Prof. Roseanna Hopper

ALY 6015
Final Project
Portuguese Bank Marketing Data Set

By

Sunil Raj THOTA

Nalini MACHARLA

Introduction

Portuguese Bank’s
Direct Marketing Campaigns

Data Exploration

Data Modelling

Data
Preprocessing

UCI Machine Learning Repository
Open-source

Data Analysis

Prediction
Decision-making

Business Problem

Actions to take for Revenue Decline?

Predict if the client will subscribe (yes/no) a term deposit (variable y)?

Identify existing Clients?

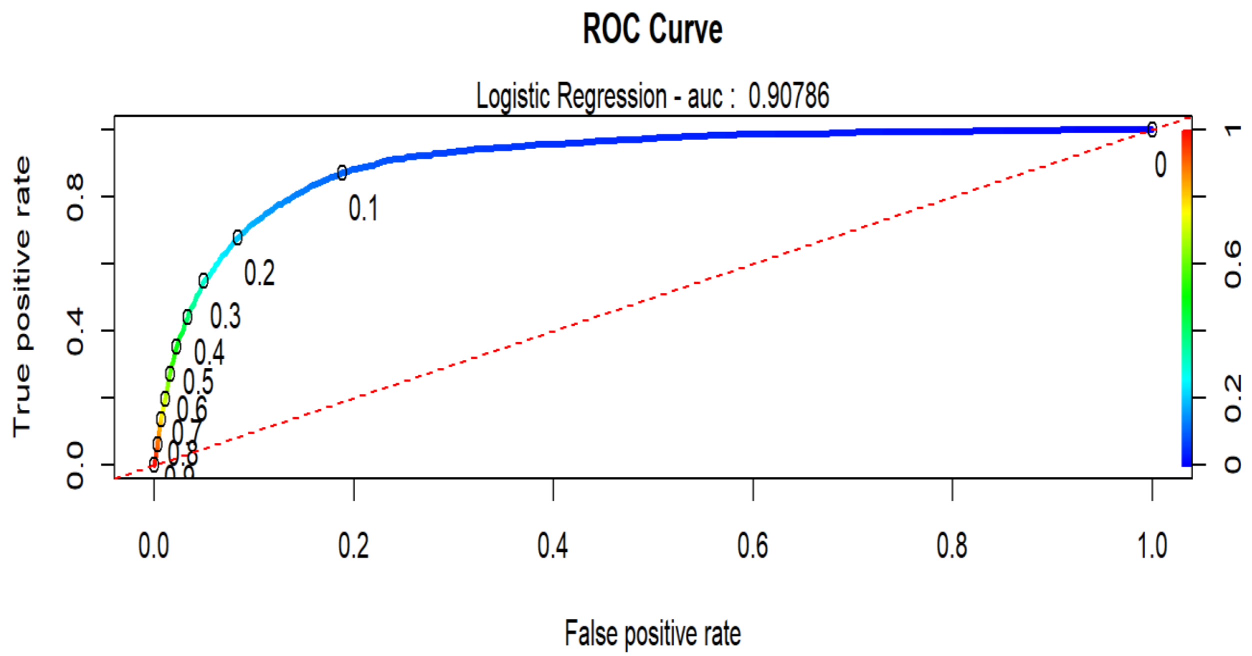
Which Feature Selection Technique should be used for our data?

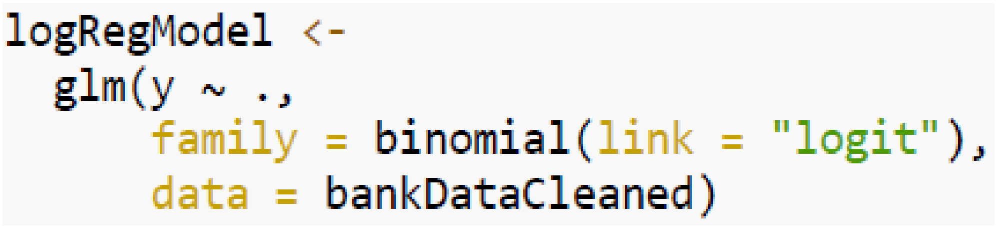
Logistic Regression

Logistic is an appropriate regression analysis to perform when the dependent variable is dichotomous or binary

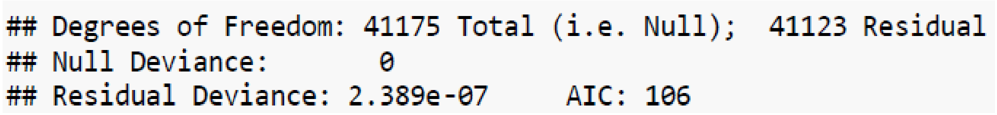
It predicts the probability of occurrence of an incident by fitting data to a logit function

So, We chose logistic regression model to discover the Client subscription with accuracy of **90.38%**

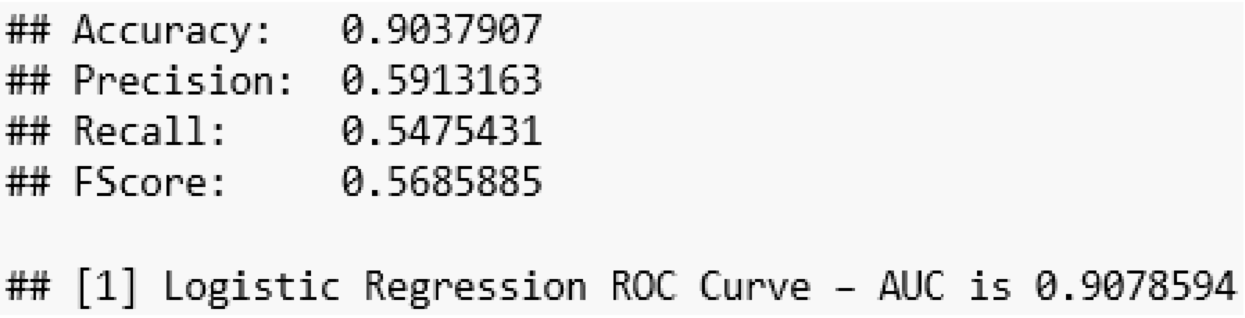




```
logRegModel <-
  glm(y ~ .,
      family = binomial(link = "logit"),
      data = bankDataCleaned)
```



```
## Degrees of Freedom: 41175 Total (i.e. Null);  41123 Residual
## Null Deviance:       0
## Residual Deviance: 2.389e-07     AIC: 106
```



```
## Accuracy:   0.9037907
## Precision:  0.5913163
## Recall:     0.5475431
## FScore:     0.5685885

## [1] Logistic Regression ROC Curve - AUC is 0.9078594
```

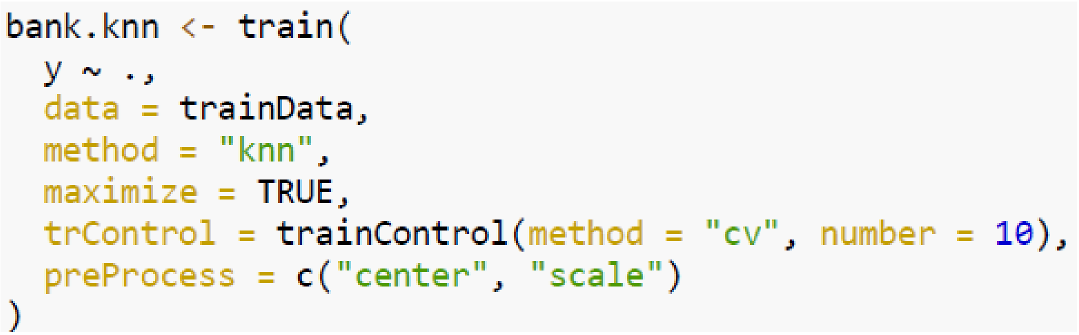
k-Nearest Neighbors

The low bias/ high variance classifiers considered is k-Nearest Neighbors, it is a supervised learning algo

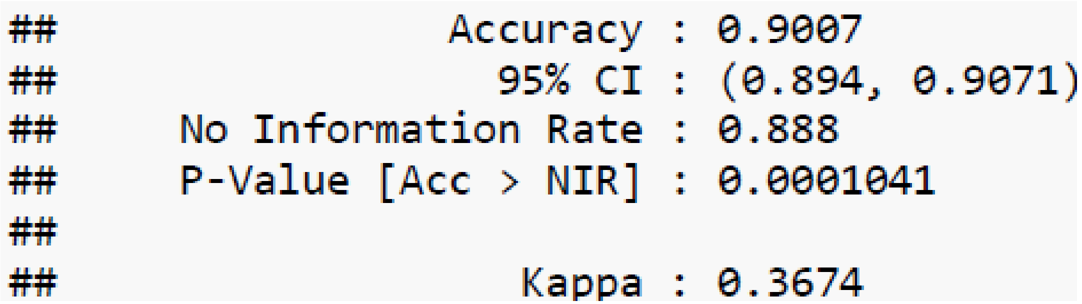
We train it under supervision using the labelled data which is already available to us

Another parameter was preProcess, where the data was centred and scaled

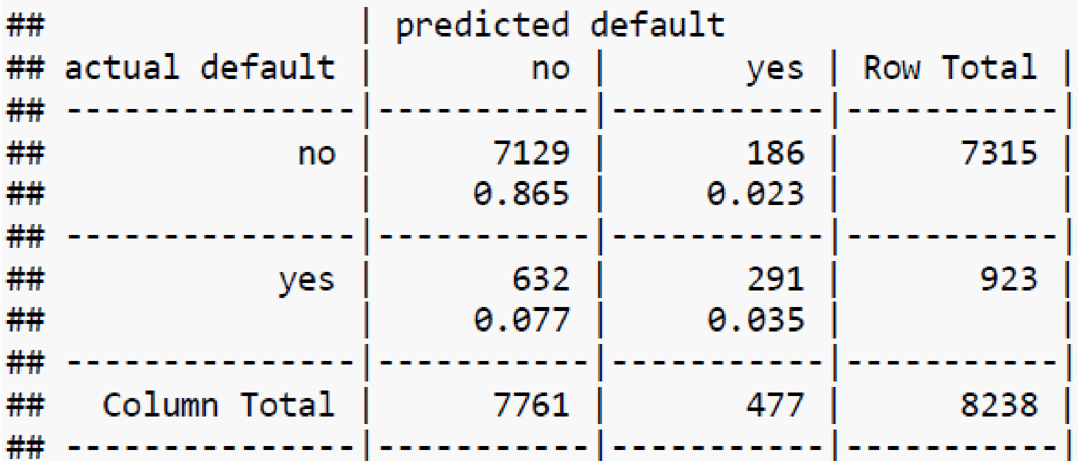
KNN method for the 80/20 test/training split's accuracy is **90.07%**



```
bank.knn <- train(
  y ~ .,
  data = trainData,
  method = "knn",
  maximize = TRUE,
  trControl = trainControl(method = "cv", number = 10),
  preProcess = c("center", "scale")
)
```



```
##                 Accuracy : 0.9007
##                   95% CI : (0.894, 0.9071)
##      No Information Rate : 0.888
##      P-Value [Acc > NIR] : 0.0001041
##
##                    Kappa : 0.3674
```



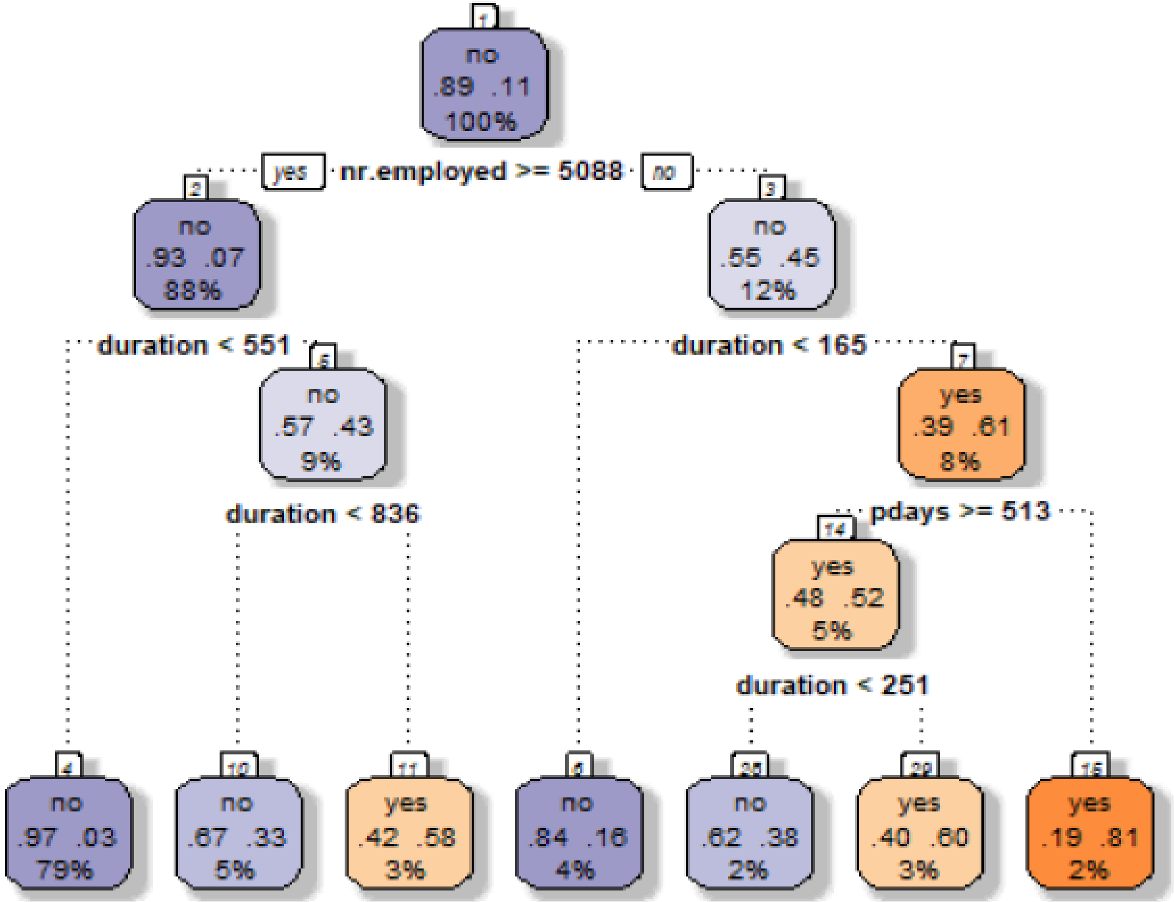
```
##                | predicted default
## actual default |        no |       yes | Row Total |
## ---------------|-----------|-----------|-----------|
##             no |      7129 |       186 |      7315 |
##                |     0.865 |     0.023 |           |
## ---------------|-----------|-----------|-----------|
##            yes |       632 |       291 |       923 |
##                |     0.077 |     0.035 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |      7761 |       477 |      8238 |
## ---------------|-----------|-----------|-----------|
```

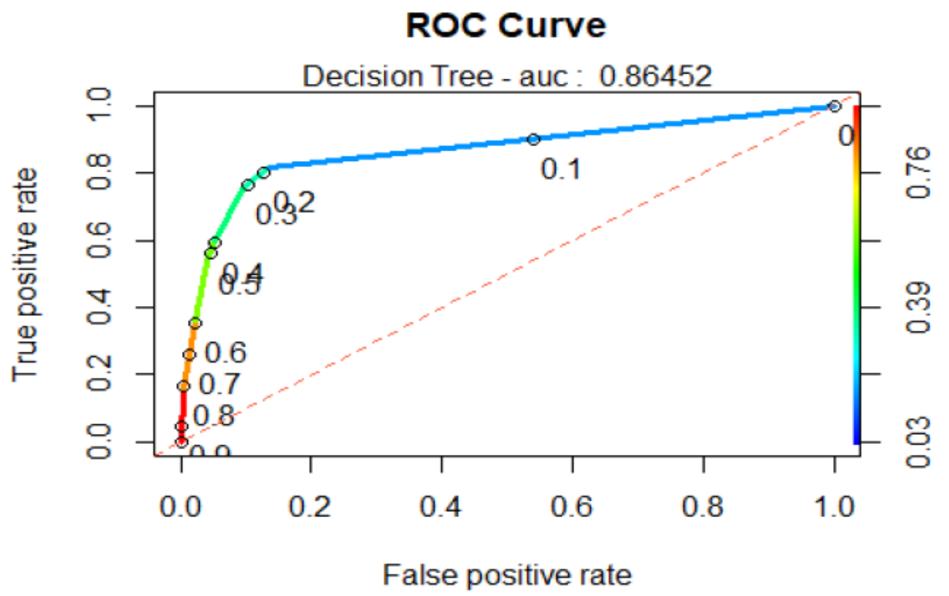
Decision Tree

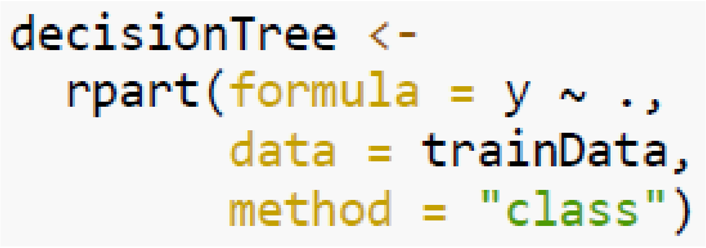
It is one of the very simplest and useful ML algorithms and used to predict a class of the given data

The results show that the model are fitted to evaluate train data considering that errors is so low 8.1%

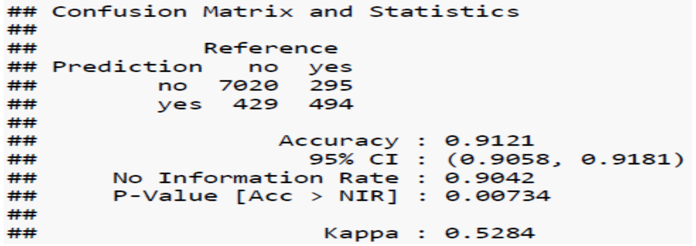
Decision Tree seems to be a better classifier with the accuracy achieved with this model is about **91.21%**







```
decisionTree <-
  rpart(formula = y ~ .,
                  data = trainData,
                  method = "class")
```



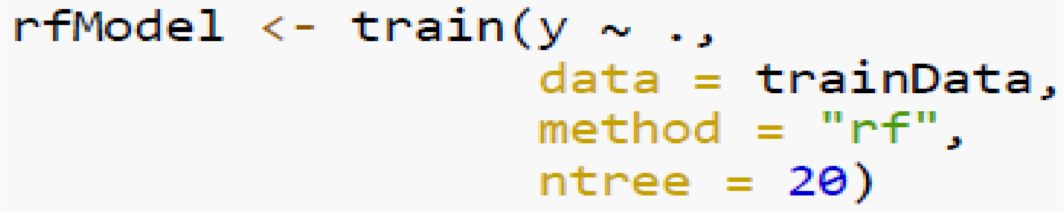
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  7020  295
##        yes  429  494
##
##                Accuracy : 0.9121
##                  95% CI : (0.9058, 0.9181)
##     No Information Rate : 0.9042
##     P-Value [Acc > NIR] : 0.00734
##
##                   Kappa : 0.5284
```

Random Forest

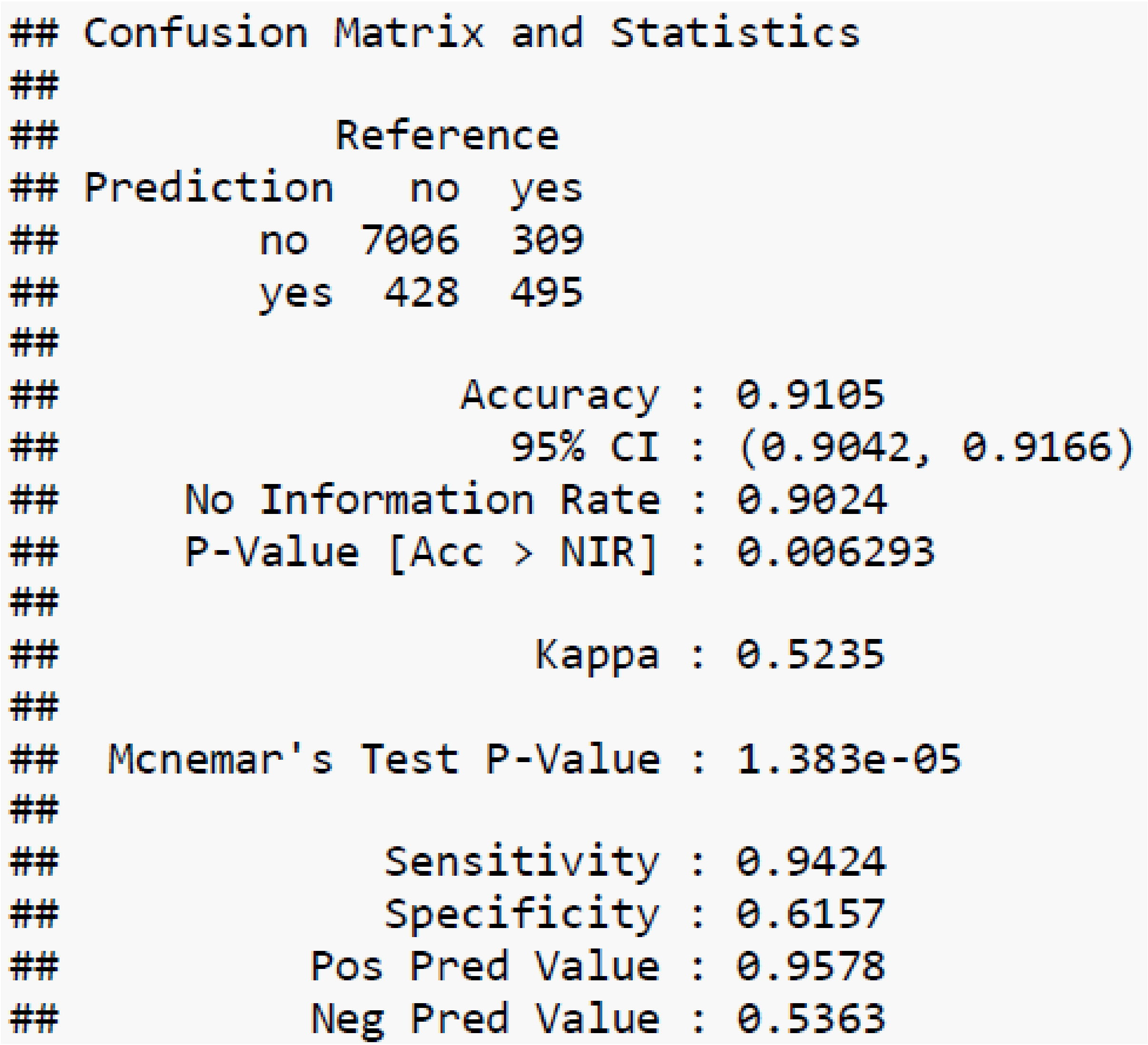
The random forest then combines the output of individual decision trees to get the ultimate output

This process of mixing the output of multiple individual models is called as Ensemble Learning

After rigorous training and testing with 20 dimensions, we obtained an accuracy of **91.05%**



```
rfModel <- train(y ~ .,
                 data = trainData,
                 method = "rf",
                 ntree = 20)
```



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  7006  309
##        yes  428  495
##
##                Accuracy : 0.9105
##                  95% CI : (0.9042, 0.9166)
##     No Information Rate : 0.9024
##     P-Value [Acc > NIR] : 0.006293
##
##                   Kappa : 0.5235
##
##  Mcnemar's Test P-Value : 1.383e-05
##
##             Sensitivity : 0.9424
##             Specificity : 0.6157
##          Pos Pred Value : 0.9578
##          Neg Pred Value : 0.5363
```

Conclusion

MODEL	ACCURACY (%)	RANK
Logistic Regression	90.38	3
k-Nearest Neighbors	90.07	4
Decision Trees	**91.21**	**1**
Random Forests	91.05	2

The Decision Tree Model produces the most accurate predictions of client will subscribe (yes/no) a term deposit information

The accuracy of the predictions are verified with a Probability Value of **0.00734** and a 95% confidence interval of **0.9058** to **0.9181**

For better understanding we can go further by building **XG Boost, Ada Boost, GBM, Light GBM,** and **Neural Network** Models and figure out the best accurate predictor and use it in the Bank's Marketing Campaign