Assignment 4

CS 484

Submission Deadline: 04/24 11:59 pm

Problem 1: Clustering (20 marks)

For this task, you will perform various clustering-related operations over datasets lab04_dataset_1.csv and lab04_dataset_2.csv, using sklearn's clustering module.

1. Dataset lab04_dataset_1.csv have two variables *x1* and *x2*. Apply KMeans algorithm on the two-dimensional data and output the resulting clusters using a scatterplot. You will apply KMeans over several clusters ranging from cluster-count K = 2 to 6. Make sure for every iteration of different cluster-count, your scatterplot should use K colors to clearly distinguish the data points belonging in their respective K clusters. Also, compute the Silhouette score for each of those K clusters and plot that score against K. (5 marks)

2. Dataset lab04_dataset_2.csv have two variables *x1* and *x2*. Again, apply KMeans algorithm on the two-dimensional data with clusters ranging from K = 2 to 4 and output the resulting clusters using scatterplots. Do the cluster outputs you obtained using KMeans for this dataset make sense? (5 marks)

3. The data in the lab04_dataset_2.csv forms 4 concentric rings rather than being well-separated clusters. So ideally, we would want 4 clusters representing the 4 concentric rings. KMeans is not well-suited to handle data like this. Use SpectralClustering to cluster the data. Show the results for clusters K = 2 to 4 (5 marks)

4. It is possible that SpectralClustering although an improvement over KMeans is still not able to create 4 clusters corresponding to the 4 concentric rings. Explore the other sklearn clustering algorithms to see which one can produce 4 clusters corresponding with the 4

concentric rings. Hint: I mentioned this algorithm during our class while discussing density-based clustering. (5 marks)

Problem 2: Neural Network (30 marks)

For this task, you will perform various neural network-related operations over datasets lab04_dataset_3.csv and lab04_dataset_4.csv, using sklearn's neural network module.

1. You will train a Multi-Layer Perceptron neural network for the task of classification on the dataset lab04_dataset_3.csv using MLPClassifier. The inputs to your MLPClassifier are *alcohol, citric_acid, free_sulfur_dioxide, residual_sugar, sulphates*, while the output is *quality_grp*, which has two categories, namely, 0 and 1. Use a train-test split of 80-20. For the learning task, you will train neural network models with different architectures:
    a. Activation function = [logistic, relu, tanh]
    b. Hidden layers = [1, 2, 3, 4, 5]
    c. Neurons per layer = [2, 4, 6, 8]

   So, basically in the first iteration you will create a learning model using the neural network architecture [logistic, 1, 2], in the second iteration you will use [logistic, 1, 4], all the way to [tanh, 5, 8]. For each of these learned models, compute the Misclassification Rate on the test set. Once you are done with all the training, you should output a dataframe with columns *Activation function, Hidden layers, Neurons per layer, Misclassification Rate*, where each row will correspond with the individual training models. Since the total count of activation functions, hidden layers and neurons are 3, 5, 4 respectively, the number of rows in your output dataframe should be 3 x 5 x 4 = 60. Also use max_iter=10000 and random_state=2023484 inside MLPClassifier definition (10 marks)

2. Create a scatterplot of the Misclassification Rate, make sure that the misclassification rates are distinguishable by different colors according to their activation function. So, the scatterplot should have 3 colors differentiating the misclassification rates associated with the 3 activation functions. (3 marks)

3. The model with the lowest Misclassification Rate is the best neural network. Output the model parameters (activation function, hidden layers, neurons) of this neural network. In the case of ties, choose the network with fewer neurons overall. (2 marks)

4. You will train a Multi-Layer Perceptron neural network for the task of regression on the dataset lab04_dataset_4.csv using MLPRegressor. The inputs to your MLPRegressor are *housing_median_age, total_rooms, households, median_income* and the output is *median_house_value*. First, normalize the dataset, and then create an 80-20 train-test split. In a similar manner to the previous classification task, you will once again learn multiple neural network models of varying architectures.

    a. Activation function = [relu, tanh]
    b. Hidden layers = [2, 3, 4]
    c. Neurons per layer = [4, 6, 8]

    For each of these learned models, compute the Root Mean Square Error. Once you are done with all the training, you should output a dataframe with columns *Activation function, Hidden layers, Neurons per layer, Root Mean Square Error*, where each row will correspond with the individual training models. Since the total count of activation functions, hidden layers and neurons are 2, 3, 3 respectively, the number of rows in your output dataframe should be 2 x 3 x 3 = 18. Also use random_state=2023484 inside MLPRegressor definition (10 marks)

5. Create a scatterplot of the Root Mean Square Error, make sure that the root mean square errors are distinguishable by different colors according to their activation function. So, the scatterplot should have 2 colors differentiating the root mean square errors associated with the 2 activation functions. (3 marks)

6. The model with the lowest Root Mean Square Error is the best neural network. Output the model parameters (activation function, hidden layers, neurons) of this neural network. In the case of ties, choose the network with fewer neurons overall. (2 marks)