CITS5508 Machine Learning
Semester 1, 2022

Lab Sheet 4

Assessed, worth 15%. Due: 11:59pm, Friday 29th April 2022

1 Outline

This lab sheet consists of two small projects. In the first project, you should train a Voting regressor for a small regression problem; in the second project, you should train (i) two Random Forest models using the original data and using the reduced-dimensional data respectively; and (ii) a Bagging regressor. This lab sheet is a good practical exercise to test your understanding of the techniques covered in Chapters 6–7.

2 Submission

Put your implementation for the two projects below into a single Jupyter Notebook file. You should name your file as **lab04.ipynb** and submit it to cssumit (https://secure.csse.uwa.edu.au/run/cssubmit) before the due date and time shown above. You can submit your file multiple times. Only the latest version will be marked.

3 Project 1

The **Concrete Slump Test** dataset

https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test

is a small dataset suitable for regression. For any new test instance, our aim is to use our trained model to predict its *28-day Compressive Strength* (Mpa) value. The *slump_test.data* dataset can be downloaded directly from the link below:

https://archive.ics.uci.edu/ml/machine-learning-databases/concrete/slump/

Although the name of the file ends with *.data*, it is actually a *csv* file.
NOTE: Save the downloaded file (*slump_test.data*) to the same directory with your Jupyter Notebook file. Do not rename or modify the file in any way.

Tasks

This project includes the following tasks:

- Firstly, inspect the data and perform data cleaning if needed. As described on the web page above. there are 7 input variables (i.e., feature columns) and 3 output variables. Our interest is the *28-day Compressive Strength* output, so the other two output variables should be dropped. Perform some basic visualisation and determine whether any additional features should be removed also. You should write a Python function for this data cleaning step.
- Perform a 80/20 random split on the dataset to form a training set and a test set. For your Voting regressor, use the following as the 3 base estimators: (i) a linear SVM regressor, (ii) a linear regressor (using the `LinearRegression` class), and (iii) a Stochastic Gradient Descent regressor. You can manually tune a few hyperparameters for each of these regressors.
- Train the base estimators and the Voting regressor on the training set and compare their predicted *28-day Compressive Strength* values for the test set. Report their RMSEs and illustrate the predicted values versus the ground truth values of all the test instances.

4 Project 2

An *Abalone* dataset is available in the UCI Machine Learning Repository website below:

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

It has 4177 instances with 8 attributes and a column that describes the *age*, represented in terms of the *number of rings*, of the abalones. For any new test instance, we want our regressor to be able to predict this value. The dataset in csv format is in the *abalone.data* file which can be downloaded from

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

You should save the file *abalone.data* to the same directory with your Jupyter Notebook file. Do not rename or modify the file in any way.

Tasks

- Read in the contents of the file and perform the usual data inspection and cleaning if needed. Perform appropriate data preprocessing to the data, e.g., you can either drop the text column or convert it into numerical values. Do an 85/15 random split to form a training set and a test set.
- Implement a Random Forest regressor with 500 estimators. You can manually experiment with various hyperparameters such as *min_samples_leaf*, *max_features*, *max_samples*, *bootstrap*, etc. Train your Random Forest regressor on the training set and test it on the test set. Report its RMSE for the predictions on the test set. Note: as the ring values must be integers, the predicted results from your Random Forest regressor must be firstly rounded to the nearest integer before the RMSE computation.
- Use the *feature importances* obtained from the training process to trim the feature dimension of the data. In your Python code, you should retain only those features whose *importance* values are above 5% (i.e., 0.05)[1]. You can either write your own Python code or use the function `SelectFromModel` from the `sklearn.feature_selection` package to work out which feature(s) can be removed.

  Report what features were retained and what features were removed in the above process. What is the total feature importance value that is retained after your dimension reduction step?
- Repeat the training and prediction processes above on the reduced-dimensional data. We expect to see a slight increase of RMSE for the reduced-dimensional data. In many real applications, the feature dimension of the data may be reduced drastically with only a slight increase in the prediction error. We won't see this in this small dataset as the feature dimension is already quite small.
- Finally, compare the performance of the two Random Forest regressors. Illustrate in a diagram or two the prediction errors of all the test instance from one of the above models, e.g., you can try computing the average error for each ring value.

  Do large or small (or both) ring values tend to have large average errors? Is a large error for a ring value related to insufficient training instances for that ring value?
- Implement a Bagging regressor with 500 SVM regressors as the base estimator. Again, choose some reasonable hyperparameter values manually for the SVM regressors. For the Bagging regressor, you should use the same common hyperparameter values (e.g., *max_features*, *max_samples*, *bootstrap*, etc) as your Random Forest regressor. Train your Bagging regressor using the full-dimensional training set and test it using the full-dimensional test set. Report the RMSE of the predictions for the test set. Illustrate in a diagram the predicted ring values versus the ground truth ring values of all the test instances.
- Compare the performance of your first Random Forest regressor with the Bagging 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] If a no features are removed with the 0.05 threshold value, then increase the threshold slightly. For the exercise in this project, we want to experiment with the effect of reducing feature dimension.