So, we use Maximum likelihood method to estimate the model parameters. Because, it has no distributional restrictions on the independent variables and gives stable estimates. The estimation process is described below,

The log likelihood function of Y is given by,



Now, squaring both sides of (1), we get,



From (2), we have,



Now, we proceed to find the score functions and variance-covariance matrix. The log likelihood function in OPEF form is given by,



Now, we assume that,

 [from (\*)]

Now, putting these values in (5) and (6), we get the score equations as, 

Similarly, 

Therefore, with the sample information functions I11, I12, I21, I22, we can write down the Fisher’s information matrix, which basically gives us the estimate of variance and covariance of α and β.



The logit link for Bernoulli distribution is given by,



Logistic regression is often used for its simple interpretation. If the explanatory variable is a **dummy** or **binary** variable then we have,



Therefore, estimate of α is the log of odds2 or in other words α is the log of the odds of success when X is 0. That is why it is called baseline or log of background odds. On the other hand, estimated β is the sample log odds ratio if X is a dummy variable. It can be interpreted as the change in log odds ratio for unit change in the value of X. Then, the fitted logistic regression model is given by,

Now, we move on to the case where there exists not one, but multiple number of covariates. Let, there are p predictors or independent variables, namely X1, X2,…,Xp. Then, the logistic regression model is given by under logit transformation,



Similar to the single predictor setup, here also we derive the score equations and they are solved iteratively as we get the estimates of βi s.

When we work in this multivariate setup, all the predictor variables may not be binary in reality. They may be given in ordinal or interval scales also. In my project work, my aim is to do a comparative analysis of how these two models perform when the independent variables are mostly ordinal in nature. Then, the interpretation of the regression will not remain the same as the binary predictor case.

In our area of discussion, I will stick to the strong assumption that the ordinal variables are equally spaced in their levels. Actually it does not matter even if the levels of the ordinal variables are not equally spaced because the results are very insensitive on this assumption except some extreme cases. **Statistician David J. Pasta** proposed the idea that we can treat the ordinal variables as continuous variables. He argued that we also often do not know that the continuous predictors have a linear relationship with the predictors. He said, “**We do not know that unit change in continuous predictors has the same effect no matter whether it is a change between two relatively low values or a change between two relatively high values. I am squarely in the camp that says “everything is linear to a first approximation**” **and therefore I am very cheerful about treating ordinal variables as continuous”**. But, we do take the relationship as linear because we want our interpretations simplified. After all, statistics is a real life science and we need to make our conclusions as simple as possible that even a layman can understand it easily. Later on, two statisticians **Long and Freese** agreed to Pasta’s proposition and added that the interpretations of regression coefficients get simpler if we consider ordinal variable as continuous variables but they also added that it involves the strong assumptions that the levels of those ordinal predictors are equally spaced or it has a linear effect on the response variable. Hence, the interpretation for regression coefficient corresponding to an ordinal predictor will remain same in this project work. We could also transform the ordinal variables to binary dummy variables with proper technique. But if we transform a large number of ordinal covariates into dummy variables then we would lose a lot of information about the data and as statisticians we should avoid the loss of information. That’s why considering them as continuous variables instead of making them binary predictors is preferable here.

When it comes to working with the Framingham dataset, I applied the following steps to fit the logistic model,

1. First, we drop the rows that contain missing observations using the ***drop.na*** function in R software.
2. After that, to make most of the predictor variables ordinal in nature, we combine different columns setting appropriate levels by column engineering. For example, let us take the two columns namely ***currentSmoker*** and ***CigsPerDay***. Now, the first is a binary predictor and the later is a continuous variable. Both of these are dependent on each other because an individual who shows non-zero ***CigsPerDay***has to have ***currentSmoker*** value as 1. So, there is no need of two separate variables. We can make a single ordinal variable with the information in hand. We categorize people who do not smoke cigs, people who smoke 1 to 10 cigs per day, people who smoke 11 to 20 cigs per day and people who smoke more than 20 cigs per day. Then we use dummy digits 1,2,3,4 to assign these 4 levels respectively. Similarly, we combine other columns as well. As a result, the variable number reduced to 11 from 15 and most of them become ordinal in nature.
3. After the primary adjustments on the data, we divide the data into two parts randomly. One part is called train set which contains majority of the data and the other part is called test data. At first, I divide the data such that the train data contains 75% of the data points and other 25% is contained by the test dataset.

* **TRAIN DATA**: This division of the data consists of the majority of the data points. It is mainly used to formulate the estimates of the model parameters and fit the model.
* **TEST DATA**: It consists of a small random part of the whole data. This division is used later in the analysis to test the accuracy of prediction of the model. We compute the responses using the estimated logit model and check the proportion of the correct predictions.

We split the data in test and train set using the ***sample.split*** function inside the library ***caTools*** in R software.

1. After splitting the data in 3:1 ratio, we proceed to fit the logistic model using the ***glm*** function in R software selecting the family as binomial. We take down the summary report of the logistic function using ***summary*** function.
2. Now, as the fitting is complete, we go for checking the prediction accuracy of the model. So, we run the test data’s covariates on the model and check the proportion of estimated responses which match with the actual response values. As because, the estimated responses are risks of having CHD or in other words they are nothing but probability values p^, so we need to set a threshold risk. Above this threshold risk, the estimated response will be considered as 1 and 0 otherwise. Then,

Prediction accuracy = (No. of correct predictions) / (Total number of predictions)

1. After checking the accuracy it is noted. Then, we try to improvise the model using some technique so that we can get the maximum prediction accuracy. We will be following more than one step to improve the model. In each step, we will try to get maximum prediction accuracy. After all the steps, I will discuss about some other important accuracy measures through confusion matrix, ROC curve etc.

* **IMPROVING THE LOGISTIC MODEL** :

There are several techniques through which one can improve a logistic regression model. I will take up mainly three methods here.

1. **ELIMINATING INSIGNIFICANT PREDICTORS**:

At first, we use all the 11 predictors in hand for building up the logistic model. But, from the summary table of the logistic regression we may find that some covariates are insignificant. The insignificance of an independent variable can be decided by looking at the p-value of the variable. In inferential analysis if the p-value is higher than the usual significance level 0.05, then the data provides enough evidence to reject the null hypothesis that the covariate has a no-zero correlation with the response. So, if some of the predictors show p-value greater than 0.05, then we can straight away declare that the predictors are insignificant. It may also happen that a predictor shows a p-value which is very close to 0.05. Our main aim in this portion is how to improve the prediction given by the model. So, if we see that omitting that one variable also increases the prediction accuracy significantly then we can omit it from the model and refit the equation afresh. We will apply this concept on the Framingham data later on the analysis portion.

1. **CHANGING THE THRESHOLD RISK**:

When we fit a binary logistic regression model to a data, we get estimates of the response variable as probability. In particular, if we fit the model in our Framingham heart study data, then we would have probabilities or risk of having CHD in 10 years as our estimated response. So, we need to fix a threshold risk level, above which we can classify the response as 1 and assign 0 below that level. Now, primarily I have set the threshold risk level at 0.5. At that level we record the prediction accuracy. Generally, in medical sciences 0.5 is taken as the lower bound of the threshold risk. Then we have to take values higher than 0.5. I will take an array of values from 0.4 to 0.6 in difference of 0.05 as my threshold risk values. Then for each value of threshold risk, we store the corresponding prediction accuracy values. Then, we draw a graph taking the threshold risk values on the x-axis and accuracies on the y-axis. The peak of the graph will give us the maximum prediction accuracy obtained from changing the threshold risk value. It should be mentioned as the discussion progresses, that this step should be performed with the prediction equation obtained after eliminating the insignificant predictors.

1. **CHANGING THE DATA SPLITTING RATIO**:

When we split the data into train and test datasets, we have to make sure that the train data has enough proportion of the data that the logistic model can be a good predictive model. And we also have to make sure that there are enough data points on which we can predict and check the accuracy. Primarily, I have taken 75% of the data points for the train dataset and remaining 25% for the test dataset. Now, to ensure that there exist enough data points in the train set, I set the lower bound for the train data proportion at 75% and I vary it from 75 to 85 percent with 1% increment in each step. Correspondingly, the test data proportion will reduce from 25% to 15%. In each step, we record the corresponding prediction accuracies. Then, we draw a graph taking the train data percentages in the x-axis and prediction accuracies in the y-axis. The peak of the graph gives the maximum prediction accuracy possible with respect to changing the data splitting ratio. Also, it should be mentioned that the model used here to optimize is the one obtained after improvisation in step 2.

Now that we have described how to improve the predictive model, we can proceed to checking different types of accuracy measures.

* **ACCURACY MEASURES**:

Here in my project, I will discuss about three accuracy measures in particular. They are,

1. **CONFUSION MATRIX**:

After fitting the binary logistic model to the data and estimating the predictions or outcomes for the test data, we categorize them to 0 and 1 based on the prefixed threshold risk level. Then, we make a table which helps us to find the baseline accuracy and prediction accuracy from the test data. This 2\*2 table consists of frequencies of concordant and discordant pairs between actual responses and predicted responses. The diagonal of this table represents the concordant pairs, that is the correct predictions and the off diagonals represent frequencies of discordant pairs or in other words wrong predictions. The confusion matrix looks like below,

|  |  |  |  |
| --- | --- | --- | --- |
| Predicted | Actual | Yes (1) | No (0) |
| Yes (1) | | N11 | N12 |
| No (0) | | N21 | N22 |

Table:1

Where, N11 = frequency of predicted positive outcomes when the individuals are actually diagnosed with CHD

N12 = frequency of predicted positive outcomes when the individuals are not diagnosed with CHD

N21 = frequency of predicted negative outcomes when the individuals are actually diagnosed with CHD

N22 = frequency of predicted negative outcomes when the individuals are not diagnosed with CHD

Therefore, N11 and N22 denote the numbers of right predictions and N12 and N21 denote the numbers of wrong predictions.

And let, total number of individuals in the test dataset is N= N11+ N12+ N21+ N22

The confusion matrix gives us four measures of accuracies. These are,

* **PREDICTION ACCURACY**

= (Total Number of right predictions) / (Total number of predictions)



* **BASELINE ACCURACY**:



* **TPR**: True positive rate or TPR is the proportion of rightly predicted Y=1 values which coincide with the actual Y value. This is also called **sensitivity or Recall** of the model.



* **FPR**: False positive rate or FPR is the proportion of those actual Y=1 responses which are predicted as Y=0 or wrongly predicted. (1-FPR) is termed as ‘Specificity’.



* **TNR**: True negative rate or TNR gives the proportion of the number of correct predictions of the actual Y=0 or negative responses. This is also called the **specificity** of the model.



It is evident that, FPR+TNR = 1 or FPR = (1-TNR)

We will be using these formulae later in the analysis part.

1. **ROC CURVE**:

The term ROC stands for **Receiver operating characteristic curve.** It is a graph diagram with FPR in the x-axis and TPR in the y-axis which measures the performance of a classification model at different threshold risks. When the threshold is set at a low level then more number of individuals would be classified as positive and when the threshold is set at a high level then less number individuals are classified as positive.

In the ROC curve, we see a trade-off between sensitivity and specificity (TPR and 1-FPR) of a diagnostic model operator. A 45 degree line is drawn, which represents the baseline for any model. If a model show a 45 degree ROC curve then the model yields TPR=FPR at every point and it is termed as “**Useless model**”. The performance of a diagnostic test or model is directly proportional to the distance of its ROC curve from the 45 degree diagonal line. In other words, farther the ROC curve from the diagonal line better is the performance. A diagnostic model would be called ‘perfect model’ if its ROC curve is parallel to the x-axis. ROC curve gives us a sorting based algorithm for accuracy measure called AUC.

**AUC MEASURE**: The term AUC stands for area under the ROC curve. It measures the total 2D area under the ROC curve. It varies from 0 to 1. If all the predictions of a classification model wrong, then the AUC measure is 0. And if all the predictions on the test data are correct, then AUC is 1. There are two reasons behind AUC being an appropriate measure of accuracy. Those are,

1. It does not measure the absolute values, but it measures how well the predictions are ranked. So, AUC is scale-invariant.
2. It does not depend on the threshold classification level. It measures the accuracy of the model irrespective of threshold risk level.

Looking at these perks, I will use AUC as a comparative measure for my two classification models.

* **RANDOM FOREST MODEL**:

Random forest model is an example of supervised model based on machine learning algorithms which can be used for both classification and regression problems. The word ‘forest’ in its name signifies that the model uses an ensemble of multiple decision trees. Building up multiple decision trees helps the model to get a stable and accurate prediction and the decision trees are trained by ‘bagging’ method. To understand how a decision tree works, let us take a real life example,

**REAL LIFE EXAMPLE OF DECISION TREE**:

Suppose I ask one of my friends to order food online for both of us. Now, the friend here now exactly the things he wants to eat. But, he has hardly any clue about my food choices. So, he asks me about my liking and disliking about food and he looked for my past orders on the delivery app. Then, based on the information in hand, he decides the order. It is clear that he has built rules to guide his decision about the food delivery, which is necessarily the main approach for random forest model.

Now, in context with our data in hand, let there is an ensemble of three decision trees which is used for our prediction. If two or more of these decision trees predicts the response to be 1, then the ensemble gives out the response as 1. That is, the ensemble works by the majority rule for classification purposes. In case of regression problems, it gives the average of all the estimates coming out from the multiple decision trees as our output. Decision tree might suffer from over-fitting problems. But, when random forest works, it combines smaller decision trees or sub trees and averages the result which cancels out the over-fitting. This is the main advantage of using random forest model over decision tree.

We use the same steps involved in fitting the logistic model for building up the random forest model on this data. We use the ***randomForest*** package in R software to carry out the classification.

Now, we will be discussing how we use random forest as a classification model works and a tree is grown.

1. **RANDOM RECORD SELECTION**: (BOOTSTRAP AGGREGATING OR BAGGING)

When the algorithm forms a standard decision tree, it is trained using exactly 63.2% or approximately 2/3 fraction of the training data set as default, which is selected using simple random sampling with replacement from the training set. The ‘Random’ part in the name of the model can be justified by this random selection of training data and also random variables for building a tree.

1. **SELECTION OF RANDOM VARIABLES**:

The algorithm also does not use all the predictor variables to split the node. It samples randomly, say m number of variable from all the independent variables. After that, it examines every variable and picks those which form the best split possible and is used to split the node. Now, the choice of m is purely subjective, it changes depending on the data. But, this m remains fixed throughout the growing of the forest. Instead of this subjectivity, if we simply use the algorithm then it considers a default m set by the machine itself.

Let, the total number of predictor variables is n.

Then, for classification purpose by default m = [sqrt(n)]

And for regression purposes, m = [n/3], where [.] denotes the greatest integer function.

1. **OOB ERROR RATE**:

After ***bagging*** the 63.2% of the training data and building the forest, the algorithm applies that ensemble of trees to the remaining 36.8% ***out-of-bag*** data. The misclassification rate in that application of trees is known as OOB (Out-of-bag) error. Suppose, we grow 500 trees in our forest, then the aggregate of the misclassification rate gives us the OOB error of the model. This internal error component validates the model’s performance on the test dataset.

Every tree in the forest classifies the OOB data. Suppose we have grown 500 trees in the forest. Each of these 500 trees has classified a binary dependent variable in the OOB data. We count the YES votes and the No votes. Let, 350 of the trees have voted YES and rest of them have voted NO. Then, the ensemble will predict YES as the class. This is called RF score which is decided by the majority rule. In case of regression, the RF would be the probability of positive output estimated by the votes given by the trees. In the previous example, the RF score for regression would be (350/500) = 0.7

**IMPROVING OR TUNING THE RANDOM FOREST MODEL**:

In case of tuning random forest model we will mainly focus on reducing the OOB error rate of the model. So, the methods we will use here will not be the same as those were in case of improving logistic regression. Random forest, being a classification model, it does not deal with threshold risk value. So, we incorporate mainly three methods for tuning. Those are,

1. **OPTIMAL NUMBER OF DECISION TREES**:

When we apply the algorithm of random forest, the machine uses ensemble of 500 decision trees by default. But we can change the number of trees to check if the OOB error can be deduced or not. We vary the tree number from 100 to 1000 with difference of 100 in each step and note down the OOB error. We then plot the OOB error rates against the number of trees in a graph and find the number of trees for which the y value is least or the curve reaches to its lowest point.

1. **OPTIMAL NUMBER OF VARIABLES FOR SPLITTING A NODE**:

As mentioned before, the algorithm does not take all the predictors into account to split a node and it takes square root of the total number of predictors by default. We can also change this number of variables (m\_try, say) and find for which m\_try, it gives minimum OOB error rate. For this we use the ***tuneRF*** function in the ***randomForest*** package of R software which automatically searches the best m\_try value for us with respect to OOB error estimates. We also plot the OOB error rates against the m\_try values to visualize the best m\_try value from the graph.

1. **JUDGING THE SIGNIFICANCE OF THE PREDICTORS**:

In case of Random Forest classification model, we do not get any regression summary table like logistic regression model, from which we can look at the p-values and decide which predictors seem insignificant in the model. To judge the importance, we use the ***importance*** function, which determines the reduction in model accuracy if a predictor variable is omitted from the model. Higher the reduction in model accuracy, higher is the significance of the predictor. We also use the ***varImpPlot*** function to plot the reduction against various predictors in ascending order in a graph. Then, the predictor in the top position will be the most significant predictor, whereas the bottom one will be taken as the least significant among all the other predictor variables.

* **FALSE ALARMS AND USE OF ROC CURVE (FUTURE STUDY)**:

There are two kinds of errors that happen in a classification based problem. In context with the data in hand, let some individual did not develop CHD in the 10 year follow up but the model prediction was the opposite. This is called a ‘**False alarm**’. And the second error is if the model predicts Y=0, that is a person will not be diagnosed with CHD and the very person later in 10 years follow up time is diagnosed with CHD. This is quantified as FNR or false negative rate. Now, in medical sciences FNR is **way more serious error** than a false alarm. Because, a person with high risk of developing a disease should take caution immediately to prevent it from happening. But, it doesn’t matter if a person receiving the false takes caution or not because the person is not going to develop the disease.

To consider a model as a good classification model is both the errors are low. But is we compare the importance of these two errors, FNR should surely be way ahead of false alarms. So, minimizing the FNR should be our first priority as this study is about a lethal disease. This concept has a strong resembles with the concept of type I error and type II error in hypothesis testing, where setting type I error fixed a low level become more important and relevant than simultaneous decrement of both the errors.

To achieve this goal, we use the ROC curve. First, we fix the value of FNR as we desire. The curve has FPR in the x-axis and TPR in the y axis. We find a point on the graph that has the maximum value (or our desired level) of FPR and we note down the value of threshold risk level at that point. Now, in the model prediction we fix that value of threshold to make sure the FPR remains our desired level in our prediction. As a repercussion, the accuracy may drop significantly because of this but we would be able to save people from misleading low threats.

**This approach can be a pathway for future study**. After fitting the model, we first fix the threshold classification level at the level in which FPR remains at our desired level and then we try to find a way how to increase both the baseline and prediction accuracy efficiently.

**ANANLYSIS AND DISCUSSION**

Our main aim for this study is to do a comparison of performance of logistic regression model and random forest on a data which has several ordinal predictors. In order to that, we first combine the columns and make new columns or predictors which are ordinal in nature. The summary of the column engineering done on the data is given in the table below,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NEW COLUMNS | COLUMNS THAT ARE COMBINED | LEVEL 1 | LEVEL 2 | LEVEL 3 | LEVEL 4 |
| Smoking\_  status | cigsPerDay and currentSmoker | People who do not smoke | People who consume 1 to 10 cigs daily | People who consume 11 to 20 cigs daily | People who consume more than 20 cigs daily |
| Diabetes\_  Status | glucose  And  diabetes | People who have Glucose level Less than 140 mg/dl and are non diabetic | People who have glucose level less than 140 mg/dl and are diabetic | People who have glucose level greater than 140 mg/dl and are non diabetic | People who have glucose level greater than 140 mg/dl and are diabetic |
| Cholesterol | totChol | People who have cholesterol level less than 170 mg/dl (Hypocholesterolemia) | People who have cholesterol level greater than 170 mg/dl and 200 mg/dl | People who have cholesterol level greater than 200 mg/dl (Hypercholeterolemia) | ----- |
| BP\_status | sysBP,  diaBP,  prevalentHyp  and  BPMeds | People who are normotensive and do not take BP medicine | People who are normotensive and take BP medicine | People who are hypertensive and do not take BP medicine | People who are hypertensive and take BP medicines |

Table: 2

**INDEX**: People who have systolic blood pressure less than 120 mmHg and diastolic blood pressure less than 80 mmHg are called ‘***Normotensive***’. People who have systolic blood pressure greater than 120 mmHg or diastolic blood pressure greater than 80 mmHg are called ‘***Hypertensive***’.

**LOGISIC REGRESSION**:

After trimming down the number of predictors, we have ***TenYearCHD*** as our binary response variable and 10 predictor variables in hand. Now, we are to fit a logistic model on this through estimating the regression coefficients minimizing the log-likelihood function. Using R software, we get the summary of the fitted logistic regression equation is given by,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficients | Estimate | Std. Error | z value | Pr(>|z|) |
| Intercept | -8.5740795 | 0.7193585 | -11.919 | < 2e-16 \*\*\* |
| Sex | 0.4691334 | 0.1225970 | 3.827 | 0.000130 \*\*\* |
| Age | 0.0760964 | 0.0073047 | 10.418 | < 2e-16 \*\*\* |
| Education | -0.1282835 | 0.0575861 | -2.228 | 0.025902 \* |
| prevalentStroke | 0.7895168 | 0.5237305 | 1.507 | 0.131686 |
| BMI | 0.0269943 | 0.0137485 | 1.963 | 0.049596 \* |
| heartRate | 0.0006205 | 0.0047536 | 0.131 | 0.896142 |
| Smoking\_staus | 0.2494770 | 0.0563523 | 4.427 | 9.55e-06 \*\*\* |
| Diabetes\_Status | 0.4075046 | 0.1150033 | 3.543 | 0.000395 \*\*\* |
| Cholesterol | 0.0031428 | 0.0012725 | 2.470 | 0.013521 \* |
| BP\_status | 0.1862030 | 0.0715940 | 2.601 | 0.009300 \*\* |

Table: 3

**INDEX**: In the table above, the number of ‘\*’ is directly proportional to the extent of significance of a predictor.

**INTERPRETATION**:

The regression coefficients of a logistic regression equation come out as the log odds ratio of the response variable. Suppose, βi denotes the regression coefficient corresponding to the ith predictor variable. Then it can be interpreted, that for one unit change in predictor will result in change in log odds ratio by the quantity βi, when all other predictor values are fixed.

Let us consider a real example from our data. Suppose we consider the continuous predictor ***Age.*** The estimated regression coefficient corresponding to age has come out to be 0.0760964. If we fix the values of all other predictors except age, then it can be said that the log odds ratio increases by the quantity 0.0760964 for one year increase in age. We can also write the interpretation with respect to odds. Since, the intercept or baseline odds has come out to be -8.5740795. Therefore the odds can be written as,



Where, k is a constant representing the effect of all the other fixed predictor variables.

Therefore, from the above expression it is clear that the odds of developing CHD in the 10 years time increases by the quantity e0.0760964 for one year increase in age.

Now, let us take a binary predictor in consideration from the data, say ***sex***. It can take either 1 or 0 (for male and female respectively). The regression coefficient corresponding to sex is 0.4691334. So, with respect to odds it can be written as,



Where, k’ denotes a constant representing the effect of all the other fixed predictor variables.

Therefore, the odds of developing CHD in 10 years is e0.4691334 times more in case of a male individual than a female individual if other predictors are kept fixed.

At last, we come to the case when the predictor variable is ordinal in nature. Let us take the predictor ***Smoking\_status***. The regression coefficient has come out to be 0.2494770. Here comes a strong assumption of linearity in this case. We assume that a the multiplied quantity by which the odds increases is same when we move from level 1 to 2 and 2 to 3 and so on if all other covariates are kept fixed. In other words, we consider that the levels of the ordinal predictors are equally spaced. The quantity by which the odds of developing CHD increases from a non smoker individual to an individual who consumes 1 to 10 cigs per day is the same increase in odds if we compare an individual who smokes 11 to 20 cigs per day than an individual having 1 to 10 cigs per day. We write in terms of odds,



Where, K” denotes the constant representing the effect of all the other fixed predictor variables.

Therefore, the odds of developing CHD in 10 years follow up time for an individual who smokes 1 to 10 cigs per day is e0.2494770 times more than a non smoker individual and as we increase the levels, the odds gets multiplied by the same quantity each time.

Similarly, we can interpret rest of the regression coefficients also whether the predictor is ordinal or binary or continuous in nature. Now, we shall move on to calculate the accuracy measures for the model.

**ACCURACY MEASURES**:

After fitting the regression equal, we use the model to predict the outcomes for the test dataset. The test data involves 914 individuals in total selected randomly from the main data. Once the prediction is done by setting the threshold risk for classification at 0.5, we build the confusion matrix.

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 772 | 3 |
| 1 | | 132 | 7 |

Table: 4

The accuracy measures are given by,

* Prediction accuracy = (772+7) / (772+132+3+7)

= 779 / 914 = 0.8523

Therefore, the fitted logistic regression model can predict whether a person is going to develop CHD or not with 85.23% accuracy rate.

* Baseline accuracy = (772+3) / (772+132+3+7)

= 775 / 914 = 0.8479

* Misclassification rate = (3+132) / (772+132+3+7)

=0.1477

Therefore, the predictive model predicts a person’s possible CHD status in 10 years with the risk of 14.77% of being wrong.

* TPR = proportion of cases when the model predicts that an individual will develop CHD in 10 years and he or she actually develops CHD in that follow-up time span.

= (7 / 139) = 0.0504

It suggests that the predictive model is able to predict 5.04% of the cases correctly when the person actually develops CHD in the 10 years of follow up time, which is a very alarming result.

* FPR = proportion of cases when the model predicts that an individual will develop CHD in 10 years and he or she does not develop CHD in that follow-up time span.

= 3 / 775 = 0.0038

Therefore, the model predicts 0.38% out of all the actual positive cases as false alarms.

* TNR = proportion of cases when the model predicts that an individual will not develop CHD in 10 years and he or she does not develops CHD in that follow-up time span.

= (1 – FPR) = (1-0.0038) = 0.9962

Out of all the cases where the subject does not develop CHD, the model can predict them with 99.62% accuracy.

* Precision = Proportion of correct prediction when the model predicts that an individual will be diagnosed with CHD in 10 years

= 7 / 10 = 0.7

* Prevalence = Proportion of actual positive cases out of all cases

= (132+7) / (772+132+3+7)

= 0.1520

It is evident from the measures, that the overall prediction accuracy of the logistic model is pretty high. But if we dig deeper into the measures, we notice that TPR is very low for the model, that the actual positive cases have a very high risk of being predicted incorrectly or as negative cases. Since we are dealing with a clinical data and with a very serious disease as our response, so we should be cautious about the FNR = (1-TPR). We will discuss how to minimize FNR error using ROC curve later in the analysis.

The model shows,

Null deviance = 2341.9, Residual deviance = 2096.6,

AIC (Akaike information criterion) = 2118.6, this measure is similar to adjusted R2, but works like just the opposite from it.

These three measures can be taken up as comparative accuracy measures because, lesser these quantities get, better is the fit.

Now, let us move on to draw the ROC curve and calculate the AUC measure.

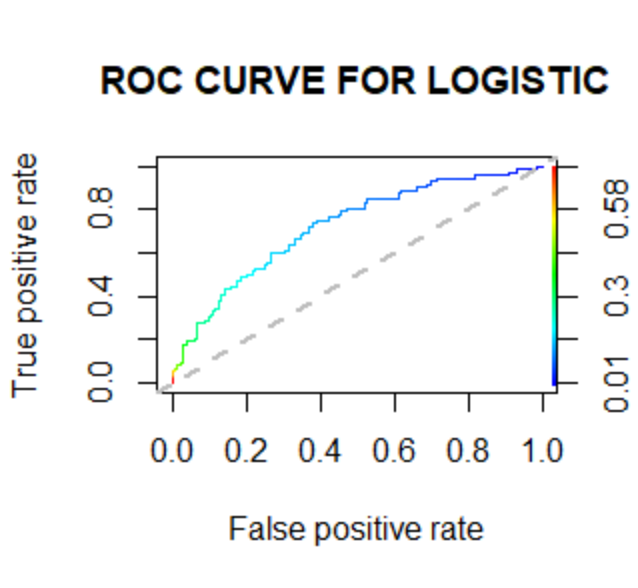


Fig. 16

The above figure gives the ROC curve for the logistic regression which plots the TPR against TPR for every value of threshold classification risk p from 0 to 1. I have used the ***ROCR*** library in R to obtain the ROC curve and AUC measure of accuracy.

**The area under the ROC curve or AUC** = 0.7152564

It measures how much the predictions are in agreement with the actual responses. Since, this measure is independent of the threshold classification level p, we can use it to compare the performance of the two model in the last section of my study after improving both the models to their highest accuracy.

**IMPROVING THE LOGISTIC REGRESSION MODEL**:

We will use the techniques discussed previously in the methodology section to improve the logistic model in terms of prediction accuracy.

1. We know, that if the p-value of a predictor variable is greater than the significance level of the model, then we can consider the predictor variable as a insignificant predictor. The usual significance level is taken as 0.05.

Now, from the summary table of the model, it is evident that the predictors

***prevalentStroke*** and ***heartRate*** are insignificant because they show p-values higher than the significance level 0.05. Now, we refit the model and see if the accuracy measures improve or not. The summary of the refitted model is given below,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coefficients | Estimate | Std. Error | z value | Pr(>|z|) |
| Intercept | -7.885468 | 0.547524 | -14.402 | < 2e-16 \*\*\* |
| Sex | 0.468289 | 0.121248 | 3.862 | 0.000112 \*\*\* |
| Age | 0.076004 | 0.007273 | 10.451 | < 2e-16 \*\*\* |
| Education | -0.145897 | 0.057065 | -2.557 | 0.010567 \* |
| BMI | 0.235664 | 0.055410 | 4.253 | 2.11e-05 \*\*\* |
| Smoking\_staus | 0.235664 | 0.055410 | 4.253 | 2.11e-05 \*\*\* |
| Diabetes\_Status | 0.431174 | 0.114273 | 3.773 | 0.000161 \*\*\* |
| Cholesterol | 0.003234 | 0.001265 | 2.556 | 0.010587 \* |
| BP\_status | 0.221873 | 0.069733 | 3.182 | 0.001464 \*\* |

Table:5

These estimates of regression coefficients can be interpreted in the same way as before. After removing the insignificant variables, we get same prediction accuracy as before. But, as a measure of comparison, we can say that AIC has decreased, that is more variation of the response variable is explained by the predictive model.

The change in values of accuracy measures before and after omitting insignificant variables is given in the table below,

|  |  |  |  |
| --- | --- | --- | --- |
| INSIGNIFICANT VARIABLES | AIC VALUE | NULL DEVIANCE | RESIDUAL DEVIANCE |
| BEFORE REMOVING | 2118.6 | 2341.9 | 2096.6 |
| AFTER REMOVING | 2116.7 | 2341.9 | 2098.7 |

Table:6

So, removing insignificant variable has made our model stable a little bit more, but there was no change in the prediction accuracy. Therefore we move on to the next step.

1. We set the threshold value at 0.5 at first, for prediction. Now, we vary it from 0.4 to 0.6 with difference of 0.01 and plot the accuracies corresponding to every p to get the clear picture. The graph (obtained using Minitab) is given below,

 Fig.17

From the diagram above, it seems clear that the prediction accuracy attained at p=0.5 is the maximum prediction accuracy (85.23%) that can be attained. But, there are 3 more values of p at which the prediction accuracy becomes highest. Now, the question arises is which value of p should be taken to make the model optimum. For that, we need to dig deeper into other accuracies and error for these values. The confusion matrices are given by,

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 770 | 5 |
| 1 | | 130 | 9 |

Table: 7 (for p=0.46)

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 773 | 2 |
| 1 | | 133 | 6 |

Table: 8 (for p=0.53)

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 774 | 1 |
| 1 | | 134 | 5 |

Table: 9 (for p=0.56)

Then, the measures of accuracy and errors are given in the table below,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Values of p | TPR | FPR | FNR | TNR | Precision |
| 0.46 | 0.0647 | 0.0064 | 0.9352 | 0.9936 | 0.6429 |
| 0.50 | 0.0504 | 0.0038 | 0.9496 | 0.9962 | 0.7000 |
| 0.53 | 0.0432 | 0.0026 | 0.9568 | 0.9974 | 0.7500 |
| 0.56 | 0.0359 | 0.0012 | 0.9641 | 0.9988 | 0.8333 |

Table:10

In a clinical study, as it has been stated earlier, between the two errors FNR is more serious than false alarms. Since, the prediction accuracies are basically the same and maximum, we choose that value of threshold risk for which the FNR error is least. From the above table, we notice that p=0.46 gives the minimum FNR value than the other three. So, we choose p=0.46 as our threshold risk for betterment in prediction with maximum accuracy and minimum FNR possible.

1. Now, we reach at the last step of improving the model where we change the percentage composition of the train and test dataset and see which composition gives us the maximum prediction accuracy. Here, we use p=0.46 as our threshold risk which is obtained in the last step. To ensure that both train and test dataset have adequate number of data points for good fitting and prediction, we set the lower bound at 75% for train set and vary it till 90%. As we get the accuracies for different compositions, we plot percentages for train set in the x-axis and the accuracies in the y-axis in a graph. We find the point in the x-axis, for which the curve gives maximum value for y that is accuracy. The graph is obtained using the Minitab software and shown below,



Fig.18

We have varied the percentage composition of train data from 75 to 90 percent with the difference of 0.5 percent in each step. So, there are 31 data points in total. From the graph above, it is evident that the maximum accuracy we can get is 0.8526316 against the train data composition 87%. So, we choose 87% of the whole data for the train data to build the model and the other 13% data points for our test data set to predict upon. The increment in the prediction accuracy is not that significant, but even a small betterment matters in case we are predicting a deadly disease status of a patient.

Now, we refit the logistic model with the significant variables and best choice of p and best percentage composition of train and test data sets. Then, after predicting on the test dataset, we get the following confusion matrix,

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 398 | 5 |
| 1 | | 67 | 5 |

Table:11

Then, the measures of accuracies are given by,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prediction accuracy | Baseline accuracy | Misclassification rate | TPR | FPR | TNR | precision | prevalence |
| 0.8484211 | 0.8484211 | 0.1515789 | 0.069444 | 0.0124 | 0.98759 | 0.5 | 0.141053 |

Table: 12

Also, AUC Measure = 0.7425903

**MAXIMIZING THE TRUE POSITIVE RATE FOR A CLINICAL DATA**:

Until now, we have been working towards improving the prediction accuracy or overall accuracy. But, we have a clinical data in hand and we are to predict the status of a lethal disease. So, a diagnostic model for a clinical data cannot be considered as good model if it cannot predict the actual positive cases with high accuracy, which is by definition the true positive rate. Or we can say that the main goal here is to minimize FNR, for which TPR automatically gets maximum. So, let us fix FNR at desired level, say 0.01. Now, we have to see which threshold risk value gives us FNR<=0.01 or TPR>0.99 from the training set and we will incorporate that threshold risk to predict on the test data and validate the result. We will also be interested to see, in order to maximize the TPR how much the overall prediction accuracy decreases. Let us plot the TPR for different threshold risk values in a graph and see where we can get the desired TPR value.

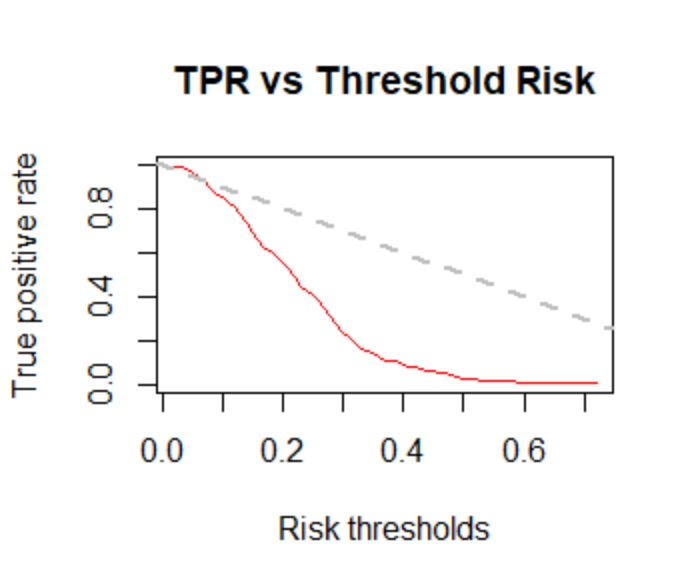


Fig.19

Now, from the graph it is clear, if we want to get TPR>=0.99 then we have to set p=0.03

Let us incorporate this threshold risk value to predict on the test dataset. The confusion matrix is given by,

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 5 | 398 |
| 1 | | 0 | 72 |

Table: 13

Therefore, prediction accuracy = 0.1621, which a very large decrement form the previous diagnostic model using p=0.46

Though, overall prediction accuracy decreases a lot for this method, we can now safely predict the actual CHD positive cases with desired accuracy. Later, we apply this method for random forest model too and compare both the models for maximum prediction accuracy.

**RANDOM FOREST CLASSIFICATION MODEL**:

Now, we have to fit Random Forest classification model to the data in hand after trimming down the number of independent variables. In order to do so, we first classify our response variable into two categories where ‘no’ denotes the value 0 and ‘yes’ denotes the value 1. We fit the model using the ***randomForest*** package in R software. Unlike logistic regression, the random forest model, being a classification model gives us categorical outputs instead of risk probabilities. Again, we will not get any algebraic form of the prediction model because it works with the help of decision trees or in other words, a specific machine algorithm. The summary of the model is given by,

Number of trees used: 500

No. of variables tried at each split: 3

OOB (Out of bag) estimate of error rate = 15.74%

**ACCURACY MEASURES**:

After building the classification model we use the model to predict on the test dataset. The confusion matrix is given by,

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 770 | 5 |
| 1 | | 135 | 4 |

Table: 14

The accuracy measures are given in the table below,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prediction accuracy | Baseline accuracy | Misclassification rate | TPR | FPR | TNR | precision | prevalence |
| 0.8468271 | 0.8479212 | 0.1531728 | 0.02878 | 0.006451 | 0.993549 | 0.555556 | 0.1520788 |

Table: 15

These measures can be interpreted in the similar way as before.

**IMPROVING THE RANDOM FOREST MODEL**:

In this section, we mainly focus on the methods in which we can improve our random forest classification model. We will apply the methods which we discussed earlier and we will consider OOB error rate as our comparative performance measure among the different random forest classifiers.

1. **FINDING THE OPTIMAL NUMBER OF DECISION TREES**:

We have used an ensemble of 500 decision trees in our initial stage of building of the model. No, we will vary the number of tress from 100 to 1000 with the difference of 100 in each step and note down the OOB error rates. Then, we plot the error rates against the number of trees in a graph and see which ensemble of trees gives us minimum OOB error rate. The graph is given below,



Fig.20

From the graph above, it is clear that the minimum OOB error rate can be obtained if we use 600 decision trees in our model. The minimum OOB error rate is 15.56%.

1. **FINDING THE OPTIMAL SPLIT**:

The number of independent variables used in each split is taken as default (mentioned previously, here which is 3) when we use the software. But, there always exist an optimal split for which the OOB error rate becomes the least. To achieve the best split, we use the ***tuneRF*** function inside the randomForest package. The arguments of this function is as follows,

x: matrix of predictor variables

y: response vector = TenYearCHD

ntreeTry: Number of decision trees while tuning = 600 (optimal number of trees derived earlier)

StepFactor: For each iteration, mtry (number of variable in each split) is inflated by this value =1.5

improve: This is the minimum by which the OOB error rate should necessarily change to continue searching and going to the next iteration = 0.001 (Basically it is a measure of precision one can set as he/she desires)

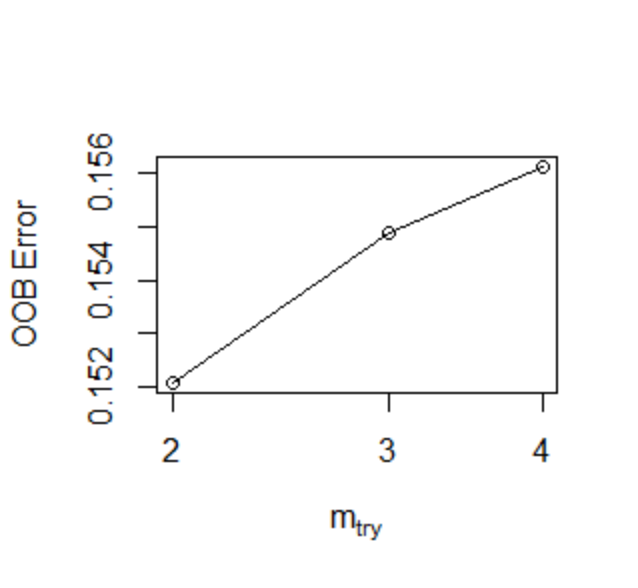
We plot the different OOB error rates against the splits in a graph and visualize, 

Fig.21

When the number of variables is 2 in each split, we are getting the minimum OOB error rate form the graph above. Also, after m\_try=3, the OOB error rate has stabilized. So, we will use 2 as our best split from here on. We also tabulate the error rates below,

|  |  |
| --- | --- |
| NO. OF VARIABLES IN EACH SPLIT | OOB ERROR RATE |
| 2 | 0.1520578 |
| 3 | 0.1548853 |
| 4 | 0.1561420 |

Table:16

1. **SIGNIFICANCE OF THE INDEPENDENT VARIABLES**:

Unlike logistic regression, we do not get regression table and corresponding p-values from which we can judge whether a variable is significant or not. So, we apply the method of checking the decrease in model accuracy and also we check the decrease in Gini impurity index if we drop the concerned independent variable. We have used the function ***Importance*** in randomForest package to get the measurements. The decrements are given in the table below,

|  |  |  |
| --- | --- | --- |
| VARIABLES | MEAN DECREASE ACCURACY | MEAN DECREASE GINI |
| Sex | 6.309296 | 13.63784 |
| Age | 21.104469 | 96.07777 |
| Education | -2.098408 | 25.80510 |
| prevalentStroke | 6.001735 | 3.87781 |
| BMI | 1.289480 | 99.93384 |
| heartRate | 1.177757 | 72.12392 |
| Smoking\_status | 3.626379 | 27.58894 |
| Diabetes\_status | 8.049203 | 12.34996 |
| Cholesterol | 9.151177 | 12.34996 |
| BP\_status | 2.839353 | 18.08076 |

Table:17

We plot the tow measures of significance using ***varImpPlot*** function in the randomForest package. The graphical representation below shows the two measure of significance simultaneously for each attribute in our data,

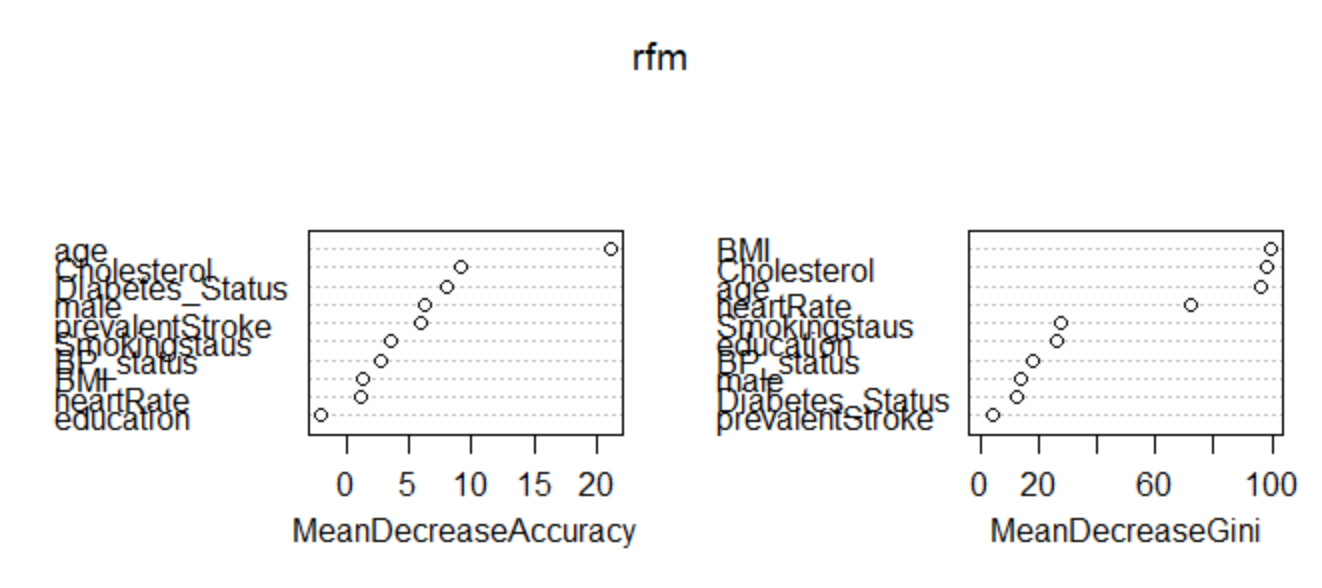


Fig.22

We know higher the value of decrease in model accuracy, higher the significance of the independent variable in the model. Therefore, age is the most significant variable and education and heartRate is the least significant variable according to our random forest model.

1. In this project work, our main purpose is to compare the performance of the logistic regression and random forest classification model on the data. But, in case of improving random forest model, we have to focus on minimizing the OOB error rate. So, changing the percentage composition of train and test data set in order to decrease error is beyond our scope. But, we fix the percentage at 87% for training set because it gave optimum accuracy in case of logistic regression.

At last, we refit the model removing the insignificant variables and use it to predict on the test dataset again using the derived best split, changed percentage composition of train data and optimal number of decision trees.

The results after refitting the model and prediction are given by,

OOB estimate of error rate=15.17%

The confusion matrix is given by,

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | No (0) | Yes (1) |
| 0 | | 402 | 1 |
| 1 | | 68 | 4 |

Table: 18

The different measures of accuracies after improving the OOB error rate of the model is given in the table below,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prediction accuracy | Baseline accuracy | Misclassification rate | TPR | FPR | TNR | precision | prevalence |
| 0.8547368 | 0.8484211 | 0.1452632 | 0.05555556 | 0.0024813 | 0.997518 | 0.8 | 0.14315789 |

Table: 19

Now, we move on to derive the ROC curve for the random forest model and the AUC (area under the ROC curve) measure of accuracy. The ROC curve is shown below,

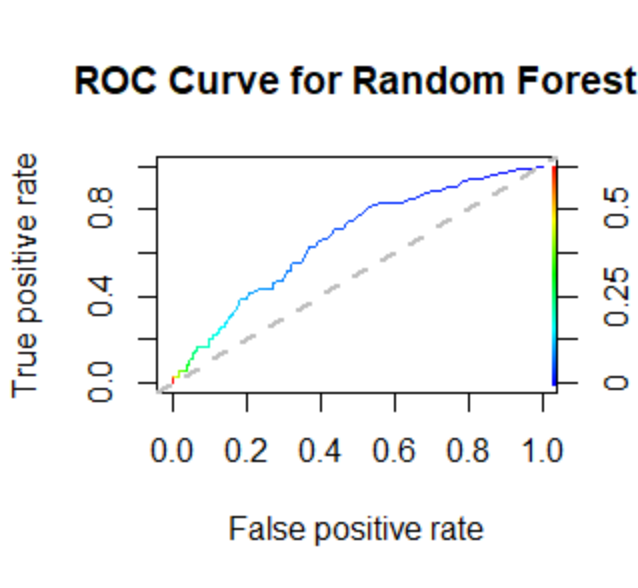


Fig.23

Also, AUC measure = 0.6562931, which we will use later as a measure of comparison with the logistic regression model in the conclusion part.

**SUMMARY AND COMPARISON OF ACCURACY**

After improving and tuning both logistic regression model and Random Forest classification model, we summarize our results or accuracy measures in the table below,

|  |  |  |  |
| --- | --- | --- | --- |
| MODELS | Prediction accuracy | Baseline accuracy | Misclassification rate |
| LOGISTIC | 0.8484211 | 0.8484211 | 0.1515789 |
| RANDOM FOREST | 0.8547368 | 0.8484211 | 0.1452632 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODELS | TPR | FPR | TNR | FNR |
| LOGISTIC | 0.069444 | 0.0124 | 0.98759 | 0.930556 |
| RANDOM FOREST | 0.055556 | 0.0025 | 0.99752 | 0.944444 |

|  |  |  |
| --- | --- | --- |
| MODELS | Precision | Prevalence |
| LOGISTIC | 0.5 | 0.141053 |
| RANDOM FOREST | 0.8 | 0.143158 |

Table: 20

From the table above, we can see that Random Forest model shows slightly higher prediction accuracy, equal baseline accuracy and lesser prevalence, less precision and more misclassification rate than the Random Forest model. From these accuracy measures, we can safely say that Random Forest model performs better than Logistic model for this data in terms of prediction accuracy.

If we dig deeper and see for the major error in the models, we see that logistic regression shows less FNR than Random Forest model (For this the other less serious error or FPR is less in case of Random Forest model). So, the former one also stands out here in case of error minimization in the model. Also, even if Random Forest model shows more prediction accuracy and less error, it shows less TPR than Logistic regression model. That suggests that logistic regression model can better predict the actual positive cases than Logistic regression model, which is desirable in case of clinical study.

We also compare the AUC values after tuning for both the models and they are summarized in the table below,

|  |  |  |
| --- | --- | --- |
| MODELS | LOGISTIC | RANDOM FOREST |
| AUC VALUES | 0.7425903 | 0.6562931 |

Table: 21

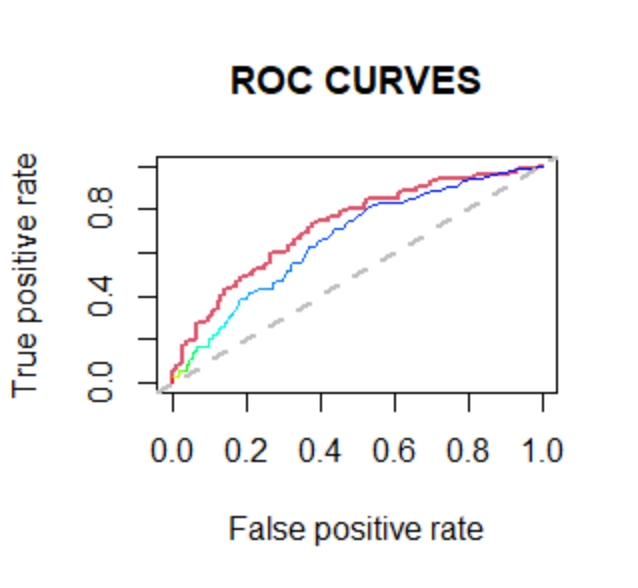
To visualize the difference in AUC or in other words, to visualize the difference in area under ROC curves, we overlay the two ROC curves on a single diagram, which is given below, 

Fig. 24

In the above diagram, the red and blue curves respectively denote ROC curve for Logistic regression model and Random Forest classification model. By the nature of an ROC curve, we know that the performance of a model deteriorates as the ROC curve gets nearer to the 45 degree line. Here, the ROC curve of the Random Forest model is closer to the 45 degree line than that of the ROC curve for Logistic regression model. Again, from the AUC measure we can validate that Logistic regression model performs better in terms of error minimization on this data because it shows greater AUC value than Random Forest model (It is also clear from the diagram because the ROC curve for logistic regression has more area underneath it).

**MAXIMIZING TPR**: **(SUBJECTIVITY IN THE MODELS)**

Random Forest, being a classification model does not incorporate any threshold risk, because unlike logistic regression, it gives us “yes” or “no” instead of risk probabilities. In case of Logistic regression model, we have to set the boundary for decision ourselves for the model. So, we have the freedom of changing or tuning the necessary errors necessary for a data. Logistic regression model involves an additional subjectivity in case of data. For example, handling the Framingham heart disease data, we were required to predict the actual CHD positive cases with maximum accuracy, which is a typical requirement for a clinical data. Logistic regression gave us that freedom to maximize the TPR at a desired level using a specific value of threshold risk, but Random Forest model failed to give us the scope to do so. So, about the matter of subjectivity, Logistic regression model proves to be a lot more flexible and relevant to this particular data.

**CONCLUSION**

“We are moving slowly in to an era, when big data is the starting point, not the end”

This was the statement Pearl Zhu, author of the famous book series “Digital Master” used about big data in his book once. The statistical world has always been full of debates. For example, the proverbial debate about “Frequentists or Bayesian” is practiced around the world even now. But, a new debate which has emerged recently in the world of data is “Are Data science and Statistics same”?

A data scientist deals with multiple machine learning models and judges by the prediction accuracy which model is good for a data. Whereas, a statistician takes one model and improve it step by step until all the specifications about the model and assumptions on data are satisfied. While data scientists are working with huge datasets and predict something, statisticians focus on how much information they can gather from a small datasets. But, they have a common ground too. Sometimes, a data scientist has hardly any clue what to do with a dataset without the insights from a statistician and if we give a statistician a dataset with billion rows and 10000 variables, he would have a hard time handling it without a data scientist. That’s why a machine learning model said to be mathematical model or algorithm which only focuses on the prediction accuracy, it does not take into account the subjectivity of the data whereas statistical modeling is all about revealing information of the data in hand.

In this project work, we have observed that in terms of prediction accuracy, Random Forest Model outplays Logistic regression model, but in terms of sensitivity and specificity in the model, Logistic regression performs better. But, in context of our clinical data, better sensitivity is what we aim for in the model, so logistic regression model would be a better choice. Also, Logistic regression gives us the independence to make TPR as high as we desire. Though we have to compromise with the prediction accuracy in that case, but it adds subjectivity in the model. One would the opportunity to modify the model as he desires. Random Forest model or any other machine learning algorithm can give us more accurate prediction introducing additional randomness in the model and using complex algorithms, but it fails to give major insights about the data that a statistical model can do.

Since, the prediction for both the model is pretty close and in addition to that, Logistic regression model gives better sensitivity and specificity than Random Forest model and we can tune the Logistic regression model for desired sensitivity by changing threshold risk, I can conclude that Logistic regression model is a better choice for analyzing Framingham Heart disease data.

**REFERENCE:**

* Categorical data analysis- Alan Agresti
* David G. Kleinbaum - Logistic Regression
* Samuel Kotz - Breakthroughs in Statistics\_ Volume II\_ Methodology and Distribution (Springer Series in Statistics Perspectives in Statistics) (1993)
* Random Forest with R – Robin Genuer Jean & Michel Poggi
* Exploratory data analysis – Tukey
* Data Analysis: An introduction – Lewis & Beck
* <https://www.listendata.com/2014/11/random-forest-with-r.html?m=1>
* <https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/>
* <https://builtin.com/data-science/random-forest-algorithm>
* <https://www.guru99.com/r-random-forest-tutorial.html>
* <https://towardsdatascience.com/random-forest-in-r-f66adf80ec9>
* Data source: <https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset>

**ACKNOWLEDGEMENT**

I would like to convey my sincere gratitude to my supervisor and mentor Prof. Surabhi Dasgupta for continuously sharing her expertise and helping me throughout the whole process whenever I needed any guidance. I would not be able to finish this project work if she did not help me with ideas and study materials constantly.

I would also like to extend my thanks to all of my other professors of the Department of Statistics, St. Xavier’s College for their encouragement and kind co-operation.

Last but not the least, my special thanks go to my parents and classmates for staying by my side in this process and helped to keep my mind straight and focused.

**APPENDIX:**

I have used mainly R for coding and Minitab for some diagrams.

**R-Code:**

rm(list=ls())

df=read.csv("C:/Users/dell/Desktop/DISSERTATION files/framingham.csv")

df1=na.omit(df)

dim(df1)

data=data.frame(df1)

## DATA VISUALIZATION ##

table2=table(data$male,data$TenYearCHD)

barplot(table2,beside = TRUE,col=c("orange","blue"),main="Gender vs CHD status",legend=rownames(table2),xlab="Gender")

x=data$age

min(x)

max(x)

breaks=seq(32,72,10)

length(breaks)

tags=c("[32-42]","[42-52]","[52-62]","[62-72]")

bin\_age=cut(x,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_age)

tbl\_age=table(data$TenYearCHD,bin\_age)

barplot(tbl\_age,main="Age vs outcome",xlab="Age(in years)",col=c("orange","blue"),legend=rownames(tbl\_age),beside=TRUE)

table3=table(data$TenYearCHD,data$education)

barplot(table3,beside = TRUE,col=c("orange","blue"),legend=rownames(table3),main="Education level vs CHD status",xlab="Education level")

table4=table(data$currentSmoker,data$TenYearCHD)

barplot(table4,beside = TRUE,col=c("orange","blue"),legend=rownames(table4),main="Smoking status vs CHD status",xlab="Smoking status")

x1=data$cigsPerDay

max(x1)

breaks=c(0,10,20,70)

length(breaks)

tags=c("[0-10]","[10-20]","[20-70]")

bin\_cigs=cut(x1,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_cigs)

tbl\_cigs=table(data$TenYearCHD,bin\_cigs)

barplot(tbl\_cigs,main="Cigs per day vs CHD status",xlab="cigs per day",col=c("orange","blue"),legend=rownames(tbl\_cigs),beside=TRUE)

table5=table(data$BPMeds,data$TenYearCHD)

barplot(table5,beside = TRUE,col=c("orange","blue"),legend=rownames(table5),main="BP Meds vs CHD status",xlab="BP Meds")

table6=table(data$prevalentStroke,data$TenYearCHD)

barplot(table6,beside = TRUE,col=c("orange","blue"),legend=rownames(table6),main="Stroke history vs CHD status",xlab="Stroke history")

table7=table(data$prevalentHyp,data$TenYearCHD)

barplot(table7,beside = TRUE,col=c("orange","blue"),legend=rownames(table7),main="Hypertension history vs CHD status",xlab="Hypertension history")

table8=table(data$diabetes,data$TenYearCHD)

barplot(table8,beside = TRUE,col=c("orange","blue"),legend=rownames(table8),main="diabetes status vs CHD status",xlab="Diabetes status ")

x2=data$totChol

min(x2)

max(x2)

breaks=c(113,170,200,600)

length(breaks)

tags=c("[-170]","[170-200]","[200-]")

bin\_chol=cut(x2,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_chol)

tbl\_chol=table(data$TenYearCHD,bin\_chol)

barplot(tbl\_chol,main="Cholesterol level vs CHD status",xlab="Cholesterol level",col=c("orange","blue"),legend=rownames(tbl\_chol),beside=TRUE)

x3=data$sysBP

min(x3)

max(x3)

breaks=c(83.5,120,140,295)

length(breaks)

tags=c("[-120]","[120-140]","[140-]")

bin\_sys=cut(x3,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_sys)

tbl\_sys=table(data$TenYearCHD,bin\_sys)

barplot(tbl\_sys,main="Systolic BP vs CHD status",xlab="Sys BP(mg/dl)",col=c("orange","blue"),legend=rownames(tbl\_sys),beside=TRUE)

x4=data$diaBP

min(x4)

max(x4)

breaks=c(48,80,100,142.5)

length(breaks)

tags=c("[-80]","[80-100]","[100-]")

bin\_dia=cut(x4,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_dia)

tbl\_dia=table(data$TenYearCHD,bin\_dia)

barplot(tbl\_dia,main="Diastolic BP vs CHD status",xlab="Dia BP(mg/dl)",col=c("orange","blue"),legend=rownames(tbl\_dia),beside=TRUE)

x5=data$BMI

min(x5)

max(x5)

breaks=c(15.54,25,35,56.8)

length(breaks)

tags=c("[-25]","[25-35]","[35-]")

bin\_BMI=cut(x5,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_BMI)

tbl\_BMI=table(data$TenYearCHD,bin\_BMI)

barplot(tbl\_BMI,main="BMI vs CHD status",xlab="BMI(kg/meter squared)",col=c("orange","blue"),legend=rownames(tbl\_BMI),beside=TRUE)

x6=data$heartRate

min(x6)

max(x6)

breaks=c(44,65,80,143)

length(breaks)

tags=c("[-65]","[65-80]","[80-]")

bin\_hr=cut(x6,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_hr)

tbl\_hr=table(data$TenYearCHD,bin\_hr)

barplot(tbl\_hr,main="Heart rate vs CHD status",xlab="Heart rate(beats/min)",col=c("orange","blue"),legend=rownames(tbl\_hr),beside=TRUE)

x7=data$glucose

min(x7)

max(x7)

breaks=c(40,100,140,394)

length(breaks)

tags=c("[-100]","[100-140]","[140-]")

bin\_glu=cut(x7,breaks=breaks,labels=tags,include.lowest=TRUE)

summary(bin\_glu)

tbl\_glu=table(data$TenYearCHD,bin\_glu)

barplot(tbl\_glu,main="Glucose level vs CHD status",xlab="Glucose level(mg/dl)",col=c("orange","blue"),legend=rownames(tbl\_glu),beside=TRUE)

##COMBINING VARIABLES USING COLUMN ENGINEERING##

cig=data$cigsPerDay

smo=data$currentSmoker

smost=array(dim=1)

for(i in 1:3658)

{

if(smo[i]==0)

{

smost[i]=1

}else

{

if(cig[i]>=1 & cig[i]<=10)

{

smost[i]=2

}else

{

if(cig[i]>=11 & cig[i]<=20)

{

smost[i]=3

}else

{

if(cig[i]>20)

{

smost[i]=4

}

}

}

}

}

data <- subset(data, select = -c(cigsPerDay,currentSmoker))

data["Smokingstaus"]=smost

dim(data)

glu=data$glucose

dia=data$diabetes

diast=array(dim=1)

for(i in 1:3658)

{

if(glu[i]<=140 & dia[i]==0)

{

diast[i]=1

}else

{

if(glu[i]<=140 & dia[i]==1)

{

diast[i]=2

}else

{

if(glu[i]>140 & dia[i]==0)

{

diast[i]=3

}else

{

if(glu[i]>140 & dia[i]==1)

{

diast[i]=4

}

}

}

}

}

data <- subset(data, select = -c(glucose,diabetes))

data["Diabetes\_Status"]=diast

dim(data)

chol=data$totChol

cho=array(dim=1)

for(i in 1:3658)

{

if(chol[i]<=170)

{

cho[i]=1

}else

{

if(chol[i]>170 & chol[i]<=200)

{

cho[i]=2

}else

{

if(chol[i]>200)

{

cho[i]=3

}

}

}

}

data <- subset(data, select = -c(totChol))

data["Cholesterol"]=chol

dim(data)

sys=data$sysBP

dia=data$diaBP

med=data$BPMeds

A=matrix(c(sys,dia,med),ncol=3)

bpst=array(dim=1)

for(i in 1:3658)

{

if(A[i,1]<=120 & A[i,2]<=80)

{

if(A[i,3]==0)

{

bpst[i]=1

}else{bpst[i]=2}

}else

{

if(A[i,1]>120 || A[i,2]>80)

{

if(A[i,3]==0)

{

bpst[i]=3

}else{bpst[i]=4}

}

}

}

data <- subset(data, select = -c(sysBP,diaBP,BPMeds,prevalentHyp))

data["BP\_status"]=bpst

dim(data)

##FOR RANDOM FOREST CLASSIFICATION MODEL ONLY##

chd=data$TenYearCHD

A=array(dim=1)

for(i in 1:3658)

{

if(chd[i]==1)

{

A[i]="yes"

}else

{

A[i]="no"

}

}

data$TenYearCHD=as.factor(A)

dim(data)

str(data)

library(caTools)

##DATA SPLITTING IN 3:1 RATIO##

set.seed(1000)

split = sample.split(data$TenYearCHD, SplitRatio = 0.75)

train = subset(data, split==TRUE)

test = subset(data, split==FALSE)

##LOGISTIC REGRESSION##

logistic = glm(TenYearCHD ~ ., data = train, family=binomial)

summary(logistic)

#PREDICTION FROM LOGISITIC MODEL##

prediction = predict(logistic, type="response", newdata=test)

table(test$TenYearCHD, prediction > 0.5)

#Calculating AUC

library(ROCR)

pred1 = prediction(prediction, test$TenYearCHD)

as.numeric(performance(pred1, "auc")@y.values)

##DRAWING ROC CURVE##

perf1 = performance(pred1,"tpr","fpr")

plot(perf1,main="ROC CURVE FOR LOGISTIC")

##IMPROVING THE LOGISTIC REGRESSION MODEL##

logistic1 = glm(TenYearCHD ~.-prevalentStroke-heartRate, data = train, family=binomial)

summary(logistic1)

##PREDICTION AFTER REMOVING INSIGNIFICANT VARIABLES##

prediction1 = predict(logistic1, type="response", newdata=test)

table(test$TenYearCHD, prediction1 > 0.5)

##FINDING OPTIMAL THRESHOLD RISK VALUE FOR WHICH PREDICTION ACCURACY IS MAXIMUM##

acc=array(dim=1)

p=seq(.4,.6,.01)

length(p)

p[1]

for(i in 1:length(p))

{

A=table(test$TenYearCHD, prediction1 >p[i] )

acc[i]=(A[1,1]+A[2,2])/914

}

acc

plot(p,acc)

accuracy\_table= data.frame(p,acc)

accuracy\_table

##FINDING MINIMUM FNR THROUGH CONFUSION MATRIX FOR THE THREE OPTIMAL THRESHOLDS##

table(test$TenYearCHD, prediction1 > 0.46)

table(test$TenYearCHD, prediction1 > 0.53)

table(test$TenYearCHD, prediction1 > 0.56)

##FINDING THE BEST SPLIT OF TRAINING AND TESTING DATA SET FOR MAXIMUM PREDICTION ACCURACY##

ratio=seq(0.75,0.90,0.005)

ratio[1]

accuracy=array(dim=1)

for(i in 1:length(ratio))

{

set.seed(1000)

split = sample.split(data$TenYearCHD, SplitRatio = ratio[i])

train1 = subset(data, split==TRUE)

test1 = subset(data, split==FALSE)

logistic1 = glm(TenYearCHD ~.-prevalentStroke-heartRate, data = train, family=binomial)

prediction1 = predict(logistic1, type="response", newdata=test)

A=table(test$TenYearCHD, prediction1 > 0.46)

accuracy[i]=(A[1,1]+A[2,2])/(A[1,1]+A[2,2]+A[2,1]+A[1,2])

}

accuracy

plot(ratio,accuracy)

table= data.frame(ratio,accuracy)

table

##DATA SPLITTING USING OPTIMAL SPLITTING RATIO##

split = sample.split(data$TenYearCHD, SplitRatio = 0.87)

train2 = subset(data, split==TRUE)

test2 = subset(data, split==FALSE)

## FITTING OF THE FINAL LOGISTIC MODEL WITH OPTIMAL DATA SPLITTING AND WITHOUT INSIGNIFICANT PREDICTORS ##

logistic\_final = glm(TenYearCHD ~.-prevalentStroke-heartRate, data = train2, family=binomial)

summary(logistic\_final)

# PREDICTION USING THE FINAL MODEL AND OPTIMAL THRESHOLD RISK VALUE ##

prediction\_final = predict(logistic\_final, type="response", newdata=test2)

A1=table(test1$TenYearCHD, prediction\_final > 0.46)

A1

## FINAL AUC MEASURE ##

library(ROCR)

ROCRpred\_final = prediction(prediction\_final, test$TenYearCHD)

as.numeric(performance(ROCRpred\_final, "auc")@y.values)

## MAXIMIZING TPR OR MINIMIZING FNR ##

prediction\_tpr = predict(logistic\_final, type="response", newdata=train2)

tpr=array(dim=1)

p=seq(0.02,0.72,.01)

length(p)

for(i in 1:length(p))

{

A=table(train2$TenYearCHD, prediction\_tpr >p[i] )

tpr[i]=(A[2,2])/(A[2,2]+A[2,1])

}

tpr

##PLOTTING TPR AGAINST DIFFERENT THRESHOLD VALUES##

plot(p,tpr,type='l',xlab="Risk thresholds",ylab="True positive rate",col="Red",main="TPR vs Threshold Risk")

abline(a=1,b=-1,lwd=2,lty=2,col="gray")

##PREDICTING ON TEST SET USING FINAL MODEL AND ESTIMATED p WITH DESIRED TRP##

prediction\_maxtpr = predict(logistic\_final, type="response", newdata=test2)

table(test2$TenYearCHD, prediction\_maxtpr >0.02)

##RANDOM FOREST MODEL##

library(dplyr)

library(randomForest)

rfm=randomForest(TenYearCHD~.,data=train,ntree=500 ,importance=TRUE)

rfm

##PREDICTION USING RANDOM FOREST MODEL##

CHD\_pred=predict(rfm,test)

test$CHD\_pred=CHD\_pred

A=table(test$TenYearCHD,test$CHD\_pred)

A

#FINDING OPTIMAL NUMBER OF VARIABLE IN EACH SPLIT

mtry=tuneRF(train[-7],train2$TenYearCHD,ntreeTry = 600,stepFactor = 1.5,improve = 0.001, trace=TRUE,plot=TRUE)

best\_m=mtry[mtry[,2]==min(mtry[,2]),1]

mtry

best\_m

##JUDGING IMPORTANCE OF THE PREDICTOR VARIABLES THROUGH MEAN DECREASE ACCURACY##

importance(rfm)

varImpPlot(rfm)

##REFITTING THE MODEL USING BEST SPLIT AND OPTIMAL NUMBER OF DECISION TREES##

rfm1=randomForest(TenYearCHD~.-education-BMI,mtry=best\_m,ntree=600,data=train2,importance=TRUE)

rfm1

CHD\_pred=predict(rfm1,test2)

test2$CHD\_pred=CHD\_pred

A=table(test2$TenYearCHD,test2$CHD\_pred)

A

#CALCULATING AUC MEASURE##

pred1=predict(rfm1,type = "prob",newdata=test2)

library(ROCR)

pred2 = prediction(pred1[,2], test$TenYearCHD)

as.numeric(performance(perf2, "auc")@y.values)

##DRAWING ROC CURVE##

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

plot(pred3,main="ROC Curve for Random Forest",colorize=TRUE)

##OVERLAYING TWO ROC CURVES FOR COMPARISON##

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

plot(perf2,add=TRUE,colorize=TRUE)

abline(a=0,b=1,lwd=2,lty=2,col="gray")