EE2211: Spring 2023
Assignment 2 (15%)
Deadline: 17th Mar 2023 at 23:59 (Friday of Week 9)

Tasks: Write a single python file for multi-class classification:

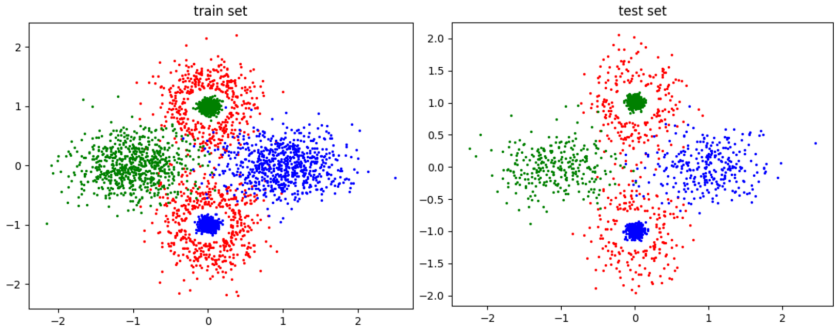
1) **Data Splitting:** The dataset is available in the template file "A2_StudentMatriculationNumber.py", which contains 3 classes of training samples for classification. Please split the dataset into two sets: 70% of samples for training, and 30% of samples for testing.
   NOTE 1: Please use "from sklearn.model_selection import train_test_split" to split the data into the training and test sets.
2) **One-hot Encoding:** Generate one-hot encoded labels for both the training set and the test set.
3) **Polynomial Features:** Use the training and test samples to generate polynomial features (use "from sklearn.preprocessing import PolynomialFeatures") with different orders from 1 to 30.
   NOTE 2: The offset/bias term will be automatically generated by PolynomialFeatures.
4) **Classification:** Use ridge regression with λ=0.0001 for classification (based on one-hot encoding) and compute the classification accuracies on the training and test sets with different orders. Please find the best classifier (polynomial order) according to the classification accuracy.
   NOTE 3: If the number of rows in the training polynomial matrix (also known as the design matrix) is less than or equal to the number of columns, then use the dual form of ridge regression (Lecture 6). If not, use the primal form (Lecture 6).

Outputs: Please submit a single python file with filename "A2_StudentMatriculationNumber.py". It should contain a function `A2_MatricNumber` that returns the following **numpy.array or scalar**:

- **X_train**: the training data matrix with dimensions (number_of_training_samples × 2). (1%)
- **Y_train**: the training targets (containing values 0, 1 and 2) of length number_of_training_samples. (1%)
- **X_test**:the test data matrix with dimensions (number_of_test_samples × 2). (1%)
- **Y_test**: the test targets (containing values 0, 1 and 2) of length number_of_test_samples. (1%)
- **Y_train_onehot**: one-hot target matrix (containing only values 0 and 1) with dimension (number_of_training_samples × 3).(1%)
- **Y_test_onehot**: one-hot target matrix (containing only values 0 and 1) with dimension (number_of_test_samples × 3).(1%)
- **best_order**: the best polynomial order (an integer scalar). (1%)
- **best_P_train**: the training polynomial features obtained with the best order. (1%)
- **best_P_test**: the test polynomial features obtained with the best order. (1%)
- **best_wp**:the estimated coefficients of classifier obtained with the best order. (2%)
- **best_train_acc**: the training accuracy (accuracy = #correct / #total) obtained with the best order. (2%)
- **best_test_acc**: the testing accuracy (accuracy = #correct / #total) obtained with the best order. (2%)

Step-by-step instructions:

0. **Python Library Installation:** pip / conda install scikit-learn matplotlib numpy
1. **Data Splitting:** In the template file "A2_StudentMatriculationNumber.py", we use the function "make_dataset" to generate a 2-D dataset (X, Y) with N samples for classification. X is a [N × 2] matrix and Y is a label vector of length N. Please split the dataset to into two sets: 70% for training set (X_train, Y_train) and 30% for testing set (X_test, Y_test). Visualization code is already implemented in the template, which automatically saves two figures "./train_set.png" and "./test_set.png" to your current folder.



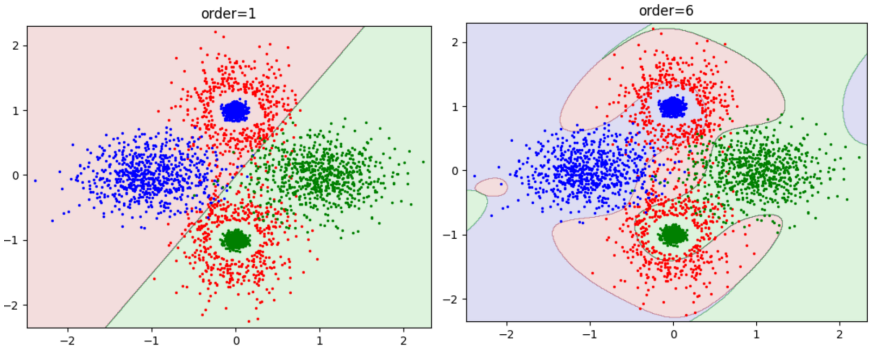
2. **One-hot Encoding:** In this step, we transform the real-valued labels in Y_train and Y_test into one-hot encoded vectors. To be precise, label 0 should be transformed to [1, 0, 0], label 1 should be transformed to [0, 1, 0], and label 2 should be transformed to [0, 0, 1]. The encoded labels Y_train_onehot and Y_test_onehot should be matrices with dimensions (number_of_samples × 3).

3. **Polynomial Features:** The dataset in this assignment is not linearly separable. In this case, the use of polynomials can help map the 2-D data into a higher-dimensional space, where the training samples may be separable. In this step, let us transform our 2-D data (X_train, X_test) to polynomial features (P_train, P_test) using "sklearn.preprocessing.PolynomialFeatures" with different orders. For example:

A polynomial function of order 2 with $d = 1$ variables

$$f_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2 \qquad \mathbf{w} = (w_0, w_1, w_2)$$

4. **Classification:** Classification is performed on the polynomial features (P_train, P_test) and the one-hot encoded labels (Y_train_onehot, Y_test_onehot). With polynomial orders from 1 to 30, you will eventually get 30 classifiers with coefficients "wp". **Please choose the best classifier** according to your training or testing accuracy and return the corresponding "best_order", "best_P_train", "best_P_test", "best_wp", "best_train_acc", "best_test_acc" as mentioned in the "Output" section. In this part, code for visualization is provided to illustrate the non-linear decision boundaries of classifiers. All figures that are automatically generated will be saved as "decision_boundary_order=#ORDER.png".



Submission:

Please use the python template provided to you. Do not comment out any lines. Remember to rename both "A2_StudentMatriculationNumber.py" and "A2_MatricNumber" using your student matriculation number. For example, if your matriculation ID is A1234567R, then you should submit "A2_A1234567R.py" that contains the function "A2_A1234567R". Please do NOT zip/compress your file. Please test your code at least once.

Points will be deducted if instructions are not followed. The way we would run your code will be as follows:

```
>> import A2_A1234567R as grading
>> X_train, Y_train, X_test, Y_test, Y_train_onehot, Y_test_onehot,
best_order, best_P_train, best_P_test, best_wp, best_train_acc,
best_test_acc = grading.A2_A1234567R()
```

NOTE: The PNG files used for visualizing your results will not affect the final grading and are only intended for you to verify your program. Visualization requires the following variables to be defined in your program: X_train, Y_train, X_test, Y_test, Y_train_onehot, Y_test_onehot, order, P_train, P_test, wp, train_acc, and test_acc. If your program does not generate any images, please check your variable names to ensure they are correct.

Gentle Reminders:

1. Please make sure you replace "StudentMatriculationNumber" and "MatricNumber" with your matriculation number!
2. Date of release: 20th Feb 2023 (Monday of Recess Week)
3. Submission: Canvas/EE2211/Assignments/Assignment 2
4. The submission folder in Canvas will be closed on **17th Mar 2023 at 23:59** (Friday of Week 9) sharp! No extensions will be entertained.