BA_FinalExam

Snehitha Anpur

2022-11-25

Data Loading

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.2

## Loading required package: lattice
```

```
Churndata=read.csv("D:\\MSBA\\rTutorial\\Rtutorial\\Churn_Train.csv") # Reading CSV file

load("Customers_To_Predict.RData") # Loading the data of RData file

set.seed(1234) # Setting Seed value
```

Converting the categorical data of character type to factor type

```
Churndata$state = as.factor(Churndata$state)

Churndata$area_code = as.factor(Churndata$area_code)

Churndata$international_plan = as.factor(Churndata$international_plan)

Churndata$voice_mail_plan = as.factor(Churndata$voice_mail_plan)

Churndata$churn = as.factor(Churndata$churn)
```

Data Cleaning

```
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.2.2

##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
colMeans(is.na(Churndata))*100 # Checking for null values percentage
```

```
##                         state                account_length
##                      0.000000                     15.031503
##                     area_code            international_plan
##                      0.000000                      0.000000
##               voice_mail_plan         number_vmail_messages
##                      0.000000                      6.000600
##             total_day_minutes               total_day_calls
##                      6.000600                      6.000600
##              total_day_charge             total_eve_minutes
##                      6.000600                      9.030903
##               total_eve_calls              total_eve_charge
##                      6.000600                      6.000600
##           total_night_minutes             total_night_calls
##                      6.000600                      0.000000
##            total_night_charge            total_intl_minutes
##                      6.000600                      6.000600
##              total_intl_calls             total_intl_charge
##                      9.030903                      6.000600
## number_customer_service_calls                         churn
##                      6.000600                      0.000000
```

```
Imputed_Churn <- mice(Churndata, m=2, maxit = 10, method = 'pmm', seed = 500) # Imputing the null value
```

```
##
##  iter imp variable
##   1   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   1   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   2   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   3   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   4   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
```

```
##   4   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   5   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   6   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   6   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   7   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   7   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   8   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   8   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   9   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   9   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charge
##   10   1  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charg
##   10   2  account_length  number_vmail_messages  total_day_minutes  total_day_calls  total_day_charg
```

```
## Warning: Number of logged events: 2
```

```
Imputed_churndata <- complete(Imputed_Churn,2) # Using the 5th dataset for this project

mice:::find.collinear(Imputed_churndata) # Checking for the Collinearity or correlation
```

```
## [1] "total_night_charge" "total_intl_charge"
```

```
Cleaned_Churndata= Imputed_churndata[,-c(7,15,18)] # Removing the Correlated columns
```

Data Exploring

```
library(corrplot)
```

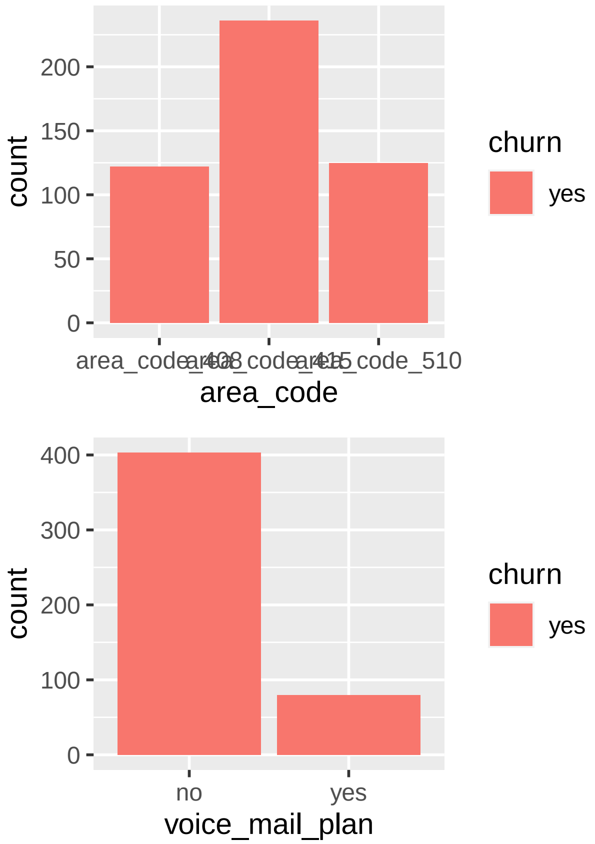
```
## Warning: package 'corrplot' was built under R version 4.2.2
```

```
## corrplot 0.92 loaded
```

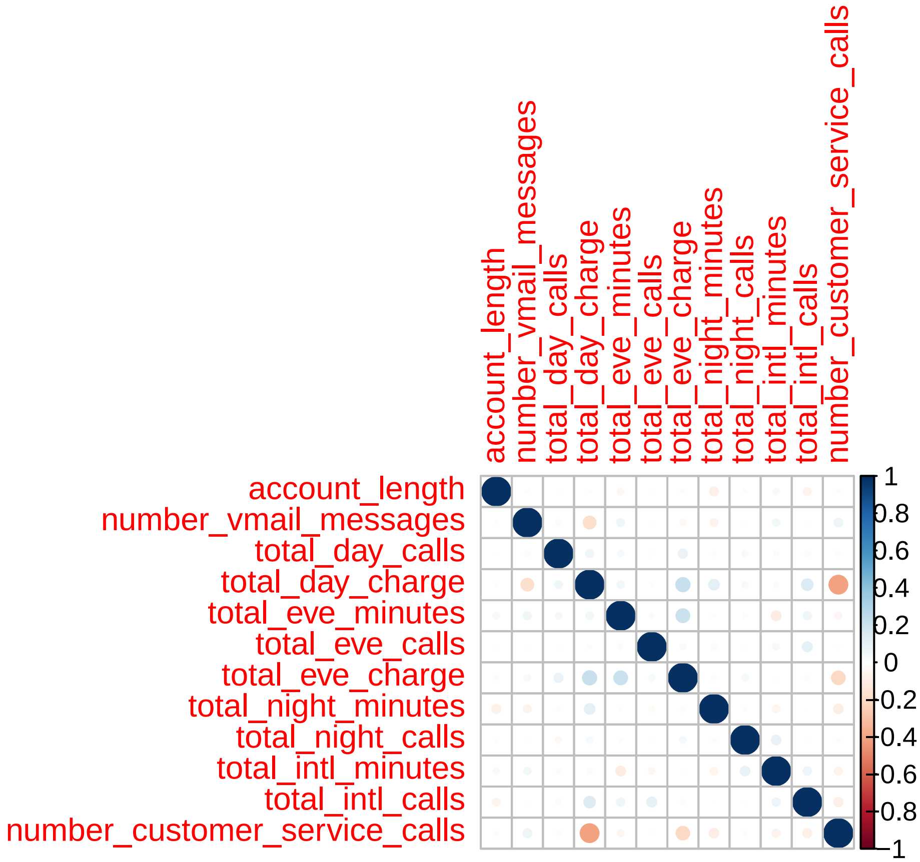
```
library(ggplot2)

library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.2.2
```

```
churn_yes = Cleaned_Churndata[Cleaned_Churndata$churn=='yes',] # Filtering the data for Churn="yes"

Area_code = ggplot(churn_yes, aes(x=area_code, fill=churn)) + geom_bar(position="dodge")

International_plan = ggplot(churn_yes, aes(x=international_plan, fill=churn)) + geom_bar(position="dodg

Voice_mail_plan = ggplot(churn_yes, aes(x=voice_mail_plan, fill=churn)) + geom_bar(position="dodge")

plot_grid(Area_code,International_plan,Voice_mail_plan) # plotting the Categorical Variables
```




```
p=table( churn_yes$churn,churn_yes$state)

corrplot(cor(churn_yes[, c(2,6:16)])) # Correlation plot for the numerical variables
```



Data Partition

```
Test_Data_label = createDataPartition(Cleaned_Churndata$churn,p=0.30,list = FALSE)  # Creating the Part

Train_Data = Cleaned_Churndata[-Test_Data_label,] # Train Data

Test_Data = Cleaned_Churndata[Test_Data_label,] # Test Data
```

Data is partitioned as Train and Test data to check for the Model which suits best for this dataset

Data Modelling

Multiple Regression

In Multiple Regression , It deals with Continuous Variables, Where as our dataset has binary target values. With this type when we use Anova method we see sum of squares has high for residuals. Hence Mutliple Regression is not the best model for this dataset

Logistic Regression

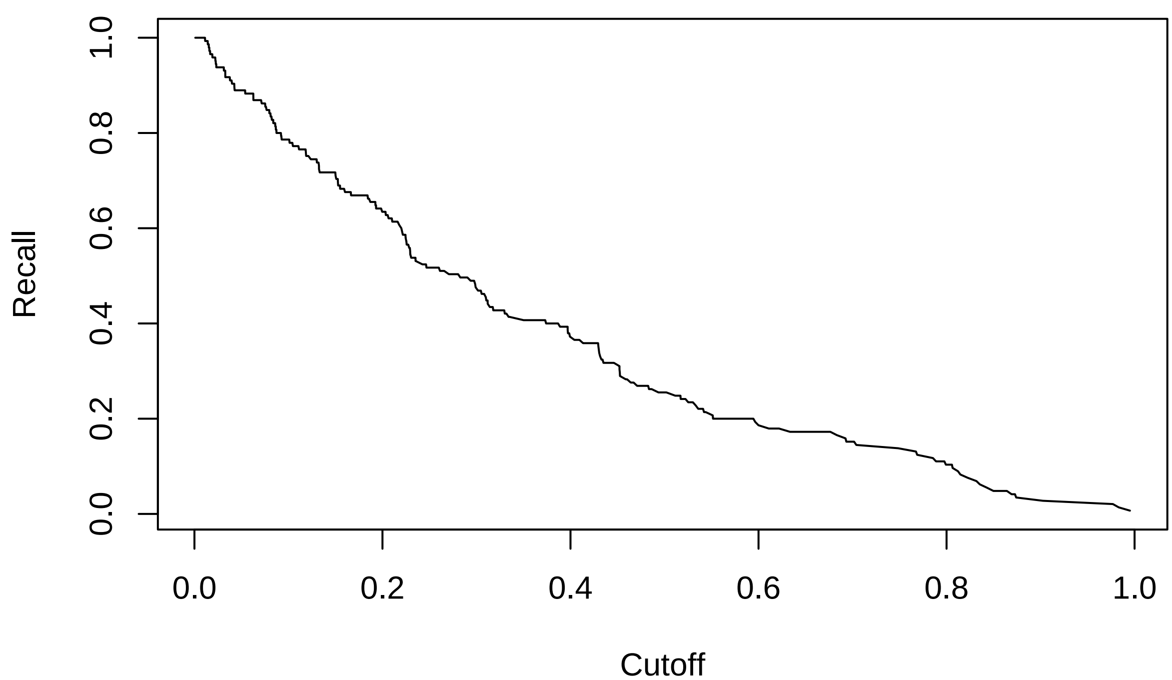
```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.2.2
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.2.2
```

```r
LR = glm(churn ~ ., data = Train_Data,family = "binomial")# Running Logistic Regression Model

Predict_LR = predict(LR, newdata = Test_Data,type = "response") #Predicting with test data

pred = prediction(Predict_LR,Test_Data$churn)

recall_perf = performance(pred, measure = "rec") # Measuring the Recall Performance

plot(recall_perf)
```




```r
Predict_LR1=  ifelse(Predict_LR>0.2,'yes','no') #Setting up cutoff value

confusionMatrix(as.factor(Predict_LR1),as.factor(Test_Data$churn)) # Running Confusion Matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  706  53
##        yes 149  92
##
##                Accuracy : 0.798
##                  95% CI : (0.7718, 0.8225)
##     No Information Rate : 0.855
##     P-Value [Acc > NIR] : 1
```

```
##
##                    Kappa : 0.361
##
##  Mcnemar's Test P-Value : 2.322e-11
##
##             Sensitivity : 0.8257
##             Specificity : 0.6345
##          Pos Pred Value : 0.9302
##          Neg Pred Value : 0.3817
##              Prevalence : 0.8550
##          Detection Rate : 0.7060
##    Detection Prevalence : 0.7590
##       Balanced Accuracy : 0.7301
##
##        'Positive' Class : no
##
```

Decision Tree

```r
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.2.2
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.2.2
```

```r
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.2
```

```r
DT = rpart(churn~., data = Train_Data, method = 'class', control=rpart.control(minsplit = 20)) # Runnin

best_CP = DT$cptable[which.min(DT$cptable[,"xerror"]),"CP"] #Finding the Best CP

Best_DT = rpart(churn~., data = Cleaned_Churndata, method = 'class',control=rpart.control(cp=.01)) # Ru

predict_DT = predict(Best_DT, newdata = Test_Data, type = 'class') #Predicting with test data

confusionMatrix(as.factor(predict_DT),as.factor(Test_Data$churn)) # Running Confusion Matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  843  45
##        yes  12 100
##
##                Accuracy : 0.943
##                  95% CI : (0.9268, 0.9565)
##     No Information Rate : 0.855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7461
##
##  Mcnemar's Test P-Value : 2.25e-05
##
##             Sensitivity : 0.9860
##             Specificity : 0.6897
##          Pos Pred Value : 0.9493
##          Neg Pred Value : 0.8929
##              Prevalence : 0.8550
##          Detection Rate : 0.8430
##    Detection Prevalence : 0.8880
##       Balanced Accuracy : 0.8378
##
##        'Positive' Class : no
##
```

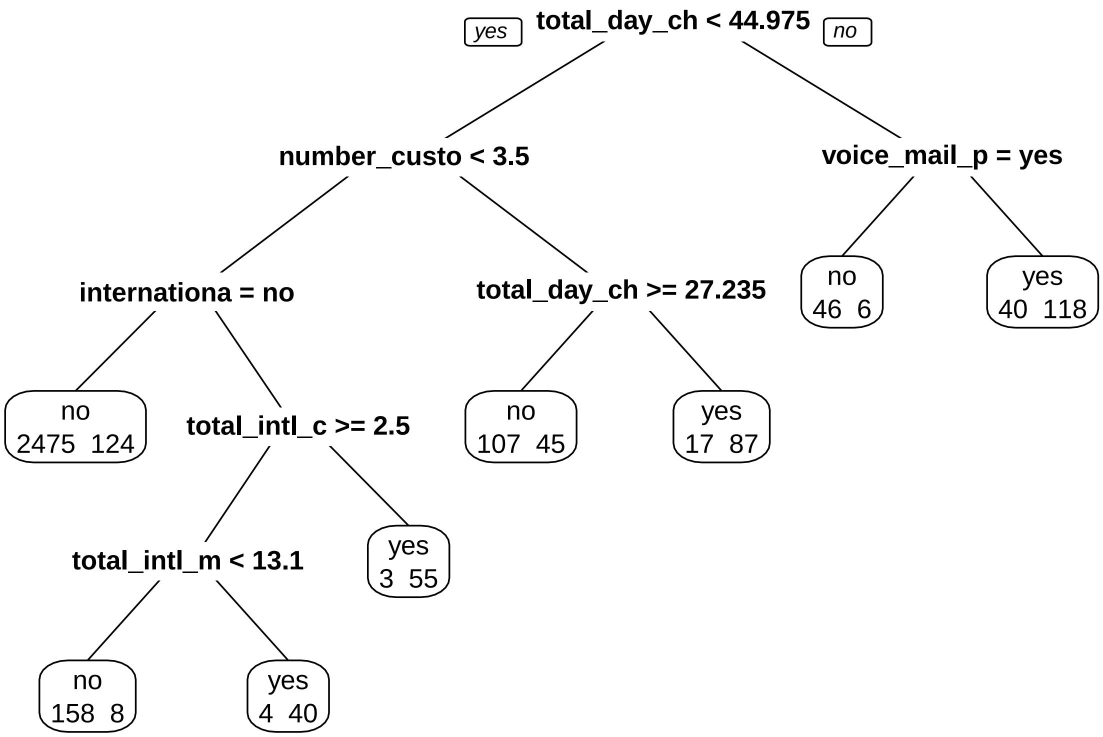
From the above Model, We can see that Decision Tree Model is the best fit for this Data set having the accuracy of 94%, Sensitivity 68.9% and specificity 98% which is better than Logistic regression having the accuracy 80%, sensitivity 60.6 % and specificity 83.7

Note: confusionMatrix function has provided sensitivity and specificity results in the reverse order

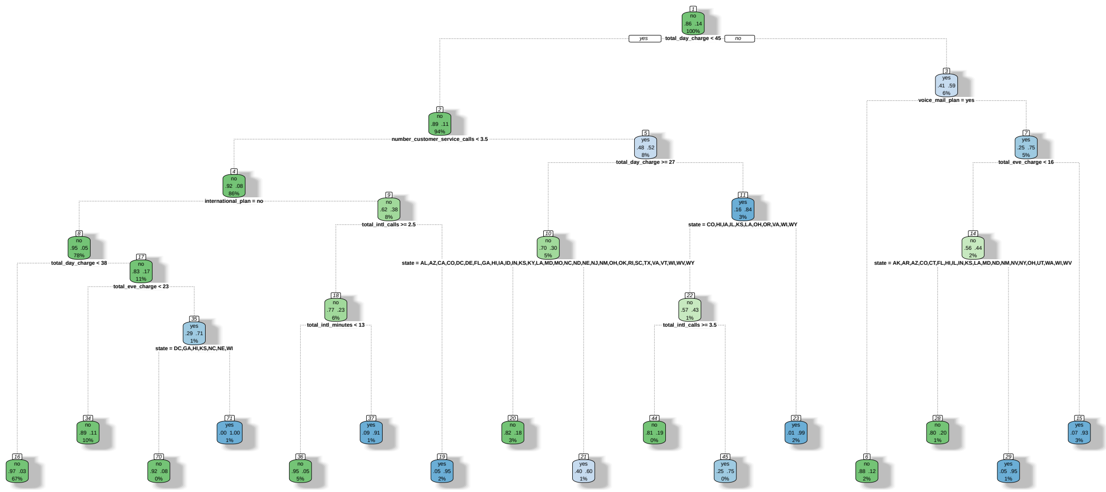
Hence, running the Decision Tree Model for the Entire Data set

Final Decision Tree

```
Churn_Model = rpart(churn~., data = Cleaned_Churndata, method = 'class') # Running Decision tree Model

best_CP = Churn_Model$cptable[which.min(Churn_Model$cptable[,"xerror"]),"CP"] # Finding best Cp

Best_Churn_Model=rpart(churn~., data = Cleaned_Churndata, method = 'class',control=rpart.control(cp=.01)

pruned_Churn_tree <- prune(Churn_Model, cp=best_CP) # Pruning the tree to avoid Over fitting

prp(pruned_Churn_tree,faclen=0,extra=1, roundint=F, digits=5)
```




```
fancyRpartPlot(Best_Churn_Model) # Running the Fancy RPlot for the Best_churn_model
```



Rattle 2022−Dec−11 17:18:02 91837

```
predict_churn = predict(Best_Churn_Model, newdata = Customers_To_Predict, type='class')

predict_churn = as.data.frame(predict_churn)

Customers_To_Predict = cbind(Customers_To_Predict,predict_churn) # Binding Customers_To_Predict with th
```