API 222 Problem Set 2

Machine Learning and Big Data Analytics: Spring 2024

Due at 11:59am on March 7 - submit on Gradescope

This problem set is worth 30 points in total. To get full credit, submit your code along with a write-up of your answers. This should either be done in R Markdown or Jupyter Notebook, submitted in one knitted PDF.

Final Project Groups (0 pts)

Please join one of 30 final project groups that have been created on this this Canvas page. Details about the final project can be found on this Canvas page. You just need to form your group by March 7. The main project milestones will be after the midterm. We recommend forming groups of 5 students. All students working together should join the same group. PhD students need to work individually (see details on Canvas). If you are a PhD student or are otherwise working alone, please form a group by yourself. Please email Jacob if you have questions about this assignment or the final project.

Conceptual Questions (15 pts)

1. Consider the four main classification methods that have been presented thus far this semester: logistic regression, k-Nearest Neighbors, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA. Which of these methods may be appropriate if you know the decision boundary between the classes is linear? (3pts)

LDA and logistic regression are appropriate if the decision boundary is linear.

2. Suppose you had the following data and you are using KNN Regression with Euclidean distance. Consider the prediction problem where you want to predict Y for the data point X1 = X2 = X3 = 0.

X1	X2	X3	Y
0	3.5	2	2
1	2.1	3	1
2	4.7	1	3
1	3.9	1	2
0	2.9	2	4
1	1.5	2	1
1	3.5	4	2

(a) Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0. (1pt)

```r
# Data
data <- data.frame(X1 = c(0, 1, 2, 1, 0, 1, 1),
                   X2 = c(3.5, 2.1, 4.7, 3.9, 2.9, 1.5, 3.5),
                   X3 = c(2, 3, 1, 1, 2, 2, 4),
                   Y = c(2, 1, 3, 2, 4, 1, 2))

# compute distance from (0, 0, 0)
data$distances <- sqrt((data$X1 - 0)^2 + (data$X2 - 0)^2 + (data$X3 - 0)^2)
data
```

```
##   X1  X2 X3 Y distances
## 1  0 3.5  2 2  4.031129
## 2  1 2.1  3 1  3.796051
## 3  2 4.7  1 3  5.204805
## 4  1 3.9  1 2  4.148494
## 5  0 2.9  2 4  3.522783
## 6  1 1.5  2 1  2.692582
## 7  1 3.5  4 2  5.408327
```

(b) What is your prediction with K = 2? Why? (1pt)

```r
# KNN with K = 2
knn <- data[order(data$distances), ][1:2, ]
mean(knn$Y)
```

```
## [1] 2.5
```

(c) If the Bayes decision boundary in this problem is highly nonlinear, then would we expect the best value for K to be large or small? Why? (1pt)

If the Bayes decision boundary is highly nonlinear, we would expect the best value for K to be small. This is because a small value of K will be more flexible and will be able to capture the nonlinearity in the data.

3. Consider we conduct a research study analyzing the risk factors for developing prostate cancer among men, with variables X_1 = age (years), X_2 = family history of prostate cancer (0 = no, 1 = yes), X_3 = smoking status (0 = non-smoker, 1 = smoker), and Y = probability of developing prostate cancer. A logistic regression analysis is performed, resulting in estimated coefficients $\hat{\beta}_1 = 0.06$, $\hat{\beta}_2 = 1.2$, $\hat{\beta}_3 = 0.8$, and $\hat{\beta}_0 = -3.5$.

(a) Interpret $\hat{\beta}_2$. (1 pt)

The estimated coefficient $\hat{\beta}_2 = 1.2$ indicates that family history of prostate cancer is associated with a 1.2 increase in the log odds of developing prostate cancer, when controlling for age and smoking status.

(b) Estimate the probability that a 60-year-old man with a family history of prostate cancer who is a smoker develops prostate cancer. (2 pts)

To estimate the probability, we use the inverse logit function:

$$P(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2+\hat{\beta}_3 X_3}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2+\hat{\beta}_3 X_3}}$$

where $X_1 = 60$, $X_2 = 1$, and $X_3 = 1$.

```
# estimate probability
X1 <- 60
X2 <- 1
X3 <- 1

prob <- exp(-3.5 + 0.06*X1 + 1.2*X2 + 0.8*X3) /
           (1 + exp(-3.5 + 0.06*X1 + 1.2*X2 + 0.8*X3))

print(paste0(round(prob*100,2), '%'))
```

```
## [1] "89.09%"
```

The estimated probability that 60-year-old man with a family history of prostate cancer who is a smoker develops prostate cancer is 89.09%.

4. k-fold cross-validation

(a) Briefly explain how k-fold cross-validation is implemented. (2pts)

In k-fold cross validation, a training set is randomly divided into k groups of equal size. The first group (or fold) is treated as the validation set, and the model is fit on the remaining k - 1 groups (training set).The MSE is calculated using the validation set. This procedure is repeated k times and on each occasion the validation set and training sets will be different than the previous one. We then take the average of all the MSE's as the final MSE.

(b) What are the advantages of k-fold cross-validation relative to the validation set approach? (1pt)

k-fold cross validation yields less variance in test error estimate and a more accurate test error, as entire dataset is used.

5. Suppose you want to minimize the false negative rate in your classification. You run two models: A and B. AUC for Model A is 0.7 and for Model B is 0.8. Can you conclude that you should choose Model B? Why or why not? (3 pts)

We cannot be sure. Model B seems to be performing better, but if we care more or solely about eliminating false negatives, then AUC is not the most informative metric: it is a summary of the relationship between false positives and true positives (or equivalently false negatives), not the rate of false negatives.

Applied Questions (15 pts)

Predicting Hospital Length of Stay

1. For the next portion of this assignment you will be working with the `LengthOfStay.csv` dataset. This dataset has data points on patients admitted into hospital, indicators of their health condition and how long they were admitted in the hospital.

This is an important problem in healthcare. In order for hospitals to optimize resource allocation, it is important to predict accurately how long a newly admitted patient will stay in the hospital.

What are the dimensions of the dataset? (1 pt)

```
data <- read_csv('LengthOfStay.csv')[, c(-1)]

dim(data)
```

```
## [1] 3000   21
```

2. Use the `cor()` function to display the correlations of all **continuous** variables in the dataset. Which variables is most highly correlated with `lengthofstay`? (2 pts)

```
# Correlation matrix
cor(data[,c(12:21)])
```

```
##                  hematocrit  neutrophils       sodium       glucose
## hematocrit      1.000000000  0.131460567  0.024343426  0.0016577282
## neutrophils     0.131460567  1.000000000  0.008076991  0.0080619318
## sodium          0.024343426  0.008076991  1.000000000 -0.0132213991
## glucose         0.001657728  0.008061932 -0.013221399  1.0000000000
## bloodureanitro -0.097450193 -0.059656588 -0.024281498  0.0102109173
## creatinine      0.002356703 -0.014551173  0.050811659  0.0238207285
## bmi             0.011712518  0.010711979  0.004984098  0.0001641313
## pulse           0.017631611 -0.021684801  0.006491825 -0.0184095819
## respiration     0.220649388 -0.062965424 -0.002377212  0.0069222501
## lengthofstay   -0.078573851 -0.034198747 -0.004106144 -0.0203796923
##                bloodureanitro   creatinine           bmi        pulse
## hematocrit        -0.09745019  0.002356703  0.0117125183  0.017631611
## neutrophils       -0.05965659 -0.014551173  0.0107119794 -0.021684801
## sodium            -0.02428150  0.050811659  0.0049840979  0.006491825
## glucose            0.01021092  0.023820728  0.0001641313 -0.018409582
## bloodureanitro     1.00000000 -0.017206983  0.0117521641 -0.015630253
## creatinine        -0.01720698  1.000000000 -0.0054429395  0.012015907
## bmi                0.01175216 -0.005442940  1.0000000000 -0.012001822
## pulse             -0.01563025  0.012015907 -0.0120018215  1.000000000
## respiration        0.02566680  0.006423549  0.0065377627  0.003457493
## lengthofstay       0.19183273 -0.004112273 -0.0023549638 -0.012585940
##                 respiration lengthofstay
## hematocrit      0.220649388 -0.078573851
## neutrophils    -0.062965424 -0.034198747
## sodium         -0.002377212 -0.004106144
## glucose         0.006922250 -0.020379692
## bloodureanitro  0.025666797  0.191832733
## creatinine      0.006423549 -0.004112273
## bmi             0.006537763 -0.002354964
## pulse           0.003457493 -0.012585940
## respiration     1.000000000 -0.035696740
## lengthofstay   -0.035696740  1.000000000
```

`bloodureanitro` is most highly correlated with `lengthofstay`.

Consider the prediction problem where you want to predict the length of stay for a patient (`lengthofstay`) against all other variables available in the data set.

3. Run ridge regression with cross-validation and standardized features using the canned function `cv.glmnet` from the package `glmnet`. You can use the λ sequence generated by `cv.glment` (you do not need to provide your own λ sequence). In order to receive credit for this question, make the line immediately preceding this command say `set.seed(222)` and run the two lines together. Please report all numbers by rounding to three decimal places. (2 pts)

```
set.seed(222)

# Generate the model inputs and run the model
y_vec <- data$lengthofstay
x_mat <- data[, -c(21)]

ridge_model = cv.glmnet(x = as.matrix(x_mat), y = as.numeric(y_vec),
                        alpha = 0, standardize = TRUE)
```

(a) Which λ had the lowest mean cross-validation error? (1 pt)

```
round(ridge_model$lambda.min, digits = 3)
```

```
## [1] 0.189
```

(b) What was the cross-validation error? (1 pt)

```
mse_min = ridge_model$cvm[ridge_model$lambda == ridge_model$lambda.min]
round(mse_min, digits = 3)
```

```
## [1] 4.874
```

(c) What was the standard error of the mean cross-validation error for this value ofλ? (1 pt)

```
mse_min_se = ridge_model$cvsd[ridge_model$lambda == ridge_model$lambda.min]
round(mse_min_se, digits = 3)
```

```
## [1] 0.148
```

(d) What was the largest value of λ whose mean cross validation error was within one standard deviation of the lowest cross-validation error? (1 pt)

```r
round(ridge_model$lambda.1se, digits = 3)
```

```
## [1] 2.802
```

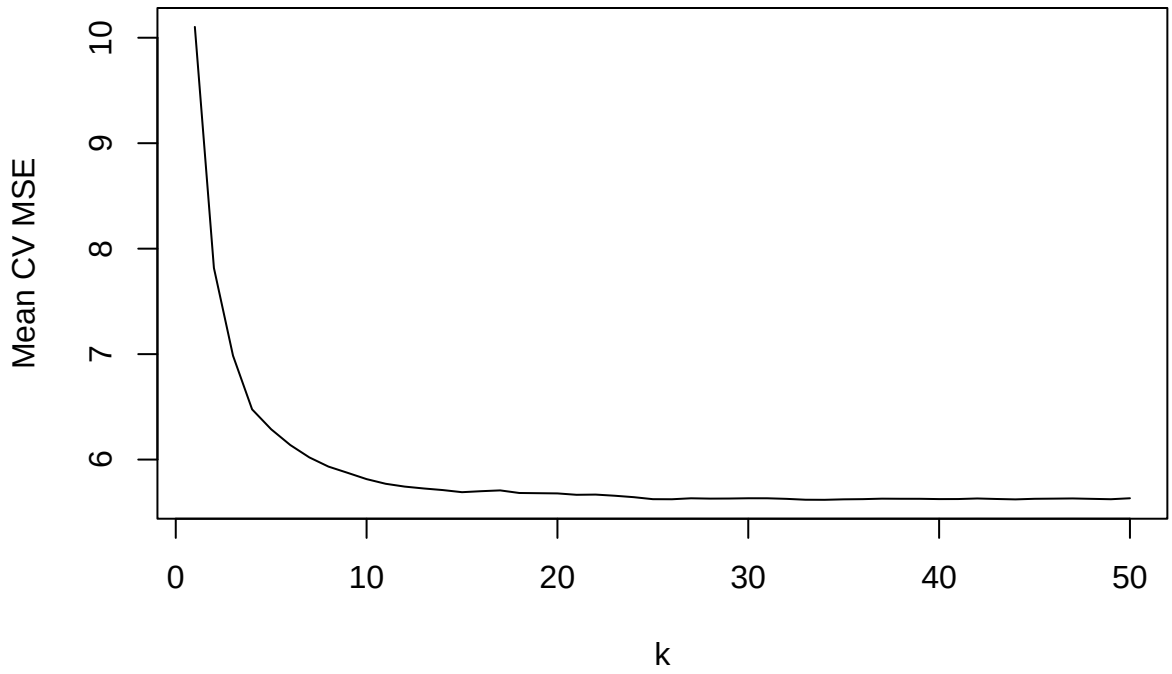
5. Now consider the same prediction problem. Implement your own 5-fold cross-validation routine for KNN for $K = 1, ..., 50$ (write the cross-validation routine yourself rather than using a canned package). Include the snippet of code you wrote here. It should not exceed 20 lines. (6pts)

```r
#------------------------- Define the inputs --------------------------------
# The feature columns of the data
x_mat <- data[, -c(21)]

# The outcome column of the data
y_vec <- data$lengthofstay

# A vector of all k values to test
k_seq = seq(1, 50)

# The number of folds to use in k-fold cross validation
kfolds = 5

#----------------------- Now the cross-validation routine ---------------------

# Create the fold IDs
fold_ids = rep(seq(kfolds), ceiling(nrow(x_mat)/kfolds))
fold_ids = fold_ids[1:nrow(x_mat)]

# Randomly rearrange the vector
fold_ids = sample(fold_ids, length(fold_ids))

# Initialize a matrix in order to store the prediction performance for each fold
CV_error_mat = matrix(0, nrow = length(k_seq), ncol = kfolds)

# Now loop through the k values
for (k in k_seq){

  # For each k, loop through the folds
  for (fold in 1:kfolds){

    ## Train the KNN model
    train_x = x_mat[which(fold_ids != fold),]
    test_x = x_mat[which(fold_ids == fold),]
    train_y = y_vec[which(fold_ids != fold)]
    test_y = y_vec[which(fold_ids == fold)]

    knn_fold_model = knn.reg(train = train_x,
                             test = test_x,
                             y = train_y,
                             k = k)

    ## Calculate and save the MSE
```

```
    CV_error_mat[k, fold]  <- mean((knn_fold_model$pred - test_y)^2)
  }

}
#-------------------------------------------------------------------------------
# Now find the mean error for each k
mean_cv_error = rowMeans(CV_error_mat)
```

(a) Plot of mean cross-validation MSE as a function of k.

```
plot(seq(50), mean_cv_error, type = "l", ylab = "Mean CV MSE", xlab = "k")
```



(b) The best k according to CV is

```
which.min(mean_cv_error)
```

```
## [1] 34
```

(c) The cross-validation error for the best k is

```r
round(min(mean_cv_error), digits = 3)
```

```
## [1] 5.619
```