Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

# Data Analytics

## Individual Project Report

**Ashutosh Sharma | 18304203 | Data Science**

# Contents

# PROBLEM STATEMENT

Which set of variables are the best to predict the response? Could Decision Trees and Random Forest on X and Y variables predict Response with some consideration of Group?

# ABSTRACT

In this project, the given data has been analyzed using R. The dataset contains 16 columns named Response, Group, X1-X7 and Y1-Y7. Each column had missing values except Response, Group, X4 & Y4. Due to major missing data, first data imputation is performed then post imputation models are built to predict the target response. The data imputation is performed using two methods i.e. Simple Random Sampling Imputation (SRSI) and a model-based approach using Multivariate Data Imputation by Chained Equation (MICE) [1]. Both imputation methods are compared, and mice approach is selected for data imputation as it provided a better result. After the data imputation, for predictive analysis of response two models are built i.e. Decision Tree (DT) and Random Forest (RF) using all the predictors. The DT model is first created without any threshold then pruned later to overcome the overfitting. Fully grown DT had an accuracy of 70%; Pruned DT model had an accuracy of 71.6% whereas the RF model had an accuracy of 81.6%. Therefore, we conclude that the RF model is suitable for predicting the Response. Furthermore, the group was not a good predictor as it was not an important feature for the RF ensemble.

# KEYWORDS

R, Missing Data, Simple Random Sampling Imputation (SRSI), Multivariate Data Imputation by Chained Equation (MICE) Decision Trees(DT), Random Forest(RF)

# 1 INTRODUCTION

Missing data is a common phenomenon that arises in all statistical analysis. Missing data could be imputed using various methods like mean imputation, random imputation, imputation using regression or predictor models could be used to impute the missing data [2]. If the missing data is less than 5%, the rows with missing's could be dropped from the statistical analysis but in this case, the missing data was more than 5%, therefore missing data must be imputed first before the data analysis. In Section 2 missing data pattern and missing data imputation is explained. In section 3, data analysis is performed using a DT, Pruned DT and RF. Section 4 consists of the conclusion of the report.

# 2 MISSING DATA

To identify the amount and pattern of missing data, MICE package in R is used to plot the pattern plot and missing matrix. Figure 2.1 shows the patterns in the missing data. The blue boxes represent the observed data and the red boxes represent the missing data. From the Figure 2.1, it could be observed that 130 rows are fully observed out of total 296 rows. The features Response, Group, Y4 and X4 have been completely observed i.e. no missing values. From the Figure 2.1, it could be also inferred that there is a connection between the missingness of features as pairs i.e. features X1-Y1, X2-Y2, X3-Y3, X6-Y6, X7-Y7 are missing together as pairs. Furthermore, features X2, X3, Y2 and Y3 are with most missing cases i.e. consists of 90 missing values.

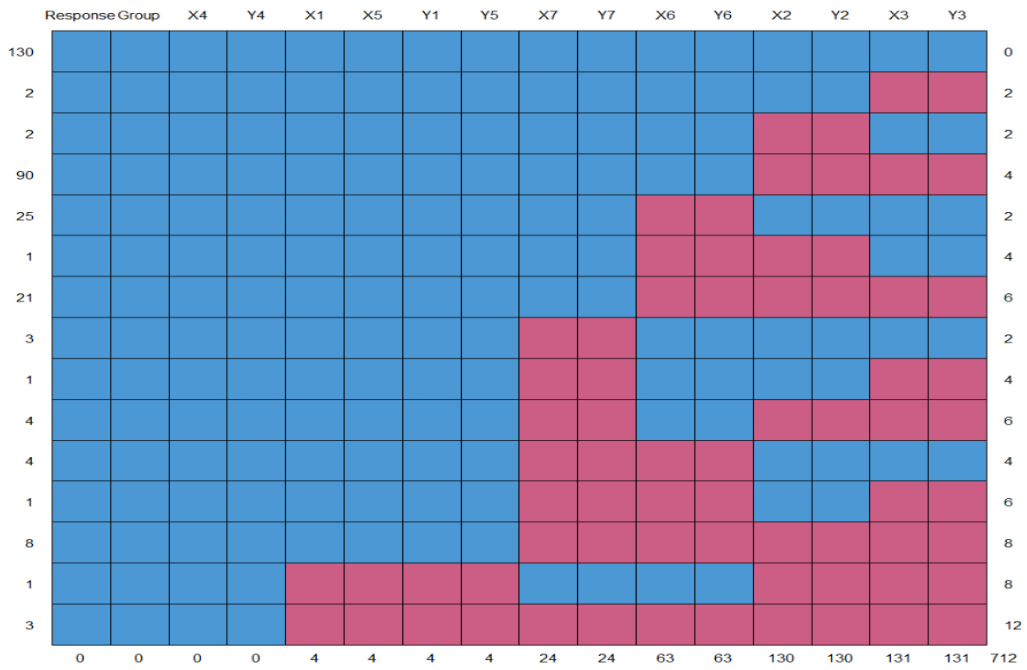


**Figure 2.1:** *Missing data pattern plot*

| | Response | Group | X4 | Y4 | X1 | X5 | Y1 | Y5 | X7 | Y7 | X6 | Y6 | X2 | Y2 | X3 | Y3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 130 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 2 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 2 |
| 90 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 4 |
| 25 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 2 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| 21 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 6 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 4 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 6 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 6 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 8 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 | 24 | 24 | 63 | 63 | 130 | 130 | 131 | 131 | 712 |

**Figure 2.2:** *Pairwise matrix when both features are missing*

As we observed that the data is missing in pairs, thus we perform a pairwise analysis of data. The $mm matrix (md.pairs) from mice package is plotted. From above Figure 2.2, it could be observed that the missingness occurs in similar number i.e. # missing of X's is equal to # missing of Y's, pairwise. Therefore, first impute the X's then use the imputed X's pairwise to impute corresponding Y's i.e. X1 used to imputed Y1, X2 used to imputed Y2 so on.

## 2.1 Original Data (Non-Missing i.e. missing data dropped)

Before imputing the missing data, we plot the histogram for the numerical columns i.e. all the X's. To plot the histogram, the missing rows are dropped and only fully observed rows are considered i.e. 130 rows. Figure 2.3 shows the histogram plot for the complete cases. From the figure, it could be inferred that the X1, X2, X3, X4 & X7 columns have left-skewed distributions whereas X5 has a bimodal and X6 has a normal distribution.

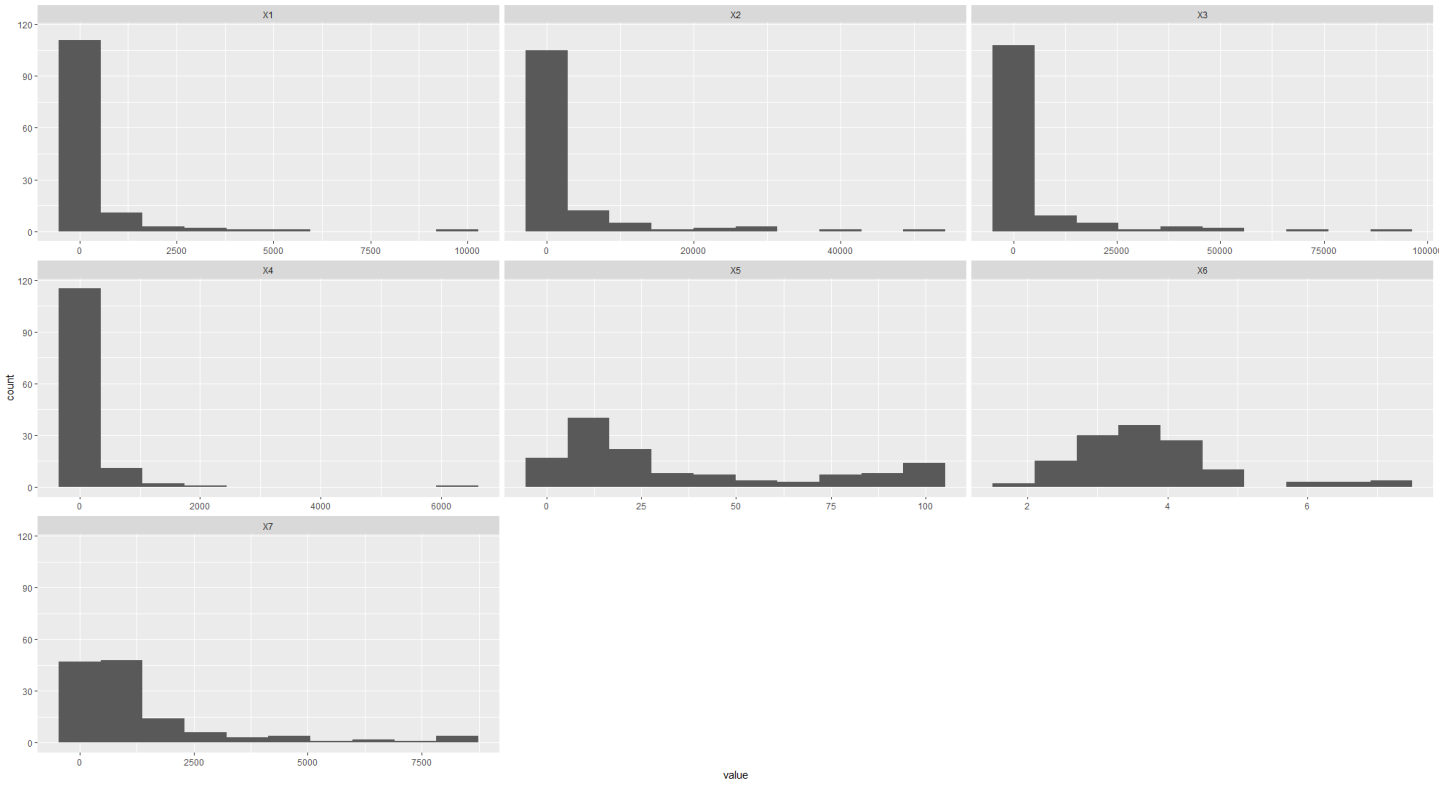


**Figure 2.3:** *Histogram without data imputation*

## 2.2 SIMPLE RANDOM SAMPING IMPUTATION (SRSI)

In this approach, missing data is imputed by randomly sampling from the observed data. All the features with missing data i.e. all features X's and Y'S are imputed directly in SRSI by randomly sampling from the observed data. SRSI is performed in the following steps:

- Repeat for each feature
- Count the number of missing values as n
- Randomly pick n samples from the observed values for the feature and impute using the selected samples

Figure 2.4 shows the histograms of SRSI. SRSI does not distort the distribution of all the columns, however, distributions of features X6 & X7 show significant change.

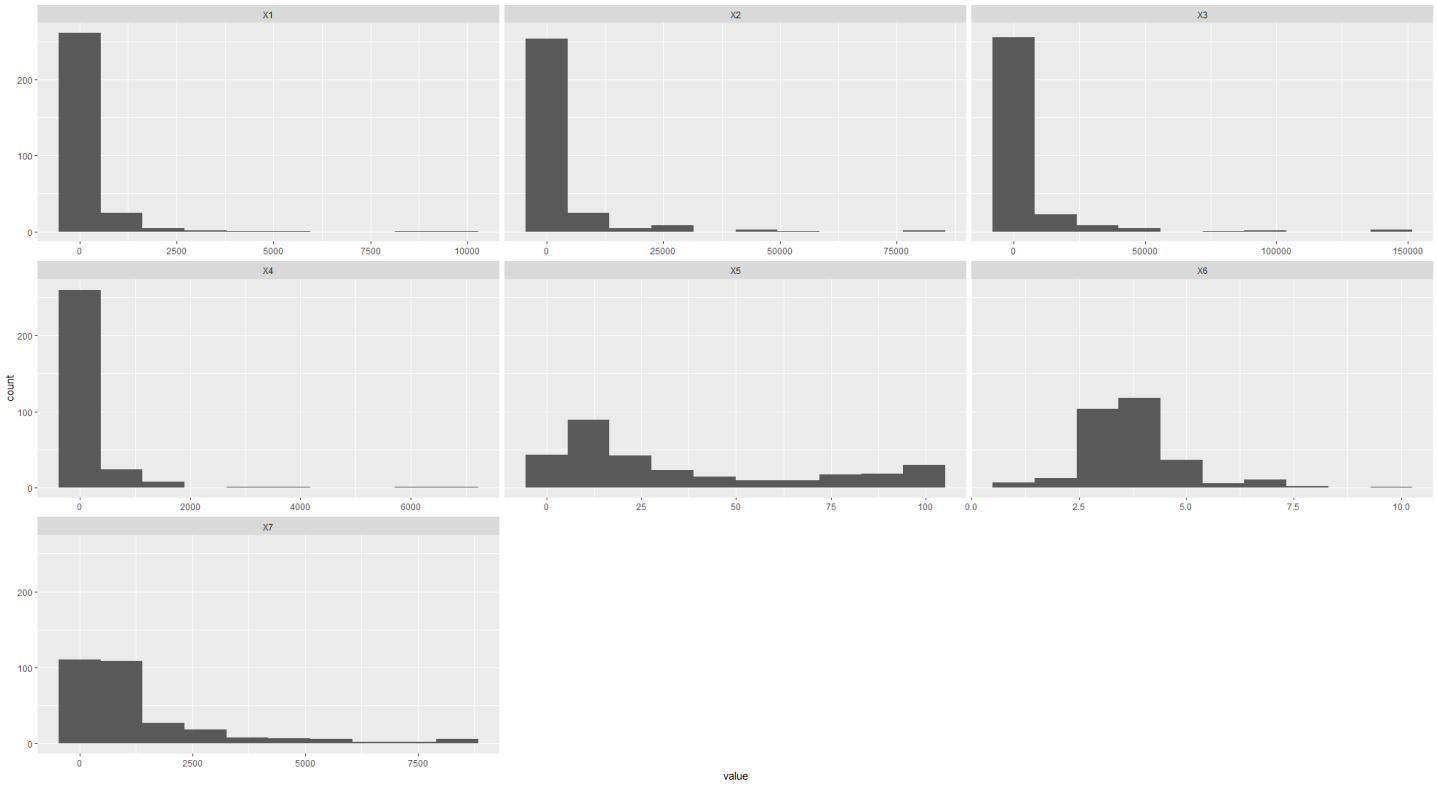


**Figure 2.4:** *Histogram of simple random data imputation*

Missing data for all X and Y features were imputed using SRSI. However, as Y's are categorical features, thus they not plotted in the histogram.

## 2.2 MULTIVARIATE DATA IMPUTATION BY CHAINED EQUATIONS(MICE)

As simple random imputation does not show a good fit for all the feature, therefore, data imputation was performed using MICE. MICE package is available in R which performs data imputation by using chained equation. MICE use unrealistic assumptions multiple times to draw the distribution of the missing data. Then the result is pooled from the many predicted distribution thus final convergence is achieved [1]. Figure 2.5 below shows the steps performed by mice for data imputation.

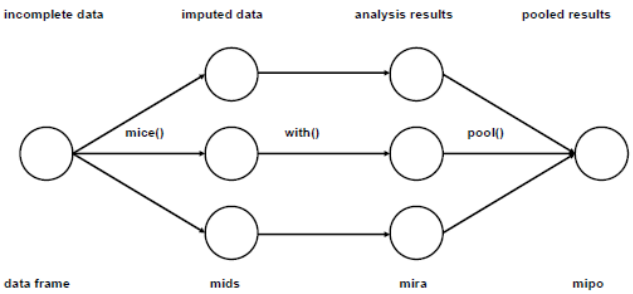


**Figure 2.5:** *MICE steps*

For mice to perform imputation hyperparameter called method needs to be specified externally by the user. RandomForest method was used to model the missing data with MICE. We first impute the missing data of columns X1,

X2, X3, X5, X6, X7 using mice. Thus, to build the RandomForest model in mice, only X's and Group is provided as the input data and other variable i.e. Y's and Response were removed. Below figure 2.6 is the histogram plot that shows the imputation of X's using mice (RandomForest).

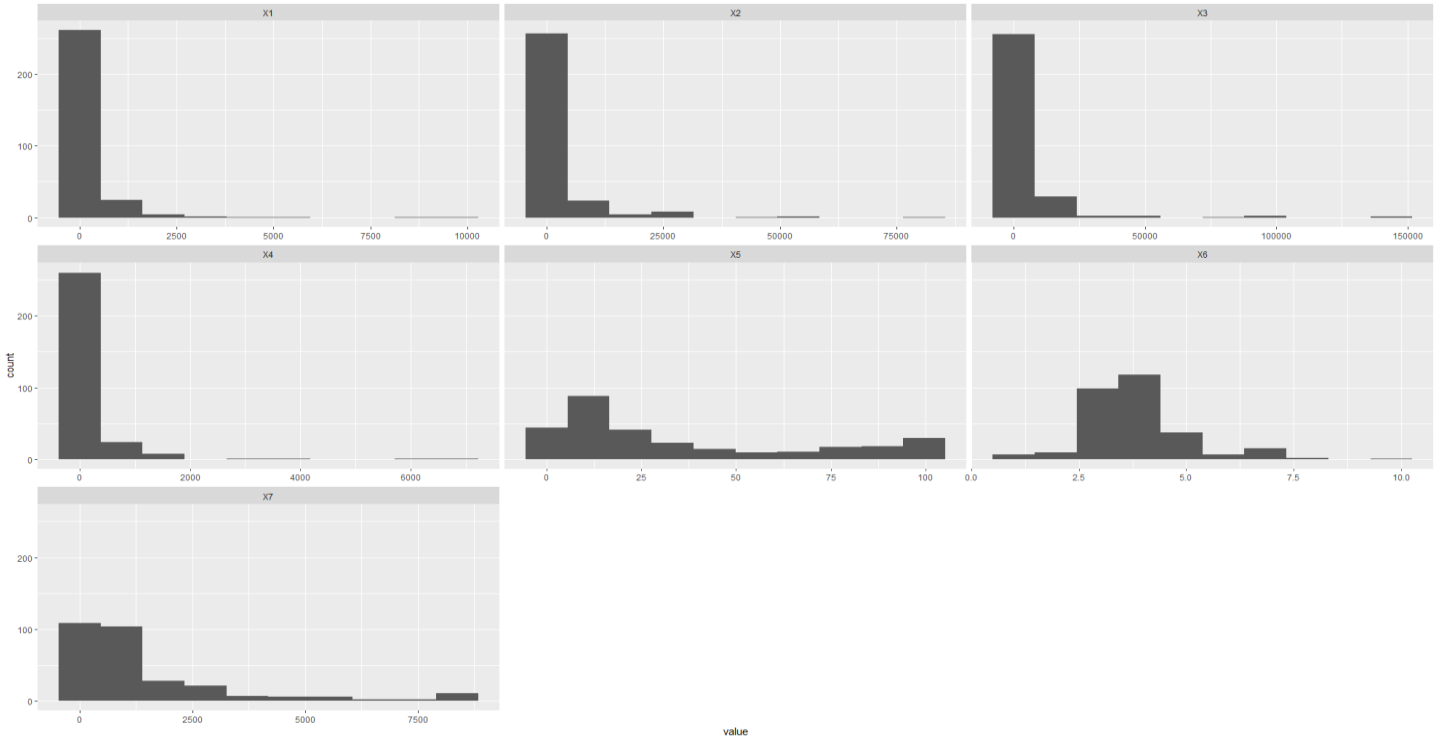


**Figure 2.6:** *Histogram of RandomForest data imputation using mice*

After imputing X's, we plotted the distribution density plot as show in the below figure 2.7. As per mice package if the imputed distribution i.e. the red lines overlap with the blue line the data is Missing at Random (MAR). From the figure it could be observed that X1, X2, X3 and X6 could be missing at random.

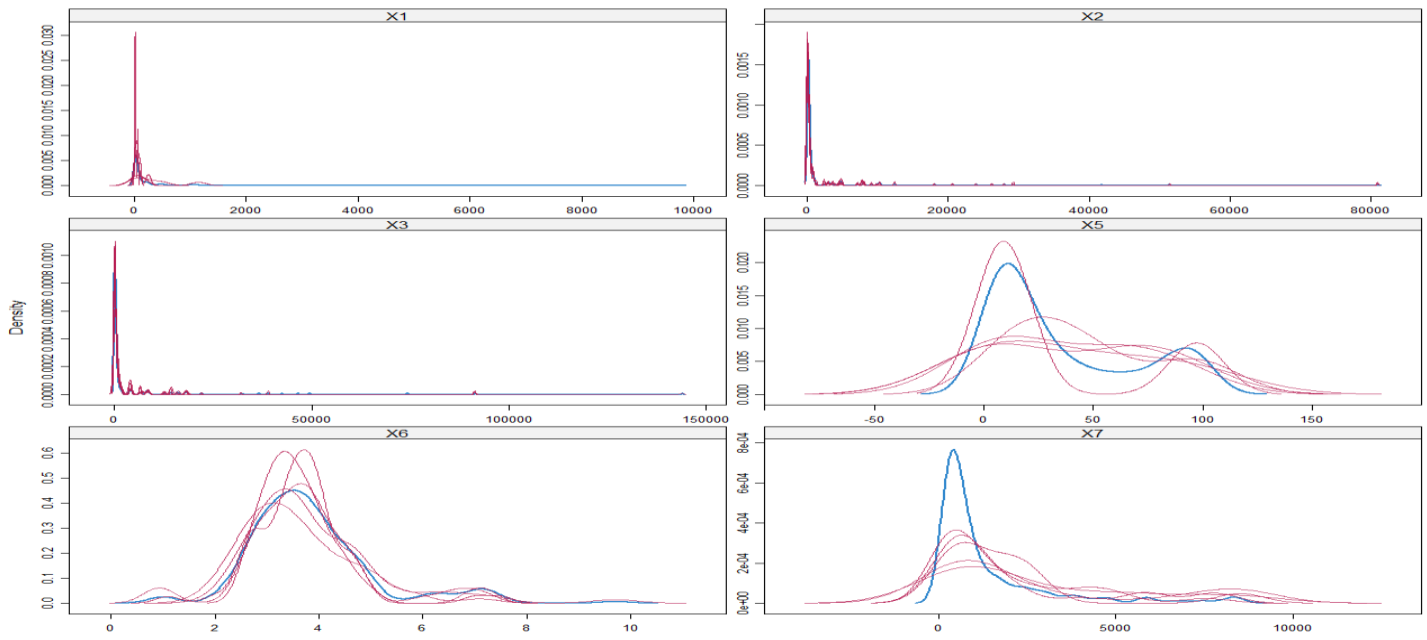


**Figure 2.7:** *mice imputation iteration density plot*

Furthermore, before imputing the Y, a manual observation of Y was performed, and it was noticed that a strong correlation exists between X variables with Y. After imputing X variable, the X features and the group were used to impute Y.

In the manual data observation, it was also noticed that all the Y variables followed simple rules as shown in below table 2.1. Thus, few features were imputed by directly using these rules. The feature X6 was imputed using RandomForest from mice, whereas X5 and X7 were imputed using Logistic regression from mice. Only X5, X6, Y5, Y6 and Group were passed as the imputation features and it was observed that the rule that was observed manually was automatically implemented perfectly by mice.

| # | Feature | Condition | Y output |
|---|---|---|---|
| 1 | X1 - Y1 | X1 < 34 | Y1 = 0 |
| | | Otherwise | Y1 = 1 |
| 2 | X2 - Y2 | X2 < 279 | Y2 = 0 |
| | | Otherwise | Y2 = 1 |
| 3 | X3 - Y3 | X3 <= 240 | Y3 = 0 |
| | | Otherwise | Y3 = 1 |
| 4 | X5 - Y5 | X5 =< 11.39 | Y5 = 0 |
| | | X5 > 11.39 AND X5<= 29.87 | Y5 = Opposite to the value of Group |
| | | Otherwise | Y5 = 1 |
| 5 | X6 - X6 | X6 <= 1.1 | Y6 = 0 |
| | | X6 > 1.1 AND X6 > 3.5 | Y6 = 1 |
| | | Otherwise | Y6 = 2 |
| 6 | X7 - Y7 | X7 <=500 | Y7 = 0 |
| | | Otherwise | Y7 = 1 |

**Table 2.1:** *mice imputation iteration density plot*

## 2.3 IMPUTATION COMPARISON

Results of both types of imputations were compared against each other and with the original fully observed data. Both the histograms of simple random imputation and mice look similar in figures 2 and figure 3. Thus, we perform an in-depth comparison of both the methods.

Results of both types of imputations were compared against each other and with the original fully observed data. Both the histograms of simple random imputation and mice look similar in figures 2 and figure 3. Thus, we perform an in-depth comparison of both the methods.

```
> #Mean of orignal data with missings removed
> summary(df_orignal[c(3:9)])
       X1               X2               X3               X4               X5               X6               X7        
 Min.   :   5.0   Min.   :    0.00   Min.   :     0   Min.   :  21.82   Min.   : 0.100   Min.   :0.900   Min.   : 110.3  
 1st Qu.:  16.0   1st Qu.:   30.25   1st Qu.:    40   1st Qu.:  50.61   1st Qu.: 9.057   1st Qu.:3.100   1st Qu.: 368.1  
 Median :  38.0   Median :  126.00   Median :   192   Median :  71.83   Median :19.300   Median :3.600   Median : 653.2  
 Mean   : 301.3   Mean   : 2908.60   Mean   :  5015   Mean   : 233.34   Mean   :35.317   Mean   :3.836   Mean   :1353.1  
 3rd Qu.: 186.0   3rd Qu.:  558.75   3rd Qu.:   880   3rd Qu.: 132.38   3rd Qu.:61.970   3rd Qu.:4.300   3rd Qu.:1519.2  
 Max.   :9743.0   Max.   :80919.00   Max.   :143856   Max.   :6864.00   Max.   :99.800   Max.   :9.700   Max.   :8491.1  
 NA's   :4        NA's   :130        NA's   :131                        NA's   :4        NA's   :63      NA's   :24      
> #Mean of missings imputed using random imputation by random sampling
> summary(df_random_imputed[c(3:9)])
       X1               X2               X3               X4               X5               X6               X7        
 Min.   :   5.0   Min.   :    0   Min.   :     0   Min.   :  21.82   Min.   : 0.100   Min.   :0.900   Min.   : 110.3  
 1st Qu.:  16.0   1st Qu.:   33   1st Qu.:    43   1st Qu.:  50.61   1st Qu.: 9.092   1st Qu.:3.100   1st Qu.: 373.1  
 Median :  36.5   Median :  134   Median :   192   Median :  71.83   Median :19.155   Median :3.600   Median : 653.2  
 Mean   : 301.0   Mean   : 3068   Mean   :  5428   Mean   : 233.34   Mean   :35.268   Mean   :3.762   Mean   :1340.3  
 3rd Qu.: 186.0   3rd Qu.:  504   3rd Qu.:   767   3rd Qu.: 132.38   3rd Qu.:61.970   3rd Qu.:4.200   3rd Qu.:1493.3  
 Max.   :9743.0   Max.   :80919   Max.   :143856   Max.   :6864.00   Max.   :99.800   Max.   :9.700   Max.   :8491.1  
> #Mean of missings imputed using RandomForest by mice package
> summary(df_rf_imputed[c(2:8)])
       X1               X2               X3               X4               X5               X6               X7        
 Min.   :   5.0   Min.   :    0.0   Min.   :     0   Min.   :  21.82   Min.   : 0.100   Min.   :0.900   Min.   : 110.3  
 1st Qu.:  16.0   1st Qu.:   30.0   1st Qu.:    40   1st Qu.:  50.61   1st Qu.: 9.008   1st Qu.:3.100   1st Qu.: 373.4  
 Median :  39.0   Median :  120.0   Median :   226   Median :  71.83   Median :19.300   Median :3.650   Median : 663.3  
 Mean   : 301.7   Mean   : 2643.4   Mean   :  4849   Mean   : 233.34   Mean   :35.380   Mean   :3.812   Mean   :1467.4  
 3rd Qu.: 186.0   3rd Qu.:  531.8   3rd Qu.:  1200   3rd Qu.: 132.38   3rd Qu.:62.400   3rd Qu.:4.300   3rd Qu.:1641.7  
 Max.   :9743.0   Max.   :80919.0   Max.   :143856   Max.   :6864.00   Max.   :99.800   Max.   :9.700   Max.   :8491.1  
```

**Figure 2.8:** *Summary of X's non-missing(original) and imputed*

The above R output figure 2.8 shows the summary of the original, simple random imputed and mice RandomForest imputed X's. The below bar graphs are plotted to compare the mean and median of data, before and after the imputation.

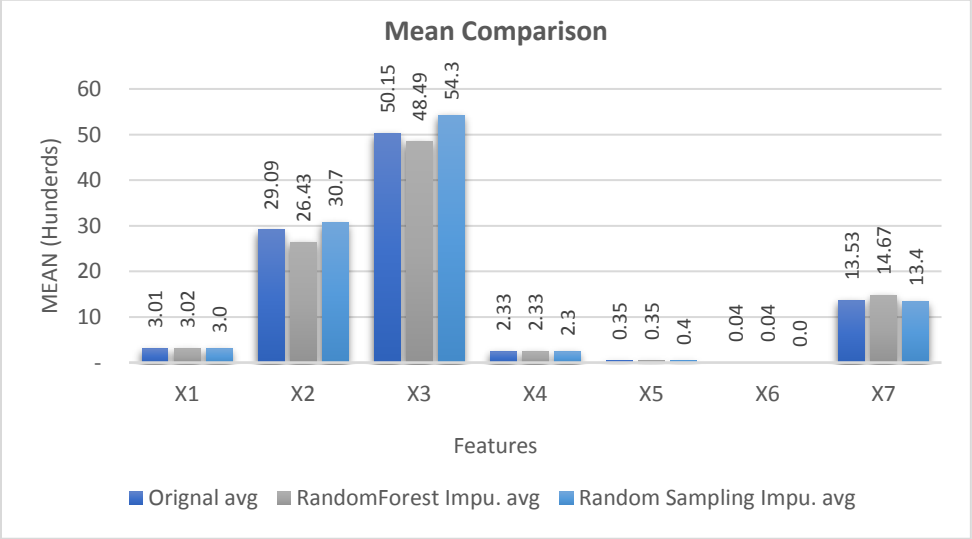


**Figure 2.9:** *Mean, bar graph of imputed and original X's*

From figure 2.10, we can observe that RandomForest imputation with mice gives us the best result as the mean and median are closest to the original non-missing data.

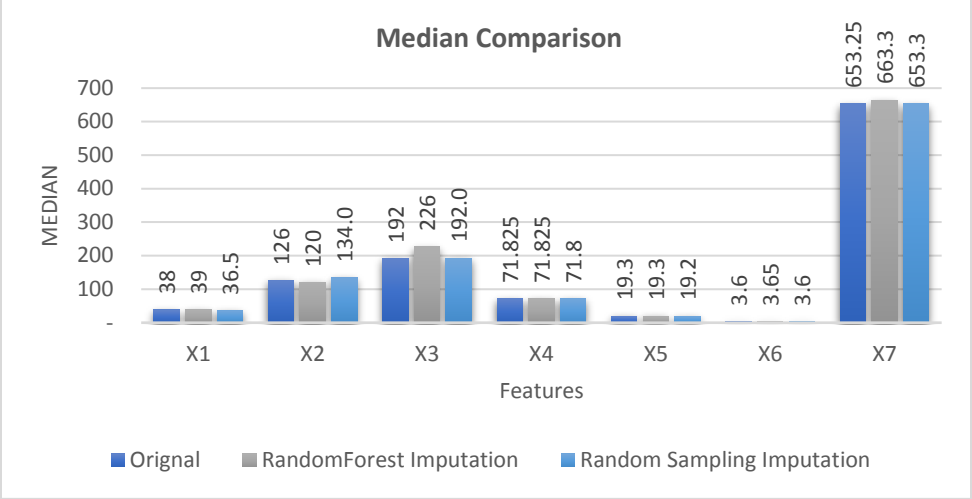


**Figure 2.10:** *Median, bar graph of imputed and original X's*

| Feature | % Δ MEAN | | % Δ MEDIAN | | % Δ MOD | |
|---|---|---|---|---|---|---|
| | RF | SRS | RF | SRS | RF | SRS |
| X1 | 0% | 0% | 3% | 7% | 0% | 0% |
| X2 | 10% | 14% | 5% | 10% | N/A | N/A |
| X3 | 3% | 11% | 15% | 18% | N/A | N/A |
| X4 | 0% | 0% | 0% | 0% | 0% | 0% |
| X5 | 0% | 0% | 0% | 1% | 0% | 0% |
| X6 | 1% | 1% | 1% | 1% | 14% | 13% |
| X7 | 8% | 9% | 2% | 2% | 97% | 3136% |

**Table 2.2:** *Percentage comparison of Mean, Median & Mod of original non-missing data with RF and SRIS*

By using the bar graph plot it was not clear to identify the best-imputed result, therefore, a % comparison table was plotted. By the above table 2, RF imputation performs better for all the X's by causing least distortion in the distribution. From the above table 2, it could also be observed that SRIS increase the mod of X7 by 3136% whereas RF increases the mod of X7 by 100%. To further compare the imputations the correlation heatmaps of all three approaches i.e. original non-missing, RF imputed, SRIS for X's have been plotted in Figure 2.11, 2.12, 2.13.

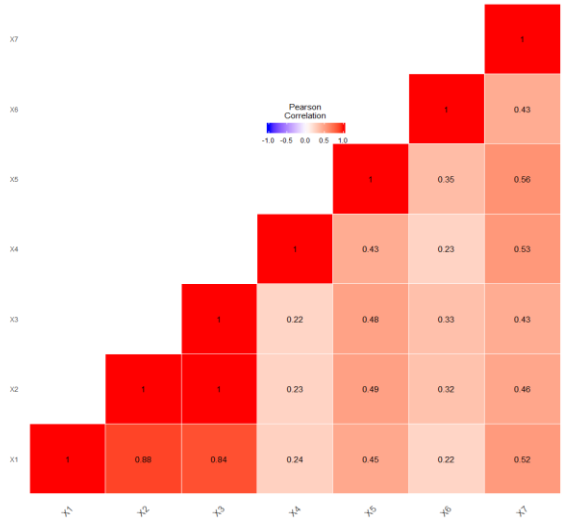

**Figure 2.11:** *Correlation heatmap of original X's*

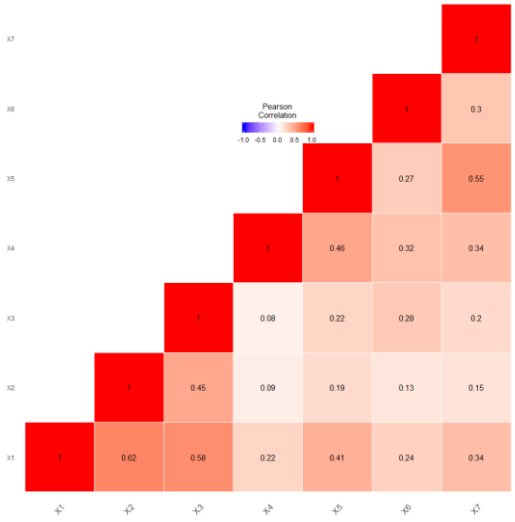

**Figure 2.12:** *Correlation heatmap of SRSI X's*

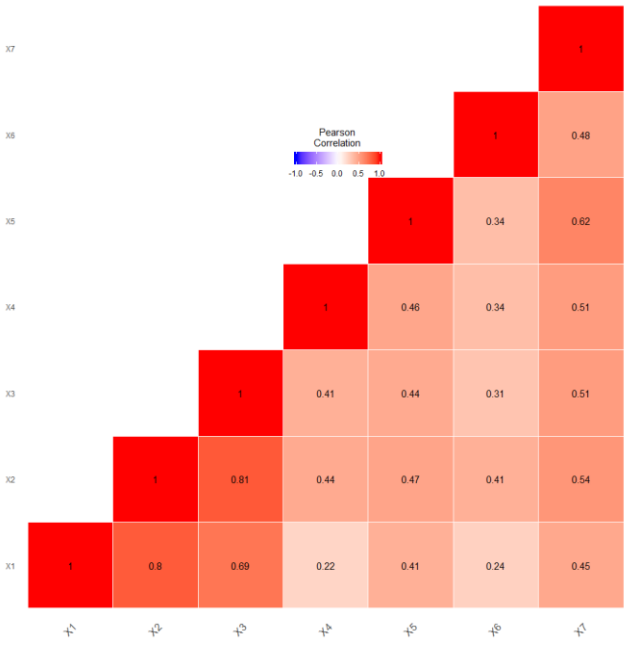


**Figure 2.13:** *Correlation heatmap of RF mice X's*

From the above correlation heatmaps, it was observed that the simple random sampling removes the correlation between the X's thus making the data meaningless.

## 2.4 IMPUTATION CONCLUSION

From the above results of the comparison, the RF imputation opted for the imputations of X's. Similarly, we used the rule mentioned in Table-1, Logistic Regression and RF from mice to impute the Y's.
Using mice RF, Logistic Regression and rule-based approach all the was successfully imputed with minimal damage to the data distributions.

# 3 DATA ANALYSIS

The data analysis is performed to build a model to predict the Response variable. Two approaches i.e. DT and RF were used to create the model. The final model is selected based on the accuracy of the model.
First, the data is split into the training set and testing set with a ratio of 8:2 i.e. 80% data for training the model and 20% for testing the model. The data split was performed by randomly sampling from the dataset without replacement. Then the model is trained using the training set. The test set is used for model evaluation.

## 3.1 DECISION TREE (DT)

Decision Trees (DT) are powerful and popular tools for classification and regression. DT represent rules, which can be understood by humans and used in a knowledge system such as a database. As in our case, the target feature response is a categorical variable, thus it was first converted to factor in R as the goal is to build a classification tree model. To build the DT model no external threshold parameter was set to stop tree creation, thus the most complex overfitted tree possible is built. Furthermore, this overfitted tree is pruned to get a better result. Below figure 3.1 show the DT plot without any threshold parameter.

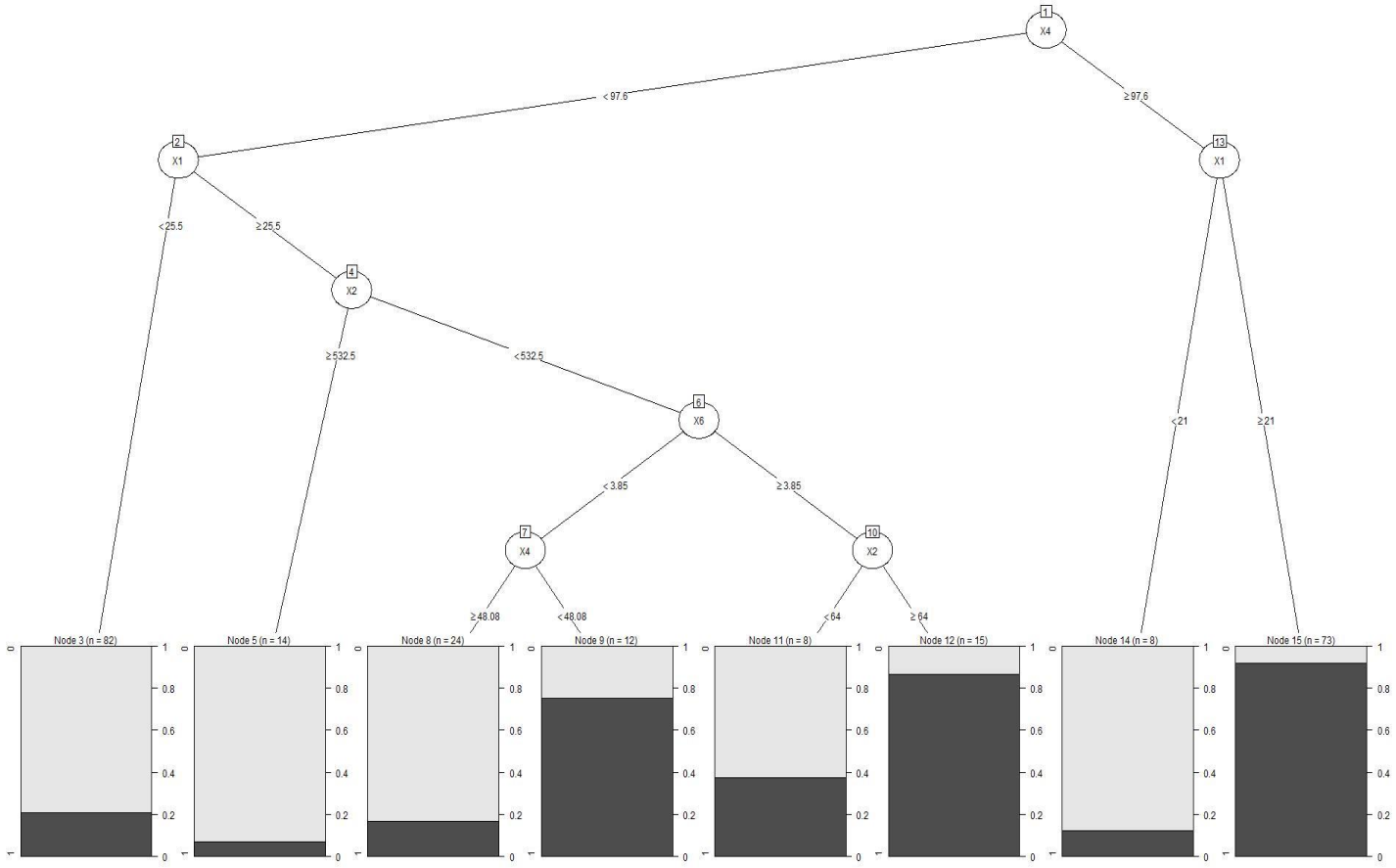


**Figure 3.1:** *DT party plot without any threshold*

The above figure 3.1 is a DT party plot using all the features as input to predict the response. It could be seen from the figure that X4 is the most important feature for predicting response followed by X1, X2, X6 and X4. The test set is used to evaluate the accuracy of this DT. This DT model achieved an accuracy of 70% in predicting the response. The below Figure 3.2 is the confusion matrix of the predictions on the test set.

| | | Predicted Response | |
|---|---|---|---|
| | | 0 | 1 |
| Actual Response | 0 | 24 | 9 |
| | 1 | 9 | 18 |

**Figure 3.2***: DT confusion matrix on test set*

As no external parameter was set to control the DT growth, it is suspected that the DT is overfitted. A CP (Complexity Paramter) plot of the above DT model was plotted to verify the same. The below figure 3.3 shows the cp plot of the above DT. From the below figure 3.3, we can observe that the cross-validation error (XERROR) is minimum at CP=0.021.

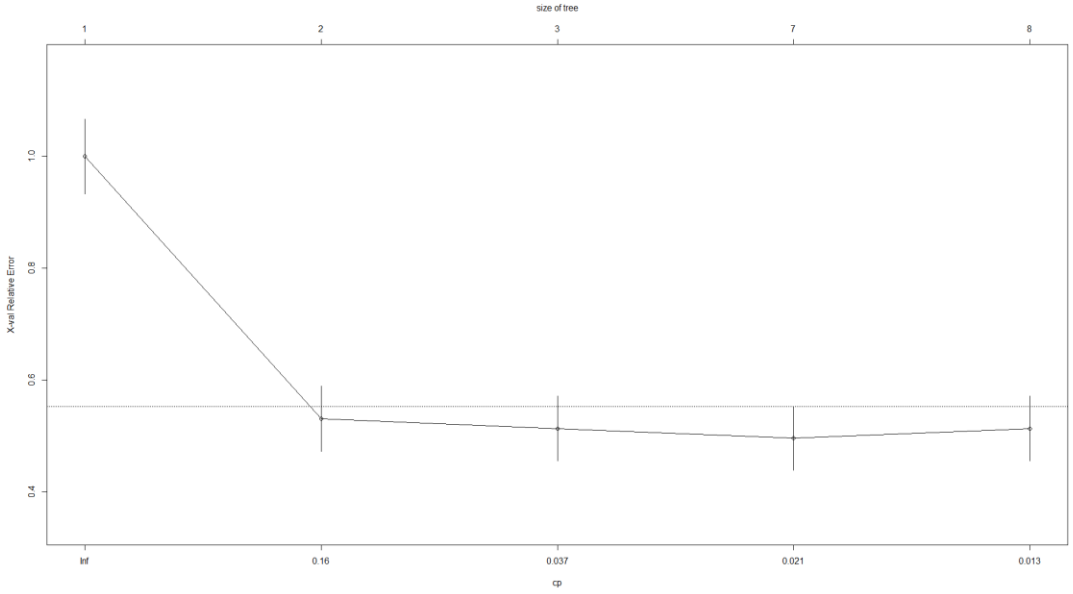


**Figure 3.3:** *CP Plot of full DT*

## 3.2 PRUNED DECISION TREE

To prune the DT, CP value is set such that the cross-validation (XERROR) is minimum i.e. CP=0.021. The DT created earlier is pruned setting this new CP threshold. The below figure 3.4 shows the DT after pruning. From the below figure it can be observed that the X2 split (10) is now removed.

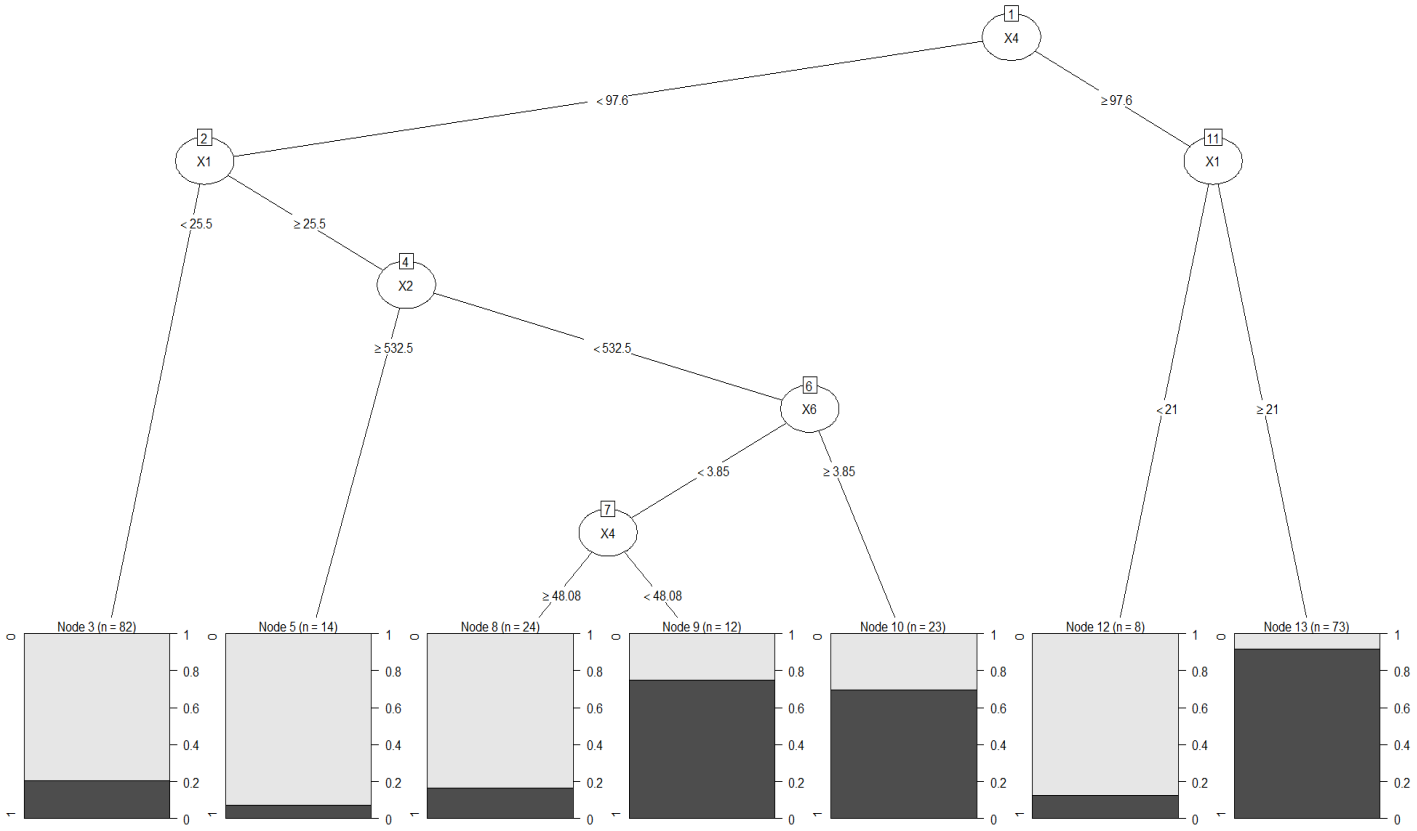


**Figure 3.4:** *Pruned DT with CP=0.021*

The accuracy of this pruned DT is calculated over the test set. This pruned DT achieved an accuracy of 71.7% in predicting the response. Thus, using DT pruning, a 1.7% accuracy increment was achieved. Below figure 3.5 shows the confusion matrix for this pruned DT.

| | | Predicted Response | |
|---|---|---|---|
| | | 0 | 1 |
| Actual Response | 0 | 24 | 8 |
| | 1 | 9 | 19 |

**Figure 3.5:** *Pruned Decision tree confusion matrix of test set*

## 3.2 RANDOM FOREST (RF)

RF (RF) is an ensemble method that uses bagged DT. RF work on the principle of random sub spacing i.e. samples are selected with replacement. In RF sampling not, all features are available while sampling. Therefore, RF provides each feature opportunity as the best split feature might not be available while sampling. RF creates many decorrelated DT and a majority voting is used from all the trees while making the prediction. RF is not prone to overfitting.

RF is applied to this data set to predict the response. RF package in R is used to build the model. The same training data, used for DT is used to build the RF model. All the available features are used for training the model. Below figure 3.6 shows the R output of the RF model and figure 3.7 shows the plot of error with the number of trees. The black line is the combined error whereas the green and red line represent 0 &1 error for Response.

```
Call:
 randomForest(formula = Response ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 25.42%
Confusion matrix:
   0  1 class.error
0 96 25   0.2066116
1 35 80   0.3043478
```

**Figure 3.6:** *RF R output*

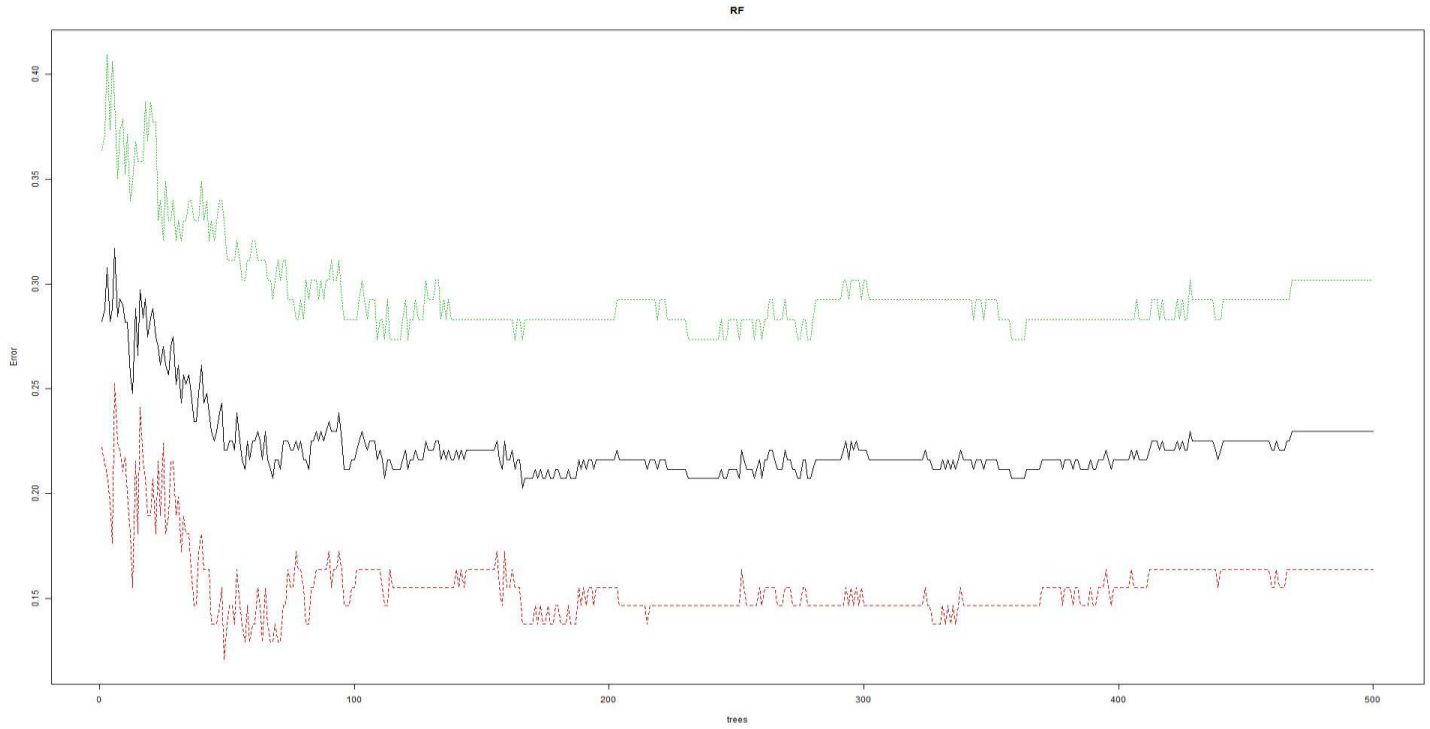


**Figure 3.7***: RF Error vs number of trees*

It can be observed from the above figures that the out of bag error reduces as the number of trees increases in the RF ensemble. The minimum out of bag (OOB) error is 25.42%. As we need to identify the best predictors for Response, variable importance plot is created. RF variable importance plot is shown in Figure 3.8. It could be observed from the figure that X4, X1, X5, X7, X6 are the top 5 predictors for the response as these features have the maximum mean Gini decrease.

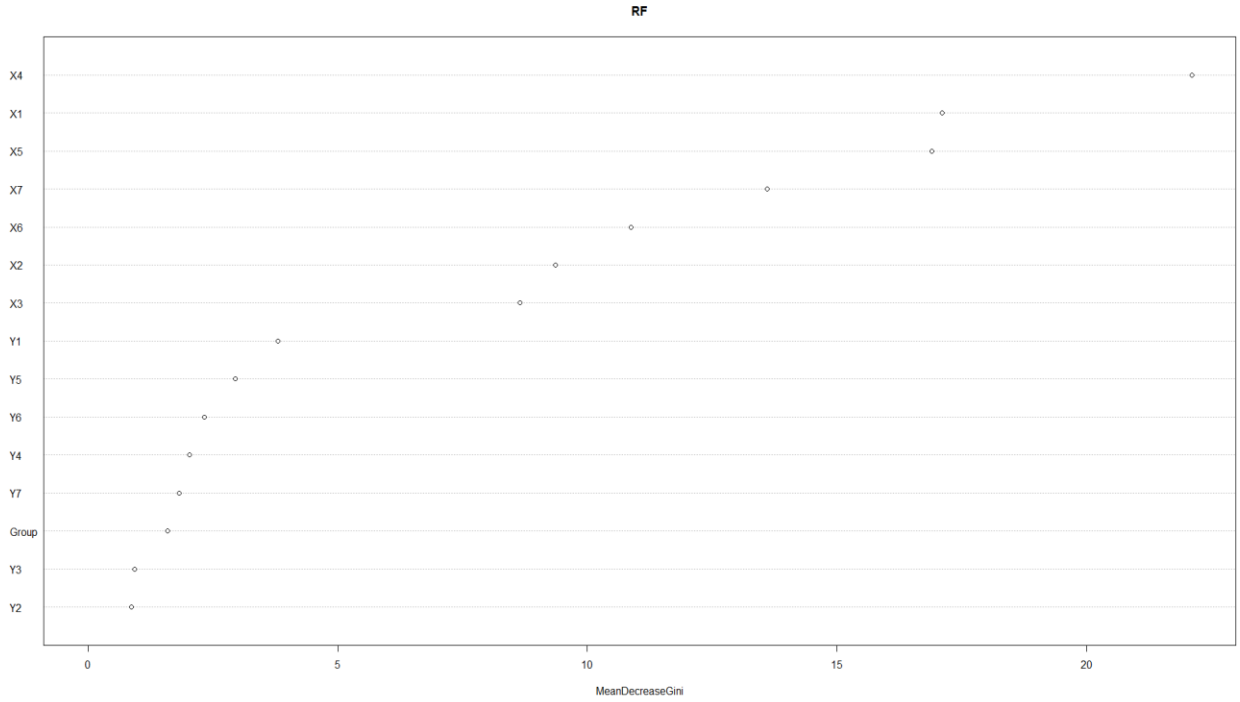


**Figure 3.8:** *RF variable importance*

The RF model is evaluated over the test set for accuracy. RF model achieved an accuracy of 81.67% on the test set. Below figure 3.9 is the confusion matrix of RF mode.

| | | Predicted Response | |
|---|---|---|---|
| | | 0 | 1 |
| Actual Response | 0 | 28 | 6 |
| | 1 | 5 | 21 |

**Figure 3.9:** *RF confusion matrix on test set*

# 4 CONCLUSION

The RF mice imputation worked better then SRSI with less distortion to the distributions. Also, it was noticed that the feature Y5 is related to X5 and group both. This rule was captured by the logistic regression imputation model implemented using mice.

The DT model achieved an accuracy of 71.7% whereas the RF achieved an accuracy of 81.7% over the test set. Thus, we can say that RF performs much better in predicting the Response.

The important predictors for the target response are as following:

1. X4
2. X1
3. X5
4. X7
5. X6
6. X2
7. X3

The group is not a good predictor for target response as the mean Gini decrease of this feature is very low in RandomForest model.

## REFRENCES


[1] v. Stef and G.-O. Karin, "mice: Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software,* vol. 45, no. 3, 2011.

[2] A. Gelman and H. Jennifer, "Missing-data imputation," in *Data Analysis Using Regression and Multilevel/Hierarchical Models*, New York, Cambridge University Press, 2012, pp. 529-544.