CITS5508 Machine Learning
Semester 1, 2024
Lab sheet 2
(Not assessed)

You will develop Python code for a simple classification task in this lab sheet. Certify that the presentation of your Python notebook is good and that you used the Markdown cells well. Make sure you properly format your plots and results. For instance, all your diagrams/plots should have proper axis labels and titles to help the reader understand what you are plotting. Another example is the confusion matrix; not showing the class names makes the confusion matrix completely useless. Use the lab sheets to learn how to improve the presentation of your notebook, as you will need this in the assessments.

Forest type mapping

In this part, we will use the *training.csv* and *testing.csv* data files supplied on LMS. These files were downloaded and slightly modified from the **Forest type mapping dataset** on the UCI Machine Learning website. You can find more details about the dataset using the link below:

http://archive.ics.uci.edu/ml/datasets/Forest+type+mapping#

The training set (*training.csv*) contains 325 instances of multivariate remote sensing data of some forest areas in Japan. There are four different forest types labelled in the first column (the column heading is '*class*'), as described in the link above. The test set (*testing.csv*) has the same format as *training.csv* and contains 198 test instances.

Tasks

Your tasks for this lab sheet are described below. You can use the file "labsheet3.ipynb" as a guide or create your notebook.

1. Read in the contents of both csv files[1].
2. Inspect what the columns are by displaying the first few lines of the file and by using the info() function. What can be observed?
3. To simplify the classification task, write Python code to remove all the columns whose names begin with `pred_minus_obs`. You should have only 9 features (`b1`, `b2`, $\cdots$, `b9`) left for both the training and test sets.
4. Write Python code to count the number of instances for each class label. Do you have an imbalanced training set?
5. Use appropriate functions to display (visualise) the different features (or attributes/columns). You can also incorporate class information. Display some plots for visualising the data. Describe what you see in your markdown cells.

[1] Since both files are in the same directory as your Jupyter Notebook file, you should be able to read each without any path name. For example `pd.read_csv('training.csv')` should work just fine.

6. We will start with a binary classification on this dataset using examples from two classes: ‘s’ (‘Sugi’ forest) and ‘d’ (‘Mixed deciduous’ forest). Write Python code to have only examples from these two classes in your training and testing sets.

7. Use the *Logistic Regression Classifier* implemented in `sklearn.linear_model` class to perform binary classification on the updated datasets.

8. Inspect some performance indicators. Show the results for:

   (a) The accuracy values for the training set and the test set.

   (b) The confusion matrix on the training and testing set.

   (c) The plot of precision versus recall for the training set.

   What do you see from inspecting these performance indicators? Which threshold values the model is using in parts (a) and (b)? How do precision and recall behave? What threshold would you choose and why?

Optional, but may be covered in assessments.

9. Examine the estimated probabilities and decision boundary of the Logistic Regression model. The tasks are:

   (a) Consider two individual features. You can choose which ones. You should then create a new version of your training set using these two features only.

   (b) Plot your Logistic Regression Classifier’s estimated probabilities and decision boundary (as in Figure 4.23 of the textbook) considering these two individual features. Therefore, you will need to create a Logistic Regression model for each of these two features and inspect the decision boundaries. Hence, you should provide two plots.

Based on your plots, comment about:

- What threshold would you choose for classification based on the predicted probabilities?
- What is the impact of changing the threshold for performance indicators such as precision and recall?
- What can you say about the overlap between classes, and how does this impact classification performance?