1) Statistical Analysis and Data Exploration

- Number of data points (houses) - **506**
- Number of features - **13**
- Minimum and maximum housing prices - **5.0, 50.0**
- Mean and median Boston housing prices - **22.5328063241, 21.2**
- Standard deviation - **9.18801154528**

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?

  Mean Squared Error. This error amplifies larger residual error more than smaller residual errors. Also there is always just 1 linear regression that minimises this error. Compared to Mean of Absolute errors, it is computationally easier to calculate, and the latter can be minimised using multiple non-unique regressions.

- Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?

  It is important to split the housing data into training and testing data as this gives estimate of performance on an independent dataset and serves as a check on overfitting

- What does grid search do and why might you want to use it?

  Grid Search is an exhaustive searching through a manually specified subset of the parameter space of a learning algorithm. It makes it easy to tune the parameters to give the best performance without us having to perform the tedious task of tuning the parameters manually.

- Why is cross validation useful and why might we use it with grid search?

  We can do random split of our data into training and testing datasets using cross validation. By using grid-search with cross validation, we can test the performance of our algorithm over multiple splits of data into training and testing set, and then find the average of all the performances.

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?

  Testing error is clearly decreasing. Training error keeps increasing slightly.

- 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?

  It is suffering from high variance/overfitting (evident from the gap between both training and testing curves) at depth 10. At depth 1, the model shows high bias/underfitting.

- 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?

  Training error keeps decrease on increasing complexity (ultimately flattens out near 0). But testing error keeps decreasing for a while and then starts climbing up again. The max-depth = 4 best generalises the dataset as this is the minima of the testing error curve

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.

  ```
  DecisionTreeRegressor(criterion='mse', max_depth=4, max_features=None,
  max_leaf_nodes=None, min_samples_leaf=1, min_samples_split=2,
  min_weight_fraction_leaf=0.0, presort=False, random_state=None,
  splitter='best')
  ```

- Compare prediction to earlier statistics and make a case if you think it is a valid model.

  Predicted Housing Price is 21.62974359
  Mean of prices of 10 feature-wise nearest houses is 21.52 and the standard deviation from the mean is 10.3099757517. The predicted house price if 0.10 bigger than the mean and this quantity (0.1) is within one-standard-deviation (10.3) from mean. Thus the model is valid.