# Customer Churn Analysis - Final Project

Ishika Gupta

04/21/22

```
# Load the dataset
data = read_csv("/Users/ishika/Desktop/Applied Data Science/Final Project /BankChurners.csv")
```

```
# Dropping unnecessay columns
data <- subset(data, select = -c(CLIENTNUM, Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts
```

```
# Churned customers are marked as 1
# Existing customers are marked as 0
data <- data %>%
    mutate(Attrition_Flag = recode(Attrition_Flag,
                         "Existing Customer" = 0,
                         "Attrited Customer" = 1))

data
```

```
## # A tibble: 10,127 x 20
##    Attrition_Flag Customer_Age Gender Dependent_count Education_Level
##             <dbl>        <dbl> <chr>            <dbl> <chr>
##  1              0           45 M                    3 High School
##  2              0           49 F                    5 Graduate
##  3              0           51 M                    3 Graduate
##  4              0           40 F                    4 High School
##  5              0           40 M                    3 Uneducated
##  6              0           44 M                    2 Graduate
##  7              0           51 M                    4 Unknown
##  8              0           32 M                    0 High School
##  9              0           37 M                    3 Uneducated
## 10              0           48 M                    2 Graduate
## # ... with 10,117 more rows, and 15 more variables: Marital_Status <chr>,
## #   Income_Category <chr>, Card_Category <chr>, Months_on_book <dbl>,
## #   Total_Relationship_Count <dbl>, Months_Inactive_12_mon <dbl>,
## #   Contacts_Count_12_mon <dbl>, Credit_Limit <dbl>, Total_Revolving_Bal <dbl>,
## #   Avg_Open_To_Buy <dbl>, Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>,
## #   Total_Trans_Ct <dbl>, Total_Ct_Chng_Q4_Q1 <dbl>,
## #   Avg_Utilization_Ratio <dbl>
```

### Necessary Data Transformation

```
# Changing data types

# Marital_status character -> factor
data$Marital_Status <- as.factor(data$Marital_Status)
summary(data$Marital_Status)
```

```
## Divorced  Married   Single  Unknown
##      748     4687     3943      749
```

```
#Income category character -> factor
data$Income_Category <- as.factor(data$Income_Category)
summary(data$Income_Category)
```

```
##        $120K +    $40K - $60K    $60K - $80K   $80K - $120K Less than $40K
##            727           1790           1402           1535           3561
##        Unknown
##           1112
```

```
#Card category character -> factor
data$Card_Category <- as.factor(data$Card_Category)
summary(data$Card_Category)
```

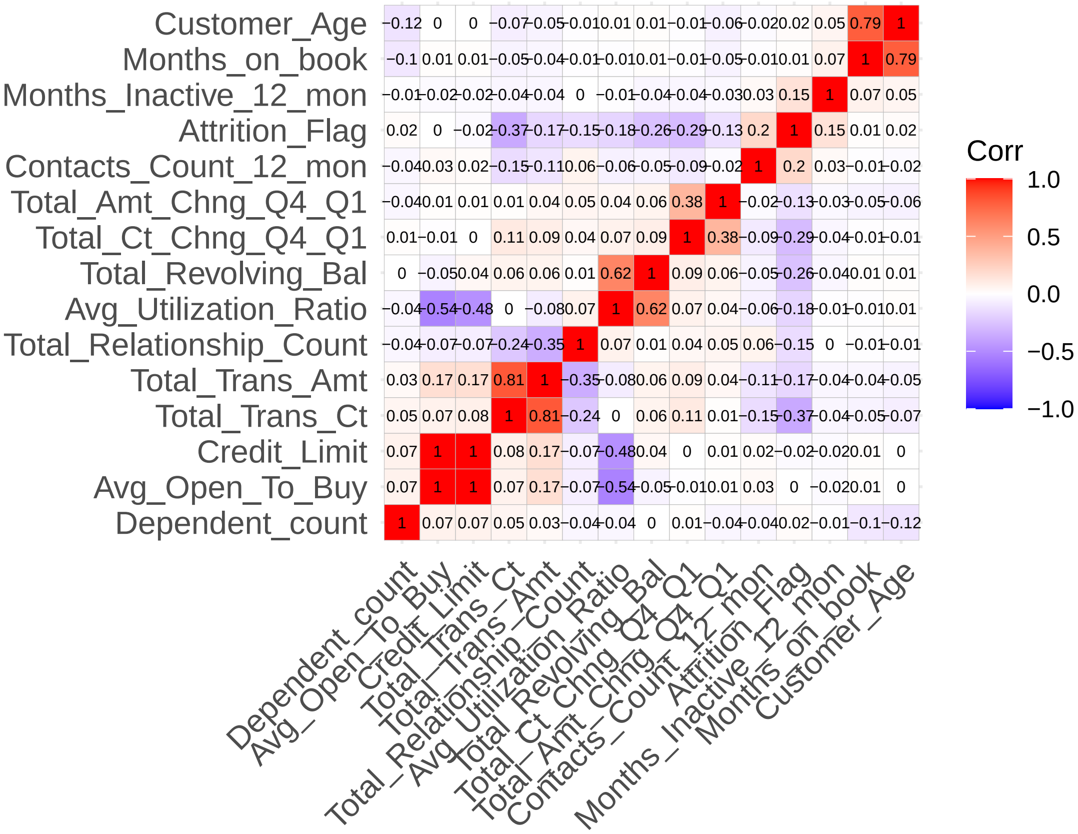
```
##     Blue     Gold Platinum   Silver
##     9436      116       20      555
```

```
# barplot of marital status by attrition flag

# Creating a vector to calculate percentages
#count <- table(data[data$Marital_Status == 'Married',]$Attrition_Flag)[1]

#count <- c(count, table(data[data$Marital_Status == 'Married',]))

#count <- as.numeric(count)

# create a dataframe
#industry <- rep(levels(adult$workclass), each = 2)
#income <- rep(c('<=50K', '>50K'), 4)
#df <- data.frame(industry, income, count)
#df
```

```
data
```

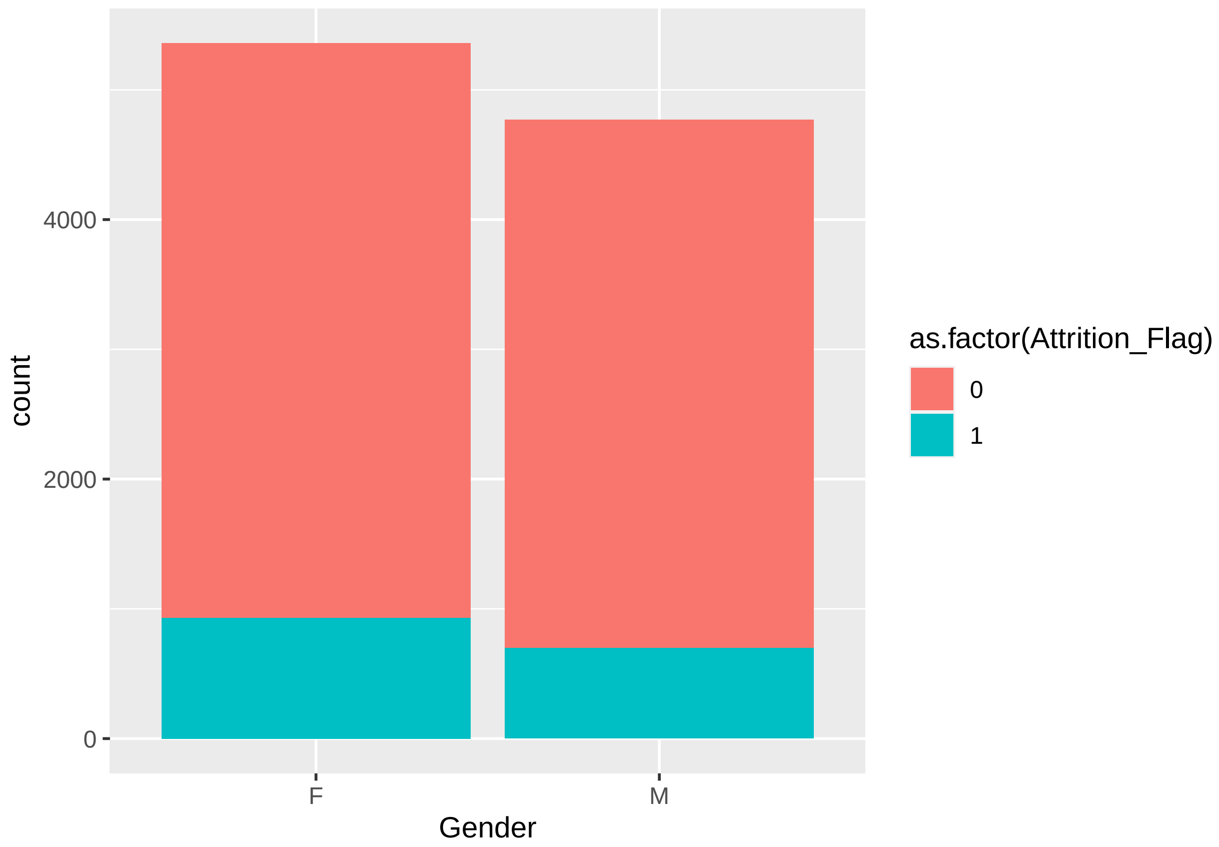
```
## # A tibble: 10,127 x 20
##    Attrition_Flag Customer_Age Gender Dependent_count Education_Level
##             <dbl>        <dbl> <chr>            <dbl> <chr>
##  1              0           45 M                    3 High School
##  2              0           49 F                    5 Graduate
##  3              0           51 M                    3 Graduate
##  4              0           40 F                    4 High School
##  5              0           40 M                    3 Uneducated
##  6              0          44 M                    2 Graduate
##  7              0          51 M                    4 Unknown
##  8              0          32 M                    0 High School
##  9              0          37 M                    3 Uneducated
## 10              0          48 M                    2 Graduate
## # ... with 10,117 more rows, and 15 more variables: Marital_Status <fct>,
## #   Income_Category <fct>, Card_Category <fct>, Months_on_book <dbl>,
## #   Total_Relationship_Count <dbl>, Months_Inactive_12_mon <dbl>,
## #   Contacts_Count_12_mon <dbl>, Credit_Limit <dbl>, Total_Revolving_Bal <dbl>,
## #   Avg_Open_To_Buy <dbl>, Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>,
## #   Total_Trans_Ct <dbl>, Total_Ct_Chng_Q4_Q1 <dbl>,
## #   Avg_Utilization_Ratio <dbl>
```

```
num_Vars <- c("Avg_Utilization_Ratio", "Total_Ct_Chng_Q4_Q1", "Total_Trans_Ct", "Total_Trans_Amt", "Tota
              "Avg_Open_To_Buy",
              "Total_Revolving_Bal",
              "Credit_Limit",
              "Contacts_Count_12_mon",
              "Months_Inactive_12_mon",
              "Total_Relationship_Count",
              "Months_on_book",
              "Dependent_count",
              "Customer_Age",
              "Attrition_Flag"
              )
df = data[num_Vars]
corr = cor(df, method = "pearson", use = "complete.obs")

ggcorrplot(corr, hc.order = TRUE, show.legend = TRUE, lab_size = 2, digits = 2,
   lab = TRUE )
```



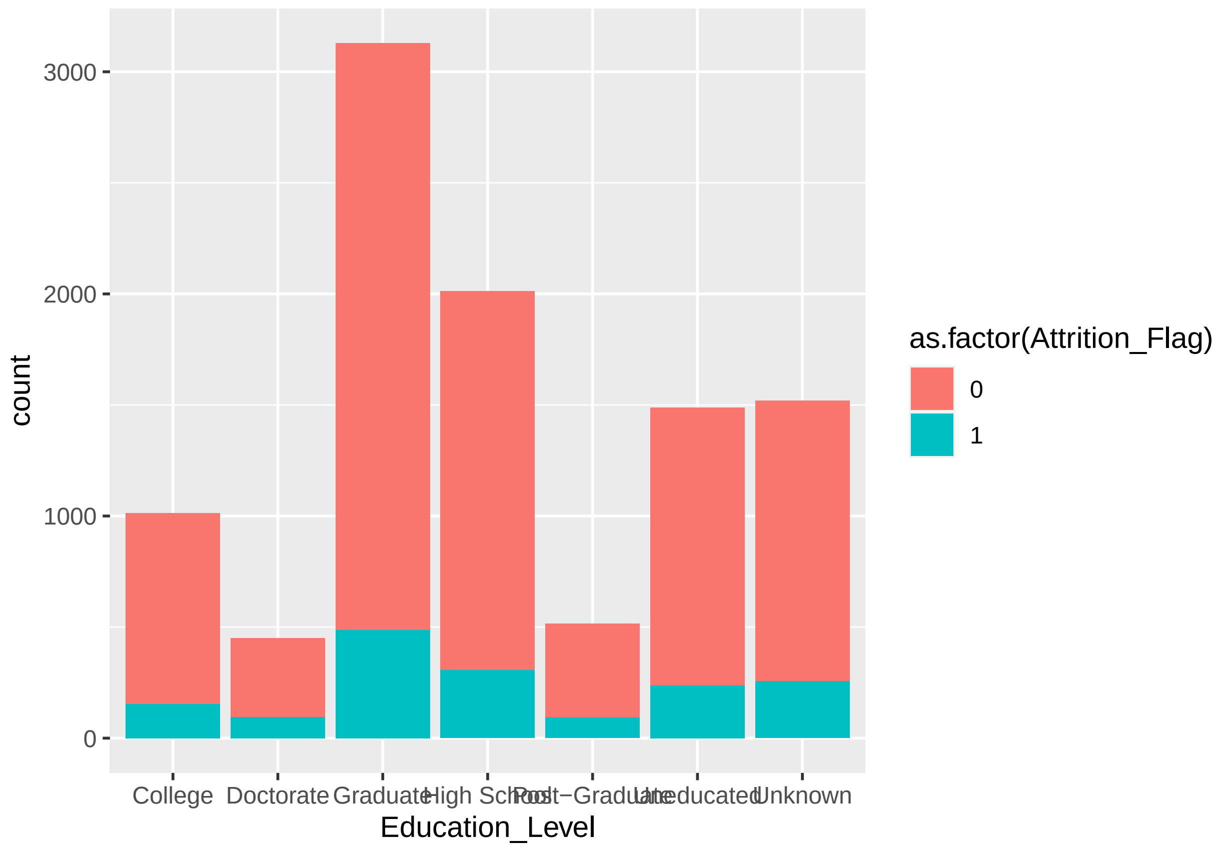
Gender Education Level Marital_Status Income_Category

```
gender_churn <- data %>% group_by(Attrition_Flag, Gender)%>%
  summarise(count = n())

ggplot(gender_churn, aes(x = Gender, y = count, fill = as.factor(Attrition_Flag))) + geom_bar(stat = "i
```

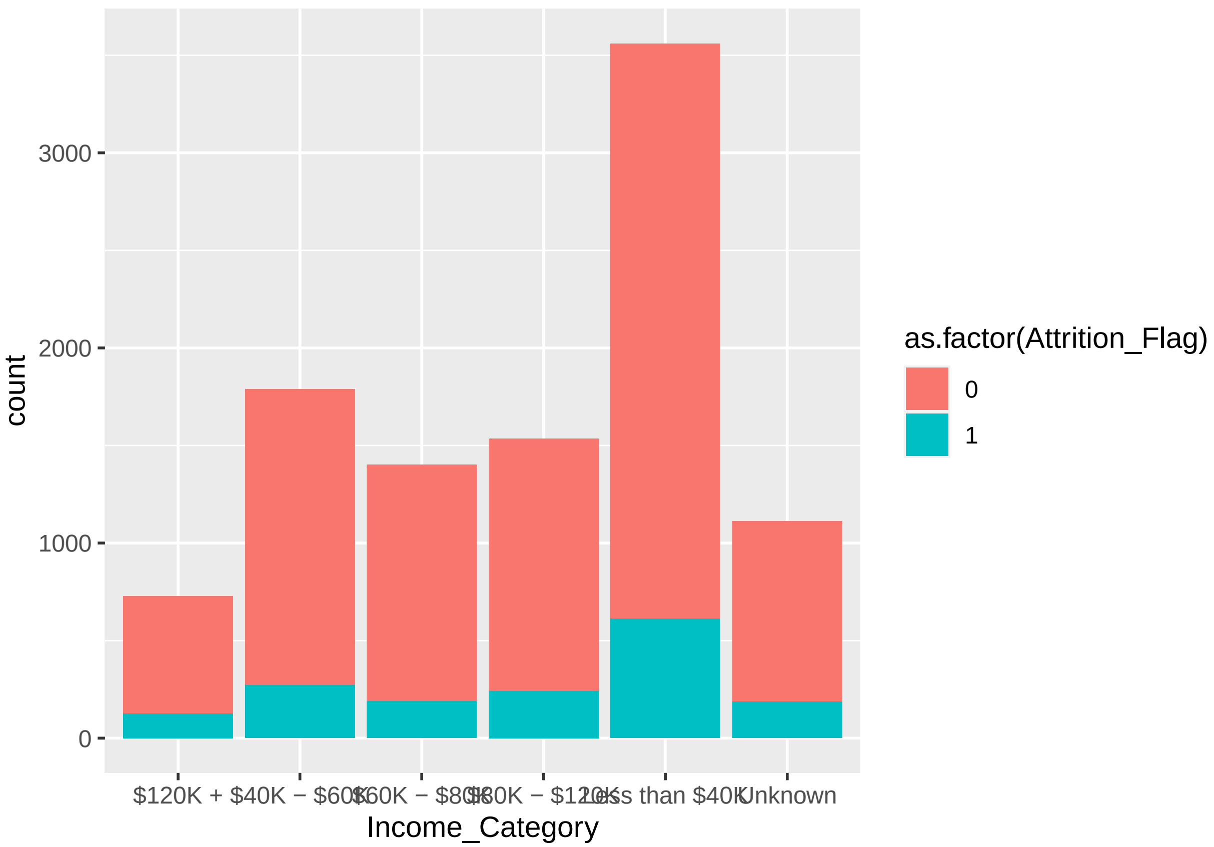


```
#ggtitle('Income Level with Workclass') + theme(axis.text.x = element_text(angle = -90)) + coord_flip()
```

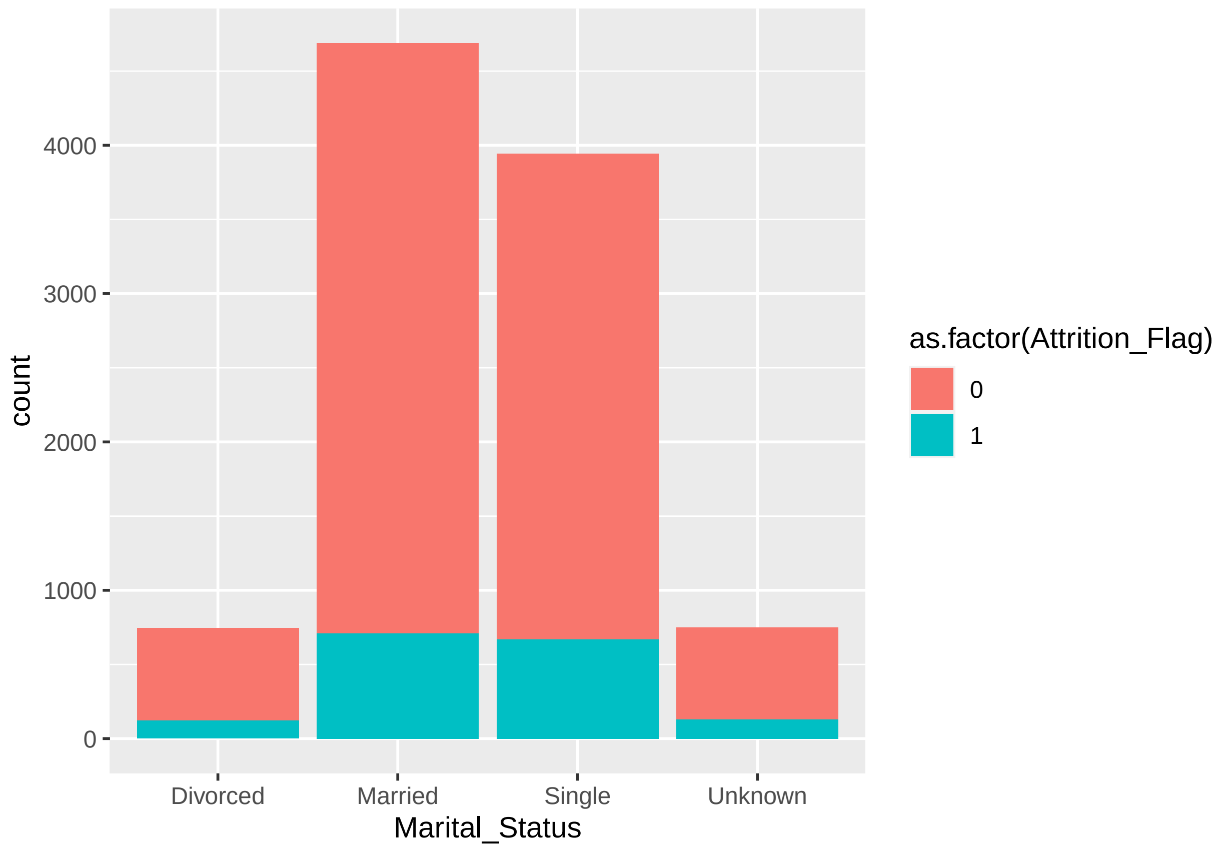
```
edu_churn <- data %>% group_by(Attrition_Flag, Education_Level)%>%
  summarise(count = n())

ggplot(edu_churn, aes(x = Education_Level, y = count, fill = as.factor(Attrition_Flag))) + geom_bar(sta
```



```
income_churn <- data %>% group_by(Attrition_Flag, Income_Category)%>%
  summarise(count = n())

ggplot(income_churn, aes(x = Income_Category, y = count, fill = as.factor(Attrition_Flag))) + geom_bar(
```



```
marital_churn <- data %>% group_by(Attrition_Flag, Marital_Status)%>%
  summarise(count = n())

ggplot(marital_churn, aes(x = Marital_Status, y = count, fill = as.factor(Attrition_Flag))) + geom_bar(
```



```
sum(is.na(data))
```

```
## [1] 0
```

```
data %>% summarise_all(n_distinct)
```

```
## # A tibble: 1 x 20
##   Attrition_Flag Customer_Age Gender Dependent_count Education_Level
##            <int>        <int>  <int>           <int>           <int>
## 1              2           45      2               6               7
## # ... with 15 more variables: Marital_Status <int>, Income_Category <int>,
## #   Card_Category <int>, Months_on_book <int>, Total_Relationship_Count <int>,
## #   Months_Inactive_12_mon <int>, Contacts_Count_12_mon <int>,
## #   Credit_Limit <int>, Total_Revolving_Bal <int>, Avg_Open_To_Buy <int>,
## #   Total_Amt_Chng_Q4_Q1 <int>, Total_Trans_Amt <int>, Total_Trans_Ct <int>,
## #   Total_Ct_Chng_Q4_Q1 <int>, Avg_Utilization_Ratio <int>
```

############– Model Building –############## ##################################

############– Random Forest –##############

```
#Splitting the train data to train_train and train_test
sample_size <- floor(0.7*nrow(data))
set.seed(154)
```

```
# randomly split train data
random_picked = sample(seq_len(nrow(data)),size = sample_size)
train =data[random_picked,]
test =data[-random_picked,]
```

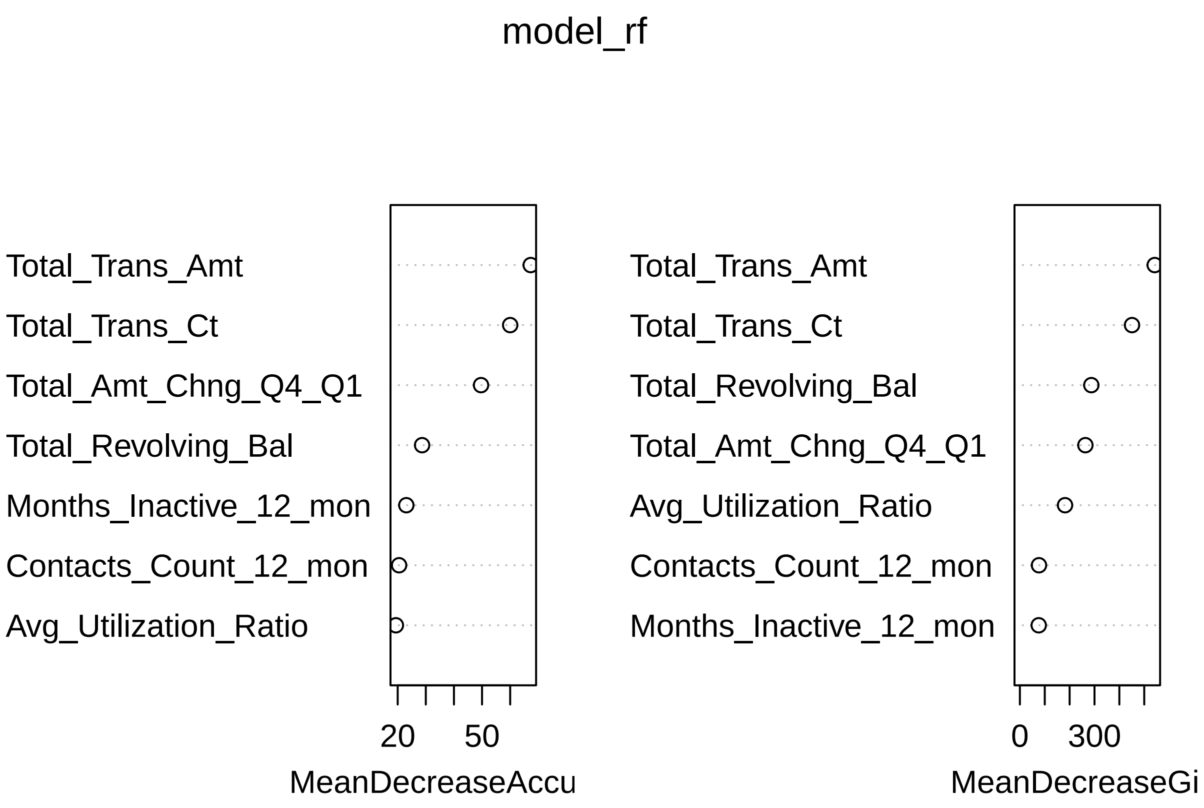
### Random Forest

Avg_Utilization_Ratio Total_Ct_Chng_Q4_Q1 Total_Trans_Ct Total_Trans_Amt Total_Amt_Chng_Q4_Q1 Avg_Open_To_Buy Total_Revolving_Bal Credit_Limit Contacts_Count_12_mon Months_Inactive_12_mon Total_Relationship_Count Months_on_book Dependent_count Customer_Age Attrition_Flag

```
# Building random forest model
model_rf <- randomForest(as.factor(Attrition_Flag) ~ Total_Trans_Ct +
                           Total_Amt_Chng_Q4_Q1 + Total_Revolving_Bal +
                           Avg_Utilization_Ratio + Total_Trans_Amt +
                           Months_Inactive_12_mon + Contacts_Count_12_mon,
                         data = train, ntree = 200, type = "class",
                         importance = TRUE)





# Using random forest model on a test data
model_rf_pred <- predict(model_rf, test, type = 'class')
model_rf
```

```
##
## Call:
##  randomForest(formula = as.factor(Attrition_Flag) ~ Total_Trans_Ct +      Total_Amt_Chng_Q4_Q1 + Tota
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 4.66%
## Confusion matrix:
##      0   1 class.error
## 0 5809 111     0.01875
## 1  219 949     0.18750
```

```
varImpPlot(model_rf)
```



```
# Confusion matrix for random forest
table(model_rf_pred, test$Attrition_Flag)
```

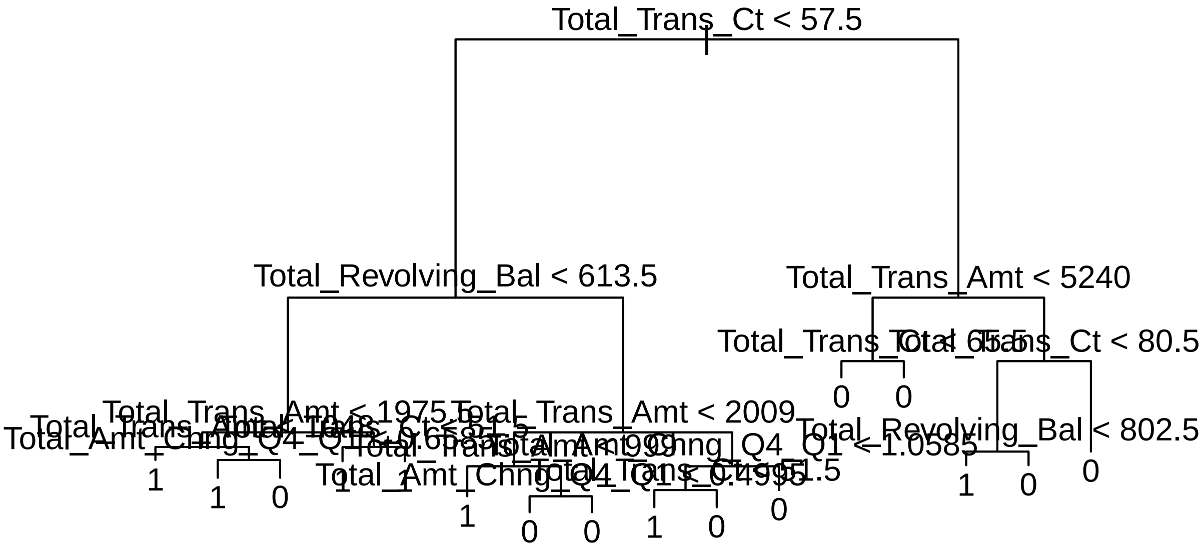
```
##
## model_rf_pred    0    1
##             0 2524   83
##             1   56  376
```

```
# Model accuracy of random forest
model_rf_accuracy <- mean(model_rf_pred == test$Attrition_Flag)
model_rf_accuracy
```

```
## [1] 0.9542613
```

###########– Decision Tree –#############

```
class_tree <- tree(as.factor(Attrition_Flag) ~ Total_Trans_Ct +
                           Total_Amt_Chng_Q4_Q1 + Total_Revolving_Bal +
                           Avg_Utilization_Ratio + Total_Trans_Amt +
                           Months_Inactive_12_mon + Contacts_Count_12_mon, train)
plot(class_tree); text(class_tree, textfont = 1)
```



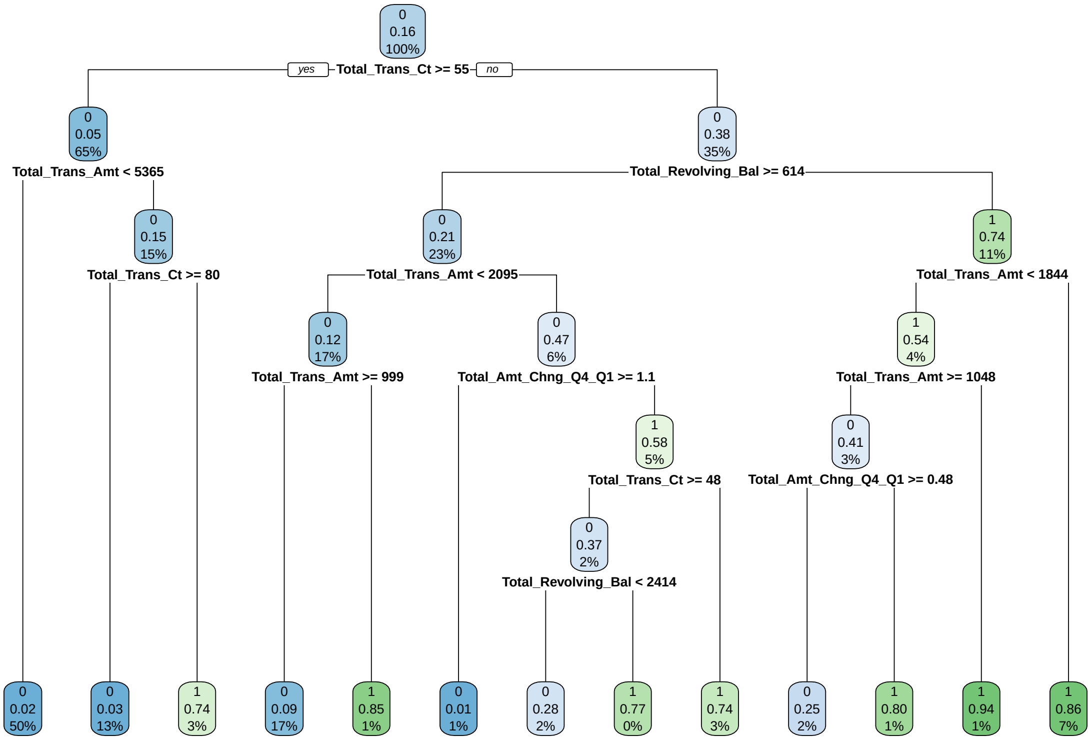
```
t1 <- table(test$Attrition_Flag, predict(class_tree, test, type = 'class'))
accuracy <- sum(diag(t1)) / sum(t1)
t1
```

```
##
##         0    1
##   0 2419  161
##   1   99  360
```

```
accuracy
```

```
## [1] 0.9144455
```

```
fit <- rpart(as.factor(Attrition_Flag) ~ Total_Trans_Ct +
                           Total_Amt_Chng_Q4_Q1 + Total_Revolving_Bal +
                           Avg_Utilization_Ratio + Total_Trans_Amt +
                           Months_Inactive_12_mon + Contacts_Count_12_mon, data = train, method = "clas
rpart.plot(fit)
```



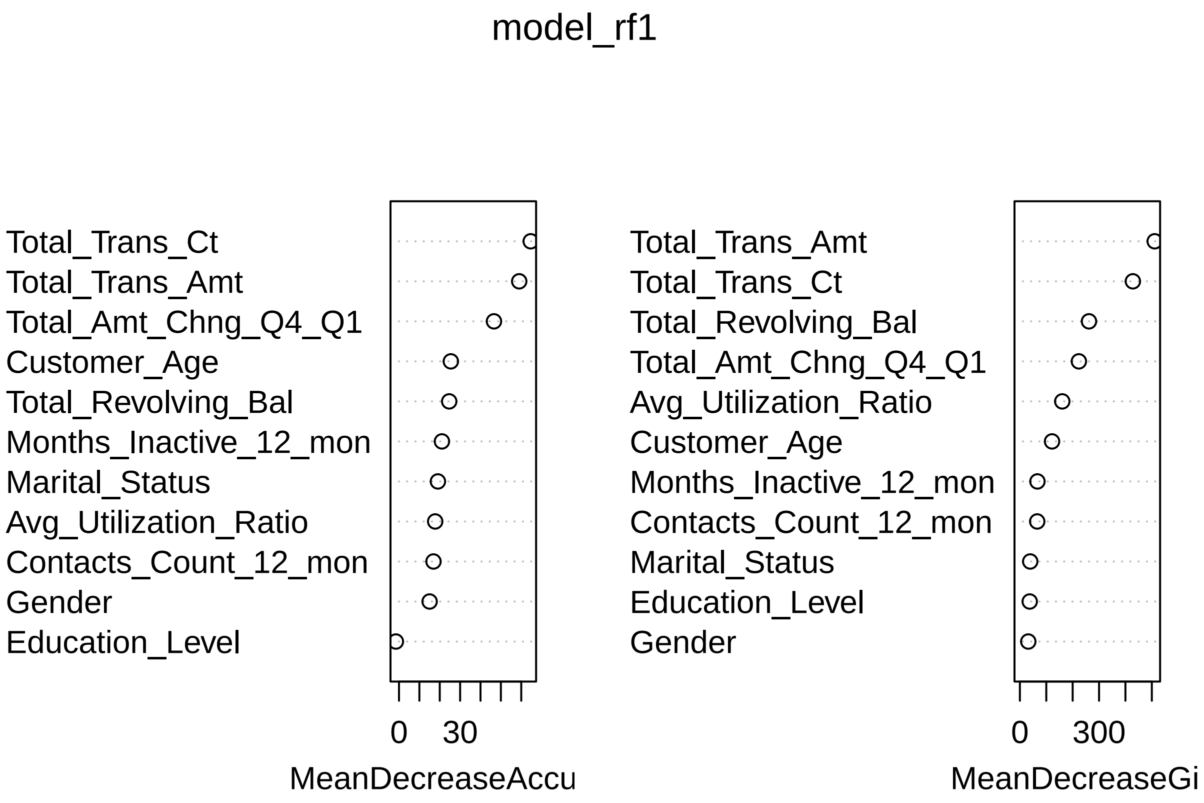
Alternative Random Forest

```
# Building random forest model
model_rf1 <- randomForest(as.factor(Attrition_Flag) ~ Total_Trans_Ct +
                           Total_Amt_Chng_Q4_Q1 + Total_Revolving_Bal +
                           Avg_Utilization_Ratio + Total_Trans_Amt +
                           Months_Inactive_12_mon + Contacts_Count_12_mon +
                            Gender + Education_Level + Marital_Status + Customer_Age,
                         data = train, ntree = 200, type = "class",
                         importance = TRUE)







# Using random forest model on a test data
model_rf_pred1 <- predict(model_rf1, test, type = 'class')
model_rf1
```

```
##
## Call:
##  randomForest(formula = as.factor(Attrition_Flag) ~ Total_Trans_Ct +      Total_Amt_Chng_Q4_Q1 + Tota
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 3
##
```

```
##         OOB estimate of  error rate: 4.36%
## Confusion matrix:
##      0   1 class.error
## 0 5829   91  0.01537162
## 1  218 950  0.18664384
```

```
varImpPlot(model_rf1)
```



```
# Confusion matrix for random forest
table(model_rf_pred1, test$Attrition_Flag)
```

```
##
## model_rf_pred1    0    1
##              0 2544   83
##              1   36  376
```

```
# Model accuracy of random forest
model_rf_accuracy1 <- mean(model_rf_pred1 == test$Attrition_Flag)
model_rf_accuracy1
```

```
## [1] 0.9608424
```