Data Mining- IDS 572
Assignment 4
Case Assignment- Predicting Customer Churn

Jigyasa Sachdeva (UIN- 664791188)
Varun Maheshwari (UIN- 671624467)

Reading the data in R:
library(readxl)
#To read excel data
Data <- read_excel("Desktop/Second Sem/Data Mining/Assignment 4/UV6696-XLS-ENG.xlsx", sheet = "Case Data")
View(Data)

Preparing the data for analysis:
library(funModeling)
#To use df_status and analyze the structure of the dataset
df_status(Data)
Output:

```
> df_status(Data)
                      variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
1                           ID       0    0.00    0    0     0     0 numeric   6347
2      Customer Age (in months)       1    0.02    0    0     0     0 numeric     61
3        Churn (1 = Yes, 0 = No)    6024   94.91    0    0     0     0 numeric      2
4             CHI Score Month 0    1193   18.80    0    0     0     0 numeric    263
5                 CHI Score 0-1    1408   22.18    0    0     0     0 numeric    242
6         Support Cases Month 0    4556   71.78    0    0     0     0 numeric     21
7             Support Cases 0-1    4062   64.00    0    0     0     0 numeric     37
8                    SP Month 0    4588   72.29    0    0     0     0 numeric     27
9                        SP 0-1    4561   71.86    0    0     0     0 numeric     81
10                   Logins 0-1    1289   20.31    0    0     0     0 numeric    294
11            Blog Articles 0-1    3632   57.22    0    0     0     0 numeric     57
12                    Views 0-1    1925   30.33    0    0     0     0 numeric   1360
13     Days Since Last Login 0-1    2665   41.99    0    0     0     0 numeric    143
```

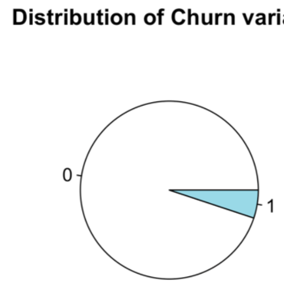
#No null values as q_na and p_na for all variables is zero
#ID is a numeric variable with 6347 unique data points and hence should be removed during analysis
#`Churn (1 = Yes, 0 = No)` is our target variable and has 2 unique levels and hence should be a factor. The target variable also constitutes of 94.91% of ‘0’s’; indicating unbalanced target variable.

#Converting data type of Churn to factor:
Data$`Churn (1 = Yes, 0 = No)` <- as.factor(Data$`Churn (1 = Yes, 0 = No)`)

1. **<u>Dependence of customer churn on customer age:</u>**

<u>Univariate Analysis:</u>

- Churn:
  pie(table(Data$`Churn (1 = Yes, 0 = No)`), main="Distribution of Churn variable")
  Output:



  As already declared: Our target variable is unevenly proportioned which might lead to unusual results during analysis.
- Age:
  library(psych) #To use describe function:
  describe(Data$`Customer Age (in months)`)
  Output:

```
> describe(Data$`Customer Age (in months)`)
Data$`Customer Age (in months)`
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    6347        0       61    0.998     13.9    12.05        1        2        5       11       20
     .90      .95
      30       36

lowest :  0  1  2  3  4, highest: 56 57 58 63 67
```

  The mean of customer age is 13.9 months and the median (0.5) is 11 months, indicating the right skewness of the variable. The minimum age is 0 months and maximum is 67 months.

According to Wall's hypothesis:
"Regarding age, I'd say that if a customer has been with us for more than 14 months, he or she knows how to use our services, is using them, gets value out it, and should be less likely to leave. Those who have been with us for less than 6 months are still perhaps only learning about the services, so I am not sure how it will go, but those between 6 and 14 months are probably the riskiest group."

Hence, <u>Converting Age into categories of less than 6 months, between 6 and 14 months, more than 14 months: New variable name- 'agecat'</u>
Conversion:
Data$agecat <- rep(0, nrow(Data))
Data$agecat[Data$`Customer Age (in months)`<=6] <- 1
Data$agecat[Data$`Customer Age (in months)`> 6 & Data$`Customer Age (in months)` <= 14] <- 2
Data$agecat[Data$`Customer Age (in months)`> 14] <- 3

Univariate of the new variable:
table(Data$agecat)

```
> table(Data$agecat)

   1    2    3 
2051 1902 2394 
```

Bivariate of the new variable with target variable:
t1 <- xtabs(~Data$agecat+Data$`Churn (1 = Yes, 0 = No)`)
prop.table(t1)*100

```
> prop.table(t1)*100
            Data$`Churn (1 = Yes, 0 = No)`
Data$agecat          0          1
          1 31.6212384  0.6932409
          2 27.6508587  2.3160548
          3 35.6388845  2.0797227
```

#Hypothesis and observation comparison:
#Hypothesis: Customers with an age of more than 14 months are least likely to leave
#Observation: 41.4% (2.07/5) population churning out belongs to this category (2nd highest)
#Hypothesis: Customers with an age between 6 to 14 months are most likely to leave
#Observation: 46.2% (2.31/5) population churning out belongs to this category (Highest)
#Hypothesis: Customers with less than 6 month age are unsure:
#Observation: 14% (0.7/5) population churning out belongs to this category (Lowest)

The over-all hypothesis was: <6 months: Unsure, 6-14: Highest, >14: Least
Observation: <6 months: Least likely to leave, 6-14: Highest, >14: Almost as likely as 6-14
Percentage of likeliness to leave: <6: 14%, 6-14: 46.2%, >14: 41.4%

2. **Run the best logistic regression model to predict customer churn**

Statistical testing before model building (at 95% Confidence Interval):
options(scipen=99) #For turning off scientific notation

Significant:

- chisq.test(Data$agecat, Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.000000000000007198
- t.test(Data$`CHI Score Month 0`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.0000000000002097
- t.test(Data$`CHI Score 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.00000001571
- t.test(Data$`Support Cases Month 0`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.00000006281
- t.test(Data$`SP Month 0`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.0000004381

- t.test(Data$`Logins 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.0004037
- t.test(Data$`Days Since Last Login 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.00005215
- t.test(Data$`Blog Articles 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.01158

Insignificant:

- t.test(Data$`Support Cases 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.5278
- t.test(Data$`SP 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.5218
- t.test(Data$`Views 0-1`~Data$`Churn (1 = Yes, 0 = No)`)
  #p-value = 0.05631

Selecting significant variables for model:
ModelData <- subset(Data, select= -c(`Customer Age (in months)`),`Support Cases 0-1`, `SP 0-1`, `Views 0-1`, ID))

<u>Logistic Regression model:</u>

<u>Train and Test set:</u>
Test <- ModelData[c(672, 354, 5203),]   #prediction to be done on these three
rown <- c(672, 354, 5203)
rownames(Test) <- c(672, 354, 5203)
#Assigning row names to the Test data
Train <- ModelData[-rown,]
mod <- glm(`Churn (1 = Yes, 0 = No)`~., data = Train, family = 'binomial')
summary(mod)

```
> summary(mod)

Call:
glm(formula = `Churn (1 = Yes, 0 = No)` ~ ., family = "binomial",
    data = Train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7745  -0.3389  -0.2966  -0.2260   2.9886

Coefficients:
                             Estimate Std. Error z value             Pr(>|z|)
(Intercept)                 -3.443575   0.173118 -19.891 < 0.0000000000000002 ***
`CHI Score Month 0`         -0.007633   0.001233  -6.190       0.000000000603 ***
`CHI Score 0-1`             -0.006838   0.002470  -2.768              0.00564 **
`Support Cases Month 0`      0.002382   0.068972   0.035              0.97245
`SP Month 0`                -0.031219   0.072957  -0.428              0.66872
`Logins 0-1`                 0.001034   0.001941   0.533              0.59436
`Blog Articles 0-1`         -0.003488   0.022666  -0.154              0.87770
`Days Since Last Login 0-1`  0.011918   0.003818   3.122              0.00180 **
agecat                       0.491719   0.082361   5.970       0.000000002368 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2552.8  on 6343  degrees of freedom
Residual deviance: 2423.5  on 6335  degrees of freedom
AIC: 2441.5

Number of Fisher Scoring iterations: 6
```

Explanation: The residual errors range from -0.7 to 2.98.
4 out of 8 variables are highly significant. Order of significance:
CHI Score Month 0 > Agecat > Days Since Last Login 0-1 > CHI Score 0-1
Residual deviance is lesser than null deviance (2423.5 < 2552.8), indicating our model is slightly better than the common classifier.
AIC value is 2441.5 which is a high value.

As observed in the first question, the target variable is very unevenly distributed; hence we need to observe the confusion matrix before moving forward with the predictions on Test Data.

Predicting using our Model on Train Data:
p <- predict(mod, data = Train, type = "response")
p1 <- ifelse(p>=0.5, "1", "0")
#Assigning class as ‘1’ if probability is greater than 0.5 and vice-versa
p1<- as.factor(p1)

Confusion Matrix:
library(caret)   #For confusionMatrix function
confusionMatrix(p1,Train$`Churn (1 = Yes, 0 = No)`, positive= '1')

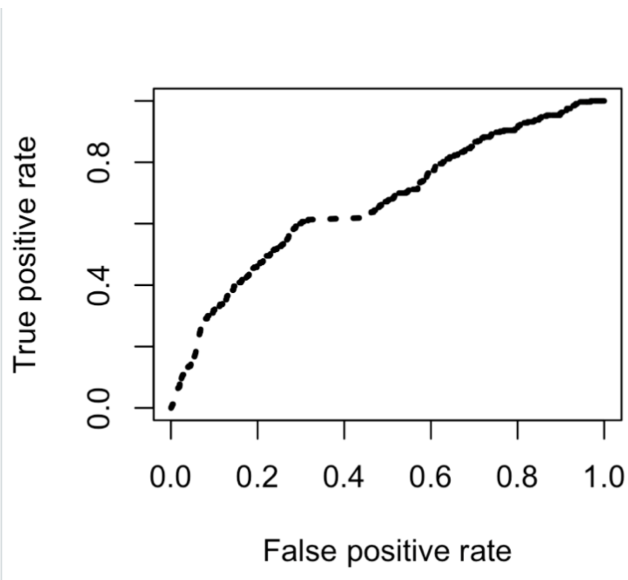
```
> confusionMatrix(p1,Train$`Churn (1 = Yes, 0 = No)`, positive= '1')
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6021  323
         1    0    0

               Accuracy : 0.9491
                 95% CI : (0.9434, 0.9544)
    No Information Rate : 0.9491
    P-Value [Acc > NIR] : 0.5148

                  Kappa : 0

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.00000
            Specificity : 1.00000
         Pos Pred Value :     NaN
         Neg Pred Value : 0.94909
             Prevalence : 0.05091
         Detection Rate : 0.00000
   Detection Prevalence : 0.00000
      Balanced Accuracy : 0.50000

       'Positive' Class : 1
```

The confusion matrix formed indicates no prediction of Customer Churn = 1. Even though the accuracy of the model attained is 94.91%, it is not helpful in determining which customer would churn as it classifies all customers as they would not churn.
With positive class as 1, Sensitivity is 0: Indicating True Positive = 0. This model is almost same as a common classifier and hence, some variation is required. We need to increase sensitivity and hence, the threshold for prediction should be changed.

Finding optimal cut-off point:

library(ROCR)   # Computing a simple ROC curve (x-axis: fpr, y-axis: tpr)

Calculating the values for ROC curve:
pred = prediction(p, Train$`Churn (1 = Yes, 0 = No)`)
perf = performance(pred,"tpr","fpr")
Plotting the ROC curve:
plot(perf, col = 'black', lty = 3, lwd = 3)




```
opt.cut = function(perf, pred){
  cut.ind = mapply(FUN=function(x, y, p){
    d = (x - 0)^2 + (y-1)^2
    ind = which(d == min(d))
    c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
      cutoff = p[[ind]])
  }, perf@x.values, perf@y.values, pred@cutoffs)}
print(opt.cut(perf, pred))
```

cutoff <- 0.05354738

Optimal threshold comes out to be 0.0535 where the trade-off between true positive rate and false positive is optimum.

<u>Predicting again with new cutoff point:</u>
p <- predict(mod, data= Train, type= "response")
predicted_class <- ifelse(p>=0.05354738, "1", "0")
predicted_class <- as.factor(predicted_class)

<u>New Confusion Matrix:</u>
confusionMatrix(predicted_class, Train$`Churn (1 = Yes, 0 = No)`, positive = '1')

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4335  136
         1 1686  187

               Accuracy : 0.7128
                 95% CI : (0.7015, 0.7239)
    No Information Rate : 0.9491
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0914

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.57895
            Specificity : 0.71998
         Pos Pred Value : 0.09984
         Neg Pred Value : 0.96958
             Prevalence : 0.05091
         Detection Rate : 0.02948
   Detection Prevalence : 0.29524
      Balanced Accuracy : 0.64946

       'Positive' Class : 1
```

Even though, our model's accuracy has reduced to 71.28%, the sensitivity has increased to 57.895%. This indicates that we are able to identify 57.895% of customers that would churn correctly, as opposed to 0% in the previous case.

Hence, the model 'mod' with prediction cut-off point as 0.05354738 is the best model to predict customer churn.

(a) 672nd observation:

a <- Test[(rownames(Test)==672),]
a1 <- predict(mod, newdata = a, type = "response")
a1
<u>Probability of the target variable of 672nd observation is 0.04448889</u>
a2 <- ifelse(a1>=0.05354738, "1", "0")
a2 #0
<u>Hence our model predicts the 672nd observation belonging to class 'No'</u>
a$`Churn (1 = Yes, 0 = No)` #0
<u>Churn of 672nd observation is 'No'</u>

(b) 354th observation:

b1 <- Test[(rownames(Test)==354),]
b2 <- predict(mod, newdata = b1, type = "response")
b2
<u>Probability of the target variable of 354th observation is 0.03414641</u>
b3 <- ifelse(b2>=0.05354738, "1", "0")
b3 #0
<u>Hence our model predicts the 354th observation belonging to class 'No'</u>
b1$`Churn (1 = Yes, 0 = No)` #0
<u>Churn of 354th observation is 'No'</u>

5203rd observation:

c1 <- Test[(rownames(Test)==5203),]
c2 <- predict(mod, newdata = c1, type = "response")
c2
<u>Probability of the target variable of 5203rd observation is 0.03257432</u>
c3 <- ifelse(c2>=0.05354738, "1", "0")
c3 #0
<u>Hence our model predicts the 5203rd observation belonging to class 'No'</u>
c1$`Churn (1 = Yes, 0 = No)` #0
<u>Churn of 354th observation is 'No'</u>

<u>Answer</u>: All three observations were classified correctly by the model as the probabilities of prediction was less than 0.0535.

3. <u>List of 100 customers with highest churn probabilities and top 3 drivers for each</u>

Finding customers in the entire dataset, hence combining both Train and Test in ModelData.
Predicting the churn probability for all the customers using the model generated above:
predictnew <- predict(mod, newdata = ModelData, type="response")
Combining the probabilities column in the dataset:
ModelData <- cbind(ModelData, predictnew)

Sorting the dataset ModelData with predicted probability in descending order:
sort <- ModelData[with(ModelData, order(-predictnew)),]
Snapshot of the dataset with sorted probabilities in descending order:

Filter

	Churn (1 = Yes, 0 = No)	CHI Score Month 0	CHI Score 0-1	Support Cases Month 0	SP Month 0	Logins 0-1	Blog Articles 0-1	Days Since Last Login 0-1	agecat	predictnew
109	0	0	-125	0	0	-8	0	6	3	0.2591387
1971	0	0	-113	0	0	-23	0	7	3	0.2430385
1672	1	2	1	0	0	0	0	61	3	0.2203650
2076	0	29	-69	0	0	0	0	31	3	0.2061139
1236	0	0	-35	0	0	0	0	31	3	0.2042982
2546	0	8	-86	0	0	-9	-1	6	3	0.2016948
1616	0	2	-42	0	0	-1	0	27	3	0.2016925
1287	0	24	-72	0	0	-6	-1	24	3	0.2016567
1929	0	7	-40	0	0	0	0	31	3	0.2011881
1862	0	0	-27	0	0	0	0	31	3	0.1955493
1574	0	3	-87	0	0	-10	-1	-1	3	0.1954166
1363	1	0	-34	0	0	-9	0	26	3	0.1922627
1693	0	0	-23	0	0	0	0	31	3	0.1912825
2838	0	0	-28	0	0	-1	0	28	3	0.1908810
1286	0	20	-77	0	0	-2	-2	12	3	0.1905191
2599	0	7	-30	0	0	0	0	31	3	0.1904230
1459	0	0	-22	0	0	0	0	31	3	0.1902269
2922	1	13	-52	0	0	-1	0	22	3	0.1898598
2080	0	4	-25	0	0	0	0	31	3	0.1886885
2680	0	2	-72	0	0	-10	-2	3	3	0.1886254
2244	0	16	-38	0	0	0	0	31	3	0.1882751
76	0	1	-70	0	0	-7	-1	3	3	0.1876433

Showing 1 to 22 of 6,347 entries, 10 total columns

<u>Subset of top 100 customers with highest churn probabilities</u>:
sorted <- sort[1:100,]

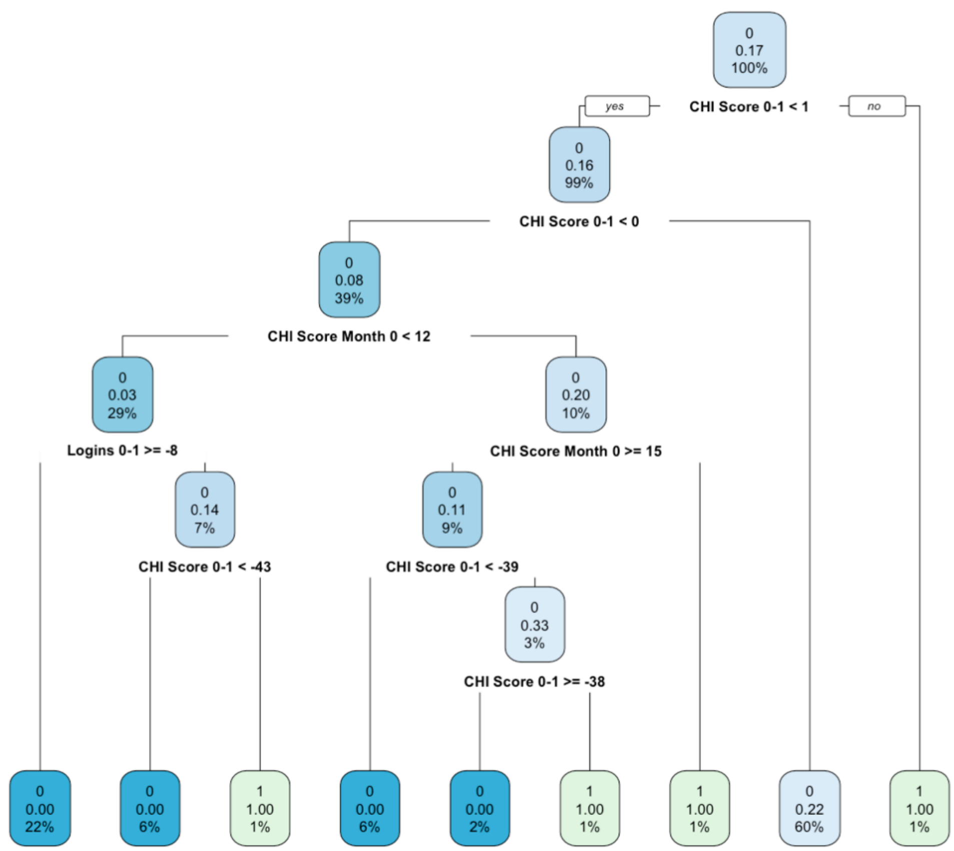
<u>Deriving top 3 drivers in the new dataset:</u>
#Removing the predicted values from the data frame to find key predictors
sortednew <- subset(sorted, select= -predictnew)

Using decision tree to predict the top 3 drivers. Using decision tree over random forest as taking all the data of top 100 customers and not train and test division.

library(rpart)
t <- rpart(`Churn (1 = Yes, 0 = No)`~ ., data= sortednew, minsplit= 0)
t

```
> t
n= 100

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 100 17 0 (0.83000000 0.17000000)
   2) CHI Score 0-1< 0.5 99 16 0 (0.83838384 0.16161616)
     4) CHI Score 0-1< -0.5 39  3 0 (0.92307692 0.07692308)
       8) CHI Score Month 0< 11.5 29  1 0 (0.96551724 0.03448276)
        16) Logins 0-1>=-8.5 22  0 0 (1.00000000 0.00000000) *
        17) Logins 0-1< -8.5 7  1 0 (0.85714286 0.14285714)
          34) CHI Score 0-1< -43 6  0 0 (1.00000000 0.00000000) *
          35) CHI Score 0-1>=-43 1  0 1 (0.00000000 1.00000000) *
       9) CHI Score Month 0>=11.5 10  2 0 (0.80000000 0.20000000)
        18) CHI Score Month 0>=14.5 9  1 0 (0.88888889 0.11111111)
          36) CHI Score 0-1< -39.5 6  0 0 (1.00000000 0.00000000) *
          37) CHI Score 0-1>=-39.5 3  1 0 (0.66666667 0.33333333)
            74) CHI Score 0-1>=-38.5 2  0 0 (1.00000000 0.00000000) *
            75) CHI Score 0-1< -38.5 1  0 1 (0.00000000 1.00000000) *
        19) CHI Score Month 0< 14.5 1  0 1 (0.00000000 1.00000000) *
     5) CHI Score 0-1>=-0.5 60 13 0 (0.78333333 0.21666667) *
   3) CHI Score 0-1>=0.5 1  0 1 (0.00000000 1.00000000) *
```

library(rpart.plot)
rpart.plot(t)



Hence the top 3 predictors are:

1. CHI Score 0-1
2. CHI Score Month 0
3. Logins 0-1