

## **Multilayer Perceptrons in sklearn**

### **Introduction**

In the lecture this week, we looked at sklearn's MLP neural network classes (MLPRegressor & MLPClassifier) to train networks based on synthetic house price data.

In this workshop, we will explore these learning algorithms with synthetic and non-synthetic data.

### **Experimenting with synthetic data models**

The lecture referenced two applications to train neural networks based on synthetic house price data. These applications are in Learning Space (1.house price regression model & 2.houseprice classification model) For each application, load them into Pycharm and experiment with the data and hyperparameters to determine the impact on application performance and accuracy.

1. Combine both applications such that they can run with the same house price data and use this to determine which learning approach (classification or regression) will give the best results for different forms of data.
2. Rework the house price generator to create a more even distribution of house data. Use the price\_histogram dictionary to hold and manage this data.
3. Use openpyxl or the csv module to output the price\_histogram dictionary data so that it can be viewed graphically with Excel's visualisation.
4. Rework the classification learning model to use the price buckets as outputs, rather than a single value. The outputs will consist of a number of nodes that relate to price ranges.

### **Experiments with Boston house price data**

In the lecture, I mentioned that the Boston house price data (from last week's sessions) wasn't particularly useful for MLP learning. Refactor your synthetic model to work with the Boston data (ml\_house\_data\_set.csv).

1. As a starting point, analyse the data to see what %age of data comes from each town/region and if any regions are strongly over or under-represented
2. Split the data into sale\_price buckets for each town to see if prices are uniform across regions
3. Replace the town / one\_hot encoding approach with a number look-up for towns
4. Create MLP solutions for this data and see how they perform, in terms of over/under-fitting and general solution performance
5. Look to develop your own approaches to make the Boston dataset produce better results