CITS5508 Machine Learning
Semester 1, 2019
Lab Sheet 3
Assessed, worth 10%. Due: 11:59pm, Friday 5th April 2019

# 1 Outline

This lab sheet asks you to use *Decision Trees* (DT) and *Support Vector Machines* (SVM) for two small projects: one is a classification and the other is a regression. The lab sheet will cover techniques and ideas from Chapters 1–6 of the textbook. For both projects, you should put the data files in the same directory as your Jupyter Notebook file.

# 2 Submission

Name your Jupyter Notebook file as **lab03.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

Your tasks for this project is to train and evaluate a DT classifier and an SVM classifier to learn to classify the Cellular Localization Sites of Proteins on some E. Coli bacteria data. There are just 8 features and 336 instances. There are 8 classes for the target given in the last column of the data file. The data file and a brief description file are available from a web repository managed by the UCI (University of California at Irvine) Centre for Machine Learning and Intelligent Systems. See

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

You will need to think about what to do with non-numerical data. Notice that although the data is not a `csv` file, the `read_csv` function can still read it without problems[1]. Out of the 8 classes, you need to remove those classes having less than 10 instances as it is not possible to classify them. Provide some plots for data visualization after you have successfully read in the data. Perform an 80/20 split of the data to form your training set and test set. To compare the performance of the two classifiers, the same training and test sets should be used for both classifiers. There is no need to do grid search for finding optimal hyperparameters for each classifier. You should be able to get reasonably good classification results (e.g., above 80% accuracy) for this dataset even using the default hyperparameter values. Based on the settings of these hyperparameters, you can then make small adjustment to see if the performance can be improved or not.

For each classifier, report the classification accuracies, F1 scores, and confusion matrices on both the training set and the test set. For each confusion matrix, you are required to display it in a diagram. You can modify the `plot_confusion_matrix` function given by the author (see the `ipynb` file for Chapter 3) so that the tick marks on both axes show the correct class labels and the numbers of the confusion matrix are also displayed.

[1] **Note:** Do not rename the data file or modify the data in the file. In the marking process, we will use the data file downloaded from the UCI website. So any renaming or modification to the data would mean your code won't run when we mark it.

For the SVM classifier, you can use any kernel function. You should repeat the above steps for your SVM classifier on both the raw data and the normalized data. Describe in your Jupyter Notebook file how your SVM classifier performed in both cases. Compare the performance of your DT classifier with the two models of your SVM classifier.

**Hints:**

- When calling the function `read_csv`, there is an optional argument that you can use to specify the character that separates the columns of the data.
- The data cleaning process for this project 2 involves removing some data instances. You can use various functions, such as `loc` and `drop`, from the `pandas.DataFrame` package to remove rows and columns of the `DataFrame` object. However, even after some rows are correctly removed, the index locations of the remaining rows in the `DataFrame` object are not automatically updated. This means that you would still be able to access the removed rows and you would see NAN (meaning *not a number*, i.e., *undefined*) for those rows. To overcome this problem, you will need to explicitly call the `reset_index` function to renumber the index locations. After performing the data split, you should check that you don't have any NAN values in both the training and test sets.

# 4 Project 2

The Bureau of Meteorology (BOM) publishes quite a lot of its data for free. After a bit of searching you should be able to find the record of daily global solar exposure amount, maximum temperature, rainfall, etc, for the Perth Metro weather station via

http://www.bom.gov.au/climate/data/index.shtml

The amount of global solar exposure is the total solar energy for a day falling on a horizontal surface and is useful for owners of solar photovoltaic panels (and solar hot water heaters). One would expect some relationship exists between this amount and the daily maximum temperature and rainfall in different seasons of the year. It would therefore be useful to train a machine learning algorithm to predict one of these values using the other two. Unfortunately, Perth had a very dry year in 2018 and many days have zero rainfall values. So the rainfall data cannot be used for our ML regression problem in this project.

The daily global solar exposure and temperature data for Perth in 2018 have been downloaded and made available on LMS (see the `project2.zip` file). The two `csv` files have 365 instances. The column headings in the files should be self-explanatory. Further explanation about the data can be found in the two `txt` files.

Your tasks for this project is to train and evaluate a DT regressor and an SVM regressor that can predict the maximum temperature value given a month, date, and solar exposure value.

Follow the same procedure as in Project 1 to read in your data, visualize it, split into training (80%) and test (20%) sets. As this is a regression problem, you only need to report the MSEs of the regressors on both the training and test sets.

Again, for the SVM regressor, you can use any kernel and you should evaluate its performance on both raw data and normalized data. Describe in your `ipynb` file which regressor (DT or SVM) gave better predictions.

# 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.