Machine Learning for Engineering Applications
Homework #2
Fall 2023

Due: **9/18/2023 by 11:59 PM – submit your zip file to CANVAS**

2.4 GHZ Indoor Channel Measurements Data

- Dataset info and download: https://archive.ics.uci.edu/dataset/480/2+4+ghz+indoor+channel+measurements

Part 1

- Use a *scikit-learn SVM model* to classify all targets.
- **Determine** which features and parameters will provide the best outcomes using PCA or LDA
- From the dataset:
    - For each location, choose a single *loc_number* folder (use this one for all models).
    - Within the folder, combine all the CSV files into a single file for that location.
        - Label the data accordingly to the dataset information.
    - You can either keep it separate or join all samples to a large master CSV file.
- In your program, print out the training and test accuracy values and the best features values via the PCA or LDA to a text file from the best model experiment.
- Generate the t-SNE and UMAP images.
- Generate the training and testing Plot Decision image.

Part 2

- Use a *scikit-learn k-NN model* to classify all targets.
- Same requirements from Part 1 – you should use the same generated dataset.

Part 3

- Use a *scikit-learn decision tree model* to classify all targets.
- Same requirements from Part 1 – you should use the same generated dataset.

Part 4

- Use a *scikit-learn random forest model* to classify all targets.
- Same requirements from Part 1 – you should use the same generated dataset.

Rubric per part:

- Header in the code (5pts)
- Comments in the code (10pts)
- SLURM bash file (5pts)
- Running code without errors (20pts)
- Executes requirements & produces required output information (60pts)
- Each part: 100 pts per part
- Total: the average of all part grades

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)
```