EECS 461/ECE 523
MACHINE LEARNING
Fall 2019

ASSIGNMENT 2

Due Date: *Saturday, December 7th, 2019, 23:59*

Assignment Submission:

1. Turn in your assignment by the due date through LMS**.**
2. Prepare a single Jupyter Notebook (.ipynb) with the answers to all questions. **Name the file as <your first name>_<your last name>_ assignment2.ipynb.**
3. Make sure to **use the sample Jupyter Notebook file provided to you as template**.

All work in questions must be your own; you must neither copy nor provide assistance to anybody else. If you need guidance for any question, talk to the instructor or TAs in office hours. You can also reach your TAs via email*****:

- Zeina Termanini at zenatermanini@std.sehir.edu.tr
- Mohammad Abunada at mohammedabunada@std.sehir.edu.tr

Late Assignment Policy: You have a total of **4 days of late assignment** turn-in allowance throughout this semester. For a single assignment, you can use **a maximum of 2 late-days.** You decide which assignments you are going to use your 4 late-days. After assignment due date/time, each 24-hours period is counted as one late date (i.e., if you submit your assignment 1 hour late or 23 hours late, you use 1 late-date). It is your responsibility to keep track of your late days. If you are late more than 2 days for any assignment or you exhausted your late days, you get 0 from the late assignment **(No exceptions)**

*** Please include a thorough description of your problem. If you are having problems with your code, include a screenshot of the exact problem. Please don't enquire if your answer is correct or not.**

Assignment Overview:

In this assignment, you will be running exploratory analysis on a dataset to better understand it and its features. You will then be processing and preparing the data to apply the machine learning knowledge you've obtained through the lectures. This will include creating, analysing and generating predictions with classification models. This assignment is mainly about the examples in chapters 3 and 4 of the course book with a different data set. Reviewing the book and the corresponding code will greatly help you. **You are expected to primarily use Scikit-Learn in the assignment.**

Data Set:

The data set provided for this assignment contains pixel information of many 28x28 grayscale images of clothing, similar to the MNIST dataset. Each image is labelled as one of the 10 different pieces of clothing:

class_labels = [T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot]

The value of a row in the column "label" corresponds to the index of the item in the class_labels list.

Data is in CSV format and has already been split into training and test sets for your convenience: train.csv: the training set, test.csv: the test set.

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pi
0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
1	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
2	6	0	0	0	0	0	0	0	5	0	...	0	0	0	30	
3	0	0	0	0	1	2	0	0	0	0	...	3	0	0	0	

DATA PREPARATION & VISUALIZATION (25 points)

In the first part of the assignment, you will analyze the dataset and preprocess it in order to prepare it for using machine learning algorithms. In this data set, our target variable is "label" while the others are our features.

(a) (5 points) Split your data into X and y:

As mentioned, each instance in the training data contains a label value along with 784 pixels. Create two pandas data frames using **train.csv**, one containing all the input features and the other containing the target label only. Name these data frames as **train_x_a** and **train_y** respectively. Repeat the same steps using **test.csv** to create the variables **test_x, test_y.**

(b) (5 points) Visualizing the data:

Write a function **plot_image** that will take a **pandas rows** of pixels as input and plot the resulting 28x28 image.

(**Hint:** You can utilize functions from the book's chapter 3 codes in this question)

(c) (5 points) Class visualization:

Using the function written in (b), plot 3 instances of each class in the training set.

(d) (10 points) Average visualization:

Using the function written in (b), plot the average image for each class in the training set. The average image for each class is calculated by taking the mean of each pixel column for that class.

BINARY CLASSIFICATION (35 points)

In the second part of the assignment, you will use Logistic Regression to perform binary classification using the pixel features in the dataset.

(e) (5 points) Binary transformation:

To run binary classification we need to first transform our class labels to binary (0 and 1). If the item is a piece of clothing (T-shirt/top, Pullover, Shirt, Trouser, Dress, or Coat), label it as 1 and and if the item is non-clothing (Sandal, Sneaker, Bag, or Ankle boot), label it as 0. Store the result in a variable called **train_y_e.** Follow the same steps to create binary labels for the test set's target column. Name this variable **test_y_e**.

(f) (5 points) Binary Classification Model:

Create a Logistic Regression model with default parameters. Perform 5-fold Cross Validation on the training data and report the mean accuracy.

(g) (5 points) Predict test data:

Train the model with **train_x_a** and **train_y_e** and predict the labels of **test_x.** Report the **accuracy, confusion matrix, precision, recall** and **f1 score** of these predictions.

(h) (10 points) Model Evaluation:

Plot the above model's precision-recall curve and ROC curve. Report the ROC area under the curve (AUC) score.

(i) (10 points) Learning curves:

Plot the learning curve of a logistic regression model with default parameters by using **train_x_a**. Increase the data size by 1000 at each step. For both train and validation, report classification accuracy. (**Hint:** You can take inspiration and modify functions from the book's chapter 4 codes in this question)

MULTICLASS CLASSIFICATION USING LOGISTIC REGRESSION (10 points)

In the third part of the assignment, you will use logistic regression to perform multiclass classification with the original dataset labels.

(j) (5 points) Training and cross validating logistic regression:
Use **train_x_a** and **train_y** to perform 3-fold cross validation on a logistic regression model with default parameters. with cv=3**.** Report the mean accuracy.

(k) (5 points) Testing Logistic Regression:
Fit a logistic regression model on the training set (**train_x_a, train_y)** and calculate its **test accuracy** by using **test_x.** Print the model's **confusion matrix** on the test set.

MULTICLASS CLASSIFICATION USING SVR (15 points)

In the fourth part of the assignment, you will use an SVM Classifier (sklearn's SVC) to perform multiclass classification using the pixel features in the dataset.

(l) (10 points) Grid Search to find best model:
In order to perform Multiclass Classification, you will need to utilize SVC. However, SVC has large hyper parameter set. To find the best combination you will be using gridsearch. Using the below parameters, run GridSearchCV (cv = 5) with an SVC model on the **train_x_a** and **train_y.** Print out the best model's parameters and accuracy.

Parameter Name	Parameter Values
kernel	linear, rbf, poly
C	0.1, 0.5, 1, 5, 10
tol	0.000, 0.001, 0.1
decision function shape	ovo, ovr

(m) (5 points) Testing the best model:

Recreate a model using the best parameters in **(l)** and calculate its **test accuracy.** Print the model's **confusion matrix** on the test set.

MULTICLASS CLASSIFICATION USING DECISION TREES (15 points)

In the fifth part of the assignment, you will use a decision trees to perform multiclass classification.

(n) (10 points) Grid Search to find best model:

Using the below parameters, run GridSearchCV (cv = 5) with a decision tree model on the **train_x_a** and **train_y.** Print out the best model's parameters and accuracy.

Parameter Name	Parameter Values
Max depth	2, 4, 10
min samples split	2, 3, 4

(o) (5 points) Testing the best model:

Recreate a model using the best parameters in **(n)** and calculate its **test accuracy.** Print the model's **confusion matrix** on the test set.

IMPORTANT NOTES

- Prepare and upload one Jupyter notebook file, which should be named as <your first name>_<your last name>_ assignment2.ipynb.
- A template Jupyter notebook file provided to you. Follow the template's structure.
- Explain your code with comments.
- **Plagiarism in any form will not be tolerated. Changing variable names is not solving an assignment.**

Wrong file name format	-10 points
Not using template	-10 points
Not using correct variable (dataframe) names	-10 points
Insufficient comments	**Any part with insufficient comments will not be graded.**