Assignment I
Machine Learning for Finance

Deadline: 2 November 2022

Instructions:
You will be graded out of 3 for each coding assignment. You will submit 1 assignment per group and all students in the group will receive the same grade. Please, submit the R code named as your group (e.g. Alpha.r, Beta.r, ...).

1 KNN and OLS: The digit identification problem

In this exercise we will compare the KNN and OLS in a classification setting.[1]

The digits classification problem aims to automate the sorting of letters by zipcode. The data from this example come from the handwritten ZIP codes on envelopes from U.S. postal mail. Each image is a segment from a five digit ZIP code, isolating a single digit. The images are 16x16 eight-bit grayscale maps, with each pixel ranging in intensity from 0 to 255. Some sample images are shown in the Figure.

The images have been normalized to have approximately the same size and orientation. The task is to predict, from the 16x16 matrix of pixel intensities, the identity of each image (0, 1, . . . , 9) quickly and accurately. If it is accurate enough, the resulting algorithm would be used as part of an automatic sorting procedure for envelopes. This is a classification problem for which the error rate needs to be kept very low to avoid misdirection of mail. The zipcode data are available with documentation here: `https://web.stanford.edu/~hastie/ElemStatLearn/data.html`. You can read the info file to understand what is the data and how it is organized.

The training data contains 7291 hand written numbers and the corresponding true identity of the number. The test data contains similar information for 2007 other numbers. The training data will be used to construct the models and the testing data will give an indication of the predictive power of the model.

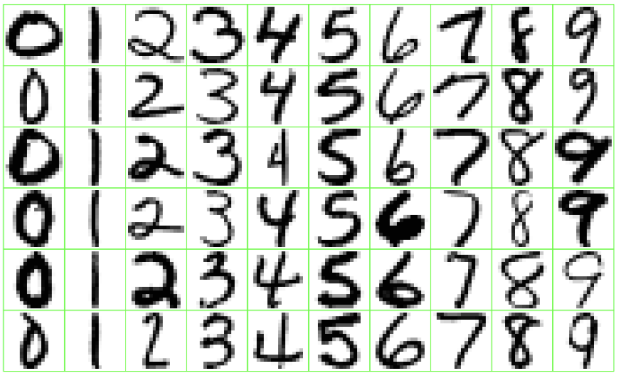
a) Plot one digit as follows:

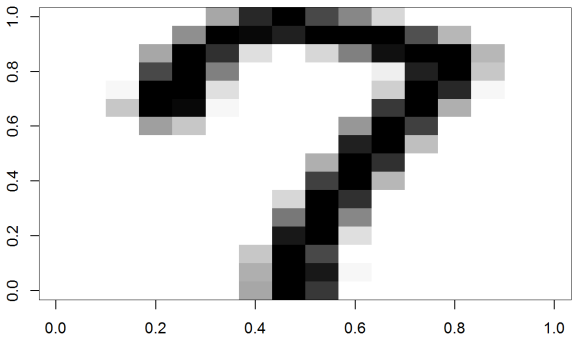
```
zipTrain <- read.csv("zip.train.gz", header = F, sep=" ")
zipTrain<-zipTrain[,1:257]
zipTest <- read.csv("zip.test.gz", header = F, sep=" ")

im <- matrix(as.numeric(zipTrain[4,2:257]), nrow = 16, ncol = 16)
image(t(apply(-im,1,rev)),col=gray((0:32)/32))
```

b) Your task is to compare the classification performance of linear regression and k–nearest neighbor classification on the zipcode data. To develop a classifier from the OLS, we will follow a linear probability model and use a simple cutoff rule matching the Bayes classifier. You may also use logit or probit models for the problem as refinements. In particular, consider the problem and training data only for the 2's and 3's, and k = 1, 3, 5, 7 and 15. Show both the training and test error for each choice.
NOTE: Always remember to set the seed for your problem for us to be able to replicate your results!

[1] HINTS: For KNN, we have the library **class** in R. First you will need to understand the organisation of R packages. Visit the landing page of the **class** package : `https://cran.r-project.org/web/packages/class`. Note the dependencies you need to install to successfully install the package. Whether these need to be installed manually or automatically may depend on your system. Install package **class** with the **MASS** dependency. The manual explains the **knn()** function with all possible options you can specify. The manual has an example, it may be helpful to work through this simple example first before moving on to the problem.

Figure 1: Handwritten digit classification





We will be referring back to this example when we study classification trees as well as neural nets. The work done to assemble the data as well as comparison of a few methods can be found in the original publication for your reference: *Denker, John S., et al. "Neural network recognizer for hand-written zip code digits." Advances in neural information processing systems. 1989.*

2 Logistic regression

In this exercise we predict the likelihood of a Shuttle disaster. This is an example showing that the Challenger disaster of January 28, 1986, might have been averted had NASA considered the warning signs. Our data contain information about the first 25 flights of the U.S. space shuttle. For each. flight, we observe the date (number of days since Jan 1st 1960), the temperature, and a measure of the number of thermal distress incidents occurred during the launch. We have three categories that describe the number of incidents: 1 = none, 2 = 1 or 2, 3 = 3 or more. **The dataset "NASA.csv" is uploaded on Luiss Learn**.

a) First reformat the variable "distress" as categorical variable

b) Use the variable "distress" as dependent variable in your logistic regression. As independent variables include date and temperature. The syntax of the glm() function is similar to that of lm() except that we must pass in the argument family=binomial in order to tell R to run a logistic regression rather than some other type of generalized linear model.

```
glm.fits=glm(Y~X1+X2,data=data,
             family=binomial)
summary(glm.fits)
```

What is the interpretation of the coefficients values?

hat is the interpretation from the coefficient values?

c) We can extract the coeffs using the coeff() function. We can also use the summary function to access parts of the model including p-values and coefficients. We can predict using the fitted model using predict() function.

```
coef(glm.fits)
summary(glm.fits)$coef
glm.probs=predict(glm.fits,type="response")
glm.probs
contrasts(Direction)
```

d) However this is performance on the training data. We need to look into performance on the test dataset. For this, we hold out data for 1986

```
train=(year<1986)
data.1986=data[!train,]
dim(data.1986)
distress.1986=distress[!train]
```

Fit a logisitic model on this dataset (similar as above) and predict "distress" for the Challenger disaster occured on Jannuary 28, 1986. What is your prediction?

3 Linear Discriminant Analysis

a) Now perform LDA on the NASA data. In R, we fit an LDA model using the lda() function, which is part of the MASS library. Notice that the lda() syntax for the lda() function is identical to that of lm(), and to that of glm() except for the absence of the family option. We fit the model using only the observations before 1986.

b) Fit LDA as you did in the previous question. Are the estimates different? Why?

c) Now do the predictions for the Challenger disaster. What is your prediction?

4 Quadratic Discriminant Analysis

Syntax for applying QDA is like that of LDA. Predict function will work the same way, we have covered this in class.

```
qda.fit=qda(distress~date+temp,data=data,subset=train)
qda.fit
qda.class=predict(qda.fit,data.1986)\$class
table(qda.class,data.1986)
mean(qda.class==data.1986)
```

Apply QDA the NASA dataset, using the years prior to 1986 as training sample. What is your prediction?

5 Non-parametric bootstrap

a) The inbuilt **boot()** function in R requires loading using **require(boot)**

b) To use it, first you need to define a function taking the original sample and index of the bootstrap sample as inputs.

Example:

```
 set.seed(20102022)
x <- rnorm(10000)
mean(x)
mean.boot <- function(x, ind) {mean(x[ind])}
boot(x, mean.boot, 100)
```

c) The results show original mean(x) and a bias and variance. With higher number of bootstrap sample, the bias will vanish - show this for above code by increasing (e).

d) Compute and interpret your results:

		observations in each draw			
		10	100	1000	10,000
bootstrap draws	10				
	100				
	1000				
	10,000				

6 Bootstrap Confidence Intervals

We can use the bootstrap estimator to estimate confidence intervals. This uses the formula for the confidence interval based on the mean and variance of the empirical distribution derived from the bootstrap. For this, we need to feed the **boot.ci()** function with the output of bootstrap.

a) Play around with the **boot.ci** function with the above example to learn how it works and how it connects to the empirical distribution. Plot the distribution and the cutoffs for the CIs you get.

b) Now set up a simple y,x vector and estimate OLS as in previous exercises. This time calculate the empirical distribution of beta through bootstrap, resampling from the original distribution. What are the bootstrap CIs? Is the coefficient of the original regression significant based on the bootstrapped CI?