# 2018R2 Data Mining (STAT5104) Assignment 2

*Yiu Chung WONG 1155017920*

```r
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
set.seed(17920)
```

## Q1

### a)

```r
d <- read.csv("tele.csv", header = TRUE, sep = ",")           #read data
d[,18] <- factor(d$Change,
                 levels = c(0, 1),
                 labels = c("stay", "change"))

inTrain <- caret::createDataPartition(d$Change,               #create index for train / test partition
                                      p = .8,
                                      list = FALSE)


d0 <- d[inTrain,]                                              #select observations for training
d1 <- d[-inTrain,]                                             #select observations for testing
```

### b)

```r
continuous <- c(5, 7, 8, 10, 11, 13, 14, 16)      #Column index for continuous variables
x0 <- d0[, continuous]                             #Select continuous variables for training data
x1 <- d1[, continuous]                             #Select continuous variables for testing data

tele.knn <- k_nn(x0, x1, d0[,18], d1[,18], v = 10)#KNN
```

```
## k= 1  error rate= 0.1887367
## k= 2  error rate= 0.2146119
## k= 3  error rate= 0.1415525
## k= 4  error rate= 0.1506849
## k= 5  error rate= 0.130898
## k= 6  error rate= 0.1232877
## k= 7  error rate= 0.1263318
## k= 8  error rate= 0.1248097
## k= 9  error rate= 0.1248097
## k= 10  error rate= 0.129376
## best k= 6  error rate= 0.1232877
```

```r
classification.table <- table(tele.knn, d1[,18])
err.rate <- (classification.table[1,2] + classification.table[2,1])  / sum(classification.table)
classification.table
```

```
##
## tele.knn stay change
##   stay    545     64
##   change   17     31
```

The error rate is (64 + 17) / 657 = 0.1232877

**c)**

```r
x0 <- d0[, -c(2, 3, 18)]#Extract continuous and integer variables (except columns 2 and 3) for training
x1 <- d1[, -c(2, 3, 18)]#Extract continuous and integer variables (except columns 2 and 3) for testing

tele.knn <- k_nn(x0, x1, d0[, 18], d1[, 18], v = 10)  #KNN
```

```
## k= 1  error rate= 0.194825
## k= 2  error rate= 0.2009132
## k= 3  error rate= 0.1385084
## k= 4  error rate= 0.1445967
## k= 5  error rate= 0.130898
## k= 6  error rate= 0.1187215
## k= 7  error rate= 0.1217656
## k= 8  error rate= 0.1187215
## k= 9  error rate= 0.1202435
## k= 10  error rate= 0.1171994
## best k= 10  error rate= 0.1171994
```

```r
classification.table <- table(tele.knn, d1[,18])
err.rate <- (classification.table[1,2] + classification.table[2,1])  / sum(classification.table)
classification.table
```

```
##
## tele.knn stay change
##   stay    555     70
##   change    7     25
```

The error rate is (70 + 7) / 657 = 0.1171994

**d)**

```r
z0 <- scale.con(x0)                                         #Scale training data
z1 <- scale.con(x1)                                         #Scale testing data

tele.knn <- k_nn(z0, z1, d0[,18], d1[,18], v = 10)          #KNN
```

```
## k= 1  error rate= 0.1872146
## k= 2  error rate= 0.1796043
## k= 3  error rate= 0.1263318
## k= 4  error rate= 0.130898
```

```
## k= 5  error rate= 0.1004566
## k= 6  error rate= 0.1004566
## k= 7  error rate= 0.0913242
## k= 8  error rate= 0.08980213
## k= 9  error rate= 0.0913242
## k= 10  error rate= 0.09284627
## best k= 8  error rate= 0.08980213
```

```r
classification.table <- table(tele.knn, as.factor(d1[,18]))
err.rate <- (classification.table[1,2] + classification.table[2,1])  / sum(classification.table)
classification.table
```

```
##
## tele.knn stay change
##   stay    554     51
##   change    8     44
```

The error rate is (51 + 8) / 657 = 0.0898021

### e)

- Error rate is the highest when only continuous variables are accounted for; lower when integer variables are includes; and lowest when variables are scaled.
- Error rate of c) is lower than b) because the integer variables are inherently correlated to the outcome. Hence adding them as predictor increases the number of criteria (dimention) of distance measure, which helps distinguishing the dissimilarities of any two observations.
- In c), variables have varying degrees of variance. For example, `Day_Mins` ranges from 7.8 to 350.8; whereas `Intl_Charge` ranges from 0 to 5.4. Since KNN typically utilises Euclidian distance as measure of dissimilarities, distances measure between neighbors are biased towards variables with higher degree of variance: a small change in a highly varying variable have a greater effect than a big change in a low varying variable. Scaling helps overcome this by normalising variance of each variable. So no single variable dominate distance mesaure. This explains lower error rate in d).

## Q2

### a)

```
x <- d[, continuous]            #Extract all the continuous variables in d
c2 <- as.factor(d$Intl_plan)    #Convese Intl_plan to category variable
c3 <- as.factor(d$Vmail_plan)   #Convese Vmail_plan to category variable
```

### b)

```
dc <- cbind(c2, c3, x)     #Combine c2, c3 and x to form a matrix dc
d0 <- dc[inTrain,]         #Extract observations for training
c0 <- d$Change[inTrain]    #Extract training label
d1 <- dc[-inTrain,]        #Extract observations for testing
c1 <- d$Change[-inTrain]   #Extract testing label
```

### c)

```
tele.nb<-e1071::naiveBayes(d0, c0)        #Naive Bayes
pr<-predict(tele.nb,d1)                   #Predict

classification.table <- table(pr,c1)
err.rate <- (classification.table[1,2] + classification.table[2,1])  / sum(classification.table)
classification.table
```

```
##         c1
## pr        stay change
##   stay     548     55
##   change    14     40
```

The error rate is (55 + 14) / 657 = 0.1050228

### d)

- The error rate of Naive Bayes approach is slightly higher than KNN with scaled variables, but lower than non-scaled KNN.
- Without having to scale data, Naive Bayes performs reasonably well compared to KNN.
- Naive Bayes does not require scaling because
    1. the priors are set based on the training data, so it will also scale those priors to match trainning data.
    2. Naive Bayes does not rely on distance measure; scaling have little effect on the computation of the posterior probability.