

### Assignment No. 3

EECS 658

Introduction to Machine Learning

Due: 11:59 PM, Thursday, October 3, 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\_Assignment3 (with your first and last name)

Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment3

#### Deliverables:

1. Copy of Rubric3.docx with your name and ID filled out (do not submit a PDF)
2. Python source code for CompareMLModelsV2
3. Screen print showing the successful execution of CompareMLModelsV2
4. Answers to the following questions for CompareMLModelsV2:
  - a. Based on accuracy which model is the best one?
  - b. For each of the 11 other models, explain why you think it does not perform as well as the best one.
5. Python source code for dbn.py
6. Screen print showing the successful execution of dbn.py
7. Answers to the following questions about dbn.py:
  - a. Does the program use k-fold cross-validation?
  - b. What percentage of the data set was used to train the DBN model?
  - c. How many samples are in the test set?
  - d. How many samples are in the training set?
  - e. How many features are in test set?
  - f. How many features are in the training set?
  - g. How many classes are there?
  - h. List the classes.

Hint: Use the Python shape method and array slices to inspect the dataset (e.g., `digits.data`, `digits.target`, `X`, `Y`, `X_train`, `X_test`, `Y_train`, `Y_test`) like this: `print(dataset.shape)` and `print(dataset[1:20,1:])`.

Note: This program uses raw numpy arrays (not pandas dataframes like we use with the Iris dataset) so the X arrays contain only the features. The X pandas dataframes we use for the Iris dataset contain the features plus the class in the last column.

#### Assignment:

- This assignment has two parts:
- Part 1: Expand our comparison ML classifiers to include SVM, Decision Tree, Random Forest, ExtraTrees, and Neural Network.
  - Enhance your CompareMLModels program and call it CompareMLModelsV2 so that it includes the SVM, Decision Tree, Random Forest, ExtraTrees, and Neural Network.
  - It should now do 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:
    - Naïve Bayesian (NBClassifier)
    - Linear regression (LinearRegression)
    - Polynomial of degree 2 regression (LinearRegression)
    - Polynomial of degree 3 regression (LinearRegression)
    - kNN (KNeighborsClassifier)
    - LDA (LinearDiscriminantAnalysis)
    - QDA (QuadraticDiscriminantAnalysis)
    - SVM (svm.LinearSVC)
    - Decision Tree (DecisionTreeClassifier)
    - Random Forest (RandomForestClassifier)
    - ExtraTrees (ExtraTreesClassifier)
    - NN (neural\_network.MLPClassifier)
  - For each of the 12 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.
- Part 2: Implement a deep learning DBN that recognizes handwritten digits from the MNIST dataset as follows:
  - Go to: <https://github.com/albertbup/deep-belief-network>
  - Download the zip file
    - Click on green Code button to the right
    - Click on Download ZIP
  - Open the zip file
  - Copy and paste the dbn folder located in the main deep-belief-network-master folder into the same directory as your Python code.
  - Scroll down and copy the Python program in Overview section that starts with "import numpy as np" and ends with "print('Done.\nAccuracy: %f % accuracy\_score(Y\_test, Y\_pred))"
  - Paste it into a file in the same directory as the dbn folder you just downloaded and name the file dbn.py.
  - dbn.py and the directory dbn should be at the same level
  - dbn.py should not be in the directory dbn
  - The code has two imports from SupervisedDBNClassification.
  - Use the one "from dbn import SupervisedDBNClassification" and comment out the other one.
  - Note: The sample code uses "from dbn.tensorflow import SupervisedDBNClassification" and has "from dbn import SupervisedDBNClassification" commented out.
  - You should now be able to double click on dbn.py and it will run
  - You do not have to comment the DBN code. (However, it might be a good way to understand the code).

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.