ISyE 7406: Data Mining & Statistical Learning
HW#3

(due @1:15pm on Thursday, Feb 13, 2020 for on-campus students,
and one week delay for DL students)

Problem 1 (50 points). In this problem, you are asked to write a report to summarize your analysis of the popular "Auto MPG" data set in the literature. Much research has been done to analyze this data set, and here the objective of our analysis is to predict whether a given car gets high or low gas mileage based 7 car attributes such as cylinders, displacement, horsepower, weight, acceleration, model year and origin.

(a) The "Auto MPG" data set is available at UCI Machine Learning (ML) Repository:

`https://archive.ics.uci.edu/ml/datasets/Auto+MPG`

Download the data file "auto-mpg.data" from UCI ML Repository or from Canvas, and use Excel or Notepad to see the data (this is a .txt file).

There are 398 rows (i.e., 398 different kinds of cars), and 9 columns (the car attributes and name). Before we do any analysis, we need to clean the raw data. In particular, some values are missing for this dataset. Many statistical methods have been proposed to deal with missing values, and please conduct literature research by yourself. For the purpose of simplicity in this homework, here we adopt a simple though inefficient method to remove those rows with missing values. Also we remove the last column of car names, which is text/string and may cause trouble in our numerical analysis. These two deletions lead to a new cleaned data set of 392 observations and 8 columns.

To save your time, you can access the cleaned data from the file "Auto.csv" from Canvas and the R code below if you save it in the local folder of your computer, say, "C:/Temp":

```
Auto1 <- read.table(file = "C:/Temp/Auto.csv", sep = ",", header=T);
```

(b) Create a binary variable, `mpg01`, 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 `mpg01` and the other `Auto` variables.

```
mpg01 = I(Auto1$mpg >= median(Auto1$mpg))
Auto  = data.frame(mpg01, Auto1[,-1]);  ## replace column "mpg" by "mpg01".
```

(c) Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

(d) Split the data into a training set and a test set. Any reasonable splitting is acceptable, as long as you clearly explain how you split and why you think it is reasonable. For your convenience, you can either randomly split, or save every fifth (or tenth) observations as testing data.

(e) Perform the following classification methods on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (c). What is the test error of the model obtained?

(1) *LDA* (2) *QDA* (3) *Naive Bayes* (4) *Logistic Regression*

(5) *KNN* with **several** values of K. Use only the variables that seemed most associated with `mpg01` in (c). Which value of K seems to perform the best on this data set?

(6) (Optional) **PCA-KNN**. The Principal Component Analysis (PCA) or other dimension reduction methods can easily be combined with other data mining methods. Recall that the essence of the PC-reduction is to replace the p-dimensional explanatory variable $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, for $i = 1, \ldots, n$, with a new p-dimensional explanatory variable $\mathbf{u}_i = (u_{i1}, \ldots, u_{ip})$, where $\mathbf{u}_i = \mathbf{A}_{p \times p}\mathbf{x}_i$. Then we can apply standard data mining methods such as **KNN** to the first $r(\leq p)$ entries of the $\mathbf{u}_i$'s, $(u_{i1}, \ldots, u_{ir})$, to predict Y_i's. Find the testing errors when the **KNN** with different values of K (neighbors) is applied to the PCA-dimension-reduced data for different $r = p-1, p-2, \ldots, 1$.

(7) (Optional) *Any other classification methods* you want to propose or use.

Write a report to summarize your findings. The report should include (i) **Introduction,** (ii) **Exploratory (or preliminary) Data Analysis**, (iii) **Methods**, (iv) **Results** and (v) **Findings.** Also see the guidelines on the final report of our course project. Please attach your computing code for R or other statistical software (without, or with limited, output) in the appendix of your report, and please do not just dump the computer output in the body of the report. It is important to summarize and interpret your computer output results.

Problem 2. (Optional, no credit. This is a PhD IE/statistics level question). Suppose we have features $x \in \mathbb{R}^p$, a two-class response, with class sizes N_1, N_2, and the target coded as $-N/N_1, N/N_2$, where $N = N_1 + N_2$. In other words, we observe N observations (y_i, x_i) with $x_i \in \mathbb{R}^p$ and the response variable $y_i = -N/N_1$ or $y_i = N/N_2$, so that $\sum_{i=1}^N y_i = 0$.

(a) Show that the LDA rule classifies to class 2 if

$$x^T \hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2}\hat{\mu}_2^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_2 - \frac{1}{2}\hat{\mu}_1^T \hat{\mathbf{\Sigma}}^{-1}\hat{\mu}_1 + \log(\frac{N_1}{N}) - \log(\frac{N_2}{N}),$$

and class 1 otherwise. Here $\hat{\mu}_k = \frac{1}{N_k}\sum_{i \in \text{class } k} x_i$ for $k = 1, 2$ and

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-2}\Big(\sum_{i \in \text{class } 1} (x_i - \hat{\mu}_1)(x_i - \hat{\mu}_1)^T + \sum_{i \in \text{class } 2} (x_i - \hat{\mu}_2)(x_i - \hat{\mu}_2)^T\Big)$$

(b) Let us treat $y_i = -N/N_1$ or $y_i = N/N_2$ as numerical values, and consider minimization of the least squares criterion $\sum_{i=1}^N (y_i - \beta_0 - \beta^T x_i)^2$. Show that the solution $\hat{\beta}$ satisfies

$$\Big[(N-2)\hat{\mathbf{\Sigma}} + \frac{N_1 N_2}{N}\hat{\mathbf{\Sigma}}_B\Big]\beta = N(\hat{\mu}_2 - \hat{\mu}_1)$$

(after simplification), where $\hat{\mathbf{\Sigma}}_B = (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T$.

(c) Hence show that $\hat{\mathbf{\Sigma}}_B \beta$ is in the direction $(\hat{\mu}_2 - \hat{\mu}_1)$ and thus $\hat{\beta} \quad \propto \quad \hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ Therefore the least squares regression coefficient is identical to the LDA coefficient, up to a scalar multiple.

(d) Show that this result holds for any (distinct) coding of the two classes. That is, if we code the Y values into any two distinct values y_1^* and y_2^*, and do linear regression as in part (b), then the results in part (c) still holds. In other words, the specific choices of $y_1^* = -N/N_1$ and $y_2^* = N/N_2$ make the proof in (b) and (c) a little easier, but not essential.

(e) Find the solution $\hat{\beta}_0$, and hence the predicted values $\hat{f} = \hat{\beta}_0 + \hat{\beta}^T x$. Consider the following linear regression rule: classify to class 2 if $\hat{y}_i > 0$ and class 1 otherwise. Show this is not the same as the LDA rule unless the classes have equal numbers of observations.
(Fisher, 1936; Ripley, 1996)