ISyE 7406: Homework # 1

The purpose of this homework is to help you to be prepared to analyze datasets in your future studies and career. Since we are learning how to analyze the dataset, this HW (and other early HWs) will provide the detailed R codes and technical details. Hence, besides running these R codes or their extensions, we expect you to write your homework solution in the format of a report (in pdf or word) that summarizes your findings, understandings, and interpretations. In the main body of the report, please be concise (possibly 2 $\sim$ 8 pages) and easy-understanding, e.g., using the descriptive tables/figures to summarize your results (instead of blindly copying and pasting R/Python output. Of course, if you want, please feel free to include R or python codes/outputs as an appendix (as many pages as you want).

Problem (KNN). Consider the well-known *zipcode* data set in the machine learning and data mining literature, which are available from the book website: <www-stat.stanford.edu/ElemStatLearn>. You can also find it at Canvas: the training data set is the file "zip.train.csv" and the testing dataset is "zip.test.csv". In the *zipcode* data, the first column stands for the response (Y) and the other columns stand for the independent variables (X_i's). The detailed description can be found from

`http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/zip.info.txt`

Here we consider only the classification problem between 2's and 7's, e.g., denote by "ziptrain27" the training data that only includes the data when $Y = 2$ or when $Y = 7$.

(1) **Exploratory Data Analysis of Training data:** when playing with the training data "ziptrain27", e.g., **report** some summary information/statistics of training data that you think are important or interesting. Please do not copy and paste the results of R or Python codes — you need to be selective, and use your own language to write up some sentences to summarize those important or interesting results.

(2) **Build the classification rule** by using the training data "ziptrain27" with the following methods: (i) linear regression; and (ii) the KNN with $k = 1, 3, 5, 7, 9, 11, 13$, and 15. Find the **training errors** of each choice.

(3) Consider the provided testing data set, and **derive the testing errors** of each classification rule in (3).

(4) **Cross-Validation**. The above steps are sufficient in many machine learning or data mining questions when both training and testing data sets are very large. However, for relatively small data sets, one may want to do further to assess the robustness of each approach. One general approach is **Monte Carlo Cross Validation algorithm** that splits the observed data points into training and testing subsets, and repeats the above computation B times ($B = 100$ say). In the context of this homework, we can combine $n_1 = 1376$ training data and $n_2 = 345$ testing data together into a larger data set. Then for each loop $b = 1, \cdots, B$, we randomly select $n_1 = 1376$ as a new training subset and use the remaining $n_2 = 345$ data as the new testing subset. Within each loop, we first build different models from "*the training data of that specific loop*" and then evaluate their performances on "*the corresponding testing data.*" Therefore, for each model or method in part (3), we will obtain B values of the testing errors on B different subsets of testing data, denote by TE_b for $b = 1, 2, \cdots, B$. Then the "average" performances of each model can be summarized by the sample mean and sample variances of these B values:

$$\overline{TE^*} = \frac{1}{B}\sum_{b=1}^{B} TE_b \qquad \text{and} \qquad \hat{Var}(TE) = \frac{1}{B-1}\sum_{b=1}^{B}\Big(TE_b - \overline{TE^*}\Big)^2.$$

Compute and compare the "average" performances of each model or method mentioned in part (2). In particular, based on your results, **write some paragraphs to provide a brief summary** of what you discover in the cross-validation, including reporting the "optimal" choice of the tuning parameter k in the KNN method, and explaining how confident you are on the usefulness of your optimal choice in real-world applications.

Appendix: Please feel free to use the following sample R codes if you want. Of course, you are free to use Python or other softwares

```
## Below assume that you save the datasets in the folder ‘‘C://Temp" in your laptop

## 1. Read Training data
ziptrain <- read.table(file="C://Temp/zip.train.csv", sep = ",");
ziptrain27  <- subset(ziptrain, ziptrain[,1]==2 | ziptrain[,1]==7);

## some sample Exploratory Data Analysis
dim(ziptrain27);    ## 1376 257
sum(ziptrain27[,1] == 2);
summary(ziptrain27);
round(cor(ziptrain27),2);
## To see the letter picture of the 5-th row by changing the row observation to a matrix
rowindex = 5;  ## You can try other "rowindex" values to see other rows
ziptrain27[rowindex,1];
Xval = t(matrix(data.matrix(ziptrain27[,-1])[rowindex,],byrow=TRUE,16,16)[16:1,]);
image(Xval,col=gray(0:1),axes=FALSE) ## Also try "col=gray(0:32/32)"

### 2. Build Classification Rules
### linear Regression
mod1 <- lm( V1 ˜ . , data= ziptrain27);
pred1.train <- predict.lm(mod1, ziptrain27[,-1]);
y1pred.train <- 2 + 5*(pred1.train >= 4.5);
## Note that we predict Y1 to $2$ and $7$,
##   depending on the indicator variable whether pred1.train >= 4.5 = (2+7)/2.
mean( y1pred.train  != ziptrain27[,1]);
## KNN
library(class);
kk <- 1;
xnew <- ziptrain27[,-1];
ypred2.train  <- knn(ziptrain27[,-1], xnew, ziptrain27[,1], k=kk);
mean( ypred2.train != ziptrain27[,1])

### 3. Testing Error
### read testing data
ziptest <- read.table(file="C://Temp/zip.test.csv", sep = ",");
ziptest27  <- subset(ziptest, ziptest[,1]==2 | ziptest[,1]==7);
dim(ziptest27) ##345 257

## Testing error of KNN, and you can change the k values.
xnew2 <- ziptest27[,-1];         ## xnew2 is the X variables of the  "testing" data
kk <- 1;     ## below we use the training data "ziptrain27" to predict xnew2 via KNN
ypred2.test  <- knn(ziptrain27[,-1], xnew2, ziptrain27[,1], k=kk);
mean( ypred2.test != ziptest27[,1]) ## Here "ziptest27[,1]" is the Y response of the "testing" data

### 4. Cross-Validation
### The following R code might be useful, but you need to modify it.
zip27full =  rbind(ziptrain27,  ziptest27)      ### combine to a full data set
n1 = 1376;   # training set sample size
n2= 345;     # testing set sample size
n = dim(zip27full)[1];    ## the total sample size
set.seed(7406);   ### set the seed for randomization
###    Initialize the TE values for all models in all $B=100$ loops
B= 100;             ### number of loops
TEALL = NULL;       ### Final TE values
```

```
for (b in 1:B){
  ### randomly select n1 observations as a new training  subset in each loop
  flag <- sort(sample(1:n, n1));
  zip27traintemp <- zip27full[flag,];  ## temp training set for CV
  zip27testtemp  <- zip27full[-flag,]; ## temp testing set for CV
  ### you need to write your own R code here to first fit each model to "zip27traintemp"
  ### then get the testing error (TE) values on the testing data "zip27testtemp"
  ### IMPORTANT: when copying your codes in (2) and (3), please change to
  ###      these temp datasets, "zip27traintemp" and "zip27testtemp" !!!
  ###
  ### Suppose you save the TE values for these 9 methods (1 linear regression and 8 KNN) as
  ###  te0, te1, te2, te3, te4, te5, te6, te7, te8 respectively, within this loop
  ###   Then you can save these $9$ Testing Error values by using the R code
  ###   Note that the code is not necessary the most efficient
  TEALL = rbind( TEALL, cbind(te0, te1, te2, te3, te4, te5, te6, te7, et8) );
}
### Of course, you can also get the training errors if you want
dim(TEALL);  ### This should be a Bx9 matrices
### if you want, you can change the column name of TEALL
colnames(TEALL) <- c("linearRegression", "KNN1", "KNN3", "KNN5", "KNN7",
  "KNN9", "KNN11", "KNN13", "KNN15");

## You can report the sample mean/variances of the testing errors so as to compare these models
apply(TEALL, 2, mean);
apply(TEALL, 2, var);
### END ###
```