MATH 4323, Fall 2023, Homework # 2.

Name: Hyundoo Jeong
#PSID: 2212332

Instructions: Submit the solutions as a file (type it up and save as a *.pdf* or a *Word*-file, no hand-written solutions) via UH Blackboard. Keep responses brief and to the point. For code & output: include only pieces that are of utmost relevance to the question.

Conceptual.

1. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable (Y = color "Blue" or color "Red").

Obs.	X_1	X_2	X_3	Y
1	1	2	-1	Blue
2	1	0	3	Blue
3	0	3	0	Red
4	0	2	-1	Red
5	2	0	-1	Blue
6	1	4	1	Red

Suppose we wish to use K-nearest neighbors to work with this data set.

(a) Use $K = 1$ to predict the color (either "Blue" or "Red"), using the K-nearest neighbor predictions of color for each of the training observations. What is the average misclassification training error?

Answer: Training error rate is 0/6 = 0% because when k = 1, all of them are predicted as blue so average misclassification is 0%.

(b) Use $K = 3$ to calculate the K-nearest neighbor predictions for all the training observations. What is the average misclassification training error?

Answer: When K = 3, 3 nearest neighbors were chosen as Obs 1,4,3. Their labels are Blue and Red and Red. So training error is 1/3 which is 0.33%.

(c) Suppose we wish to use this data set to make a prediction for Y when $x_0 = (X_1, X_2, X_3)$, with $X_1 = X_2 = X_3 = 0$, using K-nearest neighbors.

i. Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

Answer: Obs1 has Euclidean distance:2.44949.

Obs2 has Euclidean distance:3.162278.

Obs3 has Euclidean distance:3.

Obs4 has Euclidean distance:2.236068.

Obs5 has Euclidean distance:2.236068.

Obs6 has Euclidean distance:4.242641.

ii. What is our prediction with $K = 1$? Why?

Answer: It could be either "Red" or "Blue" because Obs4 and Obs5 are nearest neighbors.

iii. What is our prediction with $K = 3$? Why?

Answer: Obs1 and Obs4 and Obs5 are chosen as prediction with K = 3. Therefore, it could be "Blue".

iv. What is our prediction with $K = 5$? Why?

Answer: Obs1,2,3,4,5 is chosen as prediction with K = 5. Therefore, it could be "Blue".

2. We now review the topic of model validation:

(a) Explain how validation set approach is implemented. What are its main disadvantages?

Answer: The validation set approach is a method used to estimate the out-of-sample-error by creating a separate validation set from the original data set. So summarize, we need to split data splitting into two parts: Training set, Validation set. Next, we fitted the model and trained, and used model to predict validation data. Then, calculating validation error and rate test error rate based on calculations of validation error.

Disadvantages: The validation estimate of test error can be highly variable, which depends on which observations are in the training and validation sets.

(b) Explain how the leave-one-out cross-validation is implemented (LOOCV). How does it improve on validation set approach?

Answer: LOOCV is a method used to estimate the performance of a prediction model.

First, we split the data into 2 parts: Training set, Test set. Use training model and predict y. Then, calculate the misclassification error. The way of improvement of

LOOCV is K-fold Cross -Validation is alternative an LOOCV.

(c) What is the main disadvantage of LOOCV approach? Hint: Imagine we have 10000 observations and 20 variables.

Answer: Performing LOOCV on datasets with many observations and variables requires significant computational resources in terms of CPU time and memory such as computationally demanding.

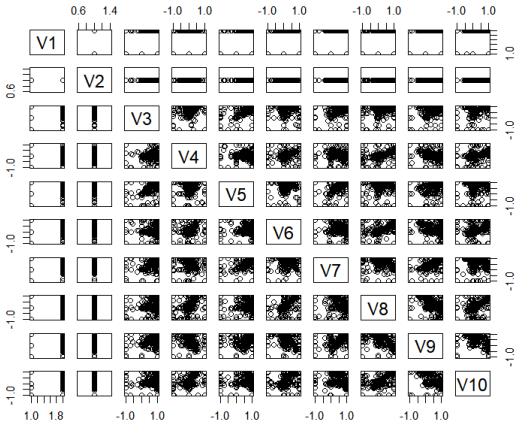
(d) Could we use validation set & LOOCV for any supervised learning approach? Or only for KNN?

Answer: Yes, we can use both the validation set and LOOCV for evaluating and assessing the performance of a wide range of supervised learning approaches.

Applied.

3. This question should be answered using the *Ionosphere* data set, which is part of the *mlbench* package. This radar data was collected by a system in Goose Bay, Labrador. The data frame consists of 351 observations on 35 independent variables. The last column in the dataframe is a categorical variable, ”Class”, defining the free electrons in the ionosphere: ”good” radar returns are those showing evidence of some type of structure in the ionosphere. ”bad” returns are those that do not.

   (a) Produce some numerical and graphical summaries of the *Ionosphere* data. Do there appear to be any patterns?



   (b) Notice that the second the column contains only one single value, so remove that column and work on the rest of the questions using the new dataset. Perform a *K*-Nearest Neighbors (KNN) algorithm with $K = 1$, where *Class* is the response, and the rest columns in the dataset as predictors.

```
[1] good bad  good bad  good bad
Levels: bad good
> head(data.frame(knn.pred,y.train))
  knn.pred y.train
1     good    good
2      bad     bad
3     good    good
4      bad     bad
5     good    good
6      bad     bad
> mean(knn.pred != y.train)
[1] 0
>
```

   (c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by KNN algorithm.

   Answer: This matrix is tested for up and down predictions Indicates that the data is not misclassified. A misclassification error occurs if there is a non-zero

value in the lower left or upper right corner.

(d) Split the data *randomly* into a training set (70%) and a test set (30%). Make sure to use *set.seed*(4323), for reproducible results. Fit the KNN model (K = 3)

Answer:

```
>
> set.seed(4323)
> n<-nrow(Ionosphere)
> train<-sample(1:n,0.7*n)
> knn.pred<-knn(train = X.train, test = X.test, cl = y.train, k = 3)
> head(knn.pred)
[1] good bad  good bad  good bad
Levels: bad good
> table(knn.pred,y.train)
        y.train
knn.pred bad good
    bad  104    2
    good  22  223
>
```

(e) Repeat (d) using K = 5.

```
> <-nrow(Ionosphere)
Error: unexpected assignment in "<-"
> n<-nrow(Ionosphere)
> train<-sample(1:n,0.7*n)
> knn.pred<-knn(train = X.train, test = X.test, cl = y.train, k = 5)
> table(knn.pred,y.train)
        y.train
knn.pred bad good
    bad  101    2
    good  25  223
>
```

(f) Repeat (d) using K = 7.

```
> set.seed(4323)
> n<-nrow(Ionosphere)
> train<-sample(1:n,0.7*n)
> knn.pred<-knn(train = X.train, test = X.test, cl = y.train, k = 7)
> table(knn.pred,y.train)
        y.train
knn.pred bad good
    bad   89    2
    good  37  223
>
```

(g) Which of these methods appears to provide the best results on this data?

I got correct prediction of each K= 3,5,7. Those of prediction is significantly bad because every result is close to the 1 which mean bad predictions. However, I got 0.931 when k = 3, and I got 0.923 when k = 5, and I got 0.888 when k = 7. So those of three, k = 7 is the best results on this data.

4. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the *Auto* data set.

(a) Create a binary variable, *mpg*01 , that contains a 1 if *mpg* contains a value above its median, and a 0 if *mpg* contains a value below its median. You can compute the median using the *median*() function. Note you may find it helpful to use the *data.frame*() function to create a single data set containing both *mpg*01 and the

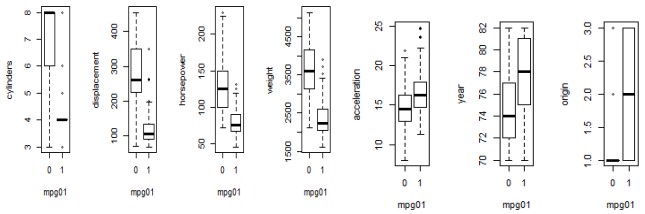
other *Auto* variables.

Answer:

```
> Auto$mpg01<-factor(Auto$mpg01)
> names(Auto)
 [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"       "acceleration"
 [7] "year"         "origin"       "name"         "mpg01"
>
```

(b) Explore the data gradually in order to investigate the association between mpgo1 and the other features. Which of the other feature seem most likely to be useful in predicting mpgOl? Sc6tterplot- and boxplot- may be useful tools to answer this question. Describe your findings.

Answer:



(c) Split the data randomly into a training set(70%) and a test set(30%). Make sure to use set.seed(1), for reproducible results.

Answer:
```
> set.seed(1)
> n<-nrow(Auto)
> train<-sample(1:n,0.7*n)
```

(d) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

Answer: I got 0.1355932, 0.1101695,0.1440678 when K = 1,3,10. So when k = 3, test error is the smallest so k = 3 is best on this data set.

(e) Are the predictor- you included into KNN model on the same scale? Proceed to scale the train & test data form parts (c) – (d) as we did in the lab. Only use the predictions that you claimed to be useful in explaining mpg01. Repeat part (d) for the scaled data.

Answer: I got 0.1271186,0.1016949,0.05932203 when K=1,3,10. Smallest value is 0.05932203 when k = 10. So K = 10 is the best number to use this model.

5) For problem 4, instead of……...

(a), Answer: I got the same value compared to problem 4 even though I used set.seed(1). So, it's 0.1271186, 0.1016949, 0.05932203 when k = 1,3,10.

(b), Answer: > cv.pred <- knn.cv(Auto[, c("cylinders","displacement", "horsepower","weight")],
+ Auto$mpg01,
+ k=1)
> mean (cv.pred != Auto$mpg01)
[1] 0.1352041
Which method win? LOOCV method win.

(c), Answer: >X.train.scaled <- scale(X.train)
> X.test.scaled <- scale(X.test,
+ center = attr(X.train.scaled, "scaled:center"),
+ scale = attr(X.train.scaled, "scaled:scale"))
> knn.pred <- knn(X.train.scaled, X.test.scaled, y.train, k=10)
> mean(y.test != knn.pred)
[1] 0.05084746
LOOCV method win too.

(D) Answer: The validation set approach is more reliable because it exposes you to more data sets.It also trains more different data inputs and applications.