USE CASE ON CLASSIFICATION OF BANK MARKETING DATASET

Team Members

Nikhil (001277045)

Manoj (001859255)

Data Mining in Engineering

(IE 7275)

Faculty Advisor

Prof. Sagar Kamarthi

Teaching Assistant

Ramin Mohamamdi

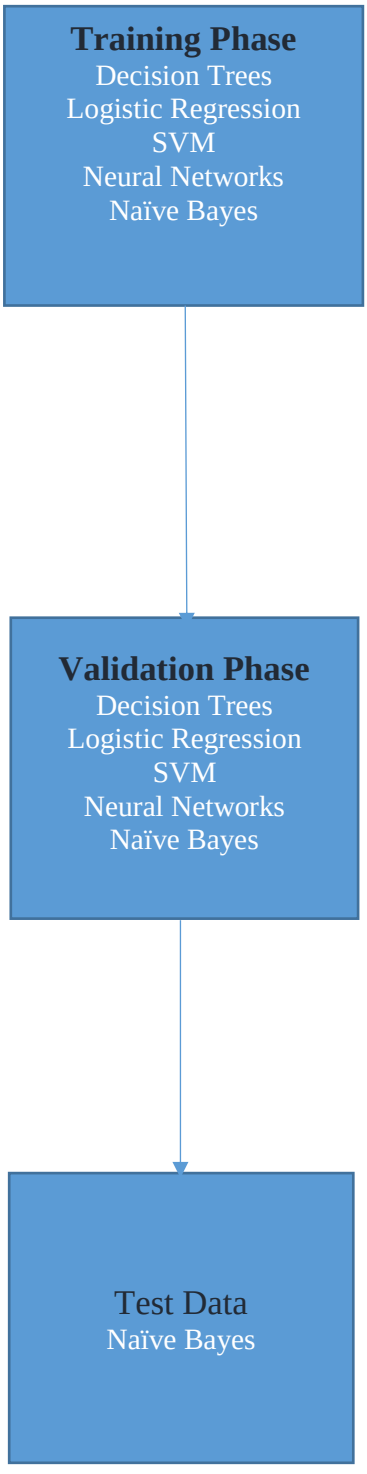
Table of Contents

Abstract

Banking Industry is prominent contributor for economy of any country. It was the main cause of the global economic crisis in 2008 because of the bad loan deposits. The project aims in finding the potential customers who are likely to take term deposit based on the marketing campaigns done by Portuguese Banking Industry. The marketing campaigns were done based on phone calls. More than one contacts to the same customer was required, in order to access if a customer would take term deposit. The dataset contains 41188 customer records with 20 predictor variable ordered by date from May 2008 to November 2010. The data was partioned to 60% of training 20% validation and 20% test data. The best model was decided by true positive and False negative critieria. Class of Interest was customer accepting the term deposit. Classifying an non-acceptor as potential customer wasn't a problem but the converse would be big loss to the marketing. Five model were implemented to on training data and the best model was decided on validation data. The best model was applied on test data.

Methodology

Machine learning methodology was used to determine the final model. The basic outline and methods used are mentioned below.

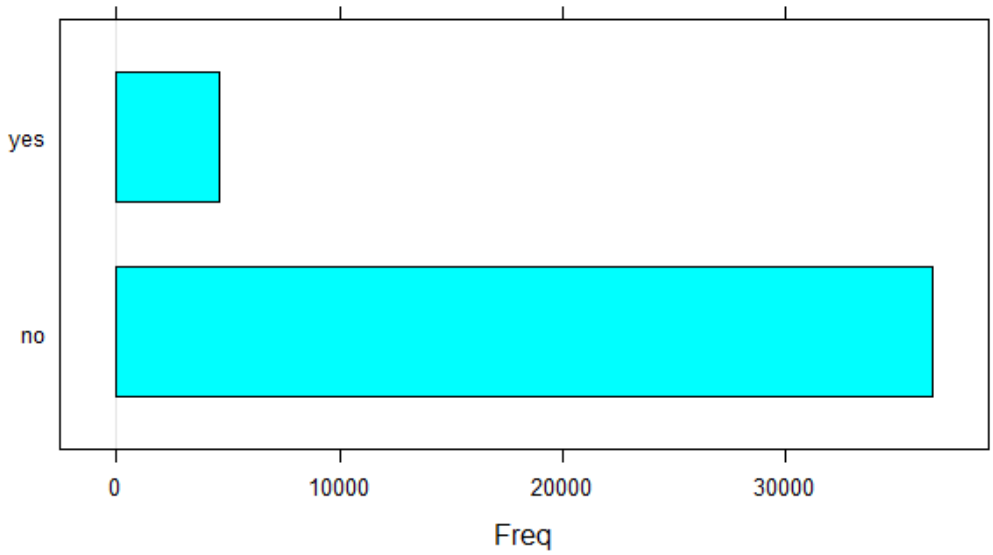


Data Description and Preprocessing

Variable Name	Type of category	Transformed categories
Age	Numeric	Not tranformed
Job	Catgegorical with admin, Blue collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown	**High-Pay-Job**(Admin, Blue-collar, management, services) **self-pay-job**(self-employed,technician, enterprenuer**) No-pay-Job**(student, technician, enterprenuer)
Maritial	Catgegorical with 'divorced','married','single','unknown'	Not transformed
Education	Catgegorical with Basic 4y, basic 6y, high school, illiterate, professional.course, university degree, unknown	**Basic Education**(Basic 4y, basic 6y, illiterate) **High School**(basic 9y, high school and unknown) **Univ&pro** (Professional Course and university Degree)
Default	Categorical-Yes, No and Unknown	Not tranformed
Housing	Categorical-Yes, No and Unknown	Not tranformed
Laon	Categorical-Yes, No and Unknown	Not tranformed
Contact	Categorical – Cellular, telephone	Cellular as 1 and telephone as 0
Month	Categorical- Jan-Nov	Q1, Q2, Q3,Q4
Day_of_Week	Categorical- Mon-Sun	Mon-Sun
Duration	Numerical variable Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.	Removed from the dataset as it us not used for predicting task.
Campaign	Numeric	Not tranformed
Pdays	Numeric	Transformed to new and previous customer. New Customers are 0 and Old customers as 1
Previous	Numeric	Not tranformed
Poutcome	Categorical Failure, Non-Existent, Success	Not tranformed
Emp.Var.rate	Numeric- Employment Variation Index	Not tranformed
Cons.price.Idx	Numeric-Consumer Price Index	Not Transformed
Cons.Conf.indx	Numeric-Consumer Confidence Indx	Not Transformed
EuriBor3M	Numeric- Eurobor 3month Rate	Not transformed
Nr.employed	Numeric-Number of employees	Not transformed.
Y (Response Variable)	Categorical- Yes(Accepted term Deposit) No(Didn’t Accept)	Transformed to 0 and 1, 1 took deposit 0 didn’t take

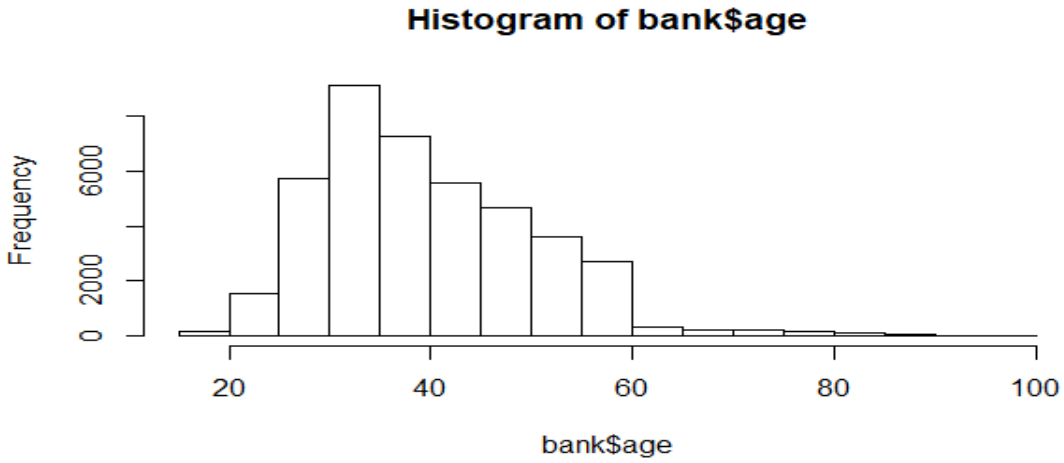
Data Visualization

Output variable (desired target):
y - has the client subscribed a term deposit? (binary: "yes","no")

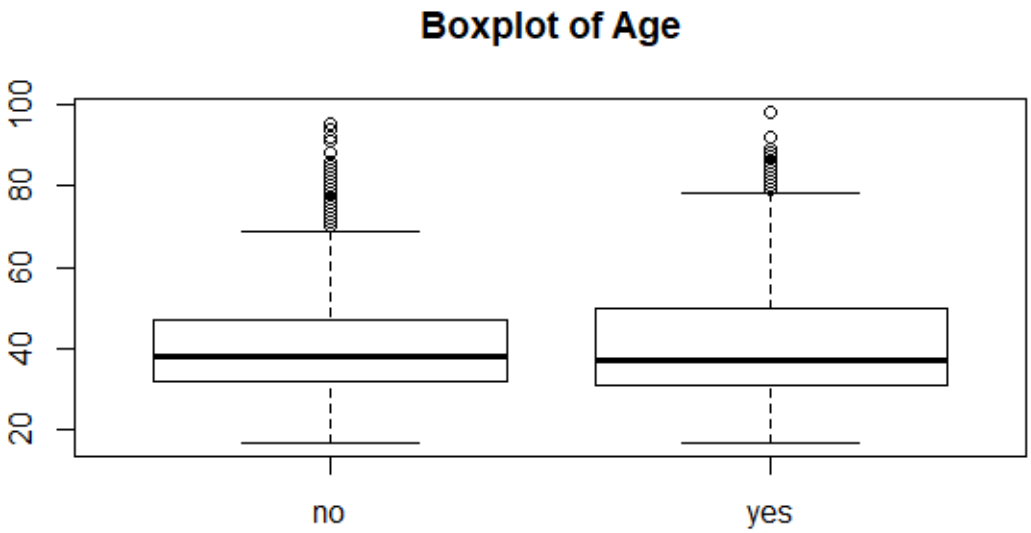


The Plot shows that most customers didn't prefer the term deposit.

The Distribution of Age

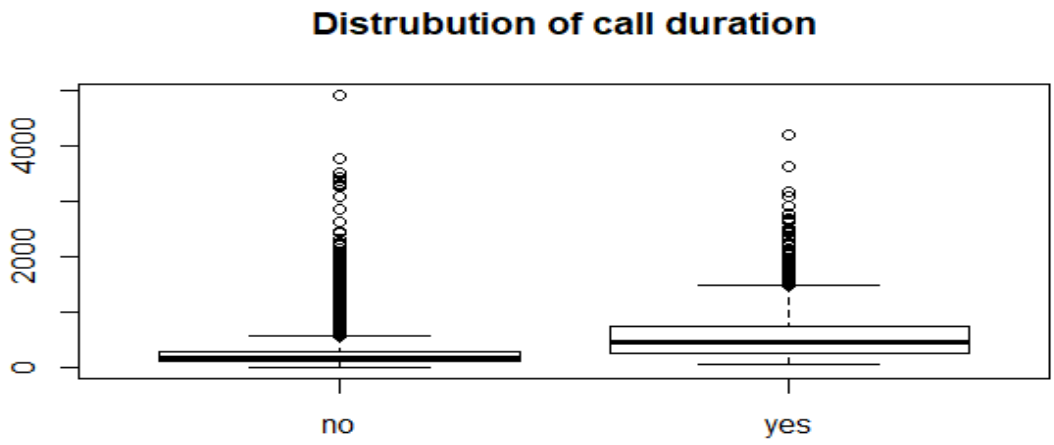


This below plot shows that the bank has contacted mostly in the age range of 20 to 60. Also we notice that maximum frequency age group is 30-35.



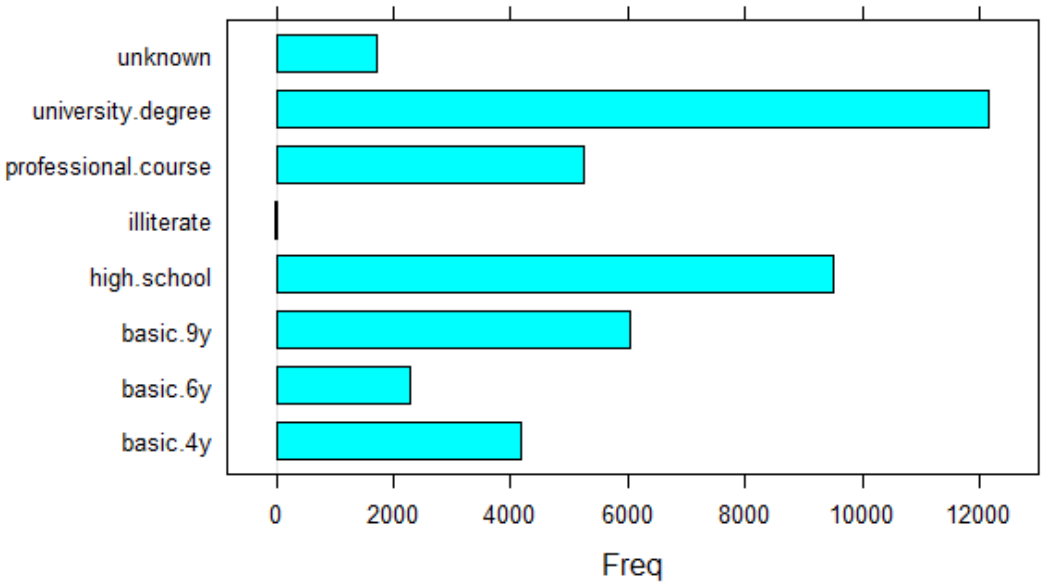
The box plot shows the distribution of age compared to term deposit. The outliers are more for customers not taking term deposit when compared to customers taking. The whiskers of NO are almost equal which mean customers not taking term deposit is equally distributed. In Yes group the upper whisker is longer. This shows that customers taking term deposit is more in the mid age group.

The Distribution of Call Duration



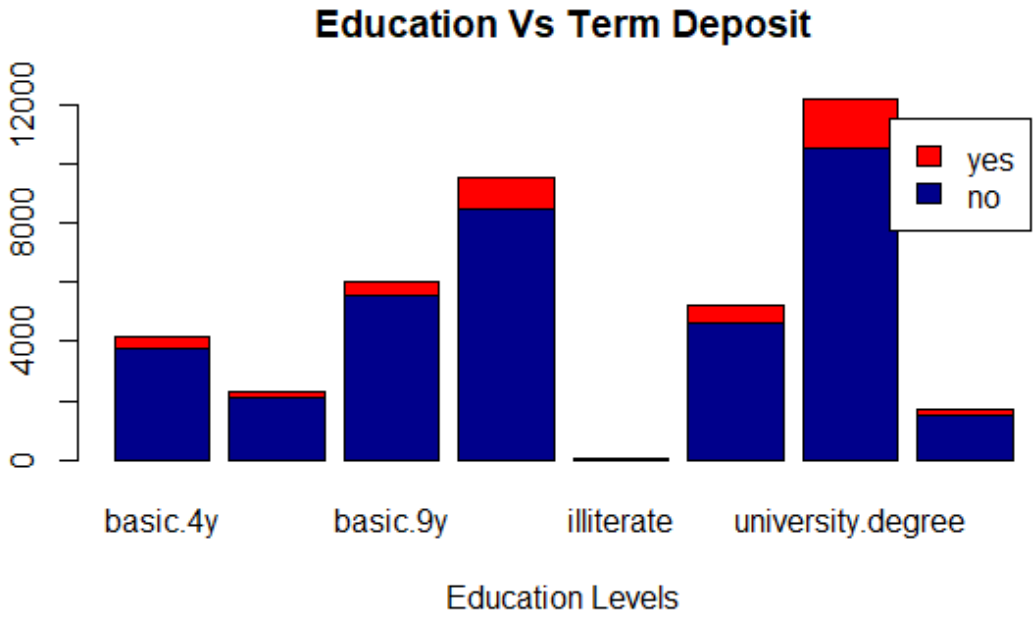
From the Above plot we infer that higher the call duration, the probability of customer accepting a term deposit is more.

Barplot of Education Variable



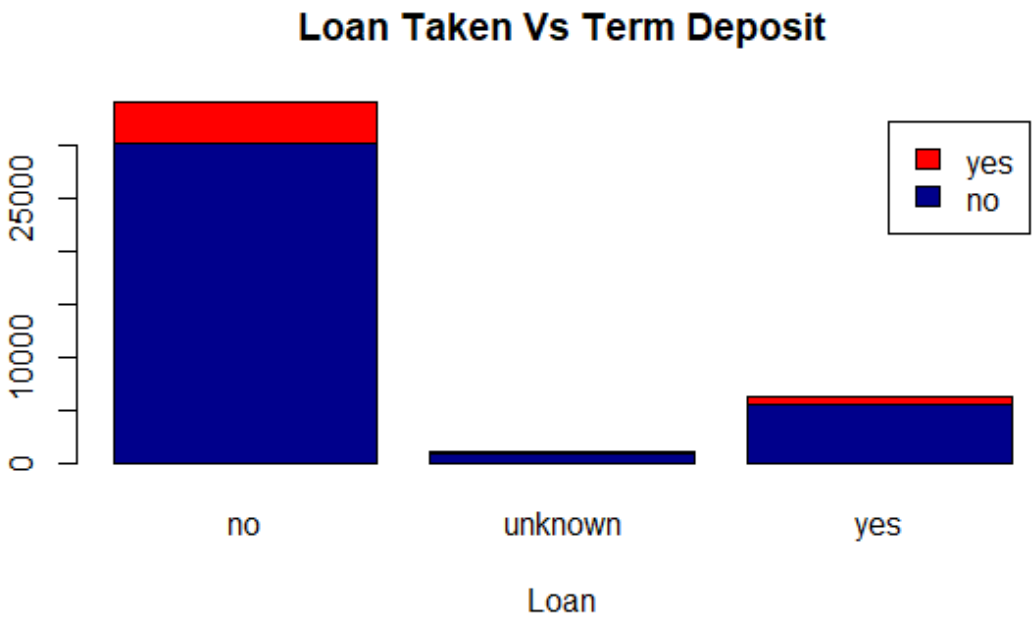
From this plot we notice that the bank contacted mostly to customers having higher education. University degree frequency is highest and the high school frequency is second highest.

The Distribution of Education Variable



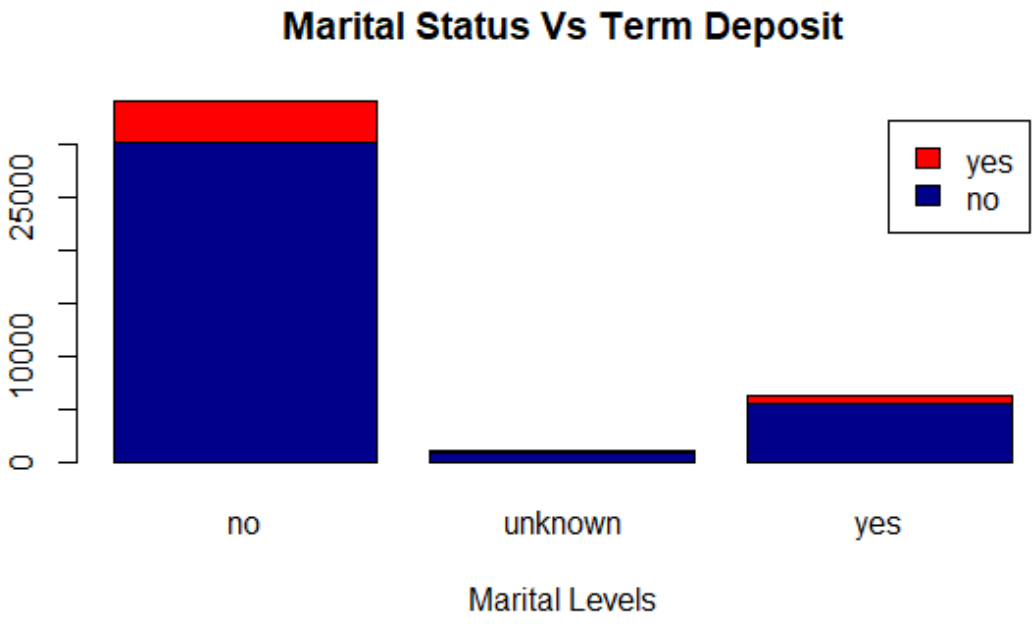
In all the education categories, the preference to term deposit is less. Compared to all education levels, customers having university degree opted more for term deposit.

Customers Having Loan vs Term Deposit



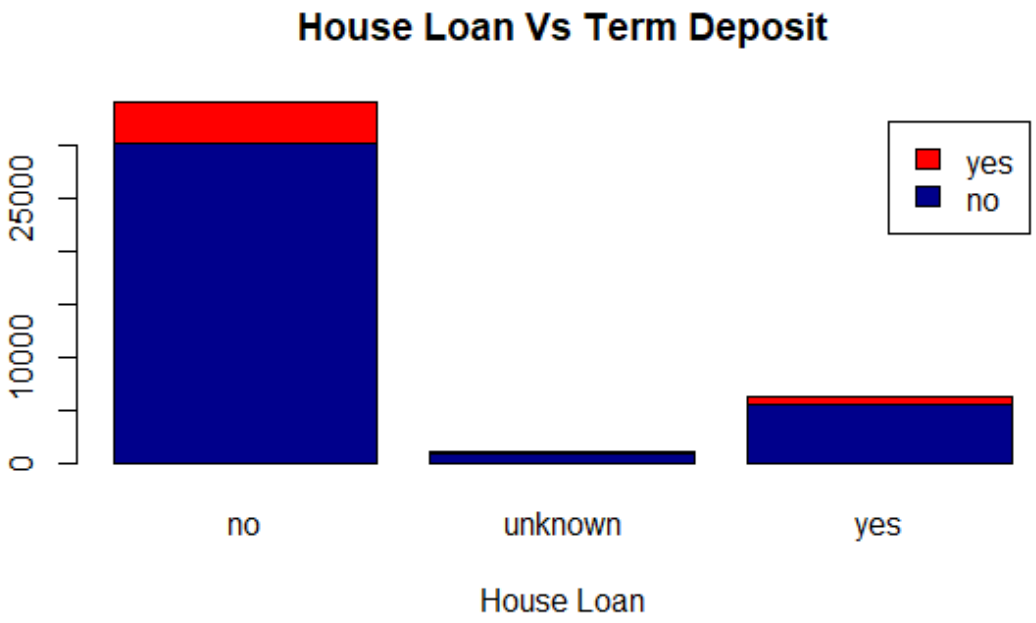
The customer having no loan prefered the term deposit when compared to customers having loan.

Customers Marital Status vs Term Deposit



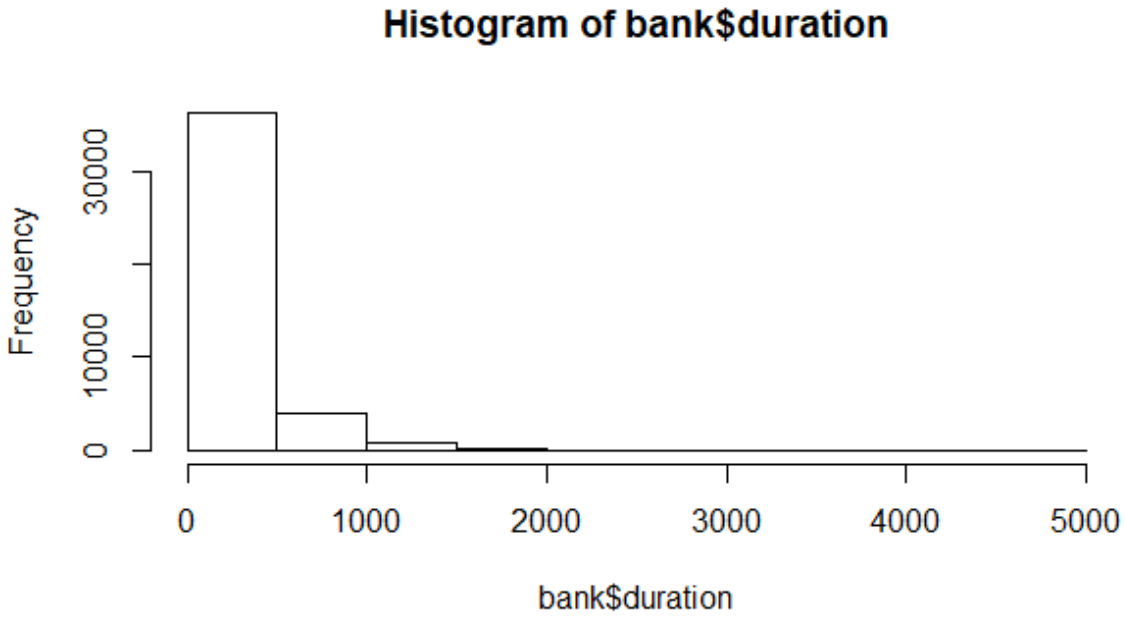
Here we see that married and single customers prefered term deposit more than divorced.

Customer Having House Loan vs Term Deposit

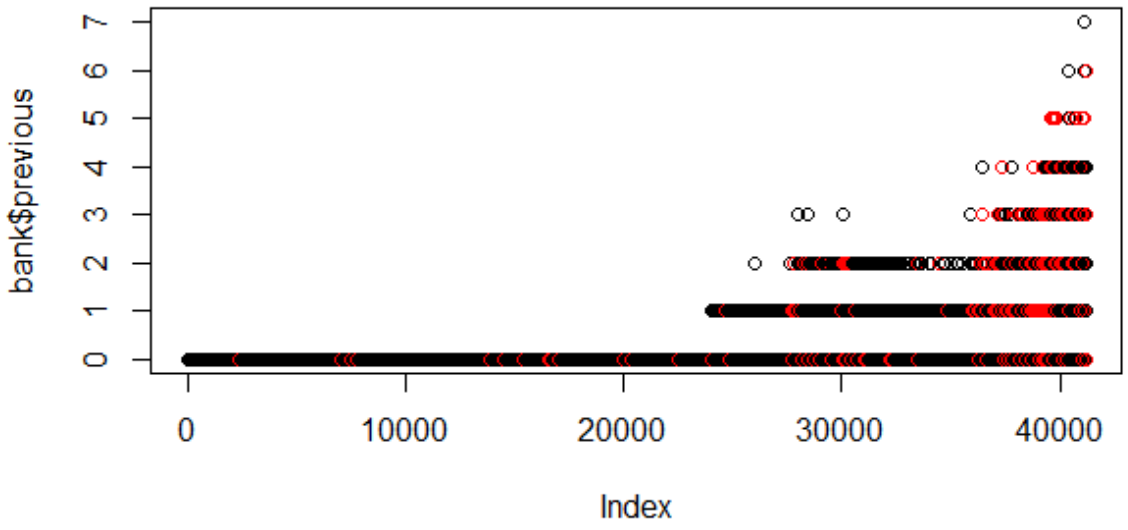


The bank contacted both the customers having house-house loans and non-house loans. Both categories preferred the term deposit almost equally.

Histogram of Customer Last Contacted duration

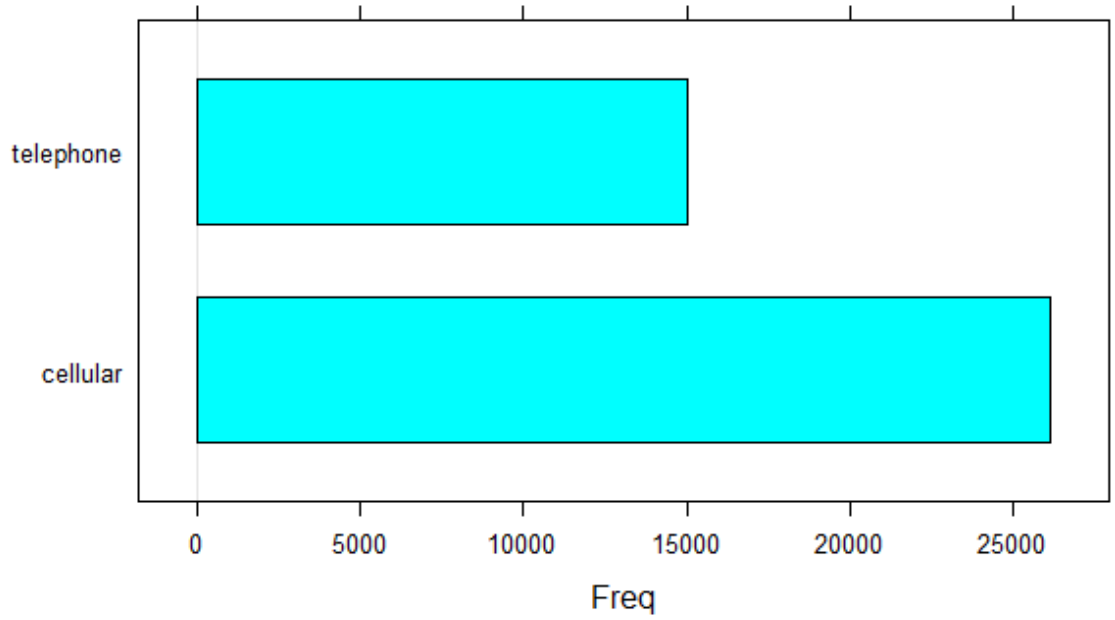


No of Contacts Performed previously vs Term Deposit

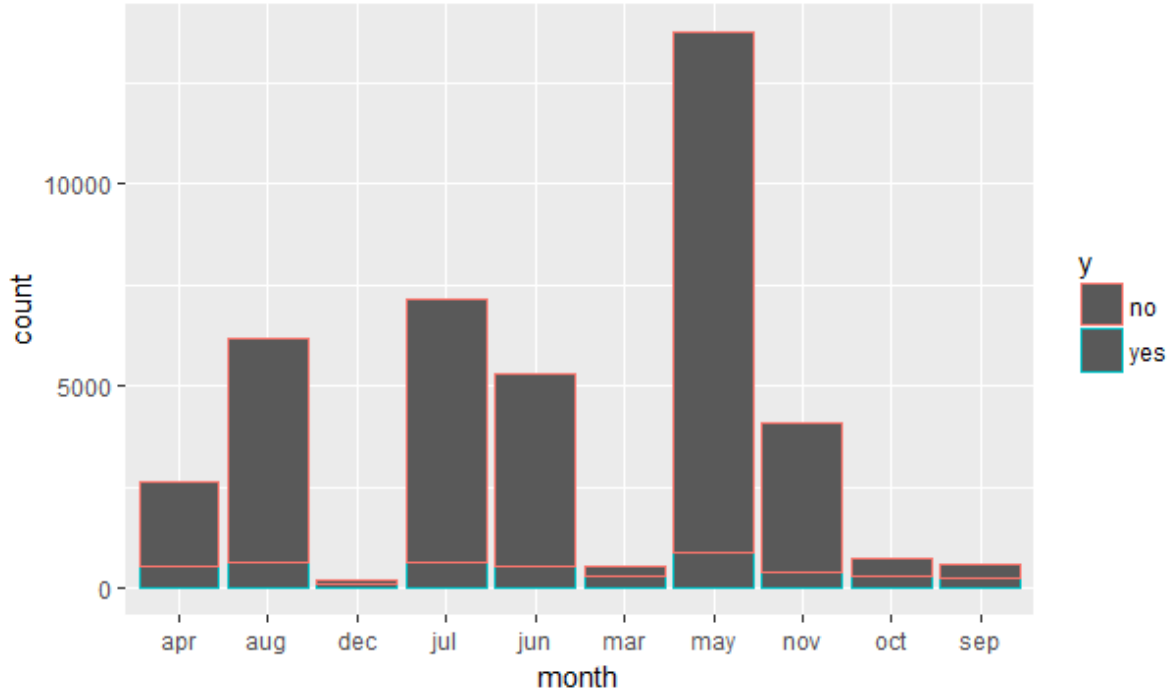


Here we notice that frequency for duration having 0 is more which indicates that bank mostly preferred new customer than existing. The scatter plot further classifies this with term deposit for new and existing customers.

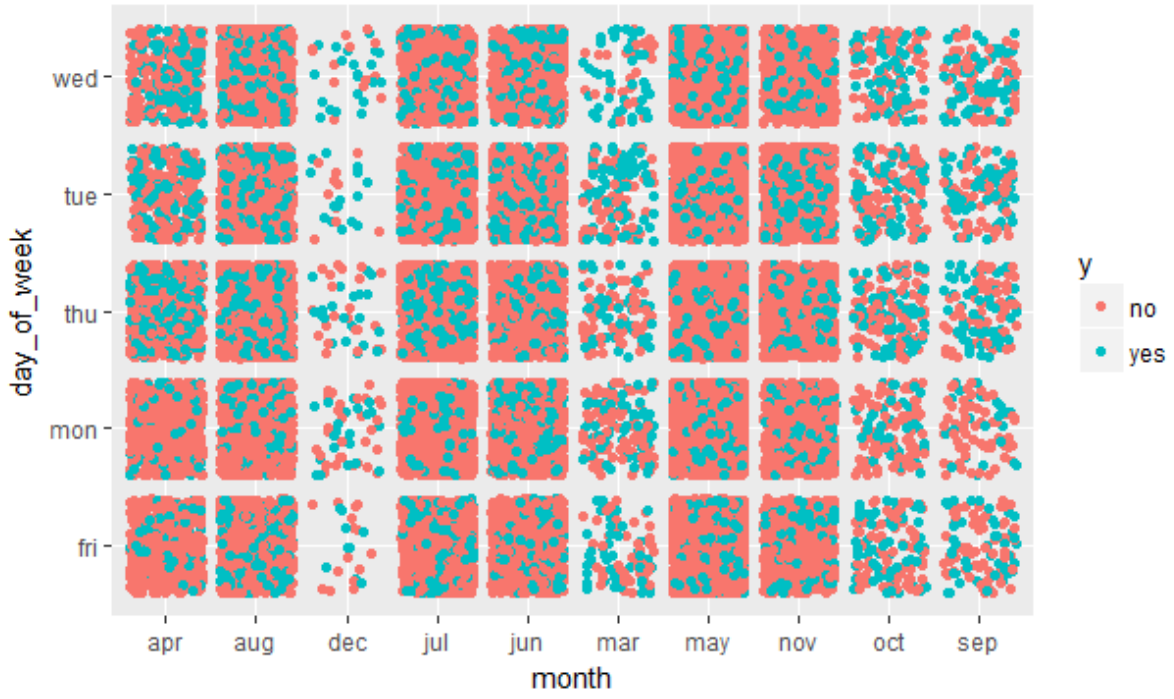
Barchart of Contact variable



Last Contact Month vs Term Deposit

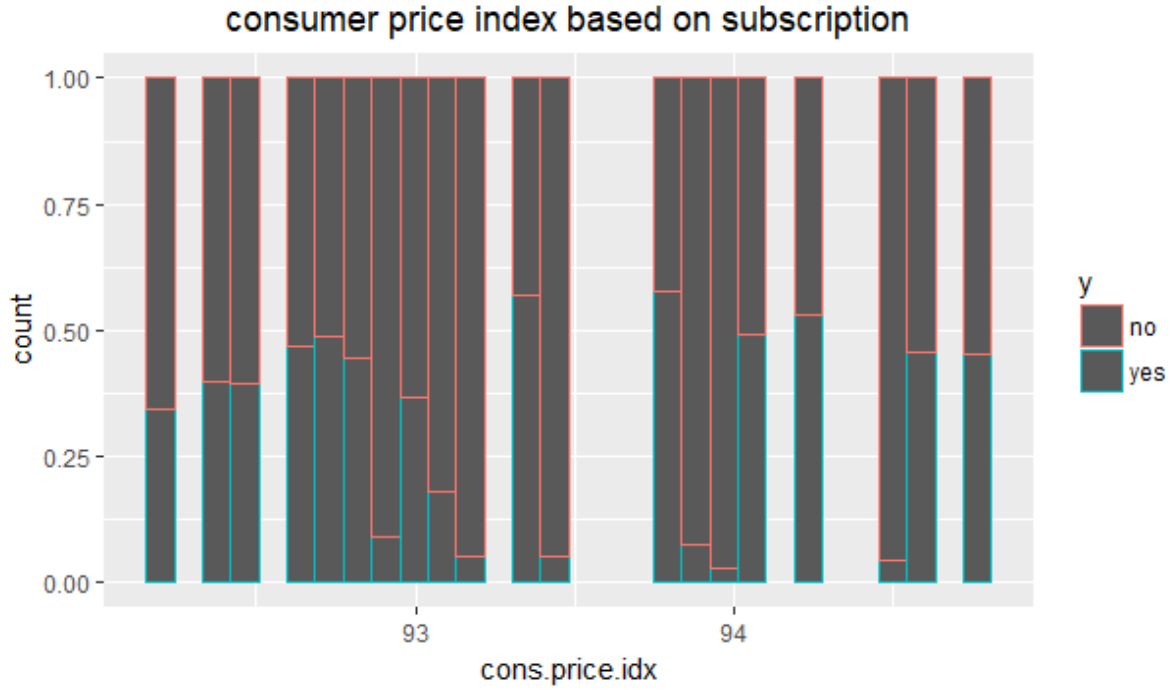


Contact days of a week vs Term Deposit



The plot of proportion table and scatterplot of month and day of the week on which people were contacted shows that the months december, march, october and september have a very high probability of people taking the plan compared to other months. But the histogram of month in accordance with y shows that very less number of people were actually contacted in those months.

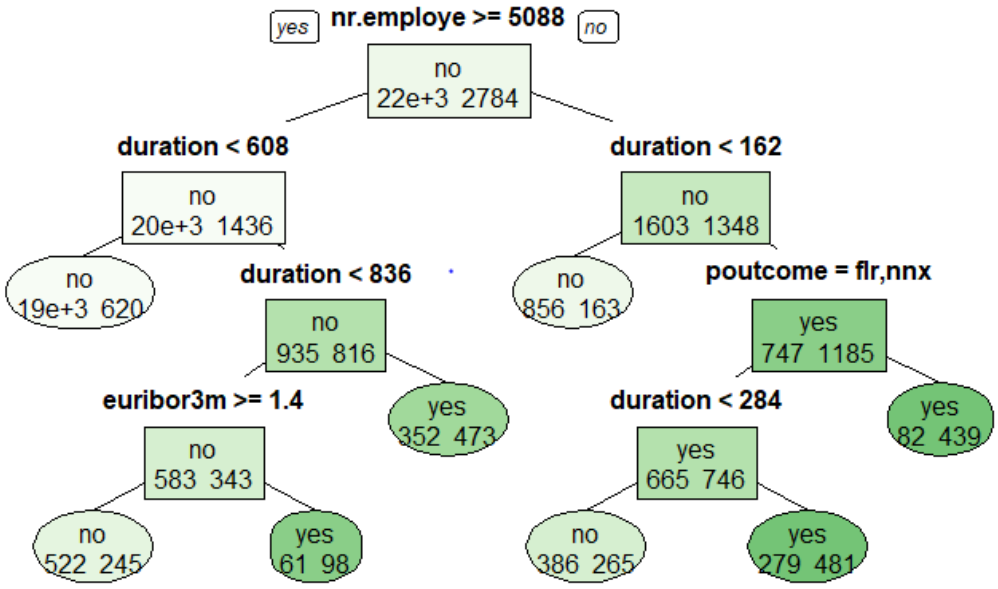
Consumer Price Index vs Term Deposit



A histogram of proportions of the variable consumer price index shows that the concentration of people taking the plan is evenly spread on both higher and lower sides of CPI, so we can't really make any generalizations about a particular group being more favorable for saying "yes"

Decision Trees

Classification tree is constructed for full grown tree. Since Decision trees are not affected by the transformations we are proceeding without any normalization.



Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7227  765
         1   83  163

               Accuracy : 0.8971
                 95% CI : (0.8903, 0.9035)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.002542

                  Kappa : 0.2419
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.17565
            Specificity : 0.98865
         Pos Pred Value : 0.66260
         Neg Pred Value : 0.90428
             Prevalence : 0.11265
         Detection Rate : 0.01979
   Detection Prevalence : 0.02986
      Balanced Accuracy : 0.58215

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predictdecision))
```

#AUC is 58.21% we got 89 percent accuracy with decision trees. But let's apply random forest to see if we can further increase our accuracy with random forest.

Random Forest.

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7136  662
         1  174  266

               Accuracy : 0.8985
                 95% CI : (0.8918, 0.905)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.000615

                  Kappa : 0.3411
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.28664
            Specificity : 0.97620
         Pos Pred Value : 0.60455
         Neg Pred Value : 0.91511
             Prevalence : 0.11265
         Detection Rate : 0.03229
   Detection Prevalence : 0.05341
      Balanced Accuracy : 0.63142

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predict_random))

Data: as.numeric(predict_random) in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.6314
```

Accuracy for decision tree is 89%.AUC is 63.3 percent. AUC has increased from 58 to 63%. From the above confusion matrix, we can see that the classes were better classified than full decision tree. Also, we see the FN value to decrease and false positive value to increase. There is an increment in the sensitivity. However, the accuracy remains same. Losing a potential customer incurs more loss to organization than incorrectly classifying a non-potential customer. The random forest reduces the loss by decreasing the False negative value.

Bagging

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7016  647
         1  294  281

               Accuracy : 0.8858
                 95% CI : (0.8787, 0.8926)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.6822

                  Kappa : 0.3149
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.30280
            Specificity : 0.95978
         Pos Pred Value : 0.48870
         Neg Pred Value : 0.91557
             Prevalence : 0.11265
         Detection Rate : 0.03411
   Detection Prevalence : 0.06980
      Balanced Accuracy : 0.63129

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predict_bag))

Data: as.numeric(predict_bag) in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.6313
```

AUC is 63% and remains the same. Accuracy is 88%. We see that bagging performed better than random forest with increment of sensitivity and decrement of false Negative.

Since this is an oversampled data with 1:9 ratio of preferred variable, we have used oversampled concept with SMOTE package to see any difference.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6211  407
         1 1099  521

               Accuracy : 0.8172
                 95% CI : (0.8087, 0.8255)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3101
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.56142
            Specificity : 0.84966
         Pos Pred Value : 0.32160
         Neg Pred Value : 0.93850
             Prevalence : 0.11265
         Detection Rate : 0.06324
   Detection Prevalence : 0.19665
      Balanced Accuracy : 0.70554

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predict_smote))

Data: as.numeric(predict_smote) in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.7055
```

AUC is 71%. Accuracy is 81%. But we see there is good improvement in sensitivity and false negative. Since our class of interest is and the oversampled concept gives us better results when compared to previous model, we finalize our decision tree as bank smote.

Logistic Regression

Logistic Regression is being applied to all the variables initially and the significant variables are taken from it and confusion matrix is created to see the accuracy.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7205  728
         1  105  200

               Accuracy : 0.8989
                 95% CI : (0.8922, 0.9053)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.0004195

                  Kappa : 0.2845
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.21552
            Specificity : 0.98564
         Pos Pred Value : 0.65574
         Neg Pred Value : 0.90823
             Prevalence : 0.11265
         Detection Rate : 0.02428
   Detection Prevalence : 0.03702
      Balanced Accuracy : 0.60058

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = predict_sig)

Data: predict_sig in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.6006
```

From the summary, we see that the residual deviance has reduced to 13590 with the cost of degree of freedom. The confusion matrix above shows that sensitivity is just 21% and false negative value is 765 which is high. AUC is 60%

However, this deviance is also large. Applying backstep regression method to find the desired variables
Confusion matrix for the best model from Backstep Regression.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7203  730
         1  107  198

               Accuracy : 0.8984
                 95% CI : (0.8917, 0.9048)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.0006969

                  Kappa : 0.2811
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.21336
            Specificity : 0.98536
         Pos Pred Value : 0.64918
         Neg Pred Value : 0.90798
             Prevalence : 0.11265
         Detection Rate : 0.02403
   Detection Prevalence : 0.03702
      Balanced Accuracy : 0.59936

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = predict_logistic_step)

Data: predict_logistic_step in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.5994
```

Accuracy 89% and AUC 60%. The above model too gives the same deviance as bank_logistic. Also the sensitivity and false negative have not improved. Applying cross-validation to check any better model

Using the cross validation in logistic regression.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7192  118
         1  726  202

               Accuracy : 0.8975
                 95% CI : (0.8908, 0.904)
    No Information Rate : 0.9612
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2823
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.63125
            Specificity : 0.90831
         Pos Pred Value : 0.21767
         Neg Pred Value : 0.98386
             Prevalence : 0.03884
         Detection Rate : 0.02452
   Detection Prevalence : 0.11265
      Balanced Accuracy : 0.76978

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predict_logistic_2))

Data: as.numeric(predict_logistic_2) in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.6008
```

#Accuracy 89%. AUC 60% and no improvement in sensitivity and false negative.

Applying Smote function

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6493  423
         1  817  505

               Accuracy : 0.8495
                 95% CI : (0.8416, 0.8571)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3648
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5442
            Specificity : 0.8882
         Pos Pred Value : 0.3820
         Neg Pred Value : 0.9388
             Prevalence : 0.1126
         Detection Rate : 0.0613
   Detection Prevalence : 0.1605
      Balanced Accuracy : 0.7162

       'Positive' Class : 1


Call:
roc.default(response = bank_val_labels, predictor = as.numeric(predict_smote_logistic))

Data: as.numeric(predict_smote_logistic) in 7310 controls (bank_val_labels 0) < 928 cases (bank_val_labels 1).
Area under the curve: 0.7162
```

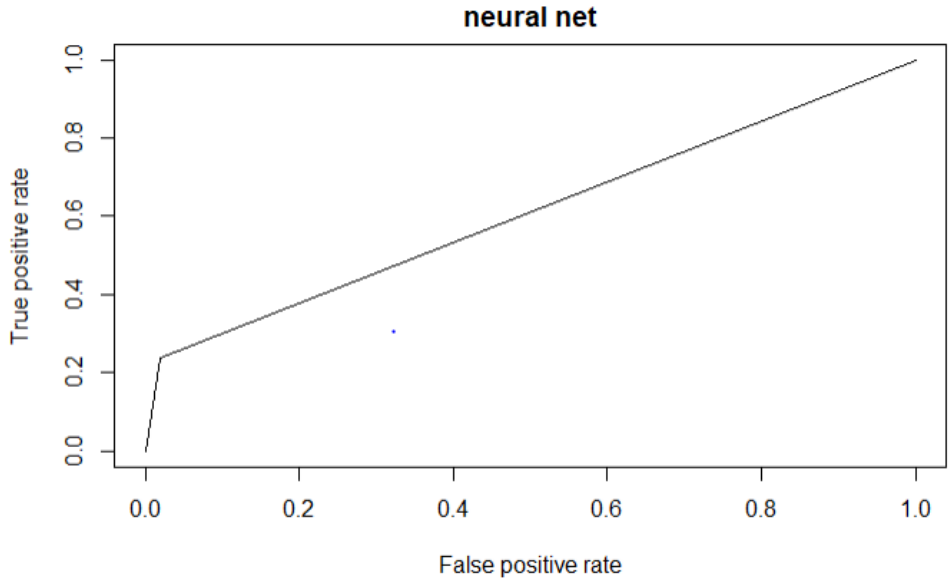
#Accuracy 85%. AUC 71%. The above confusion matrix says that sensitivity and true negative has improved has improved. But slightly lesser than decision tree smote.

Neural Networks

Normalization and created dummy variables for neural networks

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7175  707
         1  135  221

               Accuracy : 0.8978
                 95% CI : (0.891, 0.9043)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.001278

                  Kappa : 0.3005
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.23815
            Specificity : 0.98153
         Pos Pred Value : 0.62079
         Neg Pred Value : 0.91030
             Prevalence : 0.11265
         Detection Rate : 0.02683
   Detection Prevalence : 0.04321
      Balanced Accuracy : 0.60984

       'Positive' Class : 1
```

ROC Curve



ROC curve of the predicted and true values indicating the relationship between true positive rate and false positive rate. The area under the curve for the plot is 0.7386739

Trying to improve the model performance by using the function pcaNNet which applies principal component analysis to the variables before building a neural network model. And also, size of the hidden layers were reduced to 2 for the model to generalize more on future data and to avoid overfitting

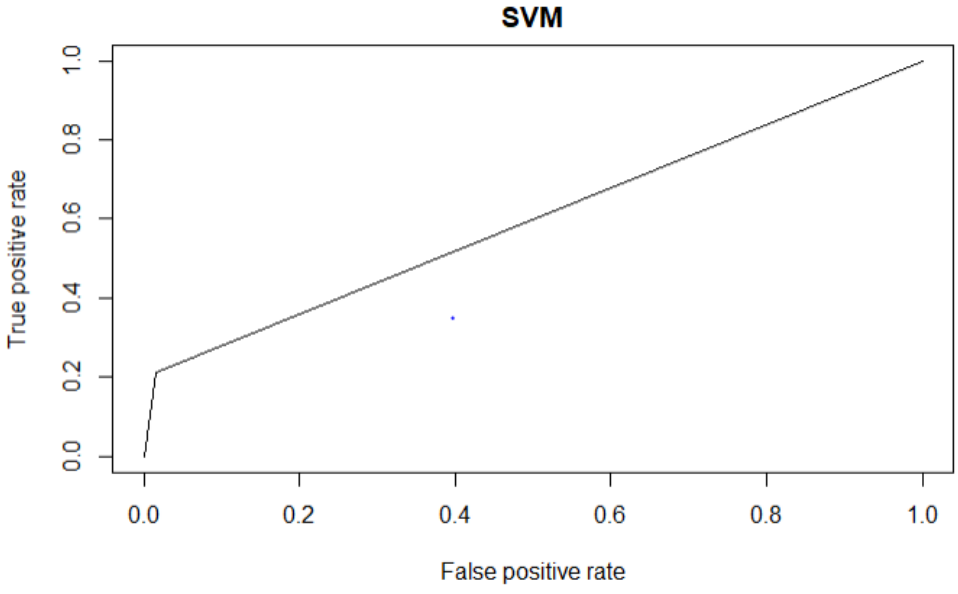
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7166  696
         1  144  232

               Accuracy : 0.898
                 95% CI : (0.8913, 0.9045)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.001007

                  Kappa : 0.3111
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.25000
            Specificity : 0.98030
         Pos Pred Value : 0.61702
         Neg Pred Value : 0.91147
             Prevalence : 0.11265
         Detection Rate : 0.02816
   Detection Prevalence : 0.04564
      Balanced Accuracy : 0.61515

       'Positive' Class : 1
```

we can see an improvement in sensitivity and false negative.

Support Vector Machine

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7202  732
         1  108  196

               Accuracy : 0.898
                 95% CI : (0.8913, 0.9045)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 0.001007

                  Kappa : 0.278
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.21121
            Specificity : 0.98523
         Pos Pred Value : 0.64474
         Neg Pred Value : 0.90774
             Prevalence : 0.11265
         Detection Rate : 0.02379
   Detection Prevalence : 0.03690
      Balanced Accuracy : 0.59822

       'Positive' Class : 1
```

ROC Curve



ROC curve of the predicted and true values indicating the relationship between true positive rate and false positive rate.

Naive Bayes model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6112  400
         1 1198  528

               Accuracy : 0.806
                 95% CI : (0.7973, 0.8145)
    No Information Rate : 0.8874
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2945
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.56897
            Specificity : 0.83611
         Pos Pred Value : 0.30591
         Neg Pred Value : 0.93857
             Prevalence : 0.11265
         Detection Rate : 0.06409
   Detection Prevalence : 0.20952
      Balanced Accuracy : 0.70254

       'Positive' Class : 1
```

Based on the sensitivity and false negative value, we choose our final model as Naive based model.

Applying naive bayes on test data.

```
[1] 0.3212075
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6142  370
         1 1167  558

               Accuracy : 0.8134
                 95% CI : (0.8048, 0.8218)
    No Information Rate : 0.8873
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3212
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.60129
            Specificity : 0.84033
         Pos Pred Value : 0.32348
         Neg Pred Value : 0.94318
             Prevalence : 0.11266
         Detection Rate : 0.06774
   Detection Prevalence : 0.20942
      Balanced Accuracy : 0.72081

       'Positive' Class : 1
```

Conclusion

From the confusion matrix above we notice that accuracy is 81% . Also the false positive value is 370 and true positive values are 558 we have used 20% for validation and 20% of test data. we got a better result for test data when compared to validation data in terms of true positive and false negative.

References

https://archive.ics.uci.edu/ml/datasets/bank+marketing
[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS. [bank.zip]