Pattern Recognition and Machine Learning

(Winter 2022)

Assignment 3: Bagging & Boosting

Early Bird Submission Deadline: Feb 6, 2022, 23:59 (No penalty)
Late Submission Deadline: Feb 7, 2022, 23:59 (20% penalty)
Final deadline: Feb 8, 2022, 23:59 (addl. 10% penalty, total penalty = 30%)

Guidelines for submission
1. Perform all tasks in a single colab file.
2. Create a report regarding the steps followed while performing the given tasks. The report should not include excessive unscaled preprocessing plots.
3. Try to modularize the code for readability wherever possible
4. Submit a zip with the colab file [.ipynb] and report [.pdf] on the classroom
5. Submit the .py file on the floated form for the laboratory
6. Plagiarism will not be tolerated

Question 1. **[50 marks]**

Ensemble Learning, a form of meta-learning, is a machine learning paradigm where multiple learners are trained to solve the same problem. In this assignment, you will code up the meta-learning algorithms Bagging (short for Bootstrap Aggregating) and Boosting.

Note: Kindly do perform the splitting of the dataset and other necessary tasks (cleaning, normalization etc.) based on prior lab experience.

Download Housing Dataset and perform the following tasks (Each task carries 5 marks)

1) Use a simple Decision Tree regressor to predict the price of a house (without any validation) and report the accuracy.
2) Perform 5-fold cross-validation to determine what the best max_depth would be for a single regression tree using the entire 'Xtrain' feature set.
3) Visualize and summarize the results across the validation sets.
4) Apply bagging to create different training datasets (select n_estimators = 10)
5) Train on different dataset to obtain different decision trees.
6) Summarize how each of the separate trees performed (both numerically and visually) using *R-squared score* as the metric. How do they perform on average?

7) Combine the trees into one prediction and evaluate it using *R-squared score.*
8) How will the results above change if 'max_depth' is increased? What if it is decreased?
9) Train random forest regressor, report mean squared error and mean absolute error. (from sklearn.ensemble import RandomForestRegressor)
10) Train Adaboost regressor, report mean squared error and mean absolute error. (from sklearn.ensemble.AdaBoostRegressor)

Note: For task 1-8 you need not have to use the sklearn library, however for 9 and 10 you can make use of sklearn library.

Question 2. **[50 marks]**

Boosting: - Boosting is an ensemble modeling technique which attempts to build a strong classifier from the number of weak classifiers. It is done by building a model using weak models in series. First, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

Note: Kindly do perform the splitting of the dataset and other necessary tasks (cleaning, normalization etc.) based on prior lab experience.
For installing XgBoost write the following command in one of the colab cell.
!pip install xgboost

For installing LightGBM write the following
!pip install lightgbm

Download Breast_cancer Dataset and perform the following tasks

1) Use a simple Decision Tree classifier to predict the outcome (without any validation) and report the accuracy.[5]
2) Perform 5-fold cross-validation to determine what the best max_depth would be for a single regression tree using the entire 'Xtrain' feature set.[5]
3) Visualize and summarize the results across the validation sets.[5]
4) Implement XGBoost in which subsample=0.7 and max_depth=4. [10]
5) Print the accuracy on the training set and test set. [5]
6) Implement LightGBM with max_depth value as 3 and choose different value for num_leaves.[7]
7) Analyse the relation between max_depth and num_leaves, and check for which value the model starts overfitting?[8]
8) Report which parameters can be used for better accuracy and also which parameter can be used for avoiding overfitting.[5]

Note: For task 1-3 you need not have to use the sklearn library.