CITS5508 Machine Learning
Semester 1, 2019
Lab Sheet 4
Assessed, worth 10%. Due: 23:59pm, Tuesday 23rd April 2019

# 1 Outline

This lab sheet consists of two small projects. The first project asks you to train an *ensemble classifier* and report its performance in terms of its F1 score on the test set. The second project asks you to train and test a *random forest regressor*, report its root mean squared error (RMSE) and experiment with dimension reduction.

This lab sheet is a good practical exercise to test your understanding of the techniques covered in Chapters 7–8.

# 2 Submission

Name your Jupyter Notebook file as **lab04.ipynb** and submit it to LMS before the due date and time shown above. You can submit your file multiple times. Only the latest version would be marked.

# 3 Project 1

In the UCI Machine Learning Repository website above, there is a Parkinsons dataset:

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

which has 23 attributes and two class labels: "healthy" and "unhealthy" (i.e., having the Parkinsons disease). The class labels appear under the *status* column. The data (as a text file in *csv* format) and description about the attributes can be downloaded, respectively, from

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

Your tasks for this project are:

- Download and visualize the data. Perform appropriate data cleaning. Are there any columns that should be removed? Perform this step in Python and provide some explanation.
- Divide the data into training and test sets (using 80/20 split).
- Implement an ensemble classifier (e.g., using the `VotingClassifier` from scikit-learn) comprising of a Logistic Regression classifier and an SVM classifier (linear or with kernel)[1] to predict each patient in the test set whether he/she is healthy or not (having the disease). Each classifier should include a regularization term. You are suggested to use `Pipeline` in your implementation. Explain in your ipynb file the hyperparameters that you use.
- Compare the F1 scores of the two classifiers versus that of the ensemble classifier in your conclusion. Display a plot showing the confusion matrix of the predicted output from the ensemble classifier (you may use the function that you wrote for the previous lab sheet.

[1] Remember that SVM is sensitive to different scales of the features.

# 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 and 29 different ring values. The data (as a text file in *csv* format) 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

Your tasks for this project are:

- Download and visualize the data. You will need to determine whether those ring values having small numbers of instances (e.g., ring values 1, 2, 3) can be grouped with the next ring value (e.g., ring value 4) or whether they should be discarded (similarly for ring values 23-29). Also, how you deal with the text column should be described in your Jupyter Notebook file. The above data cleaning process should be done using Python code. Do not manually modify the `abalone.data` file as we will use the downloaded version to mark your Python code.
- Divide the data into training and test sets (using 90/10 split).
- Train a Random Forest regressor using appropriate hyperparameter values and test it on the test set. Inspect the *feature importance* value of each feature and determine whether you could reduce to work on a lower dimensional feature space. Retrain the classifier by dropping those features whose feature importance values are below a certain threshold[2]. Compare the root mean squared errors (RMSEs) of the two regressors before and after dimension reduction on the test set. Show the absolute prediction error of each test instance in a diagram (**Hint:** Use the `bar` function of `matplotlib.pyplot`).
- Perform PCA on the data (e.g., by retaining only 99% of the variance) and retrain the Random Forest regressor (using the same hyperparameter values) on the reduced dimensional features and test the regressor on the test set. Note that you would need to reduce the dimension of the test features accordingly. Compare the RMSE of this regressor with the two RMSEs above.
- Provide a brief conclusion about the two ways of performing dimension reduction.

# 5 Presentation

Please see lab sheet 2 for tips on the presentation of your `ipynb` files.

# 6 Penalty on late submissions

See the *Unit Outline* about the penalty on late submissions.

[2]For instance, you can sort the feature importance of all the features in decreasing order and keep only those features which add up to just over 95% of the total feature importance.