

**CS 4661: Introduction to Data Science**  
**Dr. Mohammad Pourhomayoun**  
**Homework5**  
**Due Date: Sun, Dec 2**

**Question1: Handwriting Recognition using Machine Learning!**

Write and submit your python codes in “Jupyter Notebook” to perform the following tasks. Make sure to provide proper descriptions as Markdown for each section of your code.

- a- Download the dataset “Digit” from CSNS. Check out the dataset. It includes 1797 small images (8x8 pixels), each one includes a hand-written digit (0-9). You have to download the corresponding csv file that includes the labels of the images. The goal is to build a Machine Learning Algorithm that can recognize the hand-written digits!

Import the following two libraries to work with images:

```
import matplotlib.image as mpimg  
import matplotlib.pyplot as plt
```

you can use:

```
mpimg.imread(file_name) to load an image, and  
plt.imshow(image_name, cmap=plt.cm.gray_r, interpolation='nearest') to show an image.
```

Add **%matplotlib inline** at top of your code to make sure that the images will be shown inside the Jupyter explorer page.

- b- Build the feature matrix and label vector: Each image is considered as a data sample with pixels as features. Thus, to build the feature table you have to convert each 8x8 image into an array of 64 elements (i.e. 64 pixels), and put it as a row of the feature matrix with 64 feature columns.
- c- Use sklearn functions to split the dataset into testing and training sets with the following parameters: **test\_size=0.1, random\_state=2**.
- d- Use scikit-learn “Random Forest” classifier to recognize the hand-written digits based on the training/testing datasets that you built in part (c). Use this command to import and define your classifier:
- ```
from sklearn.ensemble import RandomForestClassifier  
my_RandomForest =  
RandomForestClassifier(n_estimators = 19, bootstrap = True, random_state=2)
```

Use `my_RandomForest.fit` for training your random forest classifier and `my_RandomForest.predict` for prediction. Test your Machine Learning Algorithm on testing set (from part(c)), and calculate and report the accuracy.

- e- Write some codes to find which one of the data samples (i.e. which images) have been misclassified (classified incorrectly) in your testing set. Then, use the following command to show the misclassified images:

`plt.imshow(image_name, cmap=plt.cm.gray_r, interpolation='nearest')`

**Question2 (no need for coding for Question2):** Suppose we have a dataset with 3 features:  $X_1$  = GPA,  $X_2$  = Age,  $X_3$  = Type of Position (1 for Technical positions, and 0 for Non-Technical positions), and we have built a non-linear regression model as:

$$\text{Target} = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_1 X_2 + \theta_5 X_1 X_3$$

The prediction target is “starting salary after graduation” (in thousands of dollars). Suppose we train (fit) the model, and get  $\theta_0 = 30$ ,  $\theta_1 = 20$ ,  $\theta_2 = 0.07$ ,  $\theta_3 = -30$ ,  $\theta_4 = 0.01$ ,  $\theta_5 = 10$ .

(a) Which answer is correct, and why?

- For a fixed value of Age and GPA, Technical positions earn more on average than non-technical positions.
- For a fixed value of Age and GPA, Non-Technical positions earn more on average than Technical positions.
- For a fixed value of Age and GPA, Technical positions earn more on average than Non-Technical positions when the GPA is high enough.
- For a fixed value of Age and GPA, Non-Technical positions earn more on average than Technical positions when the GPA is high enough.

(b) Predict the salary of a Technical and a Non-Technical positions with Age of 27, GPA of 4.0.

**Question3 (no need for coding for Question3):** Suppose that we would like to perform the following task using MapReduce. Please determine the input/output of **each** mapper and reducer, and all intermediate key-value pairs generated in the process of MapReduce:

- Matrix-to-Vector multiplication using MapReduce with **4 mappers** and **2 reducers**:

$$\begin{bmatrix} 2 & -3 & 1 & 5 \\ 7 & 2 & -1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ -6 \\ 9 \end{bmatrix}$$