P1 Boston House Pricing Report

1) Statistical Analysis and Data Exploration

- Number of data points (houses)? 506
- Number of features? 13
- Minimum and maximum housing prices? Min = 5.0, max = 50.0
- Mean and median Boston housing prices? Mean = 22.5328, median = 21.2
- Standard deviation? 9.1880

2) Evaluating Model Performance

- Which measure of model performance is best to use for predicting Boston housing data and analyzing the errors? Why do you think this measurement most appropriate? Why might the other measurements not be appropriate here?
  I have chosen to use mean_squared_error as the measure of model performance. This is because large errors are penalised more severely (they are squared) and therefore the model should move towards a best fit quicker.

- Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?
  This provides a way of determining how well the model works. The training data trains the model, and we can validate its performance using data it has not seen before (the test data). This was we know if the model is generalising well with the data, and can make accurate predictions on data it has not seen yet.
  If all of the data is used for training, then the model is highly likely to overfit (e.g. it just replays back the data it has been given).

- What does grid search do and why might you want to use it?
  GridSearch is a function to determine the best hyper-parameters for the model. In this case, it determines the best depth for the Decision Tree Regressor. If we did not use this, we may need to work through many different iterations of hyper-parameters using trial-and-error.

- Why is cross validation useful and why might we use it with grid search?
  Cross validation provides a way of cycling through the data to make different training and testing buckets. This provides a better way of determining the best training parameters for the model. It also ensures

that the model trains on different data sets each time so that we do not determine parameters for the model that do not generalise well.
I have used a 80/20 split between training and testing, and therefore have used 5 buckets for cross-validation (assuming that each bucket is of the same size and 1 is used for testing, this will equate to a 80/20 split as well)

3) Analyzing Model Performance

- Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?
  The general trend is to reduce the test error, however this only happens up to a point and then the curves tend to plateau.
  The training error increases as the training size increases.

- Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?
  Depth = 1 : This model suffers from underfitting as the level of error remains high for both training and test data sets.
  Depth = 10 : This model suffers from overfitting as the test error doesn't reduce even though training error is nearly at 0.

- Look at the model complexity graph. How do the training and test error relate to increasing model complexity? Based on this relationship, which model (max depth) best generalizes the dataset and why?
  Up to a point, increasing the model complexity reduces the training and test errors. However, after this point there is signs of overfitting as the training error reduces towards 0 but the test error plateaus (and potentially starts to increase again).
  From my graphs and multiple runs of the code, a max depth value of between 4 and 7 appears to work best.

4) Model Prediction

- Model makes predicted housing price with detailed model parameters (max depth) reported using grid search. Note due to the small randomization of the code it is recommended to run the program several times to identify the most common/reasonable price/model complexity.
  Prices have ranged from 18.816 (maxdepth=8) through to 21.6297 (maxdepth=4).

GridsearchCV typically picks a maxdepth of 5 (price = 20.968) or 7 (price = 19.997). What is obvious is that the lower the maxdepth value, the higher the predicted price.

- Compare prediction to earlier statistics and make a case if you think it is a valid model.
  I think it is a valid model for the following reasons:
  1) The error rate is relatively low based on the graphs, and the GridSearchCV function is picking max_depth rates that are in line with what is seen on the graphs.
  2) The house price is within the minimum and maximum that we have seen in this area.
  3) Although the predicted house price is close to the mean and median, this serves to demonstrate that it is a valid house price in Boston, but this information alone cannot determine whether it is a good prediction for this particular house.