Homework 2

Team 3

3/10/2021

## ISTM660 Homework 2 - Team 3

Pre-requisites:

# install.packages("ISLR")  
# install.packages("class")  
# install.packages("caret")  
# install.packages("rstudioapi")  
# install.packages("ROCR")  
# install.packages("pROC")

Question 1 - Predicting with Linear Regression

#HW2 - (12 points) Predicting with Linear Regression  
  
#Step 1 - Initializing the current file path & clearing variables from   
# the environment of execution.  
  
library(rstudioapi) # This is a external library of functions  
# Getting the path of your current open file  
current\_path = rstudioapi::getActiveDocumentContext()$path   
setwd(dirname(current\_path ))  
rm(list=ls())  
cat("\014")

#Step 2 - Loading the required input files for processing  
  
load('customer.rdata')  
attach(cust)  
  
#a. Specify training (75%) and testing (25%) subsets of the customer data.  
  
set.seed(25)  
testindex <- sample(nrow(cust),trunc(nrow(cust)/4))  
train<- cust[-testindex,]  
test<- cust[testindex,]  
  
#b. Determine your best linear regression model that has profit as the response variable   
# and has at minimum, timetoreturn as a predictor. How do you know this model is best?  
trainresults1 <- lm(profit~timetoreturn, data=train)  
summary(trainresults1)

##   
## Call:  
## lm(formula = profit ~ timetoreturn, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -54081 -2756 -1182 1307 44222   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3915.02 320.81 12.204 <2e-16 \*\*\*  
## timetoreturn -19.84 7.77 -2.554 0.0109 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6289 on 703 degrees of freedom  
## Multiple R-squared: 0.00919, Adjusted R-squared: 0.007781   
## F-statistic: 6.521 on 1 and 703 DF, p-value: 0.01087

trainresults2 <- lm(profit~timetoreturn+totalsales, data=train)  
summary(trainresults2)

##   
## Call:  
## lm(formula = profit ~ timetoreturn + totalsales, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -61557 -2439 -909 1444 38500   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3503.54935 335.74960 10.435 < 2e-16 \*\*\*  
## timetoreturn -25.33349 7.83125 -3.235 0.001274 \*\*   
## totalsales 0.04909 0.01293 3.798 0.000159 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6230 on 702 degrees of freedom  
## Multiple R-squared: 0.02914, Adjusted R-squared: 0.02637   
## F-statistic: 10.53 on 2 and 702 DF, p-value: 3.109e-05

trainresults3 <- lm(profit~timetoreturn+totalsales+totalrefund, data=train)  
summary(trainresults3)

##   
## Call:  
## lm(formula = profit ~ timetoreturn + totalsales + totalrefund,   
## data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -22290 -1742 -919 928 35998   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2447.79352 226.03770 10.829 <2e-16 \*\*\*  
## timetoreturn -29.57710 5.20896 -5.678 2e-08 \*\*\*  
## totalsales 0.41730 0.01506 27.714 <2e-16 \*\*\*  
## totalrefund -0.71513 0.02401 -29.781 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4142 on 701 degrees of freedom  
## Multiple R-squared: 0.5714, Adjusted R-squared: 0.5696   
## F-statistic: 311.5 on 3 and 701 DF, p-value: < 2.2e-16

trainresults4 <- lm(profit~timetoreturn+totalsales+totalrefund+lengthofrelationship, data=train)  
summary(trainresults4)

##   
## Call:  
## lm(formula = profit ~ timetoreturn + totalsales + totalrefund +   
## lengthofrelationship, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23563 -1588 -607 909 36540   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3200.47064 328.61874 9.739 < 2e-16 \*\*\*  
## timetoreturn -27.38543 5.22331 -5.243 2.1e-07 \*\*\*  
## totalsales 0.43217 0.01570 27.534 < 2e-16 \*\*\*  
## totalrefund -0.72271 0.02398 -30.132 < 2e-16 \*\*\*  
## lengthofrelationship -0.73773 0.23510 -3.138 0.00177 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4116 on 700 degrees of freedom  
## Multiple R-squared: 0.5773, Adjusted R-squared: 0.5749   
## F-statistic: 239 on 4 and 700 DF, p-value: < 2.2e-16

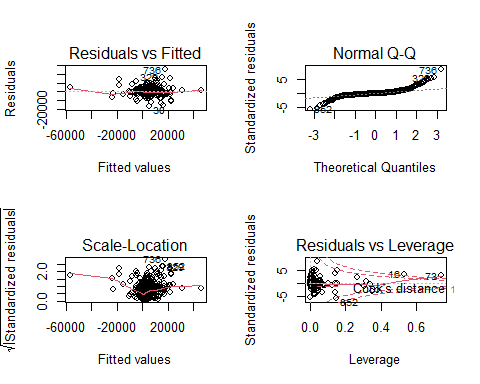
trainresults5 <- lm(profit~., data=train)  
summary(trainresults5)

##   
## Call:  
## lm(formula = profit ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -21708.3 -1332.9 -351.8 879.9 30617.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -9.486e+02 6.613e+02 -1.435 0.15188   
## lengthofrelationship -9.167e-01 2.819e-01 -3.252 0.00120 \*\*   
## itemsreturned 1.886e+01 3.608e+00 5.228 2.27e-07 \*\*\*  
## itemspurchased -6.213e+00 2.000e+00 -3.107 0.00197 \*\*   
## totalsales 3.443e-01 2.877e-02 11.967 < 2e-16 \*\*\*  
## totalrefund -6.602e-01 4.211e-02 -15.679 < 2e-16 \*\*\*  
## categories 3.677e+02 6.168e+01 5.961 4.00e-09 \*\*\*  
## timetoreturn -2.771e+01 4.646e+00 -5.965 3.90e-09 \*\*\*  
## channels 1.064e+02 1.699e+02 0.626 0.53147   
## avgprice 2.223e+01 8.400e+00 2.646 0.00832 \*\*   
## purchasefrequency 1.632e+04 1.673e+03 9.758 < 2e-16 \*\*\*  
## returnfrequency -4.235e+04 4.283e+03 -9.888 < 2e-16 \*\*\*  
## avgrefund -3.393e+00 4.446e+00 -0.763 0.44562   
## returnrate 1.467e+03 8.475e+02 1.731 0.08385 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3640 on 691 degrees of freedom  
## Multiple R-squared: 0.6737, Adjusted R-squared: 0.6676   
## F-statistic: 109.8 on 13 and 691 DF, p-value: < 2.2e-16

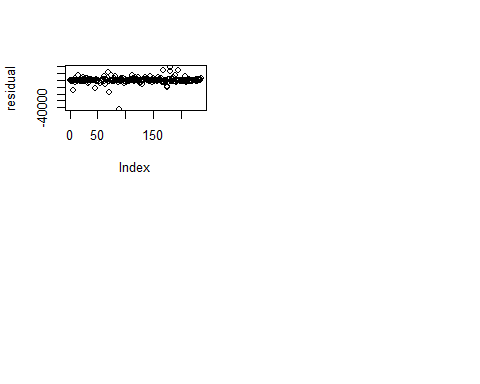
trainresults6 <- lm(profit~timetoreturn+totalsales+totalrefund+returnfrequency+purchasefrequency+lengthofrelationship+timetoreturn\*totalsales+timetoreturn\*totalrefund+lengthofrelationship\*totalrefund, data=train)  
summary(trainresults6)

##   
## Call:  
## lm(formula = profit ~ timetoreturn + totalsales + totalrefund +   
## returnfrequency + purchasefrequency + lengthofrelationship +   
## timetoreturn \* totalsales + timetoreturn \* totalrefund +   
## lengthofrelationship \* totalrefund, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15958.3 -1182.2 -311.5 745.9 24892.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.841e+03 3.027e+02 6.082 1.96e-09 \*\*\*  
## timetoreturn 5.475e+00 4.882e+00 1.121 0.262502   
## totalsales 3.955e-01 2.336e-02 16.928 < 2e-16 \*\*\*  
## totalrefund -6.207e-01 6.783e-02 -9.150 < 2e-16 \*\*\*  
## returnfrequency -2.786e+04 3.024e+03 -9.211 < 2e-16 \*\*\*  
## purchasefrequency 1.411e+04 1.088e+03 12.968 < 2e-16 \*\*\*  
## lengthofrelationship -9.622e-01 1.967e-01 -4.891 1.25e-06 \*\*\*  
## timetoreturn:totalsales -1.779e-03 4.653e-04 -3.823 0.000144 \*\*\*  
## timetoreturn:totalrefund -3.986e-03 7.281e-04 -5.474 6.14e-08 \*\*\*  
## totalrefund:lengthofrelationship 1.290e-04 2.121e-05 6.081 1.96e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3091 on 695 degrees of freedom  
## Multiple R-squared: 0.7634, Adjusted R-squared: 0.7603   
## F-statistic: 249.1 on 9 and 695 DF, p-value: < 2.2e-16

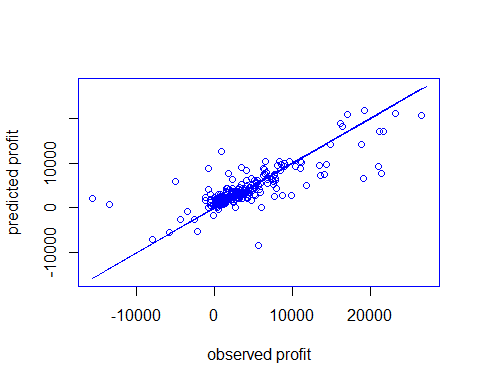
#trying different combinations of predictor variables to finally come to  
# the model which is a good fit  
  
#When we took different combinations of predictor variables, the last combination (trainresults6) gave the highest values for Multiple R-squared (0.7634) and Adjusted R-squared (0.7603).  
#This values signifies that about 76% of variance in the training set is captured by our Multiple Linear Model.  
#Also, for this model all the predictors are significant which can be inferred based on the t and Pr(>|t|) values and the p-value is almost 0 which further strengthens our hypothesis.  
  
#c. Is this a good model fit?  
  
#When we look at the t and Pr(>|t|) values for our preferred model, we can say that all the predictors variables are significant.  
#Multiple R-squared is 0.7634 and Adjusted R-squared values is 0.7603 which is good. The model is able to capture 76% of the variance in the training set.  
#The p-value is almost close to 0 and F-statistic is large which indicate better model performance.  
  
#d. Explain the relationship between timetoreturn and profit.  
  
#Based on our model, there is a direct relationship between tiometoreturn and profit.  
#If all other variables are constant then a unit increase in the timetoreturn will result in 5.47 increase in the profit which is surprising.  
  
  
#e. Are the assumptions for linear regression satisfied?  
  
par(mfrow=c(2,2))  
plot(trainresults6)



# We check for 4 assumptions of linear regression  
# 1. Mean of y is linear in x  
# 2. Error terms are normally distributed  
# 3. Error terms have constant variance  
# 4. Error terms are independent of x  
# Residual vs Fitted curve: We can see almost horizontal line with no distinctive pattern which signifies a linear relationship   
# Normal Q-Q: Most of the residual points follow a straight dashed line  
# Scale-Location: Horizontal line with equally spread points is a good indication of homoscedasticity. In our model, there is almost close to horizontal line  
# Residual vs Leverage: We can see majority of our points are in the range of -5 to 5  
  
# Based on the four plots, we can say that the assuptions of Linear regression are almost satisfied in our model  
  
# Is this a good model for predicting whether a customer is profitable (profit>$0.00) or  
# unprofitable (profit<$0.00)?  
  
predProfit = predict(trainresults6, test)  
residual = test$profit - predProfit  
plot(residual)  
  
par(mfrow=c(1,1))



par(pch=21, col="blue")  
plot(test$profit,test$profit,type='l', xlab="observed profit", ylab="predicted profit")  
points(test$profit,predProfit)



# From the above plot we can see that our model is a good predictor for profit in the range of 0 to 14000   
# It does not do a good job for profits less than 0 and for profits greater than 14000

Question 2 - Predicting with Logistic Regression

# HW2 - (12 points) Predicting with Logistic Regression  
  
#Step 1 - Initializing the current file path & clearing variables from   
# the environment of execution.  
  
library(rstudioapi) # This is a external library of functions  
# Getting the path of your current open file  
current\_path = rstudioapi::getActiveDocumentContext()$path   
setwd(dirname(current\_path ))  
rm(list=ls())  
  
#Step 2 - Loading the required input files for processing  
  
load('customer.rdata')  
attach(cust)

## The following objects are masked from cust (pos = 3):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

cust1<- cust  
  
# a. Using the ifelse command, create a new a new "class" variable for the transaction data set   
# that is a factor with value 1 if profit is negative and 0 otherwise.   
# Remove the profit column and create training and test sets as in the first question.  
  
for (i in 1:nrow(cust1))  
{  
 x = cust1[i,14]  
 if(x < 0)  
 {  
 cust1[i,15] = 1  
 }  
 else if (x > 0)  
 {  
 cust1[i,15] = 0  
 }  
}  
  
#removing the profit column  
  
colnames(cust1)[colnames(cust1) == "V15"] <- "profit\_new"  
cust1 <- cust1[,c(1:13,15)]  
  
  
set.seed(25)  
testindex <- sample(nrow(cust1),trunc(nrow(cust1)/4))  
log.train<- cust1[-testindex,]  
log.test<- cust1[testindex,]  
  
  
#b. Determine your best logistic regression model that has profit   
# as the response variable and has at minimum, timetoreturn as a predictor.   
# How do you know this model is best?  
  
trainresults=glm(profit\_new~timetoreturn+totalsales+totalrefund+timetoreturn\*totalrefund+totalsales\*totalrefund,data=log.train,family=binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(trainresults)

##   
## Call:  
## glm(formula = profit\_new ~ timetoreturn + totalsales + totalrefund +   
## timetoreturn \* totalrefund + totalsales \* totalrefund, family = binomial,   
## data = log.train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4273 -0.3513 -0.2056 -0.0539 3.4327   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.380e+00 2.901e-01 -8.205 2.31e-16 \*\*\*  
## timetoreturn 8.652e-03 4.391e-03 1.970 0.048805 \*   
## totalsales -5.115e-04 9.499e-05 -5.384 7.27e-08 \*\*\*  
## totalrefund 5.437e-04 1.100e-04 4.943 7.71e-07 \*\*\*  
## timetoreturn:totalrefund 1.346e-05 2.868e-06 4.693 2.69e-06 \*\*\*  
## totalsales:totalrefund -1.620e-09 4.551e-10 -3.559 0.000372 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 451.74 on 704 degrees of freedom  
## Residual deviance: 228.76 on 699 degrees of freedom  
## AIC: 240.76  
##   
## Number of Fisher Scoring iterations: 8

probs<-predict(trainresults,log.test,type="response")  
profitpred<-rep(0,nrow(log.test))  
profitpred[probs>0.5]<-1  
table(log.test$profit\_new,profitpred)

## profitpred  
## 0 1  
## 0 211 1  
## 1 12 11

mean(profitpred==log.test$profit\_new)

## [1] 0.9446809

#After working with various combinations of different predictor variables  
#We can say that the given model with predictor variable is a good fit for the given data  
#We are getting a model accuracy of 94.4% which is pretty good  
  
  
#c.Explain the relationship between timetoreturn and class.  
  
#Based on our model, there is a direct relationship between tiometoreturn and profit.  
#If all other variables are constant then a unit increase in the timetoreturn will result in 0.0086 increase in the profit which is surprising.  
  
  
#d. Generate an ROC curve for your model using test set predictions. What is the AUC and  
# qualitatively what does the ROC curve indicate about this model?  
  
library("pROC")

## Warning: package 'pROC' was built under R version 4.0.4

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

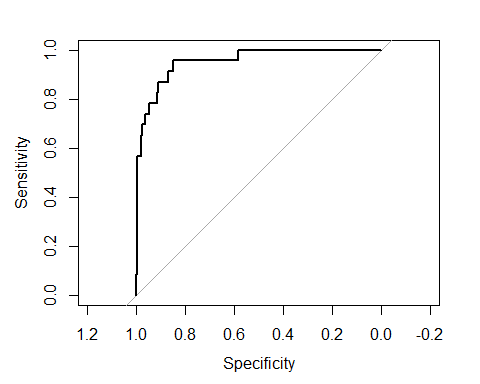
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

profit.roc<-roc(log.test$profit\_new,probs)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(profit.roc)



profit.roc

##   
## Call:  
## roc.default(response = log.test$profit\_new, predictor = probs)  
##   
## Data: probs in 212 controls (log.test$profit\_new 0) < 23 cases (log.test$profit\_new 1).  
## Area under the curve: 0.9532

#e. Is this a good model for predicting whether a customer is profitable (profit>$0.00) or  
# unprofitable (profit<$0.00)?  
  
table(log.test$profit\_new,profitpred)

## profitpred  
## 0 1  
## 0 211 1  
## 1 12 11

#We can see from the confusion metrics that our model is able to give correct result for 91.66 of the   
#times where profit is greater than 0, also it is giving a good 94.6%   
#correct results when the profit is less than 0  
  
#f. What can you say about the accuracy of predicting customers that are not profitable as the  
# threshold/cutoff probability increases?  
  
  
thresholds<-seq(from=0.1, to=0.9, by=0.05)  
acc<-rep(0,19)  
for (i in 1:19) {  
 profitpred<-rep(0,nrow(log.test))  
 profitpred[probs>thresholds[i]]<-1  
 table(log.test$profit\_new,profitpred)  
   
 acc[i]<-mean(profitpred==log.test$profit\_new)  
 acc[i]  
}   
  
acc

## [1] 0.9276596 0.9446809 0.9446809 0.9531915 0.9531915 0.9446809 0.9446809  
## [8] 0.9446809 0.9446809 0.9446809 0.9404255 0.9404255 0.9404255 0.9404255  
## [15] 0.9361702 0.9319149 0.9319149 0.9021277 0.9021277

max(acc)

## [1] 0.9531915

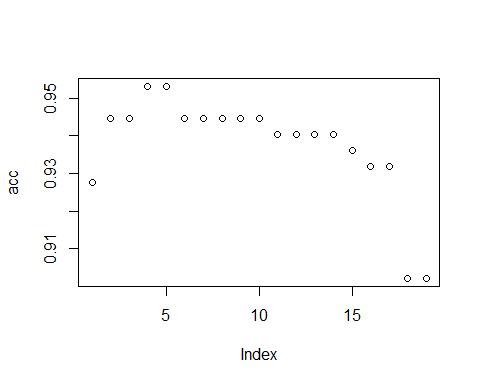
which.max(acc)

## [1] 4

thresholds[4]

## [1] 0.25

plot(acc)



As the threshold/cutoff of probability increases the prediction accuracy first increases then remain constant and finally decreases which can also be seen from the plot.

Question 3 - Predicting with KNN

# HW2 - (12 points) Predicting with KNN  
  
#Step 1 - Initializing the current file path & clearing variables from   
# the environment of execution.  
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.4

library(class)

## Warning: package 'class' was built under R version 4.0.4

library(caret)

## Warning: package 'caret' was built under R version 4.0.4

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.4

library(ROCR)

## Warning: package 'ROCR' was built under R version 4.0.4

library(rstudioapi)  
current\_path = rstudioapi::getActiveDocumentContext()$path   
setwd(dirname(current\_path ))  
rm(list=ls())  
  
#Step 2 - Loading the required input files for processing  
load("customer.rdata")  
attach(cust)

## The following objects are masked from cust (pos = 10):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

## The following objects are masked from cust (pos = 11):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

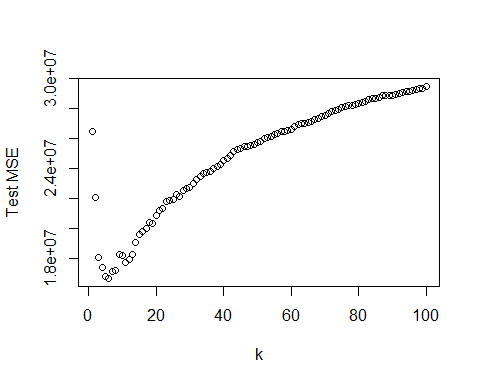
#scaling  
prep<-preProcess(cust[,c(2:13)],method=c("scale"))  
  
# a - Create the same testing and training subsets of the transaction data as   
#you did for linear regression  
  
set.seed(25)  
train<-sample(nrow(cust),trunc(nrow(cust)/2))  
train.x<-predict(prep,cust[train,c(2:13)])  
train.y<-cust[train,14]  
test.x<-predict(prep,cust[-train,c(2:13)])  
test.y<-cust[-train,14]  
  
# b - Conduct KNN regression using all the available columns for prediction.  
#What is your best KNN regression model? Justify your answer.  
mse<-rep(0,100)  
for (knum in 1:100) {  
 knn.fit<-knnreg(train.x,train.y,k=knum)  
 mse[knum]<-mean((test.y-predict(knn.fit,test.x))^2,na.rm=TRUE)  
}  
print(paste("Miniminum MSE of",as.character(round(mse[which.min(mse)])),"at k =",as.character(which.min(mse) )))

## [1] "Miniminum MSE of 16703024 at k = 6"

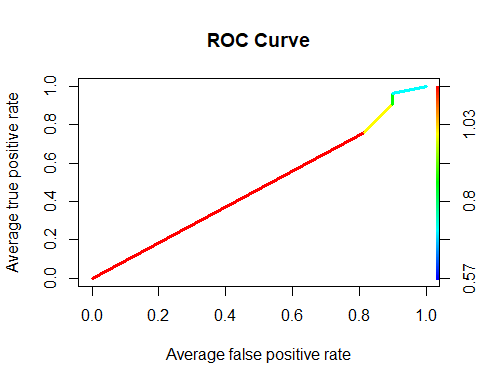
plot(1:100,mse,type="p",xlab="k",ylab="Test MSE")  
  
# At k = 6, we get the least MSE value so k = 6 is the best model  
  
# c - Is this a good model for predicting whether a customer is profitable   
#(profit>$0.00) or unprofitable (profit<$0.00)? Justify your answer.  
# This isn't a good model to predict profitability, as it is only predicting the profit   
# and not the profitability  
# also, the MSE is a high value   
  
# d - Create a new "class" column identical to what you did for logistic regression  
#(and drop profit) and recreate the corresponding training and testing subsets of the data.  
cust$class<-0  
  
for(row in 1:nrow(cust))  
{  
 x = cust[row,14]  
 if(x < 0)  
 {  
 cust[row,15] = 0  
 }  
 else if (x > 0)  
 {  
 cust[row,15] = 1  
 }  
}  
  
# removing the profit column  
cust<-cust[,c(1:13,15)]  
  
# e - Conduct KNN classification using all columns.   
#What is your best KNN classification model? Justify your answer.  
prep<-preProcess(cust[,c(2:13)],method=c("scale"))  
  
set.seed(25)  
train<-sample(nrow(cust),trunc(nrow(cust)/2))  
train.x<-predict(prep,cust[train,c(2:13)])  
train.y<-cust[train,14]  
test.x<-predict(prep,cust[-train,c(2:13)])  
test.y<-cust[-train,14]  
  
acc<-rep(0,100)  
for (knum in 1:100) {  
 knn.fit<-knn(train.x,test.x,train.y,k=knum)  
 acc[knum]<-mean(knn.fit==test.y)  
}  
print(paste("Highest accuracy of",as.character(acc[which.max(acc)]),"at k =",as.character(which.max(acc) )))

## [1] "Highest accuracy of 0.940425531914894 at k = 7"

plot(1:100,mse,type="p",xlab="k",ylab="Test MSE")



# f - Create an ROC curve for your best model. Hint: use the prob argument for KNN.   
#See documentation or search for help on how to access the values created by the prob argument.   
#They are stored as attributes of the object created when calling the knn   
knn.fit<-knn(train.x,test.x,train.y,k=7, prob=TRUE)  
prob <- attr(knn.fit, "prob")  
pred\_knn <- prediction(prob, train.y)  
pred\_knn <- performance(pred\_knn, "tpr", "fpr")  
plot(pred\_knn, avg= "threshold", colorize=T, lwd=3, main="ROC Curve")



Question 4 - Of all the models you have created (logistic, linear, knn) which is your best model and why do you consider it best?

Of all the three models that we have explored so far, we can say that Logistic regression is the best. Logistic regression gives a 94.46% prediction accuracy to predict if a customer is profitable or not for cutoff = 0.5 which is great as compared to the others. It does have higher accuracy for different cutoffs as that can lead to the problem of overfitting.

KNN classification was a good model and can be used if not for logistic regression, however, it has lower prediction accuracy than the latter. Finally, we can use Linear regression to predict actual profit but not the probability of a customer being profitable as it more suited to finding continuous data as opposed to finding discrete data, which is required in classification.

Question 5 - Classification with Linear Regression

# HW2 - Classification with Linear Regression  
# It is possible to classify customers as profitable or not profitable using a threshold level of   
# profitability much like a threshold probability for classification.   
# You would compare all the predicted customer profits to the threshold and   
# those that are below the profit threshold you would classify as unprofitable (1) or profitable (0).   
# Evaluate thresholds for profit using your best linear regression model.   
# What is your best threshold level of profit in terms of overall accuracy?   
# Is this a good method of classification? Is it better than the answer you provided to question 4?  
  
  
#Step 1 - Initializing the current file path & clearing variables from   
# the environment of execution.  
library(ISLR)  
library(class)  
library(caret)  
library(ROCR)  
library(rstudioapi)  
current\_path = rstudioapi::getActiveDocumentContext()$path   
setwd(dirname(current\_path ))  
rm(list=ls())  
  
#Step 2 - Loading the required input files for processing  
load("customer.rdata")  
attach(cust)

## The following objects are masked from cust (pos = 3):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

## The following objects are masked from cust (pos = 11):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

## The following objects are masked from cust (pos = 12):  
##   
## avgprice, avgrefund, categories, channels, itemspurchased,  
## itemsreturned, lengthofrelationship, profit, purchasefrequency,  
## returnfrequency, returnrate, timetoreturn, totalrefund, totalsales

#Creating a new class variable for classifying whether the profit column was a profit or loss.  
  
cust$class<-0  
for(row in 1:nrow(cust))  
{  
 x = cust[row,14]  
 if(x < 0)  
 {  
 cust[row,15] = 0  
 }  
 else if (x > 0)  
 {  
 cust[row,15] = 1  
 }  
}  
  
#Using the best linear model from the previous questions and splitting data into training and testing data  
  
set.seed(25)  
trainindex<-sample(nrow(cust),trunc(nrow(cust)/2))  
train<-cust[trainindex,]  
test<-cust[-trainindex,]  
  
lm\_model <- lm(profit~timetoreturn+totalsales+totalrefund+returnfrequency+purchasefrequency+lengthofrelationship+timetoreturn\*totalsales+timetoreturn\*totalrefund+lengthofrelationship\*totalrefund, data=train)  
predicted\_profit <- predict(lm\_model, test)  
  
#Creating a vector of values to put threshold data in  
  
t <- c(-1000:1000)  
mse\_result <- rep(2000, 0)  
  
#Looping through a range of threshold values (in this case, -1000 to 1000)  
  
for(row in 1:length(t)){  
 threshold <- t[row]  
 profitability <- ifelse(predicted\_profit > threshold, 1, 0)  
 mse <- mean((profitability-predicted\_profit)\*\*2)  
 mse\_result[row] <- mse  
}  
  
#Finding out the lowest mean squared error due to changing threshold.  
main\_threshold <- (t[which.min(mse\_result)])

Therefore, in terms of overall accuracy, the value of -181 is the best found threshold value as it has the lowest mse.

Linear regression is a poorer choice to go with when classifying data. Logistic regression and KNN, both do a better job of classifying data. Linear regression is great with continuous data points, but it does not do a good job of working with discrete points. That is because unlike the other two algorithms, it has nothing to do with finding probabilities.

When comparing Q5 with Q4, the preference would, thus, be to side with Q4 for classifying whether a person was profitable or not. Whereas, if we wanted to predict how profitable the customer was, linear regression would the best move.