Homework 1

Question 2.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a classification model would be appropriate. List some (up to 5) predictors that you might use.

In Financial Institutions like banks, the issue of rogue trading in financial markets is hard to detect. It requires oversight from multiple cross-functional teams such as surveillance, risk, compliance, HR, and conduct to build sufficient yet tedious controls around it. Without Big Data capabilities, the process can be extremely manual and inefficient. On top of that, bribery or corruption can be easily exploited and used against manual inspections methods where key data points are concealed under the ever-growing amount of data flowing through the organization.

In this aspect, a soft classification model could be used to assist in traditional inspection methods where extremely low-risk cases can be filtered to allow for more man-hours on high-risk ones. Working in this field, some of the key predictors I might use would include:

1. HR data on Tenure Length, Internal Employee Grade, Region, and even Pay Grade that might determine loyalty, highlight underpaid employees with higher risk, or to track high risk regions.
2. Transactional financial markets data on all trading positions executed by a trader.
3. Line Manager and subordinate data for network analyses on potential rogue trading associations across the team.
4. Conduct data on past performance in the bank.
5. Customer transactional data on suspicious purchase or misuse/transfer of funds across accounts.

Question 2.2.1

Import Packages

```
library(kernlab)
library(kknn)
```

Load data

```
data <- read.table("C:/Users/Admin/Desktop/MM/Homework 1/credit_card_data.txt")
head(data)
```

```
##   V1    V2    V3   V4 V5 V6 V7 V8  V9 V10 V11
## 1  1 30.83 0.000 1.25  1  0  1  1 202   0   1
## 2  0 58.67 4.460 3.04  1  0  6  1  43 560   1
## 3  0 24.50 0.500 1.50  1  1  0  1 280 824   1
## 4  1 27.83 1.540 3.75  1  0  5  0 100   3   1
## 5  1 20.17 5.625 1.71  1  1  0  1 120   0   1
## 6  1 32.08 4.000 2.50  1  1  0  0 360   0   1
```

Model function

```
svm_model <- function(ker, c){
  model <<- ksvm(x = as.matrix(data[,1:10]),
              y = as.factor(data[,11]),
              type = "C-svc",
              kernel = ker,
              C = c,
              scaled = TRUE)
  pred <- predict(model,data[,1:10])
  acc <- (sum(pred == data[,11]) / nrow(data))*100
  print(acc)
}
```

Running 10 models with margins C=100 to 1000 with 100 step

```
for (x in seq(100, 1000, 100)){
  print(paste("For C=", x))
  svm_model("vanilladot", x)
}
```

```
## [1] "For C= 100"
##  Setting default kernel parameters
## [1] 86.39144
## [1] "For C= 200"
##  Setting default kernel parameters
## [1] 86.39144
## [1] "For C= 300"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 400"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 500"
##  Setting default kernel parameters
## [1] 86.39144
## [1] "For C= 600"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 700"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 800"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 900"
##  Setting default kernel parameters
## [1] 86.23853
## [1] "For C= 1000"
##  Setting default kernel parameters
## [1] 86.23853
```

Accuracy at C=100:

```
svm_model("vanilladot", 100)
```

```
##  Setting default kernel parameters
## [1] 86.39144
```

Calculate a1...am and a0 at c=100

```
a <- colSums(model@xmatrix[[1]] * model@coef[[1]])
a0 <- model@b
```

V1-V10: -0.0010065, -0.0011729, -0.0016262, 0.0030064, 1.0049406, -0.0028259, 2.6002951×10^{-4}, $-5.3495514 \times 10^{-4}$, -0.0012284, 0.1063634

V0: -0.0815849

Equation of classifier: -0.0010065348V1 − 0.0011729048V2 − 0.0016261967V3 + 0.0030064203V4 + 1.0049405641V5 − 0.0028259432V6 + 0.0002600295V7 − 0.0005349551V8 − 0.0012283758V9 + 0.1063633995V10 + 0.08158492 = 0

Question 2.2.2

Using Radial Basis Kernel in SVM

Accuracy using Radial Basis Function kernel for svm:

```r
svm_model("rbfdot", 100)
```

```
## [1] 95.25994
```

Using Polynomial Kernel in SVM

Accuracy using Radial Basis Function kernel for svm:

```r
svm_model("polydot", 100)
```

```
##  Setting default kernel parameters
## [1] 86.39144
```

Question 2.2.3

Model function

```
kknn_model <- function(k){
  pred <- list()
  for (i in 1:nrow(data)){ # every row in data
    model <- kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10, #v11 a function of v1-10
                  data[-i,], # using all other data point to train
                  data[i,], # train every data point, every row
                  k = k,
                  scale = TRUE)
    pred[i] <- as.integer(round(fitted(model))) # predictions given 0-1, round predictions to either 0
  }

  acc <- sum(pred == data[,11]) / nrow(data) * 100 # compare with dataset to get accuracy
  return(acc)
}
```

Running model with 1-20 k-values

```
acc_list = list()
for (i in 1:20){ # running model with 1-20 k-values
  acc_list[i] <- kknn_model(i)  # list of accuracy for every k-value
}
acc_list
```

```
## [[1]]
## [1] 81.49847
##
## [[2]]
## [1] 81.49847
##
## [[3]]
## [1] 81.49847
##
## [[4]]
## [1] 81.49847
##
## [[5]]
## [1] 85.1682
##
## [[6]]
## [1] 84.55657
##
## [[7]]
## [1] 84.70948
##
## [[8]]
## [1] 84.86239
##
## [[9]]
## [1] 84.70948
```

```
##
## [[10]]
## [1] 85.01529
##
## [[11]]
## [1] 85.1682
##
## [[12]]
## [1] 85.3211
##
## [[13]]
## [1] 85.1682
##
## [[14]]
## [1] 85.1682
##
## [[15]]
## [1] 85.3211
##
## [[16]]
## [1] 85.1682
##
## [[17]]
## [1] 85.1682
##
## [[18]]
## [1] 85.1682
##
## [[19]]
## [1] 85.01529
##
## [[20]]
## [1] 85.01529
```

KNN-Classifier performs best at k=12 with an accuracy of 85.3211009.