Computer Science Department
California State University Channel Islands

COMP 478 - Midterm Exam 1 - Part 2
Due Date: 04/08/2022 11:59 pm
Late submissions will not be accepted or graded at all!
You are not allowed to share your solution with others!
(Please include all the files in your submission.)

Work in a group

1. (20 points) In this problem, we will write a python code to train and test four different classifiers: DT, KNN, NB, and LR for a medical diagnosis- classification task.

   (a) *Step 1*: Download the your dataset from UCI repository using the given URL:

      i. Rene and Evan: Breast Cancer - dataset1 - Recognition: `https://archive.ics.uci.edu/ml/datasets/breast+cancer`

      ii. Alex and Dominique: Heart Disease Recognition: `https://archive.ics.uci.edu/ml/datasets/heart+disease`

      iii. Jeffery and Daniel: Thyroid Disease Recognition: `https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease`

      iv. Juan-Christopher: Breast Cancer - dataset2 - Recognition: `https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)`

   (b) *Step 2*: Perform the required pre-processing steps. Note: If your data is imbalanced, use the down-sampling technique.

   (c) *Step 3*: The test dataset includes indices: 0, 5, 10, 15, 20, ... (the multiples of 5) and the rest of the data points will be your training dataset.

   (d) *Step 3*: Try the DT, LR, KNN, and NB classifiers from *sklearn* package (with random_state = 123, if applicable). Tune the hyper-parameters using CV. Note: You should not use the test set for validation step or for selecting the optimal values for the hyper-parameters.

   (e) *Step 4*: Calculate the testing accuracy, confusion matrix, precision, recall, and F-score for your classifiers and pick the best model for this problem.

   (f) *Step 5*: This time, normalize your data (Hint: use sklearn.preprocessing.scalar(), StandardScaler(), MinMaxScaler(), or MaxAbsScaler()). Does the normalization step improve your testing results?

2. (Bonus - 15 points) In this problem, we will learn how to work with a popular technique for dimension reduction, called Principal Component Analysis (PCA).

(a) Briefly explain what this method is.

(b) From sklearn.datasets import load-digits (Each data point is a 8x8 image of a digit (64 features)). Split your data into train(80% of data) and test(20% of data) via random selection.

(c) Use PCA from sklearn package to reduce the dimensionality of this dataset to 10.

(d) Print "explained-variance-ratio-" and explain what represents.

(e) Train a Logistic Regression model for the original data, and the transformed data. Fine tune the hyper-params using CV. Select the best models. Test them on the test set and compare the results. Explain why they are different.

(f) Repeat the experiments with 5 components.

(g) Compare the results of your trained classifiers with 10 and 5 components. Explain why they are different.