

Business Analytics Final Project: Group-3- Prediction

Rohith Chandra Koyyala, Manaswini Purumandla, Tejasvini Mavuleti , Riba Khan

2022-05-11

Building the Final Model to predict the churn using Test data and Decision Tree Algorithm

```
set.seed(123)

#After testing for accuracy using validation and training data ,we can use the total dataset for building the actual model to predict the churn
Model_ABC_Wireless<- rpart(churn ~ .,data=churndata_Imputed,method = 'class')

# Show the variable importance

Model_ABC_Wireless$variable.importance

##              total_day_charge              state
##              139.920014              84.068374
## number_customer_service_calls              total_eve_charge
##              82.550272              53.891680
##              international_plan              total_intl_charge
##              53.720359              51.467303
##              total_intl_minutes              total_intl_calls
##              49.739629              42.239384
##              total_day_minutes              number_vmail_messages
##              41.654025              40.725792
##              voice_mail_plan              total_eve_minutes
##              38.993936              32.954992
##              total_night_calls              total_night_charge
##              5.823237              5.186486
##              total_night_minutes              account_length
##              4.244571              3.018568
##              area_code              total_day_calls
##              2.889524              2.591913

# Show the split for variable

head(Model_ABC_Wireless$splits)

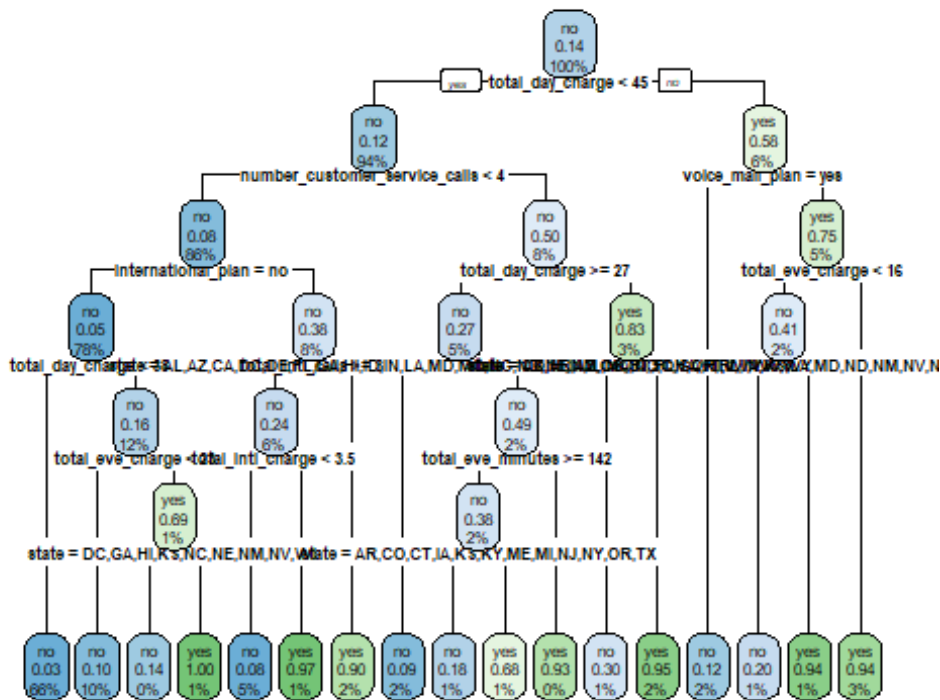
##              count ncat  improve  index adj
## total_day_charge    3333   -1  86.35605  44.975  0
## number_customer_service_calls  3333   -1  77.21135   3.500  0
## international_plan    3333    2  55.77483   1.000  0
## total_day_minutes    3333   -1  24.95119  223.250  0
```

```
## state 3333 51 14.95004 2.000 0
## number_customer_service_calls 3119 -1 80.74432 3.500 0
```

#Plot of DT

#fancyRpartPlot(Model_ABC_Wireless)

```
rpart.plot(Model_ABC_Wireless, cex=0.5)
```



#Predict the probability

```
Prob_decision_tree <- predict(Model_ABC_Wireless, newdata = churndata_Imputed,
, type = "prob")
```

#AUC Value

```
roc(churndata_Imputed$churn, Prob_decision_tree[,2])
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
##
```

```
## Call:
```

```
## roc.default(response = churndata_Imputed$churn, predictor = Prob_decision_tree[, 2])
```

```
##
```

```
## Data: Prob_decision_tree[, 2] in 2850 controls (churndata_Imputed$churn no
```

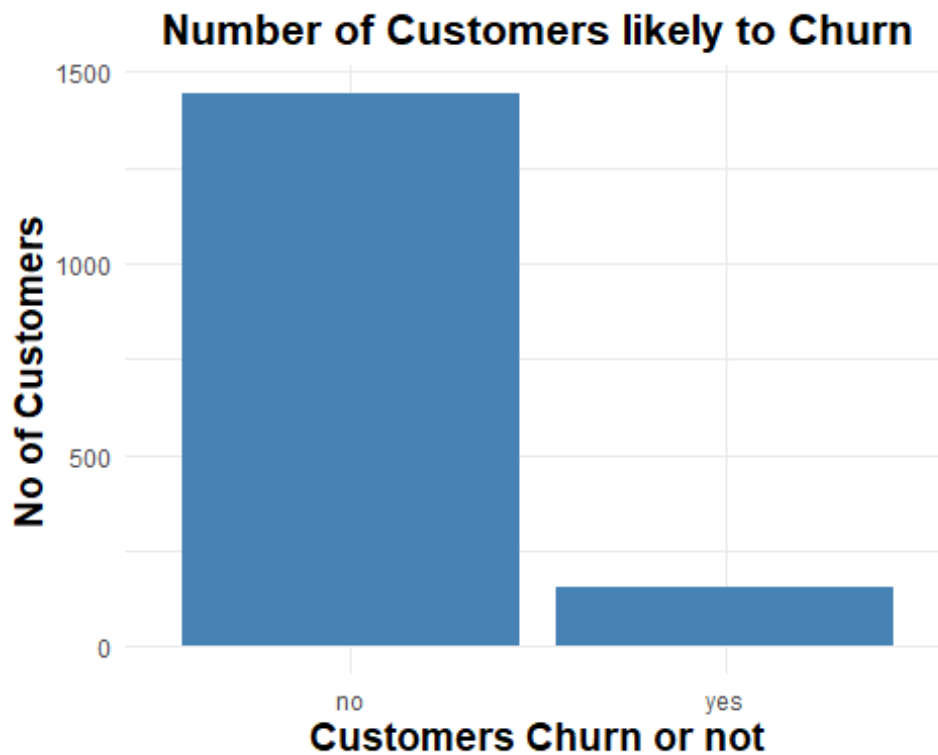
```
) < 483 cases (churndata_Imputed$churn yes).  
## Area under the curve: 0.8931
```

Prediction of the Test data

```
set.seed(123)  
load("C:/Users/kramr/Downloads/Customers_To_Predict.RData")  
  
count(Customers_To_Predict)  
  
## # A tibble: 1 × 1  
##       n  
##   <int>  
## 1  1600  
  
#summary(Customers_To_Predict)  
  
# Check for NA Values  
#colMeans(is.na(Customers_To_Predict))  
  
Churn_Prob <- predict(Model_ABC_Wireless,Customers_To_Predict,type = "prob")  
  
head(Churn_Prob)  
  
##           no           yes  
## 1 0.96969697 0.03030303  
## 2 0.96969697 0.03030303  
## 3 0.96969697 0.03030303  
## 4 0.92485549 0.07514451  
## 5 0.96969697 0.03030303  
## 6 0.04819277 0.95180723  
  
predict_churn <- predict(Model_ABC_Wireless,Customers_To_Predict,type = "class")  
head(predict_churn)  
  
##    1    2    3    4    5    6  
## no   yes  
## Levels: no yes  
  
predict_churn<- as.data.frame(predict_churn)  
  
summary(predict_churn)  
  
## predict_churn  
## no :1445  
## yes: 155
```

Plot for summary of the Test data

```
ggplot(predict_churn) +  
  aes(x = predict_churn) +  
  geom_bar(fill = "steelblue") +  
  labs(x = "Customers Churn or not",  
       y = "No of Customers", title = "Number of Customers likely to Churn") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 16L,  
    face = "bold", hjust = 0.5), axis.title.y = element_text(size = 14L, face =  
    "bold"), axis.title.x = element_text(size = 14L,  
    face = "bold"))
```



predict_churn

no :1444 yes: 156