GROUP 7

PROJECT REPORT

CUSTOMER CHURN PREDICTION

Prajwal C N & Kunal Sharma
BUSINESS ANALYTICS
MIS 64036

ABSTRACT

Churn is a problem for telecom companies because it is more difficult to acquire new customers than it is to keep existing customers from leaving. Customer Churn Modeling has received a lot of attention recently, as there are signs that existing customers generate a large portion of corporate profit. Companies are also very interested in identifying consumers who are likely to become churn, and they often use Data Mining methods to help them do so. We recognized customers who are likely to churn in this project and provided sufficient intercession to encourage them to remain based on available data.

INTRODUCTION

Customer retention and acquisition are important factors that have a direct impact on a company's profitability and striking a balance between the two is not easy. Since the churn rate influences the company's sales, customer retention is likely to be a key factor.

Customers can switch service providers for a variety of reasons, including network problems, poor customer service, and a high monthly plan. These problems can be addressed by offering discounts or improved service, among other things, to keep customers from switching service providers.

We can use this knowledge to derive patterns and forecast future outcomes in modern times when we have the analytical capabilities to interpret and analyze complex data and extract useful information.

The aim of this project is to use a predictive model to evaluate data and identify trends to predict when an established customer will switch service providers. Many predictive models, such as Nave bias, K nearest neighbor, and regression, can be used to perform our study. Here, we'll use logistic regression to construct our model.

Overview of Data

ABC wireless company has provided the following data from which we can infer:

- Demographics
    - State
    - Account length
    - Area code
    - International plan
    - Voice-mail plan
- Calling Behaviour
    - Number of messages
    - Total day minutes, Total day calls, Total day charge
    - Total evening minutes, Total evening calls and Total evening charges
    - Total night minutes, Total night calls and Total night charges
    - Total International minutes, Total International calls and Total International charges
    - Number of calls to customer service

Data Preprocessing

Importing the Churn dataset

```
Churn_Data <- read_csv("Churn_Train.csv")

# Inspecting data
head(Churn_Data)

## # A tibble: 6 x 20
##   state account_length area_code     international_plan voice_mail_plan
##   <chr>          <dbl> <chr>         <chr>              <chr>
## 1 NV               125 area_code_510 no                 no
## 2 HI               108 area_code_415 no                 no
## 3 DC                82 area_code_415 no                 no
## 4 HI                NA area_code_408 no                 yes
## 5 OH                83 area_code_415 no                 no
## 6 MO                89 area_code_415 no                 no
## # ... with 15 more variables: number_vmail_messages <dbl>,
## #   total_day_minutes <dbl>, total_day_calls <dbl>, total_day_charge <dbl>
```

```
,
## #   total_eve_minutes <dbl>, total_eve_calls <dbl>, total_eve_charge <dbl>
,
## #   total_night_minutes <dbl>, total_night_calls <dbl>,
## #   total_night_charge <dbl>, total_intl_minutes <dbl>, total_intl_calls <
dbl>,
## #   total_intl_charge <dbl>, number_customer_service_calls <dbl>, churn <c
hr>
```

Examining the dataset

```
## Rows: 3,333
## Columns: 20
## $ state                         <chr> "NV", "HI", "DC", "HI", "OH", "MO",
"NC"~
## $ account_length                <dbl> 125, 108, 82, NA, 83, 89, 135, 28, 8
6, 6~
## $ area_code                     <chr> "area_code_510", "area_code_415", "a
rea_~
## $ international_plan            <chr> "no", "no", "no", "no", "no", "no",
"no"~
## $ voice_mail_plan               <chr> "no", "no", "no", "yes", "no", "no",
"no~
## $ number_vmail_messages         <dbl> 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, NA
, 32~
## $ total_day_minutes             <dbl> 2013.4, 291.6, 300.3, 110.3, 337.4,
178.~
## $ total_day_calls               <dbl> 99, 99, 109, 71, 120, 81, 81, 87, 11
5, 1~
## $ total_day_charge              <dbl> 28.66, 49.57, 51.05, 18.75, 57.36, 3
0.38~
## $ total_eve_minutes             <dbl> 1107.6, 221.1, 181.0, 182.4, 227.4,
NA, ~
## $ total_eve_calls               <dbl> 107, 93, 100, 108, 116, 74, 114, 92,
112~
## $ total_eve_charge              <dbl> 14.93, 18.79, 15.39, 15.50, 19.33, 1
9.86~
## $ total_night_minutes           <dbl> 243.3, 229.2, 270.1, 183.8, 153.9, 1
31.9~
## $ total_night_calls             <dbl> 92, 110, 73, 88, 114, 120, 82, 112,
95, ~
## $ total_night_charge            <dbl> 10.95, 10.31, 12.15, 8.27, 6.93, 5.9
4, 9~
## $ total_intl_minutes            <dbl> 10.9, 14.0, 11.7, 11.0, 15.8, 9.1, 1
0.3,~
## $ total_intl_calls              <dbl> 7, 9, 4, 8, 7, 4, 6, 3, 7, 6, 7, NA,
4, ~
## $ total_intl_charge             <dbl> 2.94, 3.78, 3.16, 2.97, 4.27, 2.46,
2.78~
## $ number_customer_service_calls <dbl> 0, 2, 0, 2, 0, 1, 1, 3, 2, 4, 1, NA,
```

```
3, ~
## $ churn                        <chr> "no", "yes", "yes", "no", "yes", "no
", "~
```

Summary statistics of dataset

```
summary(Churn_Data)

##     state           account_length     area_code         international_pla
n
##  Length:3333        Min.   :-209.00   Length:3333        Length:3333
##  Class :character   1st Qu.:  72.00   Class :character   Class :character
##  Mode  :character   Median : 100.00   Mode  :character   Mode  :character
##                     Mean   :  97.32
##                     3rd Qu.: 127.00
##                     Max.   : 243.00
##                     NA's   :501
##  voice_mail_plan    number_vmail_messages total_day_minutes total_day_call
s
##  Length:3333        Min.   :-10.000       Min.   :   0.0    Min.   :  0.0
##  Class :character   1st Qu.:  0.000       1st Qu.: 149.3    1st Qu.: 87.0
##  Mode  :character   Median :  0.000       Median : 190.5    Median :101.0
##                     Mean   :  7.333       Mean   : 418.9    Mean   :100.3
##                     3rd Qu.: 16.000       3rd Qu.: 237.8    3rd Qu.:114.0
##                     Max.   : 51.000       Max.   :2185.1    Max.   :165.0
##                     NA's   :200           NA's   :200       NA's   :200
##  total_day_charge total_eve_minutes total_eve_calls total_eve_charge
##  Min.   : 0.00    Min.   :   0.0    Min.   :  0.0   Min.   : 0.00
##  1st Qu.:24.45    1st Qu.: 170.5    1st Qu.: 87.0   1st Qu.:14.14
##  Median :30.65    Median : 209.9    Median :100.0   Median :17.09
##  Mean   :30.63    Mean   : 324.3    Mean   :100.1   Mean   :17.08
##  3rd Qu.:36.84    3rd Qu.: 257.6    3rd Qu.:114.0   3rd Qu.:20.00
##  Max.   :59.64    Max.   :1244.2    Max.   :170.0   Max.   :30.91
##  NA's   :200      NA's   :301       NA's   :200     NA's   :200
##  total_night_minutes total_night_calls total_night_charge total_intl_minut
es
##  Min.   : 23.2       Min.   : 33.0     Min.   : 1.040     Min.   : 0.00
##  1st Qu.:167.3       1st Qu.: 87.0     1st Qu.: 7.530     1st Qu.: 8.50
##  Median :201.4       Median :100.0     Median : 9.060     Median :10.30
##  Mean   :201.2       Mean   :100.1     Mean   : 9.054     Mean   :10.23
##  3rd Qu.:235.3       3rd Qu.:113.0     3rd Qu.:10.590     3rd Qu.:12.10
##  Max.   :395.0       Max.   :175.0     Max.   :17.770     Max.   :20.00
##  NA's   :200                           NA's   :200        NA's   :200
##  total_intl_calls total_intl_charge number_customer_service_calls
##  Min.   : 0.00    Min.   :0.000     Min.   :0.000
##  1st Qu.: 3.00    1st Qu.:2.300     1st Qu.:1.000
##  Median : 4.00    Median :2.780     Median :1.000
##  Mean   : 4.47    Mean   :2.762     Mean   :1.561
##  3rd Qu.: 6.00    3rd Qu.:3.270     3rd Qu.:2.000
##  Max.   :20.00    Max.   :5.400     Max.   :9.000
```

```
##  NA's   :301      NA's   :200       NA's   :200
##     churn
##  Length:3333
##  Class :character
##  Mode  :character
##
##
##
##
```

Data cleaning and Exploratory Data Analysis

From glimpse we can see that, Some of the character variables can be converted into factors, So Converting character variables to factors.

```
Churn_Data <- Churn_Data %>% mutate_if(is.character, as.factor)
```

From summary we can see that, Churn_Data dataset has both NA and negative values, So investigating and handling further.

```
# Checking NULL values in the dataset at column level.
colSums(is.na(Churn_Data))
```

```
##                         state                account_length
##                             0                           501
##                     area_code            international_plan
##                             0                             0
##               voice_mail_plan         number_vmail_messages
##                             0                           200
##             total_day_minutes               total_day_calls
##                           200                           200
##              total_day_charge             total_eve_minutes
##                           200                           301
##               total_eve_calls              total_eve_charge
##                           200                           200
##           total_night_minutes             total_night_calls
##                           200                             0
##            total_night_charge            total_intl_minutes
##                           200                           200
##              total_intl_calls             total_intl_charge
##                           301                           200
## number_customer_service_calls                         churn
##                           200                             0
```

```r
# Checking Negative values in the dataset at column level.
sapply(Churn_Data %>% select_if(is.numeric), function(x) {
  sum(x < 0, na.rm = TRUE)
})
```

```
##                 account_length          number_vmail_messages
##                             51                            201
##              total_day_minutes                total_day_calls
##                              0                              0
##               total_day_charge              total_eve_minutes
##                              0                              0
##                total_eve_calls               total_eve_charge
##                              0                              0
##            total_night_minutes              total_night_calls
##                              0                              0
##             total_night_charge             total_intl_minutes
##                              0                              0
##               total_intl_calls              total_intl_charge
##                              0                              0
## number_customer_service_calls
##                              0
```

Since account length, and other numeric variables has few negative values, assuming them as erroneous values, we cannot void them because their corresponding churn value is “no” which means they are still associated with the provider.

```r
Churn_Data <-
  Churn_Data %>% mutate_if(is.numeric, function(x) {
    ifelse(x < 0, abs(x), x)
  })
```

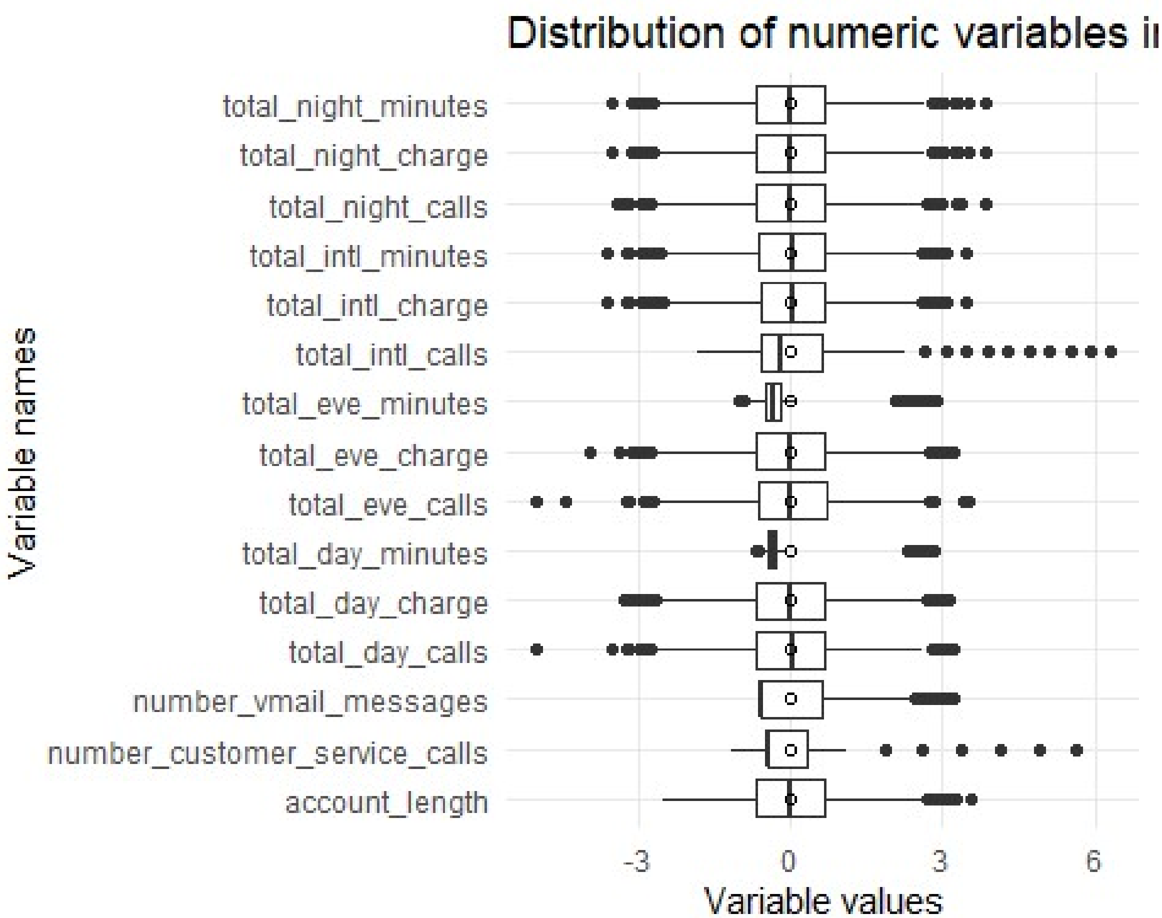
From the above plot, we see there are outliers in the data, in order to impute NA values, we have several techniques such as mean, median, KNN imputation and linear regression. Since there are many outliers in the data, its not feasible to do mean imputation. Hence using **median imputation** technique.

```r
imputation_model <- preProcess(Churn_Data %>% select_if(is.numeric),method =
"medianImpute")
data <- predict(imputation_model, Churn_Data %>% select_if(is.numeric))

Churn_Data <- Churn_Data %>% select(setdiff(names(Churn_Data), names(data)))
%>% cbind(data)
```

Visualizing distribution of Churn numeric variable.

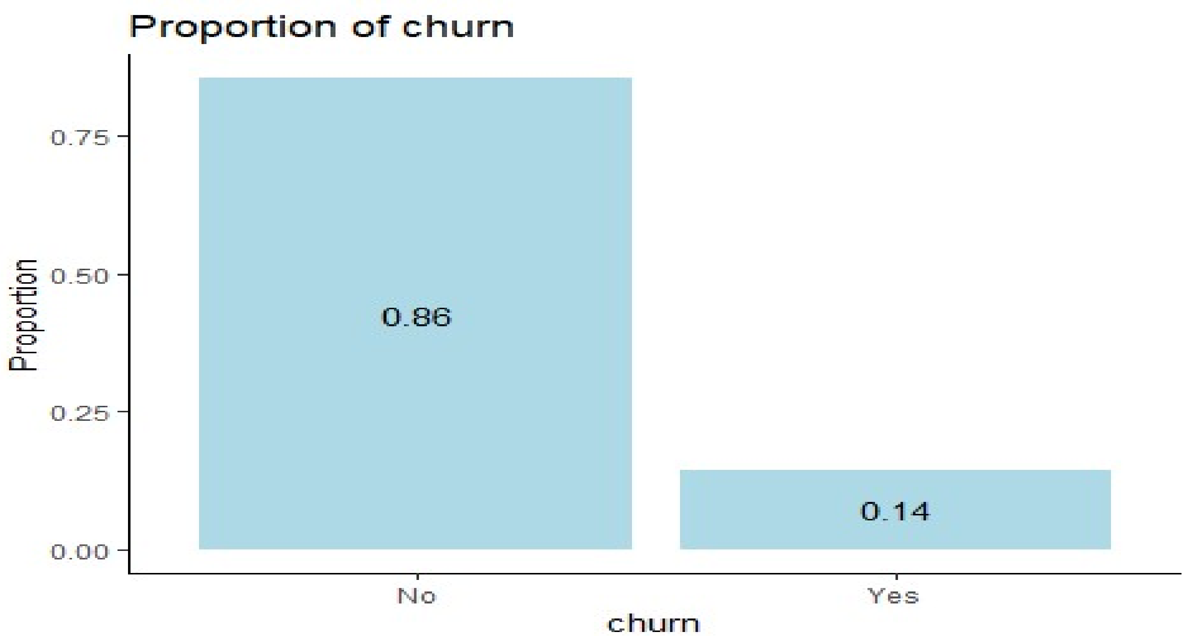
```
Churn_Data %>% select_if(is.numeric) %>% mutate_all(scale) %>% gather("featur
es","values") %>% na.omit() %>%
  ggplot(aes(x = features, y = values)) +
  geom_boxplot(show.legend = FALSE) +
  stat_summary(fun = mean, geom = "point", pch = 1) + # Add average to the bo
xplot
  scale_y_continuous(name = "Variable values", minor_breaks = NULL) +
  scale_fill_brewer(palette = "Set1") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Variable names") +
  ggtitle(label = "Distribution of numeric variables in Churn dataset")
```



From above box plot we can see that, most of the variables are not normally distributed and there are many outliers in the data.

Visualizing distribution of Churn categorical variable.

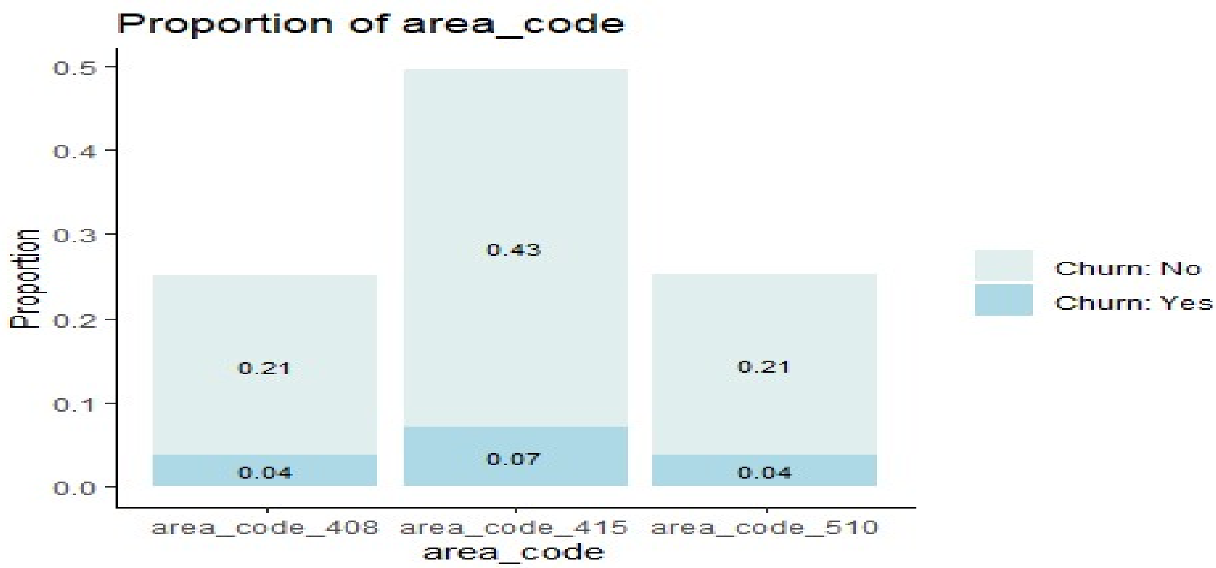
```
ggplot(Churn_Data, aes(x=churn, y=..prop..,group = 1)) +
  geom_bar(fill="light blue") +
  theme_classic() +
  geom_text(aes(label=round(..prop..,2)),stat = "count",
            position = position_stack(vjust=0.5)) +
  labs(y = 'Proportion', title = "Proportion of churn") +
  scale_x_discrete(labels = c("No","Yes"))
```



From the above graph we can see that only around 14% of population are churned and rest 86% are retained in the telecom network.

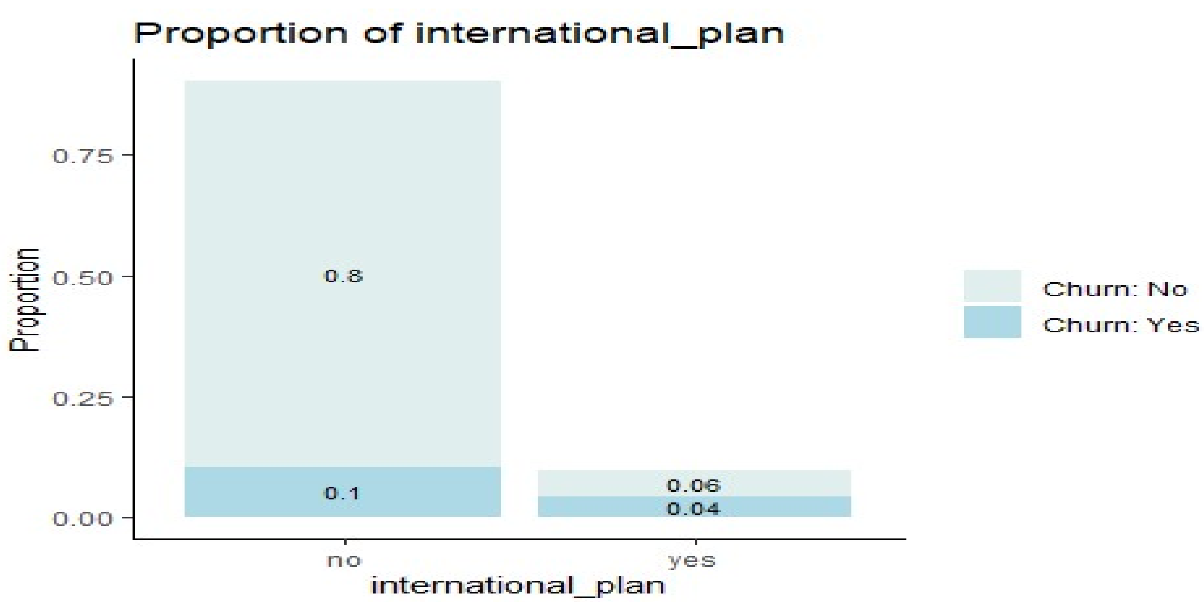
Proportion of area_code

```
as.data.frame(prop.table(table(Churn_Data[c("area_code","churn")]))) %>%
  ggplot(aes(x=area_code,y=Freq,fill=churn)) + geom_col() +
  geom_text(aes(label=round(Freq,2)),position = position_stack(vjust = 0.5),s
ize=2.8) +
  theme_classic() + labs( y = 'Proportion', title = "Proportion of area_code"
) +
  theme(legend.title = element_blank()) +
  scale_fill_manual(labels = c("Churn: No","Churn: Yes"),
                    values = c("azure2","light blue"))
```



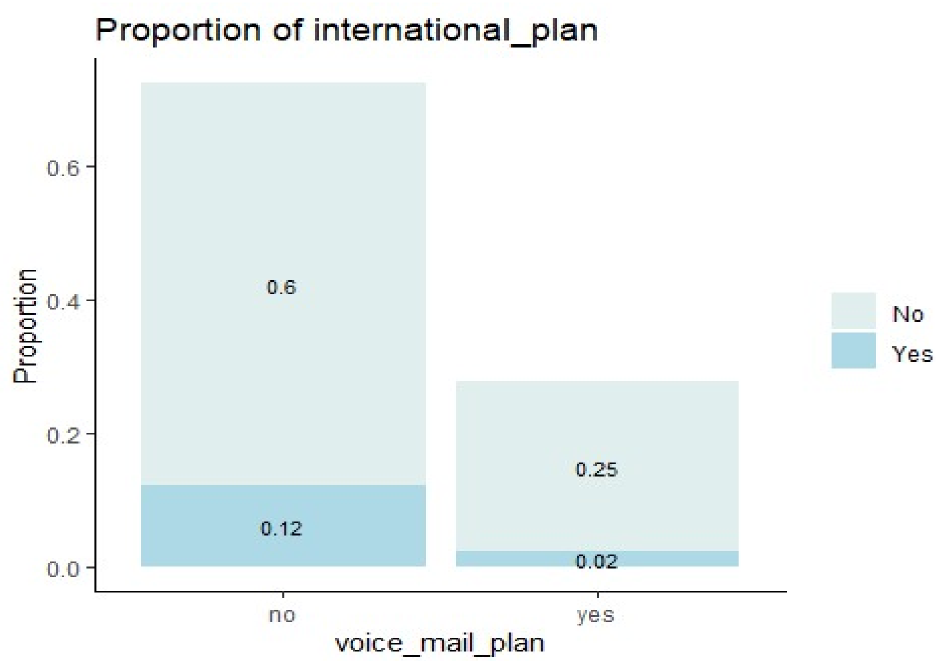
Proportion of international_plan

```
as.data.frame(prop.table(table(Churn_Data[c("international_plan","churn")])))
%>%
  ggplot(aes(x=international_plan,y=Freq,fill=churn)) + geom_col() +
  geom_text(aes(label=round(Freq,2)),position = position_stack(vjust = 0.5),s
ize=2.8) +
  theme_classic() + labs( y = 'Proportion', title = "Proportion of internatio
nal_plan") +
  theme(legend.title = element_blank()) +
  scale_fill_manual(labels = c("Churn: No","Churn: Yes"),
                    values = c("azure2","light blue"))
```



Proportion of voice_mail_plan

```
as.data.frame(prop.table(table(Churn_Data[c("voice_mail_plan","churn")]))) %>%
  ggplot(aes(x=voice_mail_plan,y=Freq,fill=churn)) + geom_col() +
  geom_text(aes(label=round(Freq,2)),position = position_stack(vjust = 0.5),size=2.8) +
  theme_classic() + labs( y = 'Proportion', title = "Proportion of international_plan") +
  theme(legend.title = element_blank()) +
  scale_fill_manual(labels = c("No","Yes"),
                    values = c("azure2","light blue"))
```

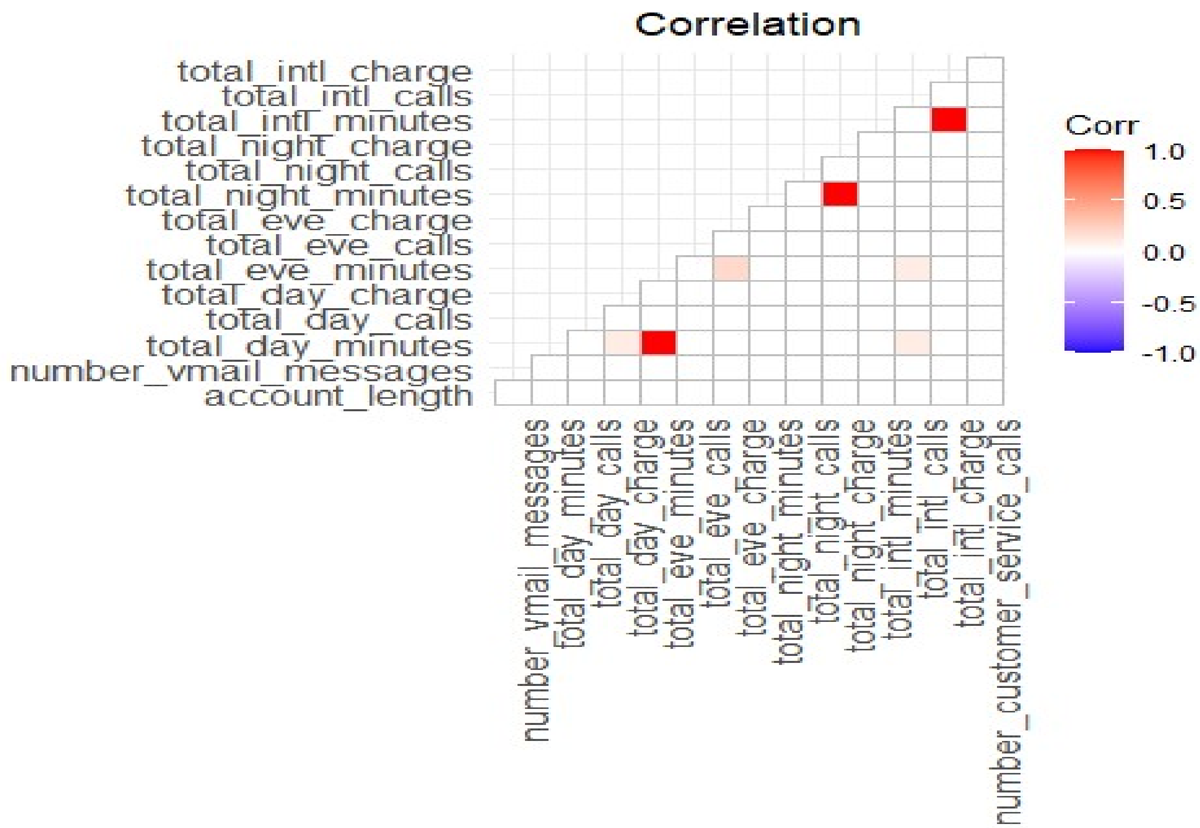


Correlation

The image below will assist us in determining the variables' correlation.

```
Churn_Data_cor <- round(cor(Churn_Data %>% select_if(is.numeric)), 1)

ggcorrplot(Churn_Data_cor,  title = "Correlation", type = "lower") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 90))
```



Total minutes and total charge for the day, evening, night, and international are strongly linked, we may omit them since they can cause "multi-collinearity" issue.

Model Strategy

The task of using a classifier to divide an example into two categories is known as binary classification. Since the target variable in this data is categorical, and outcome for this model is a likelihood or probability of odds between 0 and 1, so we will using both **logistic regression** and **decision tree** to solve this problem and comparing both models performance matrix and choosing better one.

Logistic Regression

Pre-Processing of data

Splitting dataset into training (80%) and validation (20%) sets

The training set will be used to fit our model which we will be testing over the testing set.

```
set.seed(12)
index <- createDataPartition(Churn_Data$churn, p=0.8, list=FALSE)
Churn_Data_train_df <- Churn_Data[index,]
Churn_Data_test_df <- Churn_Data[-index,]
```

Scaling train and test churn datasets

```
scaling <- preProcess(Churn_Data_train_df %>% select_if(is.numeric), method =
c("center", "scale"))
Churn_Data_train_norm <- predict(scaling, Churn_Data_train_df %>% select_if(i
s.numeric))
Churn_Data_test_norm <- predict(scaling, Churn_Data_test_df %>% select_if(is.
numeric))

Churn_Data_train_norm$churn <- Churn_Data_train_df$churn
Churn_Data_test_norm$churn <- Churn_Data_test_df$churn
```

Model Construction

```
Model_1 <- glm(churn ~ ., data = Churn_Data_train_norm , family= "binomial")
```

```
summary(Model_1)

##
## Call:
## glm(formula = churn ~ ., family = "binomial", data = Churn_Data_train_norm
)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1055  -0.5099  -0.3508  -0.2013   3.1185
##
## Coefficients: (3 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.300038   0.079590 -28.899  < 2e-16 ***
## account_length                 0.018545   0.062143   0.298  0.76538
## number_vmail_messages          0.127134   0.177494   0.716  0.47382
## total_day_minutes             -2.292991   1.250524  -1.834  0.06671 .
## total_day_calls                0.046819   0.061769   0.758  0.44847
## total_day_charge               0.853647   0.117549   7.262 3.81e-13 ***
## total_eve_minutes              2.184432   1.242318   1.758  0.07869 .
## total_eve_calls               -0.041535   0.061678  -0.673  0.50068
## total_eve_charge               0.068285   0.194636   0.351  0.72571
## total_night_minutes            1.835634  47.612807   0.039  0.96925
## total_night_calls              0.038145   0.062195   0.613  0.53967
## total_night_charge            -1.703082  47.611535  -0.036  0.97147
## total_intl_minutes            -5.955075  16.386684  -0.363  0.71630
## total_intl_calls              -0.186343   0.066424  -2.805  0.00503 **
## total_intl_charge              6.191261  16.384772   0.378  0.70553
## number_customer_service_calls  0.689509   0.056961  12.105  < 2e-16 ***
## area_code_area_code_408        0.005171   0.075775   0.068  0.94559
## area_code_area_code_415       -0.025656   0.075043  -0.342  0.73244
## area_code_area_code_510              NA         NA      NA       NA
## international_plan_no         -0.603710   0.048133 -12.542  < 2e-16 ***
## international_plan_yes               NA         NA      NA       NA
## voice_mail_plan_no             0.560696   0.181780   3.084  0.00204 **
## voice_mail_plan_yes                  NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209  on 2666  degrees of freedom
## Residual deviance: 1740  on 2647  degrees of freedom
## AIC: 1780
##
## Number of Fisher Scoring iterations: 6
```

Now we can infer from the summary of the model, the significant variables, p values, test statistics etc..

Predict values using based on Model_1.

```
pred_probs <- predict(object = Model_1,Churn_Data_test_norm, type = "response
")

# Finding accuracy for the model
# Function to find the accuracy, based on probability(0.5 - 0.9)
sequence1 <- data.frame(pred_cutoff = seq(0.5,0.9,0.1), pred_accuracy = rep(0
,5))

for (i in 1:5){
  Model_11 <- as.factor(ifelse(pred_probs > sequence1$pred_cutoff[i], "yes",
"no"))
  sequence1[i,2] <- confusionMatrix(Model_11,Churn_Data_test_df$churn )$overa
ll[1]
}

# Shows the probability with its accuracy
sequence1

##   pred_cutoff pred_accuracy
## 1         0.5     0.8678679
## 2         0.6     0.8693694
## 3         0.7     0.8663664
## 4         0.8     0.8603604
## 5         0.9     0.8588589
```

Assigning labels based on maximum probability prediction

```
Model_Pre_lables <- as.factor(ifelse(pred_probs>sequence1$pred_cutoff[which.m
ax(sequence1$pred_accuracy)] ,"yes","no"))
```

Performance Metrics

Confusion matrix for churn model.

```
confusionMatrix(Model_Pre_lables,Churn_Data_test_norm$churn)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  563  80
##        yes   7  16
##
##                Accuracy : 0.8694
##                  95% CI : (0.8414, 0.894)
##     No Information Rate : 0.8559
```

```
##     P-Value [Acc > NIR] : 0.1746
##
##                   Kappa : 0.2258
##
##  Mcnemar's Test P-Value : 1.171e-14
##
##             Sensitivity : 0.9877
##             Specificity : 0.1667
##          Pos Pred Value : 0.8756
##          Neg Pred Value : 0.6957
##              Prevalence : 0.8559
##          Detection Rate : 0.8453
##    Detection Prevalence : 0.9655
##       Balanced Accuracy : 0.5772
##
##        'Positive' Class : no
##
```

From the above confusion matrix we can see that, **Accuracy** -> `0.869` **Sensitivity** -> `0.9877` **Specificity** -> `0.1667`.

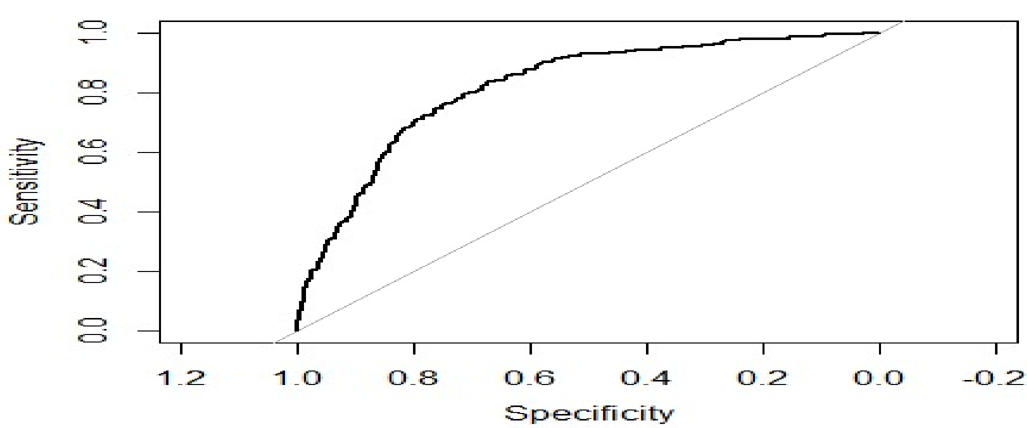
ROC Curve of the model 1

```
roc(Churn_Data_test_df$churn, pred_probs)

##
## Call:
## roc.default(response = Churn_Data_test_df$churn, predictor = pred_probs)
##
## Data: pred_probs in 570 controls (Churn_Data_test_df$churn no) < 96 cases
(Churn_Data_test_df$churn yes).
## Area under the curve: 0.7973

plot.roc(roc(Churn_Data_test_df$churn, pred_probs))
```

From the above analysis we see Area under curve (AUC) of the model is 0.7973.



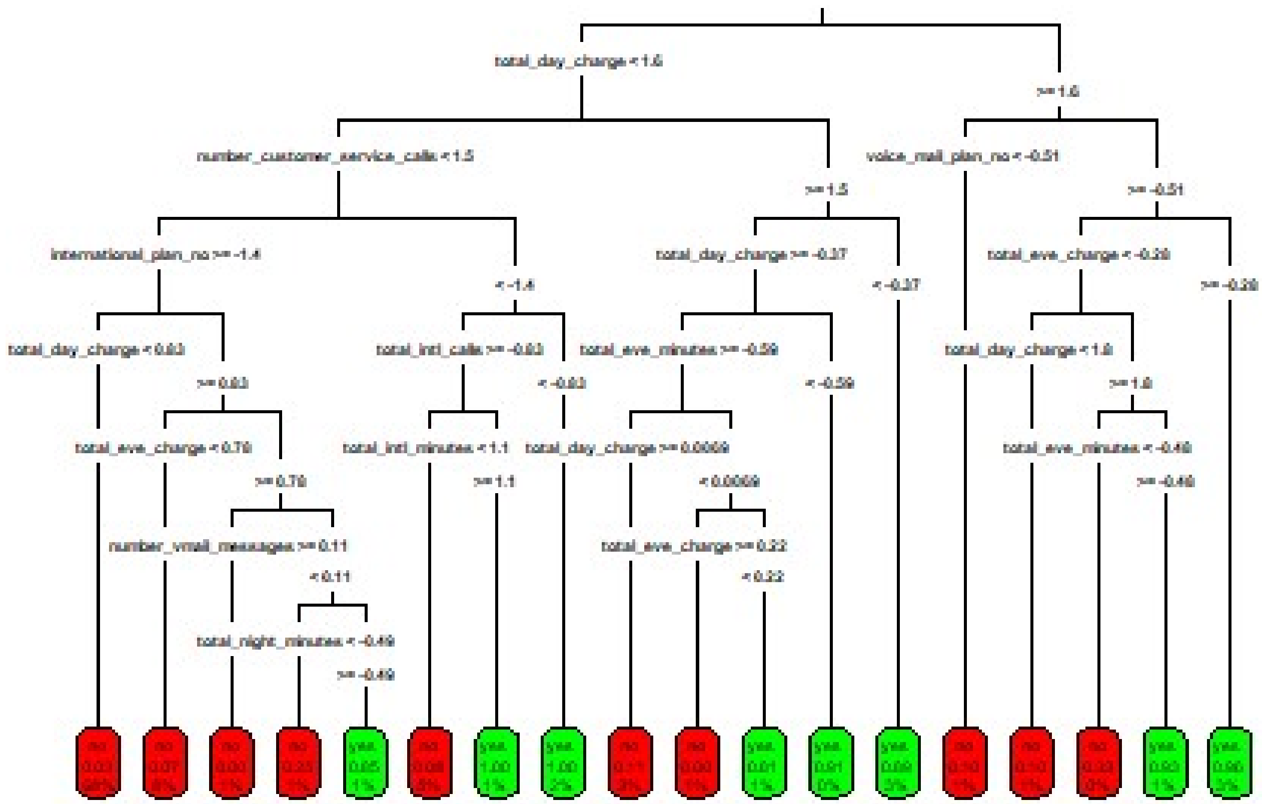
Decision Tree Classifier

Constructing decision tree model on above partitioned data

Model Construction

```
Model_2 <- rpart(churn ~ ., data = Churn_Data_train_norm, method = "class")

rpart.plot(Model_2, type = 3, box.palette = c("red", "green"), fallen.leaves
= TRUE)
```



From above, we can see the summary plot of the model where each variable is split into branches or nodes based on **Entropy value**. To use entropy to determine the optimal features is split upon, the algorithm calculates the change in homogeneity that would result from a split on each possible feature which is a measure known as **information gain**.

Predict values using based on Model_2.

```
pred_labels <- predict(object = Model_2,Churn_Data_test_norm, type = "class")
pred_probs <- predict(object = Model_2,Churn_Data_test_norm)
```

Performance Metrics

Confusion matrix for significant variable model.

```
confusionMatrix(pred_labels,Churn_Data_test_norm$churn)

## Confusion Matrix and Statistics
##
##            Reference
## Prediction  no yes
##        no  563  38
##        yes   7  58
##
##                Accuracy : 0.9324
##                  95% CI : (0.9106, 0.9503)
##     No Information Rate : 0.8559
##     P-Value [Acc > NIR] : 5.285e-10
##
##                   Kappa : 0.6837
##
##  Mcnemar's Test P-Value : 7.744e-06
##
##             Sensitivity : 0.9877
##             Specificity : 0.6042
##          Pos Pred Value : 0.9368
##          Neg Pred Value : 0.8923
##              Prevalence : 0.8559
##          Detection Rate : 0.8453
##    Detection Prevalence : 0.9024
##       Balanced Accuracy : 0.7959
##
##        'Positive' Class : no
##
```

From above confusion metric we can see that, **Accuracy** -> `0.9324` **Sensitivity** -> `0.9877` **Specificity** -> `0.6042`

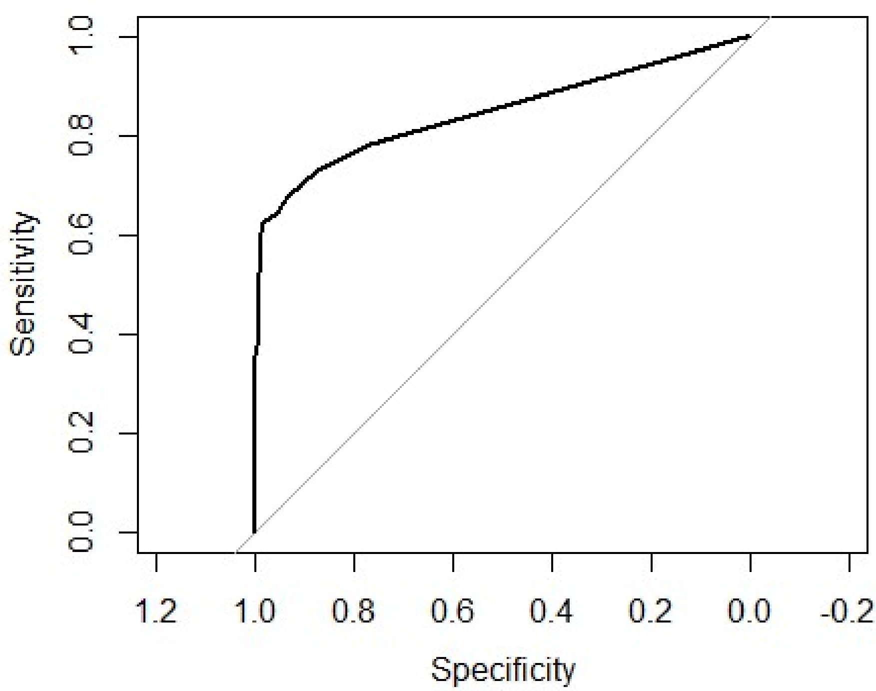
AUC of the model 2

```
roc(Churn_Data_test_df$churn, pred_probs[,2])

##
## Call:
## roc.default(response = Churn_Data_test_df$churn, predictor = pred_probs[,
2])
##
## Data: pred_probs[, 2] in 570 controls (Churn_Data_test_df$churn no) < 96 c
ases (Churn_Data_test_df$churn yes).
## Area under the curve: 0.847

plot.roc(roc(Churn_Data_test_df$churn, pred_probs[,2]))
```

From the above analysis we see Area under curve (AUC) of the model is 0.847.



Conclusion

From logistics regression and decision tree models we found that, AUC and accuracy values of decision tree are higher. Hence choosing decision tree as best model for future predictions.

Predicting Model based on Customers_To_Predict data

```
load("C:/Users/prajw/Downloads/Customers_To_Predict.RData")

Customers_To_Predict <- Customers_To_Predict %>% select(-state) %>% fastDummi
es::dummy_cols(., remove_selected_columns = TRUE)
Customers_To_Predict <- as.data.frame(scale(Customers_To_Predict))
predict_labels <- predict(object = Model_2, Customers_To_Predict, type = "cla
ss")


Customers_To_Predict <- Customers_To_Predict %>% mutate(Churn_Prob = predict_
labels)

table(Customers_To_Predict$Churn_Prob)

##
##  no yes
## 903  97
```

We're using a data set that contains a list of customers for whom we need to forecast future churn. We were able to predict that out of 1000 customers 97 customers moving from ABC wireless to another network.

Contributions

NAME	CONTRIBUTION
Prajwal C N pchamben@kent.edu	Model Building, Model Performance, Predictions and Results, Data Cleaning, Data Exploration, Documentation and presentation
Kunal Sharma Ksharm11@kent.edu	Model Building, Model Performance, Predictions and Results, Data Cleaning, Data Exploration, Documentation and presentation