Homework Week 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.

Response:

- As an HR analyst, it is very important for the business to identify what factors will make employees more likely to terminate prematurely. For which I will use a classification model that will separate employees in two different categories “likely to stay” and “likely to leave” based on the following:
    - Compensation
    - Job Type
    - Average weekly hours
    - Average commute time
    - Performance
    - Tenure
    - Promotions or progressions
    - Terminated (Classification)

Question 2.2 The files credit_card_data.txt (without headers) and credit_card_data-headers.txt (with headers) contain a dataset with 654 data points, 6 continuous and 4 binary predictor variables. It has anonymized credit card applications with a binary response variable (last column) indicating if the application was positive or negative. The dataset is the “Credit Approval Data Set” from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Credit+Approval) without the categorial variables and without data points that have missing values.

1. Using the support vector machine function ksvm contained in the R package kernlab, find a good classifier for this data. Show the equation of your classifier, and how well it classifies the data points in the full data set. (Don’t worry about test/validation data yet; we’ll cover that topic soon.)

Response: This question can have multiple responses. With a classifier of 50 the model predicts 565 points with 86.4% accuracy.

Classifier Equation: -0.0010065348z1 - 0.0011729048z2 - 0.0016261967z3 + 0.0030064203z4 + 1.0049405641z5 - 0.0028259432z6 + 0.0002600295z7 - 0.0005349551z8 - 0.0012283758z9 + 0.1063633995z10 + 0.08158492 = 0

Code:

```
#Set ksvm model1
library(kernlab)
model1 <- ksvm(as.matrix(data[ ,1:10]), as.factor(data[ ,11]), type = "C-svc",
kernel="vanilladot", C=50, scaled = TRUE)
# calculate a1…am
a <- colSums(data[model1@SVindex,1:10] * model1@coef[[1]])
a
# calculate a0
a0 <- sum(a*data[1,1:10])- model1@b
```

```
a0
# see what the model1 predicts
pred <- predict(model1,data[,1:10])

#pred
# see what fraction of the model1’s predictions match the actual classification
sum(pred == data[,11]) / nrow(data)

#General Summary
model1

#Explore attributes:
#attributes(model1)

# For example, the support vectors
alpha(model1)
alphaindex(model1)
b(model1)
```

2. You are welcome, but not required, to try other (nonlinear) kernels as well; we’re not covering them in this course, but they can sometimes be useful and might provide better predictions than vanilladot.

Response: The best classifier with a Gaussian Kernel(**“rbfdot”**) is able to predict values with a cost of is C = 50 at 93.6% percent accuracy.

3. Using the k-nearest-neighbors classification function kknn contained in the R kknn package, suggest a good value of k, and show how well it classifies that data points in the full data set. Don’t forget to scale the data (scale=TRUE in kknn).

Response: The best classifier I could find is K = 10 it has 88.2% percent accuracy. I choose this classifier because it was a recommendation made on Jalayer Academy to pick the square root of the total data points.

Code:

```
library("kknn")

#Import data without headers. (headers seem to be a problem when running the KKNN
model)

credit<-
read.table("https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/39b78ff5c5c289
81f009b54831d81649/asset-
v1:GTx+ISYE6501x+2T2017+type@asset+block/credit_card_data.txt")

# Mix the rows of the dataframe in order for the model to be able to predict the
R1/V11
#set.seed to be able to replicate results according to some of the comments of the
students.

set.seed(9850)
```

```
#runif() produces a random number for a uniform distribution.

gp<- runif(nrow(credit))

credit<- credit[order(gp),]

#Training will have 90% of the data 588 aplicants and 67 for de validation set

credit.learn<- credit[1:588, ]
credit.valid<- credit[588:654, ]

for(i in credit.learn)
  response.learn = credit.learn[-i,11]
attrs.learn = credit.learn[-i,1:10]

for(i in credit.valid)
  response.valid = credit.valid[-i,11]
attrs.valid = credit.valid[-i,1:10]

for(i in credit)
  response = credit[-i,11]
attrs = credit[-i,1:10]

model2.tknn <- train.kknn(response.learn~., data = attrs.learn, ks = 10, scale =
TRUE)

model2.tknn.q2 <- train.kknn(response~., data = attrs, ks = 10, scale = TRUE)

pred2 <- round(predict(model2.tknn, attrs.valid))

pred2.q2 <- round(predict(model2.tknn.q2, attrs))

#Compare the prediction with the value:

testresults <- sum(pred2 == response.valid)
resultprct <- testresults/ length(pred2)

testresults.q2 <- sum(pred2.q2 == response.learn)
resultprct.q2 <- testresults.q2/ length(pred2.q2)

resultprct
resultprct.q2
# Answer Q3:
# The best classifier I could find is K = 10 it has 88.2% percent accuracy. I choose
this classifier
# because it was a recomendation made on Jalayer Academy to pick the sqare root of
the total datapoints.
```