CITS5508 Machine Learning
Semester 1, 2024
Lab sheet 4
(Not assessed)

You will develop Python code to continue the classification tasks in the previous lab sheet, but now we will consider a **multiclass classification**. 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 continue to use the *training.csv* and *testing.csv* data files supplied on LMS. This is the same data used in lab sheets 2 and 3. This lab sheet builds on your work in the previous labs.

Tasks

Your tasks for this lab sheet are described below. To perform these tasks, use features b1-b9 and **all classes** in the "Forest type mapping data set".

1. Apply feature scaling using the `StandardScaler`. Remember, you should fit the scalers using the training set only, and the estimated parameters should be used to transform the training and test sets.
2. Use the Support Vector Machine Classifier implemented in the `sklearn.svm.SVC` class to perform multiclass classification using the *one-versus-one* strategy. You should look at the Scikit-learn API for this class and experiment with two hyperparameters: C and `kernel`. You should use Grid Search and 3-fold cross-validation to find the optimal values for these two hyperparameters that maximise the classification accuracy. For other hyperparameters, you can use the API default values.
3. What are the best values of these hyperparameters according to Grid Search? What is the performance in each of the validation folders using these values?
4. Train your model on the entire training set using these optimal values and inspect some performance indicators, such as:
   (a) The accuracy in the training and test sets;
   (b) The confusion matrix on the test set.
5. Repeat the steps 2-4 process *without* doing feature scaling. Compare the results.
6. In lab sheet two, you inspected the meaning and values of the different features used in this data set. Do you think you should do feature scaling in this problem? Why or why not? How do the results you obtained support your conclusions?

7. Repeat tasks 1-4 using `sklearn.linear_model.LogisticRegression` class to implement *Softmax Regression*. You should look at the Scikit-learn API for this class and experiment with Grid Search and 3-fold cross-validation with two hyperparameters: `penalty` and C. Note that C is the inverse of regularization strength for this class.

8. Repeat tasks 1-4 using the K-NN algorithm. Again, your Grid Search should consider different values of K and a 3-fold cross-validation.

9. Report a summary and comment on the results of the three techniques (SVM, Softmax Regression and K-NN).