Predicting Boston Housing Prices

1) Statistical Analysis and Data Exploration

- Number of data points (houses)?
    - 506
- Number of features?
    - 13
- Minimum and maximum housing prices?
    - Minimum: 5.00
    - Maximum: 50.00
- Mean and median Boston housing prices?
    - Mean: 22.53
    - Median: 21.20
- Standard deviation?
    - 9.19

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?
    - The best measure of model performance is the regression scoring metric: mean squared error.
    - This measurement is most appropriate because squaring the residual error:
        - Makes all values positive,
        - It also emphasizes larger errors over smaller errors.
        - It's differentiable in calculus which allows us to find the minimum or maximum value.
    - Since the data is continuous, we should select a regression scoring metric. It does not make sense to use a classification scoring metric.
- Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?
    - Splitting data into training and testing sets allows you to evaluate and compare the performance of different algorithms on data which it was not trained on.
    - If you do not split the data, you wind up with a model that overfits the data. This will not generalize well to data points outside of your data set.
- What does grid search do and why might you want to use it?
    - Grid search allows you to set different values for parameters on a machine learning algorithm. It then goes through all parameter combinations, and uses cross validation,

to determine the parameter values which performed best. This is an easy, systematic way to go through parameter combinations instead of relying on trial and error.

- Why is cross validation useful and why might we use it with grid search?
    - Cross validation allows you to split the data into k different folds, and in k different iterations selects one fold as the test set and the remaining folds as training sets. This allows you to use all the data as testing and training sets to get the maximal metrics.
    - It is useful in grid search so that we don't rely on a bad data split to determine what the optimal parameter combination is. It reduces bias by allowing the parameter combination to run across the entire data set.
    - I used a 20-fold cross-validation while implementing GridSearch because using the default value of 3-fold was not giving consistent results. When using 3-fold, the optimal max-depth ranged from 3 to 10, and a price ranging between 19 and 22. When using 20-fold, the optimal max-depth only fluctuated between 4 and 5, and the price ranged from 20.5 to 21.5.

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?
    - As the training size increases, the general trend for training error is increasing slope and then a plateau after size 200.
    - As the training size increases, the general trend for testing error is decreasing slope and a plateaus after size 20. The testing error rate fluctuates much more than the training error rate.
- 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?
    - At max depth of 1, the training and testing errors are high and converges. This is a classic sign of high bias, or underfitting.
    - At max depth of 10, the training error rate is nearly zero, and the testing error rate is much higher. This is a classic sign of high variance, or overfitting.
- 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?
    - As the model complexity increases, training error approaches zero, but testing error is much higher and plateaus around 20.
    - Max depth of 4 best generalizes the data set because it has nearly the lowest testing error rate, and is relatively low complexity. A model that best generalizes the data set will have a low testing error rate.

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.
    - Max-depth of 5
    - Predicted price of 20.967
- Compare prediction to earlier statistics and make a case if you think it is a valid model.
    - The predicted price of 20.967 is very close to the median price. It is within one standard deviation of the mean.