CITS5508 Machine Learning
Semester 1, 2020

Lab Sheet 4

Assessed, worth 10%. Due: 11:59pm, Monday 4th May 2020

1 Outline

This lab sheet consists of two small projects. In the first project, you should train an AdaBoost Regressor and a Gradient Boosting Regressor and compare their performances. In the second project, you should train two Random Forest Regressors using the original data and using the reduced-dimensional data and compare their performances. This lab sheet is a good practical exercise to test your understanding of the techniques covered in Chapter 7.

2 Submission

Name your Jupyter Notebook files as **lab04proj1.ipynb** and **lab04proj2.ipynb** for the two projects below and submit them to *cssubmit*: https://secure.csse.uwa.edu.au/run/cssubmit?p=np before the due date and time shown above. You can submit your files multiple times. Only the latest version will be marked.

3 Project 1

In the UCI Machine Learning Repository website above, there is a dataset on white wine:

https://archive.ics.uci.edu/ml/datasets/wine+quality

which has 12 attributes and a column that describes the *quality* rating of the wine. For any new test instance, we want the regressors to predict this value. In this project, you should work on the *white wine* data file only. The *winequality-white.csv* can be downloaded directly from the link below:

https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/

NOTE: Save the downloaded file (*winequality-white.csv*) to the same directory with your notebook file. Do not rename or modify the file in any way. As the file uses the semi-colon character (';') to separate the columns, you will need to use the *delimiter* argument to read in the contents.

This project includes the following tasks:

- Firstly, inspect the data and perform data cleaning if needed. Do a random 85/15 split on the data to form a training set and a test set. The same training set and test set must be used for both regressors; however, feature scaling can be optionally and independently performed for them.
- For the AdaBoost Regressor, use an SVM regressor with an RBF kernel as the base estimator. Both the AdaBoost and Gradient Boosting Regressors should have 6 estimators only, as, due to the sequential nature of these algorithms, it takes a long time to train the models.
- Note that as the wine quality ratings must be integers, the outputs from the regressors should be firstly rounded to the nearest integers and the rounded integers should be considered as the final predictions. For instance, if the predicted quality rating of an instance is 6.8 then it should be rounded to 7; if the predicted quality rating is 6.2 then it should be rounded to 6.

- Your Python code should report the MSEs (both numerically and graphically) of all the intermediate models (from the first five estimators) and the final model (from the last estimator) of each regressor on the training set and the testing set. In addition, show (using histogram(s) or bar chart(s)) the prediction errors of the final model of each regressor on the training and testing sets. In this last part of the illustration, you should use the *raw errors* (i.e., the errors can be negative, zero, or positive integers[1])
- Compare the performance of the regressors and provide a brief discussion.

4 Project 2

The **Abalone** dataset

http://archive.ics.uci.edu/ml/datasets/Abalone

is a small dataset suitable for regressing (predicting) the numbers of rings of new abalone instances. The dataset contains 8 attributes. The ring values are what we want the two regressors to predict.

The data and description about the attributes can be downloaded, respectively, from

http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data
http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.names

NOTE: Download the file *abalone.data* and save it to the same directory with your .ipynb file. Do not rename the file. Although the file name ends with *.data*, you should be to read it the same way as a *.csv* file.

- This project includes the tasks below. Read in the contents of the file and perform the usual data inspection and cleaning. Same as Project 1, do an 85/15 random split to form the training and testing sets. Implement a Random Forest regressor to predict the *ring* values of abalones in the test set. You should also set various hyperparameter values carefully to avoid overfitting the training data. Similar to Project 1, as the ring values must be integers, a rounding operation on the regressor's outputs should be carried out. Report (both numerically and graphically) the MSEs of the predictions on the training and testing sets. Finally, use histogram(s) or bar chart(s) to show the raw errors of the predictions on both sets.
- Your next task is to use the *feature importances* obtained from the training process to trim the feature dimension of the data. In your code, retain just over 95% of the feature importance. Repeat the process above on the reduced-dimensional data.
- Finally, compare the performance of the two versions of your regressor.

5 Penalty on late submissions

See the URL below about late submission of assignments:
https://ipoint.uwa.edu.au/app/answers/detail/a_id/2711/~/consequences-for-late-assignment-submission

[1] Given a test instance, if the error is negative (or positive), it means the regressor underestimates (or overestimates) the quality rating. Using the raw errors instead of absolute errors, we can see whether the regressor tends to overestimate or underestimate and by how much. e.g., if the errors are $[+1, -2, 0, -2, -2]$ for a test set of 5 instances, it means the regressor overestimates the quality rating by 1 one time, underestimates the quality rating by 2 three times, and correctly predicts the quality rating only once.