ASSIGNMENT 1

Subject : Data preprocessing, regression, and classification.
Instructor: Dr. Selim Yılmaz (selimyilmaz@mu.edu.tr)
Out Date: 11/04/2020 23:59:59
Due Date: 11/11/2020 23:59:59

DECLARATION OF HONOR CODE[1]

Student ID .....
Name .....
Surname .....

In the course of Data Mining (CENG 3521), I take academic integrity very seriously and ask you to do as well. That's why, this page is dedicated to some clear statements that defines the policies of this assignment, and hence, will be in force. Before reading this assignment booklet, please first read the following rules to avoid any possible violation on academic integrity.

- This assignment must be done individually unless stated otherwise.
- You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, you cannot copy code (in whole or in part) of someone else, cannot share your code (in whole or in part) with someone else either.
- The previous rule also holds for the material found on the web as everything on the web has been written by someone else.
- You must not look at solution sets or program code from other years.
- You cannot share or leave your code (in whole or in part) in publicly accessible areas.
- You have to be prepared to explain the idea behind the solution of this assignment you submit.
- Finally, you must make a copy of your solution of this assignment and keep it until the end of this semester.

I have carefully read every of the statements regarding this assignment and also the related part of the official disciplinary regulations of Muğla Sıtkı Koçman University and the Council of Higher Education. By signing this document, I hereby declare that I shall abide by the rules of this assignment to prevent any violation on academic integrity.

Signature .....

[1]This page should be filled and signed by your handwriting. Make it a cover page of your report.

1 Introduction

In this assignment you will practice what you have been taught in CENG3521 course. These include *i*) different preprocessing approaches which are applied on the datasets for the sake of better performance in terms of effectiveness and efficacy, *ii*) linear regression that predicts a real valued output by analysing the relation between independent and dependent variables, and *iii*) finally logistic regression which, contrary to linear regression, estimates a categorical value representing the expected class of given data object. You are expected to complete the following two phases which involve different data mining tasks.

2 Regression Task

2.1 Curse of dimensionality

The objective of this section is to analyze the degradation of the data mining algorithm as the size of the data dramatically increases. For this purpose, follow the instructions below. Note, you should apply them in a row.

1. Generate (or use ready-to-use) a toy/fictional n-dimensional dataset (i.e., $D \mid D \in \mathbb{R}^n$) having m tuples. While the first **four** dimensions are expected to be correlated with the ground truth vector (i.e., y) and the remaining ones are to be arbitrarily generated. Feel free to establish the relation between those.
2. Split D such that randomly selected 70% tuples are used for training while 30% tuples are used for testing.
3. Apply linear regression with Stochastic Gradient Descent (SGD) solver (with 1,000 iterations) to handle D.
4. For each parameter setting given in Table 1, repeat the previous steps for **five times**. Then fill each row of the Table 1 with your observations given that parameter setting.
5. Discuss the positive relation between dimension size and training time and error.
6. Discuss the negative relation between tuple size and error yield by the algorithm.
7. Discuss also positive relation between tuple size and training time.

Table 1: The effectiveness and efficacy of linear regression algorithm with respect to the varying tuple and dimension size.

	Parameter setting		**Observations***	
#	**Tuple Size (m)**	**Dimension Size (n)**	**Training Time (in ms)**	**Error (cost)**
a.	10,000	100		
b.	10,000	1,000		
c.	10,000	2,000		
d.	100,000	100		
e.	250,000	100		
f.	500,000	100		

*:**average of five runs.**

2.2 Sampling and dimensionality reduction

Here, you are to apply two strategies to better handle D under a parameter settings **c** and **f** in Table 1 to verify why it is a good idea to process a given dataset before it is given for some data mining tasks. To accomplish this phase, follow the instruction below:

1. For a given D having shapes in **c** and **f** in Table 1, apply Principal Component Analysis (PCA) and a sampling (without replacement) algorithm to, respectively, reduce n and m to each size given in Table 2.
2. Split D such that randomly selected 70% tuples are used for training while 30% tuples are used for testing.
3. Apply linear regression with Stochastic Gradient Descent (SGD) solver (with 1,000 iterations) to handle D.
4. For each parameter setting given in Table 2, repeat the previous steps for **five times**. Then fill each row of the Table 2 with your observations given that parameter setting.
5. Discuss your observation regarding the effect of these strategies on regression algorithm.

Table 2: The effect of dimensionality reduction and sampling strategies on learning time and performance.

Parameter set **c** in Table 1

Dimension Size (n)	**Training Time (in ms)**	**Error (cost)**
500		
100		
10		
4		
1		

Parameter set **f** in Table 1

Tuple Size (m)	**Training Time (in ms)**	**Error (cost)**
300,000		
150,000		
100,000		
1,000		
100		

Congratulations! You have successfully completed this part. Only one step left to be a precious part of the data science community :) Now, you can go ahead to the 2nd task.

3 Classification Task

3.1 Visualization and binary-class classification

The aim of this space is to have you gained an experience on how to visualize the tuples and decision boundary that is generated by logistic regression algorithm on the search space. To do that, follow the steps here:

1. Load a dataset from `sklearn` module (e.g., moon, circle, and the like) (or generate on your own[2]) a toy/fictional *two*-dimensional dataset D with a tuple size greater than 100 (feel free to give any arbitrary number for every class).
2. Split D such that randomly selected 70% tuples are used for training while 30% tuples are used for testing.
3. Run logistic regression algorithm with SGD solver (with 10,000 iterations) on D.
4. Through a 1×2-axis figure, visualize training and testing samples as well as a decision boundary that fits D well. An example outcome to this step is given in Figure 1. Once you output this figure save it as 'BinaryClassVisualization.pdf'; then **close** figure window.

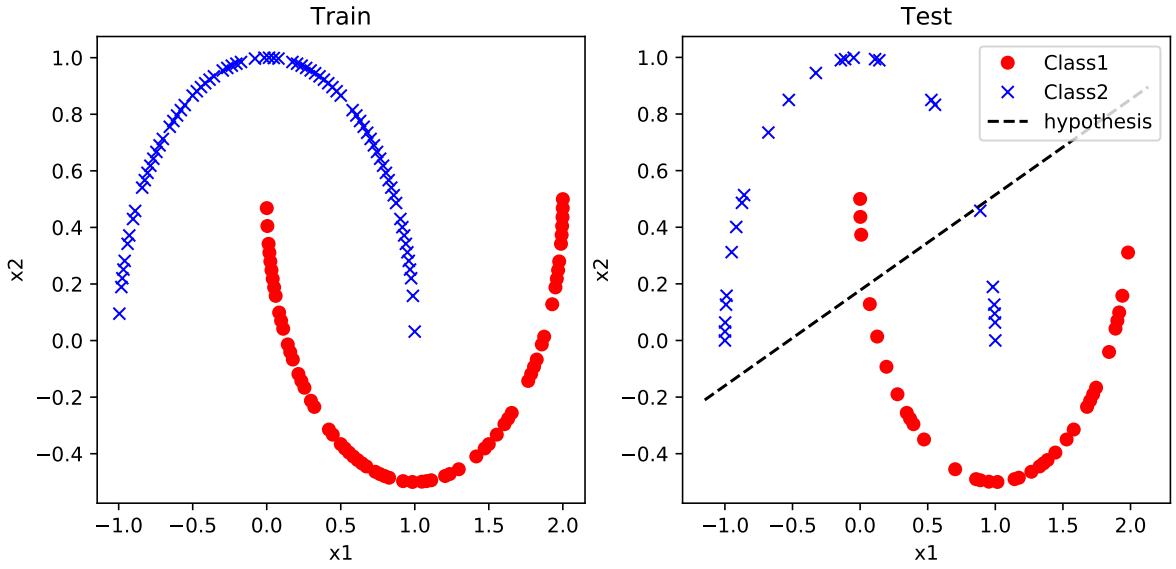

Figure 1: An example for a visualization of given two-class dataset.

3.2 Noising and multi-class classification

Noise in a dataset can adversely affect the identification performance of given algorithm. That's why data scientists often come up with various strategies to deal with it. In this task you are expected to observe degree to which a noise would affect the learning algorithm. Follow the instruction below to accomplish this task:

1. Load *digit* dataset D from `sklearn` module.
2. Split D such that randomly selected 70% tuples are used for training while 30% tuples are used for testing.
3. Add noise to P% of **training data** and **testing** data as follows:
   - Randomly select 10 dimension for a tuple on which noise is to be added.

[2] in this case you should follow a pattern that differentiates given two classes

- For each dimension j change the value of $x_j \mid x_j \in [0, 16]$ to find its noisy value $\hat{x_j}$ using following equation

$$\hat{x_j} = |x_j - 16| \tag{1}$$

4. Apply a denoising strategy[3] to get rid of such 'bogus' information resulted in environment **h.** in Table 3.
5. Write down your observations to Table 3. Discuss the results you obtain.
6. Finally, add noise on a randomly selected a tuple from D. Then, visualize original tuple and its noisy form (through Eq. 1) in a comparative way by using 1×2-axis figure. Afterward, save is as 'Noising.pdf' then **close** the figure window. An example is given in Figure 2.

Table 3: The effect of noise on classification performance.

#	Noise rate (P%)		Error
	Train	**Test**	
a.	0	25	
b.	0	50	
c.	0	75	
d.	25	0	
e.	50	0	
f.	75	0	
g.	25	25	
h.	50	50	
i.	75	75	
j.	denoised form of **h.**		

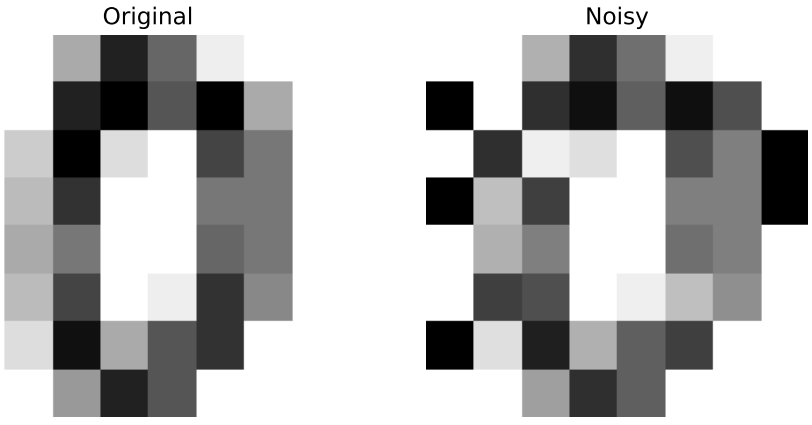

Figure 2: An example on noising effect on a digit sample.

[3] feel free to use any method in `sklearn` for denoising.

3.3 Evaluation/performance metric demonstration

At the last step of this assignment, you are to show the effectiveness of classification algorithm on a given confusion matrix with a heatmap chart as shown in Figure 3. To do that, first apply logistic regression algorithm with SGD solver (with 10,000 iterations) for one time on digit dataset. Here, you are to use *over-vs-rest* (or *one-vs-all*) trick for multi-class classification. After you train, make use of a confusion matrix to measure classification performance. Use heatmap to represent that matrix and save it as 'ConfusionMatrixHeatmap.pdf' then close the figure window.

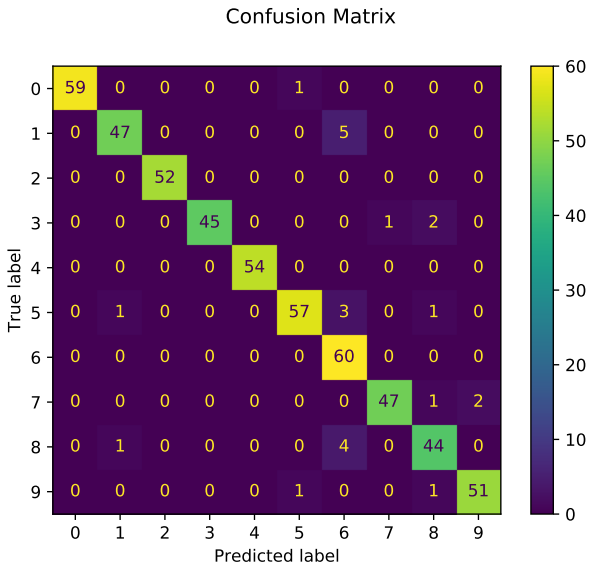

Figure 3: A heatmap chart revealing confusion matrix for digit dataset.

I would like to warmly congratulate you for your respectable effort and time put for completion of this assignment. Through this assignment, you have gained the following skills/knowledge:

- application of a linear regression algorithm with SGD solver on a given D.
- use of dimensionality reduction and sampling strategies to process D.
- interpretation of how these strategies could positively affected to some degree.
- application of a logistic regression algorithm with SGD solver on a given D to handle binary- and multi-class classification problem.
- visualization of dataset with two to three dimensions as well as decision boundary on a plot.
- observation of bad effect of noise on learning algorithm.
- interpretation of testing performance using confusion matrix through a heatmap type plot.
- finally, practicing on Python and scikit-learn library.

Notes

- Your source code should be designed as **easy-to-follow**. **Place comment** in it as much as possible. **Separate each task** through apparent patterns.
- Use LaTeX to prepare your reports. Include the observation tables here to your report. Once again, filled and signed declaration form should be first page of your report. **Reports must not exceed 5 pages in total**.
- **Do not miss** the deadline.
- **Save your work** until the end of this semester.
- The assignment must be **original**, **individual work**. **Duplicate or very similar assignments are both going to be considered as cheating.**
- You can ask your questions via **Piazza** (https://piazza.com/mu.edu.tr/fall2020/ceng3521) and you are supposed to be aware of everything discussed in Piazza.
- You will submit your work on CENG3521 course page at https://dys.mu.edu.tr with the file hierarchy as below[4]:

→ <student id>.zip
  → Assignment1.py
  → Report1.pdf

[4]do not place any file into a directory. Just compress all the files together.