Machine Learning for Engineering Applications
Homework #4
Fall 2023

Due: **10/18/2023 by 11:59 PM – submit your zip file to CANVAS**

Gas Turbine Dataset

- Dataset info and download: https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set

Part 1

- Use *k-means++* to observe clusters in the data using the LEAP cluster
- Determine the number of centroids by using the Elbow Method (*provide the plot*) for the 2011 dataset
- Use the correct number of centroids and plot the clusters with its centers and silhouettes for each individual year
- Determine the distortion score and save it to a text file for each individual year

Submit:

- Python file(s) (.py)
- The Elbow Method images (.png or .jpg)
- Cluster images (.png or .jpg)
- Silhouette images (.png or .jpg)
- Text file with the distortion scores (.txt)
- The SLURM/.OUT file(s) used for the LEAP cluster execution

Part 2

- Use *k-means++* to observe clusters in the data using the LEAP cluster
- Combine all the mini datasets into a single dataset
- Determine all the same requirements from **Part 1**

Submit:

- Python file(s) (.py)
- The Elbow Method image (.png or .jpg)
- Cluster image (.png or .jpg)
- Silhouette image (.png or .jpg)
- Text file with the distortion score (.txt)
- The SLURM/.OUT file(s) used for the LEAP cluster execution

Part 3

- Use the *Agglomerative technique* to determine the hierarchical tree using the LEAP cluster
- Determine the dendrogram plot for **Part 2**

Submit:

- Python file(s) (.py)
- Dendrogram image (.png or .jpg)
- The SLURM/.OUT file(s) used for the LEAP cluster execution

Rubric per Part and Overall:

- Header in the code (5pts)
- Comments in the code (10pts)
- Running code without errors (20pts)
- Executes requirements & produces required output information (65pts)

Template:

```
#*****************************************************************************
# Damian Valles
# ML - HW#1
# Filename:  hw1-perceptron.py
# Due:  Sept. 6, 2023
#
# Objective:
# To demonstrate header and comment expectations for assignments for the semester.
#*****************************************************************************
#Importing all required libraries
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Perceptron

#Downloading the dataset, creating dataframe
iris = datasets.load_iris()

#Separating data & target information
X = iris.data[:, [2, 3]]
y = iris.target

#Data split for training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)

#Scaling training data
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)

#Creating perceptron with hyperparameters
ppn = Perceptron(max_iter=40, eta0=0.01, shuffle=True)

#This is training the model
ppn.fit(X_train_std, y_train)

#Scaling test data
sc.fit(X_test)
X_test_std = sc.transform(X_test)

#Testing the model data
y_pred = ppn.predict(X_test_std)
```