

## Assignment No. 2

EECS 658

Introduction to Machine Learning

Due: 11:59 PM, Thursday, September 19, 2024

Submit deliverables in a single zip file to Canvas

Files in other formats (e.g., .tar) will not be graded

Name of the zip file: FirstnameLastname\_Assignment2 (with your first and last name)

Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment2

### Deliverables:

1. Copy of Rubric2.docx with your name and ID filled out (do not submit a PDF)
2. Python source code for CompareMLModels
3. Screen print showing the successful execution of CompareMLModels
4. Answers to the following questions:
  - a. Based on accuracy which model is the best one?
  - b. For each of the 6 other models, explain why you think it does not perform as well as the best one.

### Assignment:

- For this assignment, we are going to compare how well different ML classifiers classify the iris dataset.
- Write a Python program called CompareMLModels that does the following:
  - Uses 2-fold cross-validation to produce a test set of 150 samples of the iris data set with the following ML models:
    - Linear regression (LinearRegression)
    - Polynomial of degree 2 regression (LinearRegression)
    - Polynomial of degree 3 regression (LinearRegression)
    - Naïve Bayesian (GaussianNB)
    - kNN (KNeighborsClassifier)
    - LDA (LinearDiscriminantAnalysis)
    - QDA (QuadraticDiscriminantAnalysis)
    - Remember 2-fold cross-validation involves:
      - Dividing the data set into 2 folds
      - Training the model with fold 1
      - Testing the model with fold 2
      - Training the model with fold 2
      - Testing the model with fold 1
      - Concatenating the test results from the 2 folds to get a test set of 150 samples.
  - For each of the 7 models the program should display (with a label before each model's display indicating which model the results are for):
    - Confusion matrix
    - Accuracy metric
  - If the values in your confusion matrices do not add up to 150, then you did something wrong.

Rubric for Program Comments		
Exceeds Expectations (90-100%)	Meets Expectations (80-89%)	Unsatisfactory (0-79%)
Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line.	Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line.	Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line.

#### Adequate Prologue Comments:

- Name of program contained in the file (e.g., EECS 658 Assignment 1)
- Brief description of the program, e.g.,
  - Check versions of Python & create ML “Hello World!” program
- Inputs (e.g., none, for a function, it would be the parameters passed to it)
- Output, e.g.,
  - Prints out the versions of Python, scipy, numpy, pandas, and sklearn
  - Prints out “Hello World!”
  - Prints out the overall accuracy of the classifier.
  - Prints out the confusion matrix.
  - Prints out the P, R, and F1 score for each of the 3 varieties of iris.
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- Author’s full name
- Creation date: The date you first create the file, i.e., the date you write this comment

#### Adequate comments summarizing major blocks of code and comments on every line:

- Provide comments that explain what each line of code is doing.
- You may comment each line of code (e.g., using `//`) and/or provide a multi-line comment (e.g., using `/*` and `*/`) that explains what a group of lines does.
- Multi-line comments should be detailed enough that it is clear what each line of code is doing.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

#### Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
  - Your comments must be significantly different from your collaborators.
  - More scrutiny will be applied to grading your comments in particular explaining the code “in your own words”, not the source’s comments (e.g., ChatGPT’s comments).

- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.