ISYE 6501 HW1

January 2021

1 Question 2.1

Event: If a person has infected by COVID19 virus.
The result can be categorized to the following:

- Not infected ever
- Has been infected

Of course, the predictor is medical test results.

But if we want to analyze further on this, say to find who is easier to get infected. Then other attributes become valuable. Like occupancy, age, other condition the person may have, where the person lives, if the person wears masks and what kind of mask and how often does the person wear it.

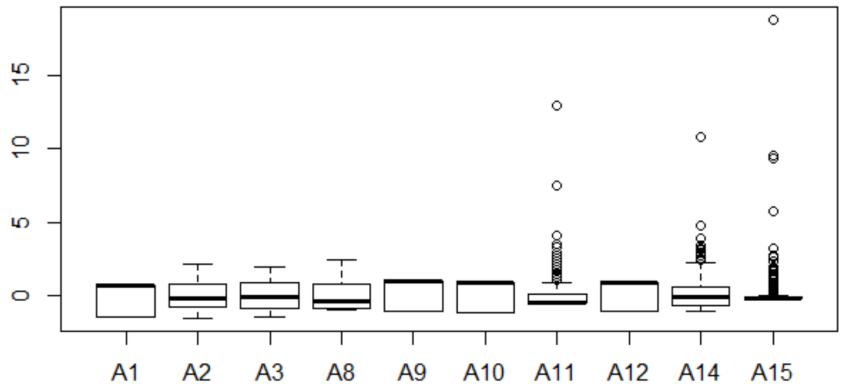
2 Question 2.2 SVM

Per Homework instruction, we know that the Credit Card Data has been cleaned and all missing value have been removed, there is not much cleaning to do with the data. From Table 1 - the summary of Credit Card Data, we can tell that the mean for R1 (response) is 0.453 which is fairly close to 0.5. Since the value for R1 is either 0 or 1, a mean close to 0.5 means that the data is pretty balanced in terms of response. So the accuracy metric should be sufficient to check how well models work.

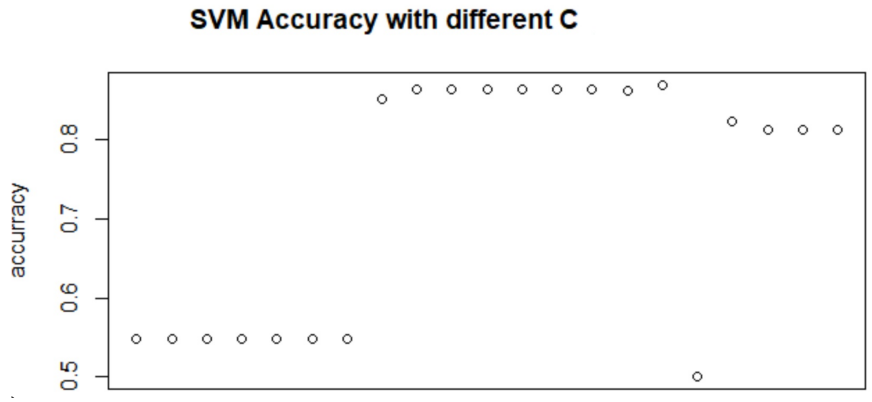
Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
A1	654	0.690	0.463	0	0	1	1
A9	654	0.535	0.499	0	0	1	1
A10	654	0.561	0.497	0	0	1	1
A11	654	2.498	4.966	0	0	3	67
A12	654	0.538	0.499	0	0	1	1
A14	654	180.084	168.316	0	70.8	271	2,000
A15	654	1,012.731	5,249.321	0	0	399	100,000
R1	654	0.453	0.498	0	0	1	1

Reviewing the box plot of the data, we can find that attributes A11, A14, and A15 have outliers at the larger number side. But they should not impact our result when using SVM method.



I have tried 21 numbers for the value C from 10^{-10} to 10^{10}, per the plot below, when $C = 10^5$ we got the best accuracy.



The equation of the classifier using SVM model is:

$$\begin{aligned} &-0.07175452x_1 - 0.10599985x_2 - 0.25131944x_3 + 0.11717882x_8 \\ &+ 1.00373549x_9 - 0.24213026x_{10} + 0.24228761x_{11} - 0.04099124x_{12} \\ &- 0.18893170x_{14} + 0.59799602x_{15} + 0.05792296 = 0 \end{aligned} \tag{1}$$

With a 0.8700306 accuracy of prediction, and $C = 100,000$.

The A9 attribute has the largest positive impact on the approval of credit card process (also our model). The A3 attribute has the largest negative impact. In the contrast, A12 and A1 have the least impact, we may do further research to find if we can remove these two attributes from our model if needed.

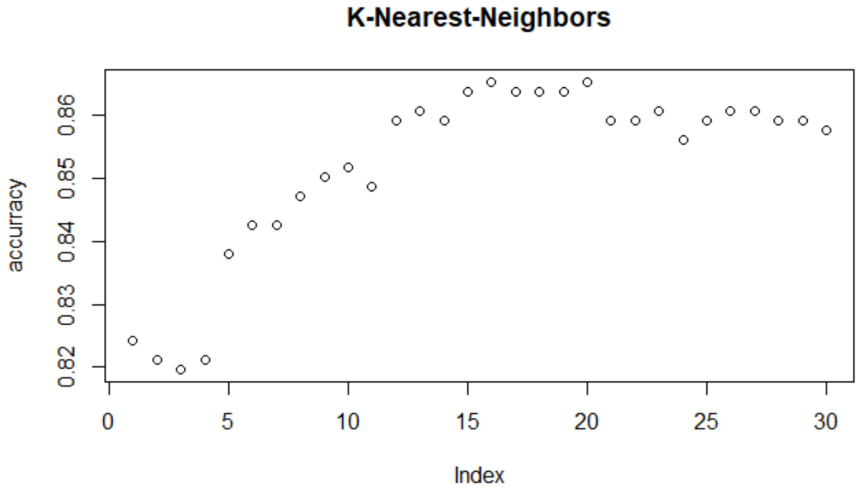
My R code is:

```
# read data
data <- read.delim2("credit_card_data_headers.txt", header = TRUE, sep = "\t", dec = ",")

# access library kernlab
library(kernlab)

# Convert data into a numerical matrix
dataNumericalMatrix <- data.matrix(data[,1:10])
```

```
# Review data with boxplot
BoxPlot = boxplot(scale(dataNumericalMatrix, center = TRUE, scale = TRUE))
BoxPlot

# create a function svmModel and pass the number c
svmModel = function(c){

  # call ksvm.  Vanilladot is a simple linear kernel.
  model <- ksvm(dataNumericalMatrix,as.factor(data[,11]),type="C-svc",
                kernel="vanilladot",C=c,scaled=TRUE)

  # calculate a1...am
  a <- colSums(model@xmatrix[[1]] * model@coef[[1]])

  # calculate a0
  a0 <- -model@b

  # see what the model predicts
  pred <- predict(model,dataNumericalMatrix)

  # see what fraction of the model's predictions match the actual classification
  acc=sum(pred == data[,11]) / nrow(data)
  return(acc)
  }

# set up a vector of 20 zeros to start
accurracy=rep(0,21)

# call svmModel for values of c in
for (i in -10:10){
    accurracy[i+11] = svmModel(10**i)
}

# review accuracies
plot(accurracy, xaxt="none", main="SVM Accuracy with different C")
```

3 Question 2.2 KNN

When using the KNN model to categorize data, I tried different k values, from 1 to 30. And I drew a plot of the accuracy for different k values, the x-axis named Index indicate value of k. As showed in the plot, when $k = 16$ and $k = 20$, the KNN model is the most accurate with an accuracy of 0.8654434. I did some further tests with larger k value, the accuracy were not as high.



Compare the 2 methods - SVM and KNN, they both have similar accuracy (0.8700306 for SVM and 0.8654434 for KNN). Correct level of flexibility is critical, with good value of C and k, both algorithms provide fairly good predictions. In this specific case, SVM takes less time of running to find the best model. SVM is not sensitive to outliers, the model shows the impact of each attributes, and it also has a slightly higher accuracy. on the other hand, KNN is sensitive to outliers, we'll need more information to determine how to deal with outliers in the data if we stick to KNN. KNN also won't directly show which attribute has more impact in the model like SVM does. Between the two of the models, I think SVM is a better choice for this data.

My R code is:

```
# read data
data <- read.delim2("credit_card_data_headers.txt", header = TRUE, sep = "\t", dec = ",")

# access library kknn
library(kknn)

# Convert data into a scaled numerical matrix
#dataNumericalMatrix <- scale(data.matrix(data[,1:10]), center = TRUE, scale = TRUE)

#Setting seed to produce reproducible results
set.seed(1)

# create a function knnModel and pass the number K
knnModel = function(K){

  # predictions: start with a vector of all zeros
  predicted <- rep(0,(nrow(data)))

  # Loop for all data points 1 --> total number of rows in dataset
  for (i in 1:nrow(data)){

    # data[-i] means we remove row i of the data when finding nearest neighbors...
    #...otherwise, it'll be its own nearest neighbor!
```

```
    model = kknn(R1~.,data[-i,],data[i,],k=K, scale = TRUE)

    # record whether the prediction is at least 0.5 (round to 1) or less than 0.5 (round to 0)
    predicted[i] <- as.integer(fitted(model)+0.5) # round off to 0 or 1
  }

  # calculate accuracy of predictions
  acc = sum(predicted == data[,11]) / nrow(data)
  return(acc)
}

# set up a vector of 30 zeros to start
accurracy=rep(0,30)

# call knnModel for values of k from 1 to 30
for (K in 1:30){
  accurracy[K] = knnModel(K)
}

# review accuracies
plot(accurracy)
title("K-Nearest-Neighbors")
```