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 price: 5.0
  - Maximum price: 50.0
- Mean and median Boston housing prices?
  - Mean price: 22.5
  - Median price: 21.2
- Standard deviation?
  - Standard deviation: 9.2

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 ended up using `mean_absolute_error` to measure my model performance. I'm not certain that it is the absolute best measure I could use, but I believe it works well. I needed to use a regression metric, since the housing prices are continuous. I could not have used, for example, a classification metric. By using the `mean_absolute_error`, I am minimizing the average (i.e., the *mean*) error. The error can be either positive or negative, so we want to minimize the *absolute* error (otherwise overestimates and underestimates might cancel each other out). The `mean_squared_error` regression metric would have worked just as well. If there were outliers, perhaps `median_absolute_error` would have worked best.
- Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?
  - You cannot evaluate performance by how well the model does on training data. If you did so, the model could just 'hard-code' the expected value for each given data point. This would amount to over-fitting.

- What does grid search do and why might you want to use it?
  - Grid search automatically fine-tunes parameters using cross-validation. Thus you can improve your model greatly with only a few extra lines of code.
- Why is cross validation useful and why might we use it with grid search?
  - Cross validation is useful because it lets you use all of your data for both testing and training. Grid search will use cross validation to fine-tune parameters.

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?
  - In general, as training size increases, training error increases and testing error decreases. They both approach the same horizontal asymptote with training error approaching the asymptote from below and testing error approaching it from above.

- 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 a tree depth of 1, the model suffers from high bias reflected in the generally high value of error in training

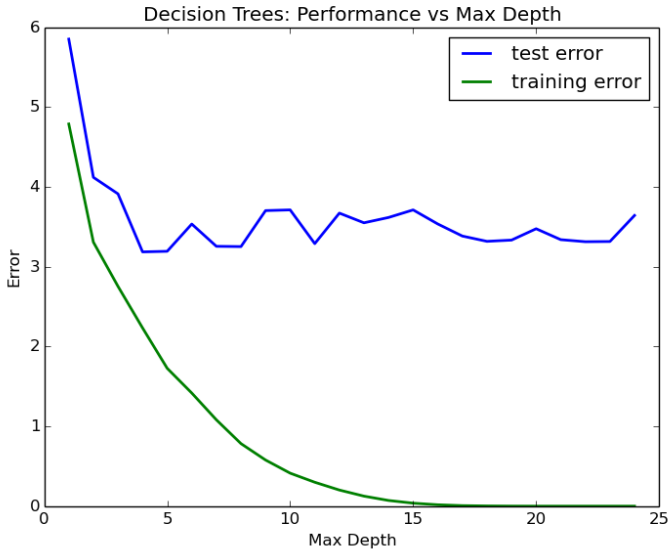
and testing and there is no improvement with new data.

For the decision tree regressor with max depth of 10, there is clear high variance/over-fitting. You can see this because the training error is quite low (close to 0 for a training size of 350), while the testing error is relatively high (about 3.0 for a training size of 350).

- 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, the training and test error both decrease. The training error decreases to about 0, while the test error decreases to about 3.2. The test error first reaches its minimum with a max depth of about 5 and does not decrease significantly (if at all) after that. Hence, the model with max depth of 5 best generalizes the data set. (We want the simplest model that minimizes error.)



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.
  - The model reports that `best_params_` are `{'max_depth': 6}`. This is quite close to my guess of 5. The model gives a predicted housing price of 20.76598639.
- Compare prediction to earlier statistics and make a case if you think it is a valid model.
  - As a sanity check, the prediction is close to both the mean and median Boston housing prices (and certainly within one standard deviation of the mean). Also, the nearest neighbors prediction average is 21.52, which is quite close to the predicted housing price. Hence, I believe we can conclude that this is a valid model.