```
---
title: "Machine Learning Final Project"
author: "Ram"
date: "12/14/2021"
output: word_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


Loading the required libraries

```{r}
library(class)
library(caret)
library(ISLR)
```


Read the required data

```{r}
survey.lung.cancer <- read.csv("C:/Users/ramne/Desktop/ML
Assignment/Final Project/survey lung cancer.csv")
View(survey.lung.cancer)
```


Creating the dummy variables for Column 'Gender'.

```{r}
library(psych)
dummy_Data <- as.data.frame(dummy.code(survey.lung.cancer$GENDER))
names(dummy_Data) <- c("GENDER_1", "GENDER_2")
```


Removing the original Column 'Gender' as we have new dummy variables
'GENDER_1' and 'GENDER_2'


```{r}
Data_without_Gender = subset(survey.lung.cancer,select = - c(GENDER))
New_data <- cbind(Data_without_Gender,dummy_Data)
New_data$LUNG_CANCER <- as.factor(New_data$LUNG_CANCER)
```

Data is split into training(60%) and validation(40%) data

```{r}
set.seed(123)
train_index <- createDataPartition(New_data$LUNG_CANCER,p=0.6,list =
FALSE)
train_data <- New_data[train_index,]
validation_data <- New_data[-train_index,]
```
```

Normalize the training data using preProcess function.

Apply the normalized model on the training and validation data using Predict Function.

```{r}



train.norm.df <- train_data
valid.norm.df <- validation_data
traval.norm.df <- New_data #(Training + Validation data)
#Use preProcess() function to normalize numerical columns from the dataset
Values_z_normalised <- preProcess(train_data[,-15],method = c("center","scale"))
#Applying the normalized model on the training, validation and test data
train.norm.df[,-15] <-  predict(Values_z_normalised,train_data[,-15])
valid.norm.df[,-15] <-  predict(Values_z_normalised,validation_data[,-15])
traval.norm.df[,-15] <-  predict(Values_z_normalised,New_data[,-15])
```

Determine the best K

```{r}
library(caret)
set.seed(123)
accuracy.df <- data.frame(k = seq(1, 15, 1), accuracy = rep(0, 15))
# compute knn for different k on validation.
for(i in 1:15) {
  knn.pred <- knn(train.norm.df[,-15], valid.norm.df[,-15],
                  cl = train.norm.df[, 15], k = i)
  accuracy.df[i, 2] <-
confusionMatrix(knn.pred,valid.norm.df[,15])$overall[1]
}
accuracy.df
plot(accuracy.df,type="o")
```

From the above results, best K is 12.

Confusion Matrix for the best K(12).

```{r}
Model.k.12 <- knn(train.norm.df[,-15], valid.norm.df[,-15],
                  cl = train.norm.df[,15], k = 12,prob = TRUE)
confusionMatrix(Model.k.12,valid.norm.df[,15])
```

Splitting the data into Training (50%), Validation (30%) and Test (20%) sets

```{r}
## Data Splitting (50% Training Data and 30% for validation data and 20% test data)
set.seed(123)
str(New_data)
```

```
test_index1 <- createDataPartition(New_data$LUNG_CANCER,p=0.2,list =
FALSE)
Test_Data2 <- New_data[test_index1,]#  (Test data)
train_vali_data <- New_data[-test_index1,]
train_index2 <-
createDataPartition(train_vali_data$LUNG_CANCER,p=0.625,list = FALSE)
train_data2 <- train_vali_data[train_index2,] # (Training data)
validation_data2 <- train_vali_data[-train_index2,]# (validation data)
 NROW(Test_Data2)
 NROW(train_data2)
 NROW(validation_data2)
```

Applying the K-nn model for the data

```{r}

# Renormalizing the (training+validation) data
set.seed(123)
Values_z_normalised2 <- preProcess(traval.norm.df[,-15], method =
c("center","scale"))
traval.norm.df[,-15] <-  predict(Values_z_normalised2,New_data[,-15])
 # Data Normalization

# Copy the original data
train.norm.df2 <- train_data2
valid.norm.df2 <- validation_data2
train_vali.norm.df <- train_vali_data
test.norm.df2 <-Test_Data2
#Use preProcess() function to normalize numerical columns from the
New_data dataset
Values_z_normalised_repartition <- preProcess(train_data2[,-15],method =
c("center","scale"))
train.norm.df2[,-15] <-
predict(Values_z_normalised_repartition,train_data2[,-15])
valid.norm.df2[,-15] <-
predict(Values_z_normalised_repartition,validation_data2[,-15])
train_vali.norm.df[,-15] <-
predict(Values_z_normalised2,train_vali_data[,-15])
test.norm.df2[,-15] <-
predict(Values_z_normalised_repartition,Test_Data2[,-15])

## Modeling k-NN for validation data
set.seed(123)
train_knn_12<- knn(train.norm.df2[,-15],train.norm.df2[,-
15],cl=train.norm.df2[,15],k=12,prob=TRUE)
valid_knn_12<- knn(train.norm.df2[,-15],valid.norm.df2[,-15],cl=
train.norm.df2[,15],k=12,prob= TRUE)
#print(ModelNew.k.1)
head(train_knn_12)
head(valid_knn_12)
actual= valid.norm.df2[,15]
mean(valid_knn_12==actual)
class_prob = attr(valid_knn_12,"prob")
head(class_prob)
# Knn for test data
```

~~~
Values_z_normalised3<- preProcess(train_vali_data[,-15], method =
c("center","scale"))
train_vali.norm.df[,-15] <-
predict(Values_z_normalised3,train_vali_data[,-15])
test.norm.df2[,-15]<- predict(Values_z_normalised3,Test_Data2[,-15])
test_knn_12<- knn(train_vali.norm.df[,-15],test.norm.df2[,-
15],cl=train_vali.norm.df[,15],k=12)
#print(Model_new3)
head(test_knn_12)
actual= test.norm.df2[,15]
mean(test_knn_12==actual)
```
~~~

Including Confusion Matrix

Accuracy of the Knn models for traning,validation and test datasets for k=12

~~~
```{r}
confusionMatrix(train_knn_12,as.factor(train.norm.df2[,15]),positive =
'1')
confusionMatrix(valid_knn_12,as.factor(valid.norm.df2[,15]),positive =
'1')
confusionMatrix(test_knn_12,as.factor(test.norm.df2[,15]),positive = '1')
```
~~~