ISYE-6501
Fall 2024

Homework 2

Question 3.1

Using the same data set (credit_card_data.txt or credit_card_data-headers.txt) as in Question 2.2, use the ksvm or kknn function to find a good classifier:
(a) using cross-validation (do this for the k-nearest-neighbors model; SVM is optional); and

SOLUTION AND ANALYSIS:

After running cross validation on k 1 through 30 and calculating their accuracies, it was found that the optimal k value is k=16 with an accuracy of 85.6%.
The code input can be found in black and the code output or results can be found in blue.

R CODE:

```
> library(kknn)
> data_df <- read.table("credit_card_data.txt", header=FALSE)
> head(data_df, 10)
   V1    V2     V3    V4 V5 V6 V7 V8  V9   V10 V11
1   1 30.83  0.000 1.250  1  0  1  1 202     0   1
2   0 58.67  4.460 3.040  1  0  6  1  43   560   1
3   0 24.50  0.500 1.500  1  1  0  1 280   824   1
4   1 27.83  1.540 3.750  1  0  5  0 100     3   1
5   1 20.17  5.625 1.710  1  1  0  1 120     0   1
6   1 32.08  4.000 2.500  1  1  0  0 360     0   1
7   1 33.17  1.040 6.500  1  1  0  0 164 31285   1
8   0 22.92 11.585 0.040  1  1  0  1  80  1349   1
9   1 54.42  0.500 3.960  1  1  0  1 180   314   1
10  1 42.50  4.915 3.165  1  1  0  0  52  1442   1
> Count <- nrow(data_df)
> list1 <- list()
> for (i in 1:30)
+ {
+   model <- cv.kknn(V11~.,
+                 data_df,
+                 kcv=10,
+                 k=i,
+                 kernel="optimal",
+                 distance = 2,
+                 scale=TRUE)
+   answer<-as.integer(model[[1]][,2]+0.5)
+   list1[i]<-sum(answer==data_df$V11)/Count
+ }
```

```
> optimal_k <- which.max(list1)
> cat("Best k=", optimal_k, "\n")
Best k= 16
> print(list1[optimal_k])
[[1]]
[1] 0.8562691
```

(b) splitting the data into training, validation, and test data sets (pick either KNN or SVM; the other is optional).

SOLUTION AND ANALYSIS:
After initiating the training and validation data using 60% and 20% of the data respectively, it was found that the best k=10. I then ran the model on the test data which was 20% of the remaining unused data.
Running the model on the test data using the best k=10 was found to give an accuracy of 92.3%
The code input can be found in black and the code output or results can be found in blue.

R CODE:

```
> library(kknn)
> data_df <- read.table("credit_card_data.txt", header=FALSE)
> head(data_df, 10)
   V1    V2     V3    V4 V5 V6 V7 V8  V9   V10 V11
1   1 30.83  0.000 1.250  1  0  1  1 202     0   1
2   0 58.67  4.460 3.040  1  0  6  1  43   560   1
3   0 24.50  0.500 1.500  1  1  0  1 280   824   1
4   1 27.83  1.540 3.750  1  0  5  0 100     3   1
5   1 20.17  5.625 1.710  1  1  0  1 120     0   1
6   1 32.08  4.000 2.500  1  1  0  0 360     0   1
7   1 33.17  1.040 6.500  1  1  0  0 164 31285   1
8   0 22.92 11.585 0.040  1  1  0  1  80  1349   1
9   1 54.42  0.500 3.960  1  1  0  1 180   314   1
10  1 42.50  4.915 3.165  1  1  0  0  52  1442   1
> Count <- nrow(data_df)
> list1 <- list()
> traindata<-data_df[1:ceiling(Count*0.6),]
> validdata<-data_df[(ceiling(Count*0.6)+1):ceiling(Count*0.8),]
> testdata<-data_df[(ceiling(Count*0.8)+1):Count,]
> for (i in 1:30)
+ {
+   model <- kknn(V11~.,
+               traindata,
+               validdata,
+               k=i,
+             kernel="optimal",
+             distance = 2,
+             scale=TRUE)
+  answer<-as.integer(fitted(model)+0.5)
+  list1[i]<-sum(answer==validdata$V11)/nrow(validdata)
+ }
> optimal_k <- which.max(list1)
> cat("Best train k=", optimal_k, "\n")
Best train k= 10
> model <- kknn(V11~.,
+            traindata,
+            testdata,
+            k=optimal_k,
+            kernel="optimal",
+            distance = 2,
+            scale=TRUE)
> answer<-as.integer(fitted(model)+0.5)
> list1[i]<-sum(answer==testdata$V11)/nrow(testdata)
> optimal_k <- which.max(list1)
> accuracy<-as.numeric(list1[optimal_k])
> accuracy<-accuracy*100
> cat("Accuracy using best k on test data is", accuracy, "%")
Accuracy using best k on test data is 92.30769 %
```

Question 4.1

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

As a quality data analyst for a hospital, a clustering model can be used to determine which nursing unit a patient should be admitted to based on the following predictors:

- Patient Medical History (chronic conditions, surgical history)
- Current Medical Status (primary diagnosis, vital signs, lab results)
- Care Needs (specialized equipment, medication requirements)
- Patient Demographics (age, gender, insurance payer)
- Hospital Resources (bed availability, staff availability)
- Risk Factors (risk of infection, fall, or pressure injury)

Question 4.2

The iris data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository. The

response values are only given to see how well a specific method performed and should not be used to build the model.

Use the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

SOLUTION and ANALYSIS:

After running the kmeans algorithm, I found that the best k value and predictor combination would be k=3 with using the predictor petal length/width because the three flowers were split the most by three clusters with 50/50 setosa in cluster 3, 48/50 versicolor in cluster 1, and 46/50 virginica in cluster 2. The matrix with k=3 using the petal predictor can be found in green in the code below.

The code input can be found in black and the code output or results can be found in blue.

R CODE:

```
> library(kknn)
> iris_df <- read.table("iris.txt", header=TRUE)
> head(iris_df, 10)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1         5.1        3.5         1.4         0.2  setosa
2         4.9        3.0         1.4         0.2  setosa
3         4.7        3.2         1.3         0.2  setosa
4         4.6        3.1         1.5         0.2  setosa
5         5.0        3.6         1.4         0.2  setosa
6         5.4        3.9         1.7         0.4  setosa
7         4.6        3.4         1.4         0.3  setosa
8         5.0        3.4         1.5         0.2  setosa
9         4.4        2.9         1.4         0.2  setosa
10         4.9        3.1         1.5         0.1  setosa
> iris_scaled_df = iris_df
> for (i in 1:4) { iris_scaled_df[,i] <- (iris_df[,i]-min(iris_df[,i]))/(max(iris_df[,i])-min(iris_df[,i])) }
> iris_cluster_sepal_1 = kmeans(iris_scaled_df[,1:2], 2, nstart = 20)
> iris_cluster_sepal_2 = kmeans(iris_scaled_df[,1:2], 3, nstart = 20)
> iris_cluster_sepal_3 = kmeans(iris_scaled_df[,1:2], 4, nstart = 20)
> iris_cluster_sepal_4 = kmeans(iris_scaled_df[,1:2], 5, nstart = 20)
> table(iris_cluster_sepal_1$cluster, iris_df$Species)
   setosa versicolor virginica
  1     50         7        1
  2      0        43       49
> table(iris_cluster_sepal_2$cluster, iris_df$Species)
   setosa versicolor virginica
  1   49       0      0
  2    1      37     16
  3    0      13     34
> table(iris_cluster_sepal_3$cluster, iris_df$Species)
     setosa versicolor virginica
  1    0      10     30
  2    0      33     19
  3   33       1      0
  4   17       6      1
> table(iris_cluster_sepal_4$cluster, iris_df$Species)
    setosa versicolor virginica
  1    1      29      9
  2   25       0      0
  3    0       2     16
  4   24       0      0
  5    0      19     25
> iris_cluster_petal_1 = kmeans(iris_scaled_df[,3:4], 2, nstart = 20)
> iris_cluster_petal_2 = kmeans(iris_scaled_df[,3:4], 3, nstart = 20)
> iris_cluster_petal_3 = kmeans(iris_scaled_df[,3:4], 4, nstart = 20)
> iris_cluster_petal_4 = kmeans(iris_scaled_df[,3:4], 5, nstart = 20)
> table(iris_cluster_petal_1$cluster, iris_df$Species)
   setosa versicolor virginica
  1   50       0      0
  2    0      50     50
> table(iris_cluster_petal_2$cluster, iris_df$Species)
   setosa versicolor virginica
  1    0      48      4
  2    0       2     46
  3   50       0      0
> table(iris_cluster_petal_3$cluster, iris_df$Species)
   setosa versicolor virginica
  1    0      42      0
  2    0       8     23
  3    0       0     27
  4   50       0      0
> table(iris_cluster_petal_4$cluster, iris_df$Species)
    setosa versicolor virginica
  1    0       0     25
  2    0       2     21
  3    0      23      0
  4    0      25      4
  5   50       0      0
```