Statistical Methods in Artificial Intelligence

Assignment 4

Deadline : 11 November 2023 11:55 P.M

Instructor : Prof Ravi Kiran Sarvadevabhatla

November 3, 2023

1 General Instructions

- Your assignment must be implemented in Python.
- Submit your assignment as a folder with a Jupyter Notebook file (.ipynb).
- While you're allowed to use ChatGPT for assistance, you must explicitly declare in comments the prompts you used and indicate which parts of the code were generated with the help of ChatGPT.
- Plagiarism will only be taken into consideration for code that is not generated by ChatGPT. Any code generated with the assistance of ChatGPT should be considered as a resource, similar to using a textbook or online tutorial.
- The difficulty of your viva or assessment will be determined by the percentage of code in your assignment that is not attributed to ChatGPT. If during the viva if you are unable to explain any part of the code, that code will be considered as plagiarized.
- Clearly label and organize your code, including comments that explain the purpose of each section and key steps in your implementation.
- Properly document your code and include explanations for any non-trivial algorithms or techniques you employ.
- Ensure that your Jupyter Notebook is well-structured, with headings, subheadings, and explanations as necessary.
- Your assignment will be evaluated not only based on correctness but also on the quality of code, the clarity of explanations, and the extent to which you've understood and applied the concepts covered in the course.
- Make sure to test your code thoroughly before submission to avoid any runtime errors or unexpected behavior.

- The Deadline will not be extended.
- Moss will be run on all submissions along with checking against online resources.
- We are aware how easy it is to write code now in the presence of ChatGPT and Github Co-Pilot, but we strongly encourage you to write the code yourself.
- We are aware of the possibility of submitting the assignment late in github classrooms using various hacks. Note that we will have measures in place for that and anyone caught attempting to do the same would be give zero in the assignment.

2 Tasks:

The tasks in Section 3 and Section 4 of the assignment need to be performed on both regression and classification tasks. The datasets for these tasks are:

- **Classification**: We use the **Wine Quality Dataset** that provides information on various contents of the wine and a general quality index(needs to be predicted).
- **Regression**: We use the **Boston Housing Dataset**, which provides a variety of features and the price of the house (needs to be predicted).

Note that creating proper modularised code i.e. creating appropriate functions that make the code short and precise is a necessity for this assignment.

3 Ensemble Learning

In this task we shall deal with two major ensemble learning methodologies: Bagging and Stacking.

3.1 Pre-Requisite

Implement the following types of models for the tasks at hand. Make sure you tune over necessary hyper-parameters and select the best-performing model.

- Decision Trees (Can use sklearn)
- Logistic and Linear Regressor for Classification and Regression respectively.
- Multi-Layer Perceptron

Note that this list of models would henceforth be referred to as **List 1 models**

3.2 Bagging

For both the classification and regression tasks do the following tasks:

1. Design a function that performs the bagging methodology of ensemble learning by taking as input the following parameters:
    - Base Estimator Model
        - **List 1 models**
    - Number of Estimators
        - Variable according to the base-estimator
    - Fraction/Number of Samples
        - 0.15, 0.25, 0.5, 0.75, 1.0
    - Bootstrap
        - True: Sampling with replacement is performed
        - False: Sampling without replacement is performed
    - Voting Mechanism:
        - Hard Voting: Every classifier has equal weight in the final output
        - Soft Voting: The final output is weighed by the confidence of the base estimator models.
2. For ease, assume the best hyper-parameters from sub-task 1 and train ensemble models across all combinations of parameters specified above and report the best performing models.
3. Plot a heatmap for the accuracies obtained by each class of base estimator models across Fraction of Samples and Number of Estimators (Keep other parameters constant and to your choice)
4. Compare the performance of each model in **List 1 models** with the best-performing ensemble model of the same class with a single side-by-side histogram

Note:

1. Voting Procedure: Every classifier specified above provides a confidence probability that can be used for soft voting. For regression, divide the dataset into train, val, and test splits and use the performance on the val set as an insight for confidence.

3.3 Stacking

Stacking is a general procedure where a learner is trained to combine the individual learners. Here, the individual learners are called the **level-0** learners, while the combiner is called the **level-1** learner, or meta-learner.

1. Design a function that performs the bagging methodology of ensemble learning by taking as input the following parameters:
    - Level-0 estimators:
        - **List 1 Models**
    - Level-1 estimators:
        - Logistic and Linear Regressor for classification and regression respectively
        - Decision Tree
    - Stacking Methodologies:
        - Stacking
        - Blending
2. For ease, assume the best hyper-parameters from sub-task 1 and train ensemble models across all combinations of parameters specified above and report the best-performing models.
3. Compare the accuracies and the training time of the best-performing models of Bagging and Stacking ensembles of each Base Estimator Model class.

4 Random Forest vs Boosted Trees

In this section, we shall compare the results of Random Forests (a Bagging technique) with multiple Boosting Techniques over the tasks of classification and regression specified above.

1. Train a Random Forest Classifier and Regressor along the same lines of the Bagging exercise (for decision tree) and report the best-performing hyper-parameters.
2. Compare the results of the best Random Forest with the following Boosted Decision Trees. Also, experiment with the number of estimators and plot their training times and accuracies.
    - Decision Trees + AdaBoost
    - Gradient Boosted Decision Trees
3. Provide an analysis of the mistakes of these models and try to explore and explain the feature similarity of the common mistakes