Machine Learning for Engineering Applications
Homework #1
Fall 2023

Due: **9/6/2023 by 11:59 PM – submit your zip file to CANVAS**

Dataset

- Use the provided dataset on CANVAS – "Data_User_Modeling_Dataset.xls"
- It has 3 tabs: 1) Information, 2) Training_Data, 3) Test_Data

Part 1

- Use a *scikit-learn perceptron model* to classify all labels
- Use the "**STG**" and "**PEG**" features to train, predict, and plot the classification.
- In your program, print out the training and test accuracy values to a text file
- Plot the classification outcome, save it as an image for submission.
- Generate and log your excel sheet to record the parameter changes vs. test accuracy (min 10 test performed to find best accuracy)
    - Find the highest test accuracy by tuning the model parameters.

Part 2

- Use a *scikit-learn logistic regression model* to classify all labels.
- Use the "**SCG**" and "**STR**" features to train, predict, and plot the classification.
- In your program, print out the training and test accuracy values to a text file
- Plot the classification image, save it as an image for submission.
- Generate and log your excel sheet to record the parameter changes vs. test accuracy (min 10 test performed to find best accuracy)
    - Find the highest test accuracy by tuning the model parameters.

Things to think about…

- Next page: it's a template in how you need to structure and organize your code for each part of the assignment.
- Make sure you include comments (#*red lines*) for each Python file (part of the grade)
- Make sure you include the header for Python file (part of the grade)
- Make sure each Python file runs on LEAP! (**big** part of the grade)
- Make sure each Python file produces the output values (part of the grade)
- Make sure each Python file produces the output plots (part of the grade)
- Submit all your required files **in one zip-file**.
- Do not send the dataset…
- **Remember**: There is a *scikit-learn website* with all of the documentation & everything else you can **Google**.

Submit:

- Python file (.py) per part
    - Do **NOT** submit .ipynb (Jupyter Notebook) files
- Excel files (.xlsx)

- Best classification plot images (.png or .jpg) per part
- Text files with the best accuracy values (.txt) per part
- SLURM bash script per part

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)
```