CITS5508 Machine Learning
Semester 1, 2024
Lab sheet 5
(Not assessed)

You will develop Python code to perform regression tasks using decision tree models. 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.

Abalone dataset

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.

Tasks

This part of the lab sheet includes the following tasks:

1. Read the file's contents and perform some data inspection/exploratory data analysis. You can use histograms, bar plots, or scatter plots. Inspect how the features are related to each other and their relationship with the target feature. Explore the dataset's structure, features, and missing values.
2. Perform appropriate data preprocessing, e.g., you may need to drop or convert the text column into numerical values.
3. Do an 85/15 random split to form training and test sets.
4. Train a Decision Tree Regressor using default hyperparameters. Inspect the obtained tree. Do you think it is reasonable?
5. Select a performance measure and inspect the model performance on the training and test data.
6. Provide a plot with the model predictions and the "true" values for the test set. Comment on the results.
7. Using 5-fold cross-validation, experiment with different hyperparameters (e.g., `max_depth`, `min_samples_split`) and observe their impact on model performance. With the optimal obtained hyperparameters, re-train the model and compare the model's performance on the training and test sets with the one using the default hyperparameters. What changed?

8. For the test set, provide a plot with the predictions of the two models (with default hyperparameters and optimal ones) and the "true" values. Comment on the results.

9. Visualize the fine-tuned decision tree model to understand how it makes predictions. Interpret the tree structure and identify important features.

10. Use an SVM Regressor for the same task, and fine-tune some hyperparameters using a 5-fold cross-validation. Compare the results of both models. Which one performed better on the test set? Why do think that is the case? Have you scaled your features for the SVM Regressor?