HW ISYE6501x Question 3.1 Part B - Week 1

A. Tse

May 15, 2018

Part B

Loading the Libraries

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

Reading the Dataset

```
credit_card <- read.csv("credit_card_data_headers.csv", header = TRUE)
head(credit_card)
```

```
##   A1    A2    A3   A8 A9 A10 A11 A12 A14 A15 R1
## 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
```

Looking at Number of Rows and Columns in the Dataset

```
nrow(credit_card)
```

```
## [1] 654
```

```
ncol(credit_card)
```

```
## [1] 11
```

RNG

```
set.seed(1452)
```

Converting to Numeric Data

```
for(s in 1:10) {
  credit_card[,s] <- as.numeric(as.character(credit_card[,s]))
}
credit_card$R1 <- as.factor(credit_card$R1)
```

Splitting the Datasets to the Training, Testing and Validation Sets

```
validex <- createDataPartition(credit_card$R1, p=0.70, list=FALSE)
val <- credit_card[-validex,]
```

```
dummi <- credit_card[validex,]
tesx <- createDataPartition(dummi$R1, p=0.85, list=FALSE)
credit_test <- dummi[-tesx,]
credit_train <- dummi[tesx,]

X_train <- credit_train[,1:10]
X_train <- as.matrix(X_train)
y_train <- as.numeric(credit_train$R1)
X_test <- credit_test[,1:10]
X_test <- as.matrix(X_test)
y_test <- as.numeric(credit_test$R1)
```

Building the Last SVM Model

```
support_mod <- ksvm(X_train, y_train, type="C-svc", kernel="rbf",
sigma=0.025, C=10)

valid_X <- as.matrix(val[,1:10])

Yval <- as.numeric(val$R1)

SupVecM <- predict(support_mod, valid_X)

conf_matr <- table(Yval, SupVecM)
conf_matr

##      SupVecM
## Yval  1  2
##     1 97 10
##     2 14 74

acru <- (sum(diag(conf_matr))/sum(conf_matr))
acru

## [1] 0.8769231
```

Building the Last KNN Model

```
knn_mod <- kknn(R1 ~., train = credit_train, test = val, scale = TRUE,
k = 40)

knn_fit <- fitted(knn_mod)

conf_matr <- table(val$R1, knn_fit)
conf_matr

##    knn_fit
##      0  1
##   0 98  9
##   1 19 69
```

```
acru <- (sum(diag(conf_matr))/sum(conf_matr))
acru

## [1] 0.8564103
```

By splitting the datasets to train, test and cross-validation, both the models for KNN and SVM don't differ much in accuracy at 87.69% and 85.64%, and it is done at 70% splitting for training and using 15% for testing and validation. k is set at 40 for the KNN model. And by setting C = 10 and sigma = 0.025, the accuracy is the most optimal. In comparision to setting it at C=100 and sigma = 0.5, there is a sight difference.