Aston University
Machine Learning

Portfolio Task 1: Supervised Learning

Instructions:
In this assessed task, you will be applying algorithms from the first family of machine learning techniques covered in this module: supervised learning. The aim of this task is to test your ability to apply machine learning algorithms to well-specified tasks and to evaluate the performance of these algorithms and to use this evaluation to improve performance.

Details:
Follow the instructions below to complete the portfolio task. The task requires you to carry out some implementation in Python and to provide a short written justification of your choices, of maximum 250 words. The recommended format for submission is a Jupyter notebook, integrating your code and written justification.

Marking:
This portfolio task is worth 15% of the overall module mark.

The mark scheme for the task is as follows:

- **50-59** Solution approaches have been applied to both sub-tasks and, where requested in the task, their performance measured. The approaches taken are broadly correct but may have some flaws in application or methodology. Model evaluation and a justification of chosen approaches have been attempted but shows limited understanding.
- **60-69** Justification in sub-task 1 shows clear understanding of the properties of the chosen algorithms. Multiple solution approaches (algorithms/models/parameter sets) have been applied to the problem in sub-task 2 and have undergone evaluation. Justification for the selected approach is evidence-based and well presented.
- **70-79** The methodology used to compare solution approaches for sub-task 2 is carefully designed and leads to well-supported conclusions. Clear understanding of experimental design is demonstrated.
- **80+** As above, but with additional evidence (for both sub-tasks) of some or all of: attention to quality throughout the implementation, thorough understanding in experimental design, excellent justification.

No specific descriptors are provided for marks below the threshold of 50. Marks in the range **0-49** are allocated where the submitted work has not reached the expectation for the threshold descriptor.

Sub-task 1:

Download the file regression1.csv from Blackboard. It contains 100 pairs of values. The values in the first column are explanatory values/observations. The values in the second column are the corresponding response variable values.

Using Python, apply least squares regression to fit a line to the data. Plot the data and the fitted line on the same axes.

Calculate the coefficient of determination (the R^2 score) of your prediction and print it.

Based on your findings, comment on whether least squares regression is well suited to this dataset. Justify your answer.

Sub-task 2:

Download the file regression2.csv from Blackboard. This dataset contains 500 rows of three values each. The values in the first and second column are explanatory variables/observations and the values in the third column are the corresponding response variable values.

You decide to use a multi-layer perceptron (MLP) with a single hidden layer to fit a model to the data. To implement this, it is recommended that you use scikit-learn's MLPRegressor. Unless you have a good reason to do otherwise, use its default parameter values (except for the size of the hidden layers – see below).

One of the parameters we can set in an MLP is the number of nodes in its hidden layer. The more nodes we have in the hidden layer, the more complexity we can capture in our model. However, you are aware of the connection between model complexity and overfitting and will want to choose a number of hidden nodes which is high enough to capture the complexity in the data, but not so high that your model overfits the data.

Based on the principles discussed in lectures, design a methodology to:

- choose an appropriate number of hidden nodes for an MLP applied to this regression problem,
- train an MLP with this number of hidden nodes and estimate its generalisation error.

Implement your methodology using Python. Explain your methodology and justify your choice of the number of hidden nodes. You may want to produce graphs to help support some aspects of this justification.