Report Summary –

Outcome of the Machine Learning Techniques on the Customer Churn details of Telecom Company data set

Date: 18th February 2018

Submitted by:

Name	SMS ID	SID
Parthasarathi Chatterjee	118284	TA17021

A. Introduction

A data set (Telecom Customer Data.csv – Refer Annexure) of 7043 customers of a telecom company and their churn details (yes-have left and No-still using the service) is provided. The dataset comprises of demographic data of the customer, services availed, type of contract, payment method, amount spent, etc.

The requirement is to develop machine learning models to predict the potential churn of the customers and accordingly devise retention strategies to retain the customers.

B. Approach

Given the data set, two machine learning supervised classification techniques (KNN & ANN) have been developed so that the telecom company can use the models as a tool to predict the potential churn of their customers and implement proper retention strategies. (Refer Annexure for the R Code)

C. Method: K Nearest Neighbour (KNN)

Step 1: Importing Data Set and Data Preparation by imputing the values in the blank cell of the data set.

Reason: The package mice has been used to impute the values in the blank field of the attribute "TotalCharges". MICE(Multivariate imputation by Chained Equations) operates under the assumption that given the variables used in the imputation procedure, the missing data are Missing At Random (MAR), which means that the probability that a value is missing depends only on observed values and not on unobserved values. MICE package imputes data conveniently for categorical, binary and continuous variables and the method adopted here is "pmm" i.e. predictive mean matching for the variables

```
library(mice)

library(caret)

library(ROCR)

library(caTools)

library(class)

library(h2o)
```

```
## Importing the Dataset

setwd("D:R")
my_datafile = read.csv('Telecom Customer Data.csv', header = TRUE)

## Imputation of the Blank field of the attribute "TotalCharges"

my_data = my_datafile[2:21]
tempData <- mice(my_data,m=5,maxit=10,meth='pmm',seed=500)
summary(tempData)
telecom_data<- complete(tempData,1)
summary(telecom_data)
write.csv(telecom_data,"Imputed Data_15.csv")
```

Outcome/ Inference:

The missing blank fields in the attribute "TotalCharges" are imputed by the above mechanism and are saved as the file Imputed_15.csv in the working directory (Refer Annexure). In addition, the attribute CustomerID is dropped from the working data set as it has no relevant significance to be used in the classification model

Step 2: Transformation of the Categorical variables by Binary Numbers (1, 0) and Dummy Variables

Reason: Since the KNN and ANN model need to be executed on numeric values, the categorical variables like Gender, Partner, PhoneService, Dependents, PaperlessBilling are transformed using binary numbers (1,0). The other categorical variables like MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StremingTV, StreamingMovies, Contract, PaymentMethod are transformed using dummy variables.

Reference R Code:

```
##Transformation of the Categorical variables by Binary numbers (1,0)

telecom_data$gender<-ifelse(telecom_data$gender=='Male',1,0)
telecom_data$Partner<-ifelse(telecom_data$Partner=='Yes',1,0)
telecom_data$PhoneService<-ifelse(telecom_data$PhoneService=='Yes',1,0)
telecom_data$Dependents<-ifelse(telecom_data$Dependents=='Yes',1,0)
```

```
telecom_data$PaperlessBilling<-ifelse(telecom_data$PaperlessBilling=='Yes',1,0)

##Transformation of the Categorical variables by Dummy Variables

telecom_data$MultipleLines1<-as.numeric(telecom_data$MultipleLines=="Yes")
telecom_data$MultipleLines2<-as.numeric(telecom_data$MultipleLines=="No")
telecom_data$InternetService1<-as.numeric(telecom_data$InternetService=="DSL")
telecom_data$InternetService2<-as.numeric(telecom_data$InternetService=="Fiber optic")
telecom_data$OnlineSecurity1<-as.numeric(telecom_data$OnlineSecurity=="Yes")
telecom_data$OnlineSecurity2<-as.numeric(telecom_data$OnlineSecurity=="No")
telecom_data$OnlineBackup1<-as.numeric(telecom_data$OnlineBackup=="Yes")
telecom_data$OnlineBackup2<-as.numeric(telecom_data$OnlineBackup=="No")
telecom_data$DeviceProtection1<-as.numeric(telecom_data$DeviceProtection=="Yes")
telecom_data$DeviceProtection2<-as.numeric(telecom_data$DeviceProtection=="No")
telecom_data$TechSupport1<-as.numeric(telecom_data$TechSupport=="Yes")
telecom_data$TechSupport2<-as.numeric(telecom_data$TechSupport=="No")
telecom_data$StreamingTV1<-as.numeric(telecom_data$StreamingTV=="Yes")
telecom_data$StreamingTV2<-as.numeric(telecom_data$StreamingTV=="No")
telecom_data$StreamingMovies1<-as.numeric(telecom_data$StreamingMovies=="Yes")
telecom_data$StreamingMovies2<-as.numeric(telecom_data$StreamingMovies=="No")
telecom_data$Contract1<-as.numeric(telecom_data$Contract=="Month-to-month")
telecom_data$Contract2<-as.numeric(telecom_data$Contract=="One year")
telecom_data$PaymentMethod1<-as.numeric(telecom_data$PaymentMethod=="Electronic check")
telecom_data$PaymentMethod2<-as.numeric(telecom_data$PaymentMethod=="Bank transfer (automatic)")
telecom_data$PaymentMethod3<-as.numeric(telecom_data$PaymentMethod=="Credit card (automatic)")
telecom_data$PaymentMethod2<-as.numeric(telecom_data$PaymentMethod=="Bank transfer (automatic)")
telecom_data$PaymentMethod3<-as.numeric(telecom_data$PaymentMethod=="Credit card (automatic)")
telecom_data$Churn<-ifelse(telecom_data$Churn=='Yes',1,0)

## Dropping the Original variables transformed by dummy variable

telecom_data$MultipleLines<-NULL
telecom_data$InternetService<-NULL
telecom_data$OnlineSecurity<-NULL
```

```
telecom_data$OnlineBackup<-NULL

telecom_data$DeviceProtection<-NULL

telecom_data$TechSupport<-NULL

telecom_data$StreamingTV<-NULL

telecom_data$StreamingMovies<-NULL

telecom_data$Contract<-NULL

telecom_data$PaymentMethod<-NULL

write.csv(telecom_data,"Transformed_Data_15.csv")
```

Outcome/ Inference

The variables as mentioned above are thus transformed. In the case of the categorical variables which are transformed by dummy variables, the original categorical categorical variable is dropped so as to avoid the multi collinearity amongst the original variable and the transformed dummy variable. The data set after transformation is saved as the file Transformed_Data_15.csv in the working directory (Refer Annexure)

Step 3: Standardized the numerical variably by Z-scale

Reason: Since the values and ranges across the continuous variables (tenure, MonthlyCharges, TotalCharges) differ substantially and thus some variables will outweigh others, the continuous variables are standardized using scaling technique under z-scale.

Reference R Code:

```
## Standardized the numerical variably by Z-scale


telecom_data$tenure<-as.numeric(scale(telecom_data$tenure))

telecom_data$tenure

telecom_data$MonthlyCharges<-as.numeric(scale(telecom_data$MonthlyCharges))

telecom_data$MonthlyCharges

telecom_data$TotalCharges<-as.numeric(scale(telecom_data$TotalCharges))

telecom_data$TotalCharges

write.csv(telecom_data,"Standardized_11.csv")
```

<u>Outcome/ Inference</u>

The continuous variables are standardized and__is saved as the file Standardized_11.csv in the working directory (Refer Annexure)

<u>Step 4:</u> Splitting the data set between training set and test set

Reason: The machine learning classification model will be built on the training set and once the model is developed, the same will be tested on the test set.

Reference R Code:

```
set.seed(8)
split = sample.split(telecom_data, SplitRatio = 0.75)
my_training_data = subset(telecom_data, split == TRUE)
my_test_data = subset(telecom_data, split == FALSE)
```

<u>Outcome/ Inference</u>

The data set is splitted in the ratio 70% and 30% among the training set and test set

<u>Step 5:</u> Building the KNN (K Nearest Neighbour) Model

Reason : The KNN Model is built on the training set and tested on the test set.

Reference R Code :

```
y_pred_data = knn(train = my_training_data[,-10],
            test = my_test_data[,-10],
            cl = my_training_data[,10],
            k = 5)

y_pred_data=as.numeric(y_pred_data)
```

<u>Outcome/ Inference</u>

The KNN Model is built

Step 6: Creation of the confusion matrix and detecting Accuracy of the Model

Reason : The confusion matrix is deduced and the accuracy of the model is calculated

Reference R Code :

##Making the confusion Matrix and calculating the accuracy

my_cm = table(my_test_data[,10], y_pred_data)

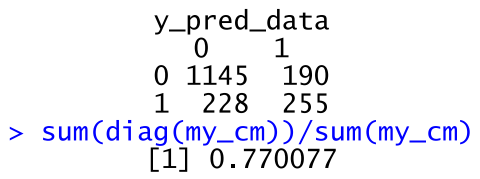
colnames(my_cm)<-c("0","1")

my_cm

sum(diag(my_cm))/sum(my_cm)

Outcome/ Inference

The confusion matrix and the accuracy of the model is as follows :



```
   y_pred_data
       0    1
  0 1145  190
  1  228  255
> sum(diag(my_cm))/sum(my_cm)
[1] 0.770077
```

The accuracy of the model thus calculated is approx. 77%

Step 7: Plot ROC Curve, Area under the Curve (AUC) and calculation of Sensitivity and Specificity.

Reason :

- **Sensitivity (also called the true positive rate, the recall) measures the proportion of positives that are correctly identified as such (e.g. the percentage customer who are correctly identified as leaving the telecom company).**
- **Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g. the percentage of customer who are correctly identified as not leaving the telecom company).**

Reference R Code :

## Plot ROCR Curve

af<-prediction(y_pred_data,my_test_data$Churn)

af

ag<-performance(af,"tpr","fpr")

plot(ag,main='ROC CURVE - KNN')

abline(a=0,b=1)

ac<-performance(af,"auc")

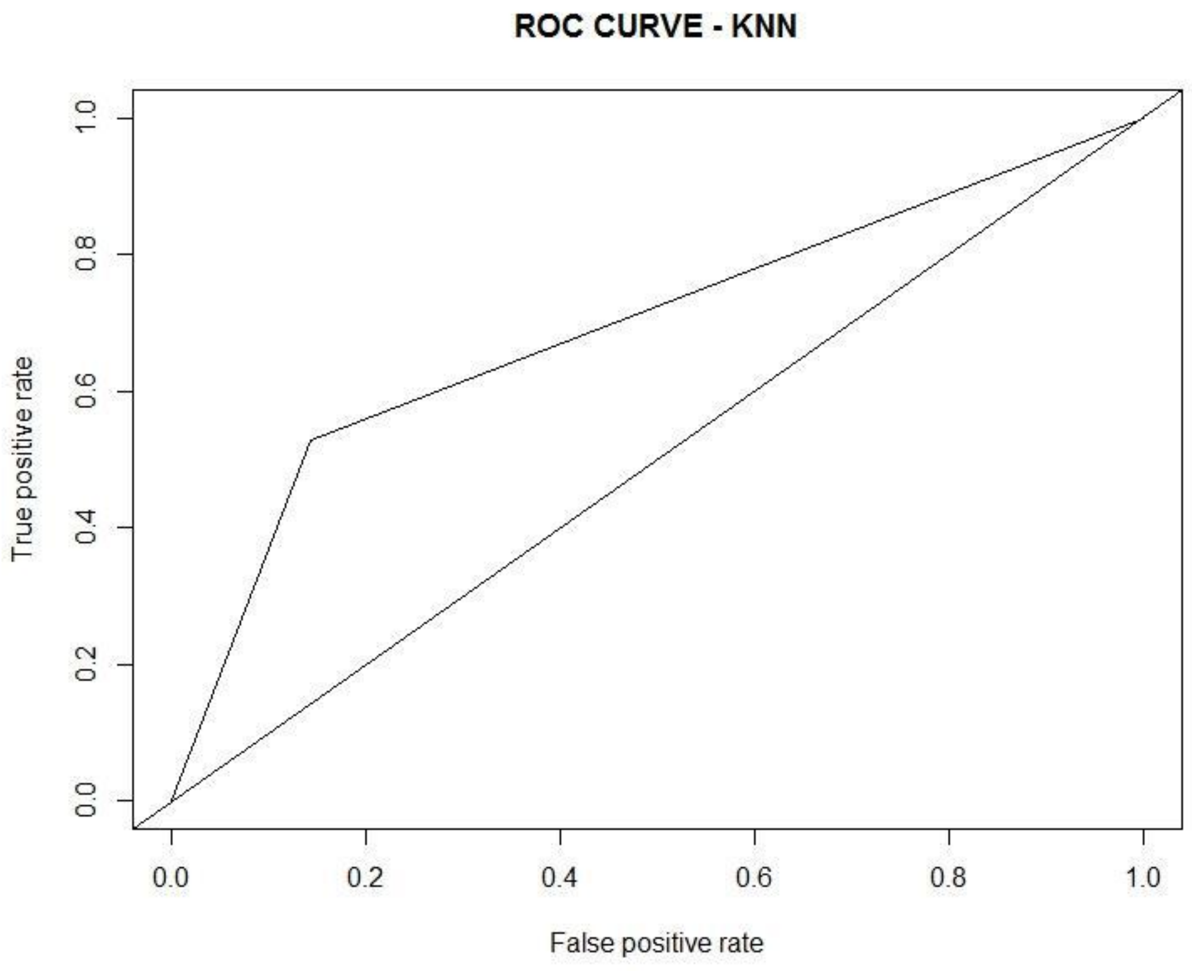
ac

## Calculating the Sensitivity and Specificity

sensitivity(my_cm)

specificity(my_cm)

Outcome/ Inference



The R output is as follows:

```
Slot "y.values":
[[1]]
[1] 0.6928141

> sensitivity(my_cm)
[1] 0.8339403
>
> specificity(my_cm)
[1] 0.5730337
```

The Area under the ROC curve calculated is 69%

The calculated value of sensitivity is 83.39%

The calculated value of specificity is 57.30%

D : Method : Artificial Neural Network (ANN)

The Step 1 to Step 4 is same as above

Step 5: Building the ANN Model

Reason : The ANN Model is built on the training set and tested on the test set.

Reference R Code :

```
## Building the ANN Classification Model

h2o.init(nthreads = -1)
classifier = h2o.deeplearning(y = 'Churn',
                              training_frame = as.h2o(my_training_data),
                              activation = 'Rectifier',
                              hidden = c(5,5),
                              epochs = 100,
                    train_samples_per_iteration = -2,seed = 1)

## Predicting the test set results
prob_pred = h2o.predict(classifier, newdata = as.h2o(my_test_data[-10]))
y_pred_test_data =  ifelse(prob_pred > 0.5, 1, 0)
y_pred_test_data = as.vector(y_pred_test_data )
```

<u>Outcome/ Inference</u>

The ANN Model is built

<u>Step 6</u>: Creation of the confusion matrix and detecting Accuracy of the Model

Reason : The confusion matrix is deduced and the accuracy of the model is calculated

Reference R Code :

```
## Making the confusion Matrix and calculating the accuracy

my_cm = table(my_test_data[,10], y_pred_test_data)
my_cm
sum(diag(my_cm))/sum(my_cm)
```

<u>Outcome/ Inference</u>

The output of the confusion matrix and AUC is as follows:

```
   y_pred_test_data
        0    1
   0 1187  148
   1  228  255
sum(diag(my_cm))/sum(my_cm)
[1] 0.7931793
```

The Accuracy of the model thus calculated is .793 or 79.31%.

<u>Step 7</u>: Plot ROC Curve, Area under the Curve (AUC) and calculation of Sensitivity and Specificity.

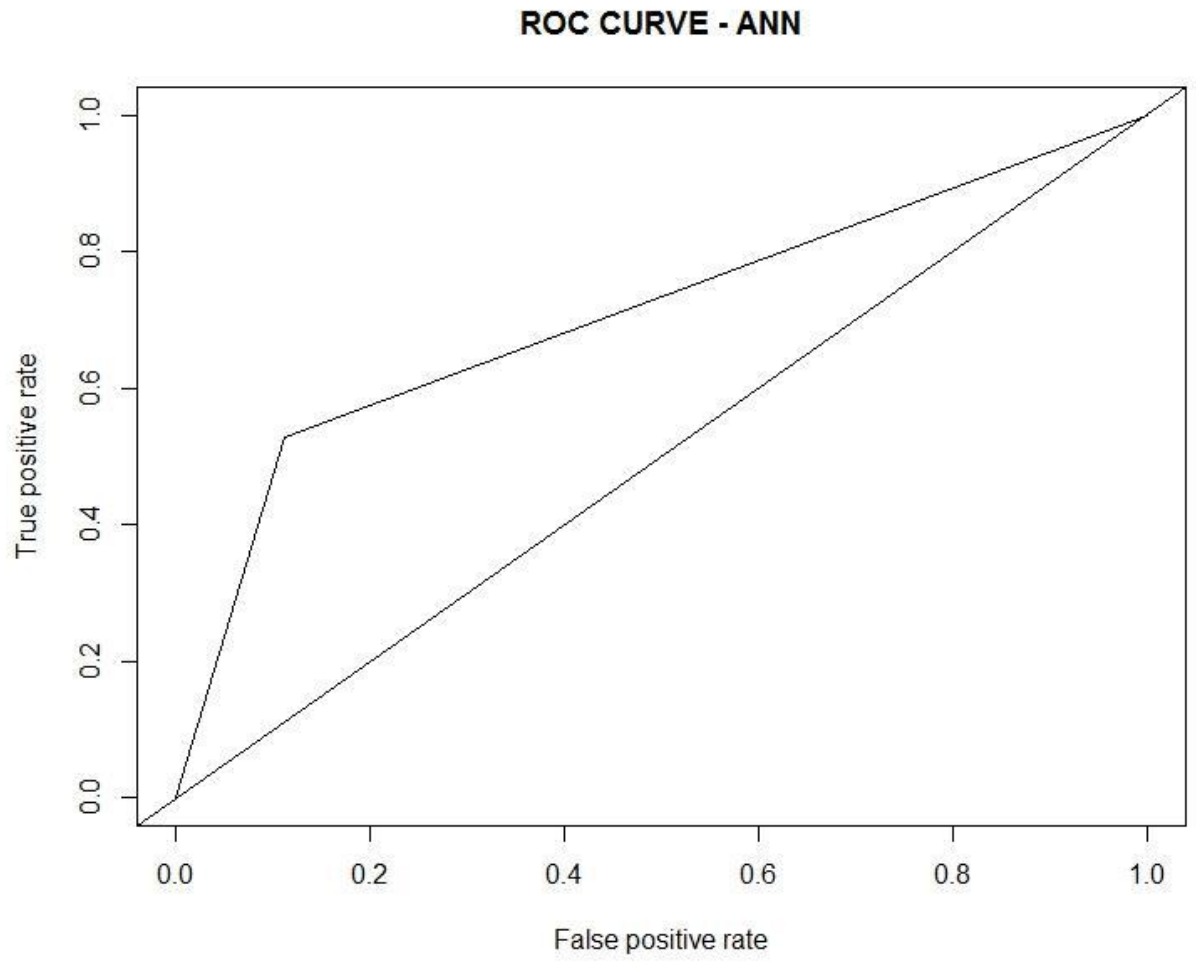
Reason :

- **Sensitivity (also called the true positive rate, the recall) measures the proportion of positives that are correctly identified as such (e.g. the percentage customer who are correctly identified as leaving the telecom company).**
- **Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g. the percentage of customer who are correctly identified as not leaving the telecom company**

Reference R Code :

```
## Plot ROCR Curve

af<-prediction(y_pred_test_data,my_test_data$Churn)

af

ag<-performance(af,"tpr","fpr")

plot(ag,main='ROC CURVE - ANN')

abline(a=0,b=1)

ac<-performance(af,"auc")

ac

## Calculating the Sensitivity and Specificity

sensitivity(my_cm)


specificity(my_cm)
```

Outcome/ Inference



The R output is as follows :

```
Slot "y.values":
[[1]]
[1] 0.7085444


> sensitivity(my_cm)
[1] 0.8388693
>
> specificity(my_cm)
[1] 0.6327543
```

The Area under the ROC curve calculated is 71%

The calculated value of sensitivity is 83.89%

The calculated value of specificity is 63.28%

CONCLUSION :

The comparisons of the performance of the two models are as follows:

	Accuracy (%)	AUC (%)	Sensitivity (%)	Specificity (%)
Model KNN	77	69.28	83.39	57.3
Model ANN	79.31	70.85	83.89	63.28

The above table shows that in every aspect of the performance criteria, ANN model is the best in comparison to KNN Model.

ANNEXURE

1. **Working Data Set**

Telecom Customer Data.csv

2. **Imputed Data Set**

Imputed Data_15.csv

3. **Transformed Data Set**

Transformed_Data_15.csv

4. **Standardized Data Set**

Standardized_11.csv

5. **R Code**

R_Code.txt