Section 1: Statistical Analysis and Data Exploration

Question	**Answer**
Number of data points (houses)?	506
Number of features?	13
Minimum housing price?	5.0
Maximum housing price?	50.0
Mean housing price?	22.5328
Median housing price?	21.2
Standard deviation of housing price?	9.1880

Section 2: Evalutating Model Performance

Q: 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?

A: First of all, this is a regression problem so it's best to use one of the Regression metrics. Housing prices are a continuous variable and it's almost always good enough to make a prediction that's close enough rather than exactly spot on. So it doesn't make any sense to use one of the Classification Metrics.

Among the Regression Metrics, there are 3 measures of error in sklearn:

- Mean Absolute Error: The mean of the absolute values of all errors.
- Median Absolute Error: The median of the absolute values of all errors.
- Mean Squared Error: The mean of the square of all errors.

Mean absolute error is fine since it converts each error to a positive number and treats all errors, whether large or small, equally. Median absolute error describes the center of a distribution of the absolute value of errors. However, I prefer using the **Mean Squared Error** because it emphasizes and penalizes the larger errors. This results in a model that is less likely to be way off in its predictions than one that is optimized using mean absolute error.

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

A: Splitting the data into Training and Testing parts allows you to first train your data using most of the points and then benchmark your model on some data points that were not used to train your model. If you didn't do this, then you would not be able to check for over or under-fitting in your model.

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

A: Grid search provides an organized and systematic way of tuning multiple sets of parameters for a classifier. It can be pictured as an N-dimensional grid (where N is the number of parameters) and each node of the grid represents a particular combination of parameters that'll be tested. This is useful when we're not sure what the optimal combination of parameters is so we'll test them one by one.

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

A: If you just randomly partition your data in to Test and Training sets, how would you know that the training set is suitable for building a model? For example, if you happened to pick mostly houses in one particular neighborhood for your training set, it probably would end up training a biased estimator. To minimize the possibility of such a drawback, we use Cross Validation which partitions the original data into several different ways. For each partition, we'll train and test the data using the respective sets and get a score for the accuracy of that particular model. Then we average the scores among all models/partitions which is a better representation of the power of this particular training algorithm when used on our data set.

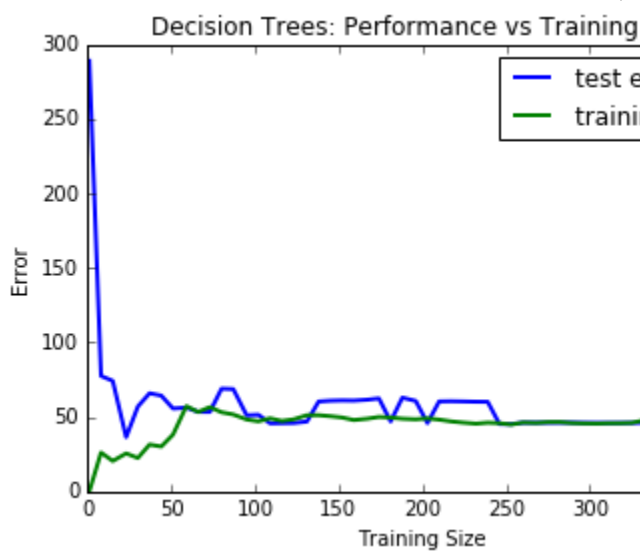
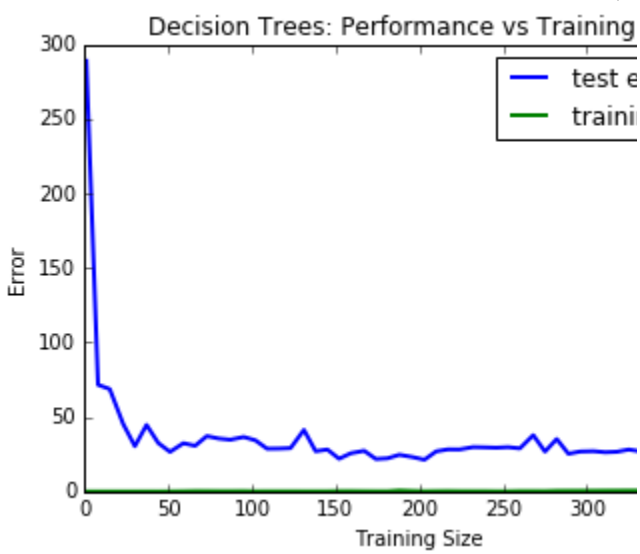
When we use CV with Grid Search, we'll test each "node" in the grid multiple times using different testing and training set splits before we assign an averaged accuracy score to that particular combination of parameters. This way, we can judge each set of parameters more fairly by minimizing the potential biases when splitting our testing and training sets.

Section 3: Analyzing Model Performance

Q: Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?

A: It seems in almost every case, the training error rises gradually while the testing error falls rapidly as the training size increases.

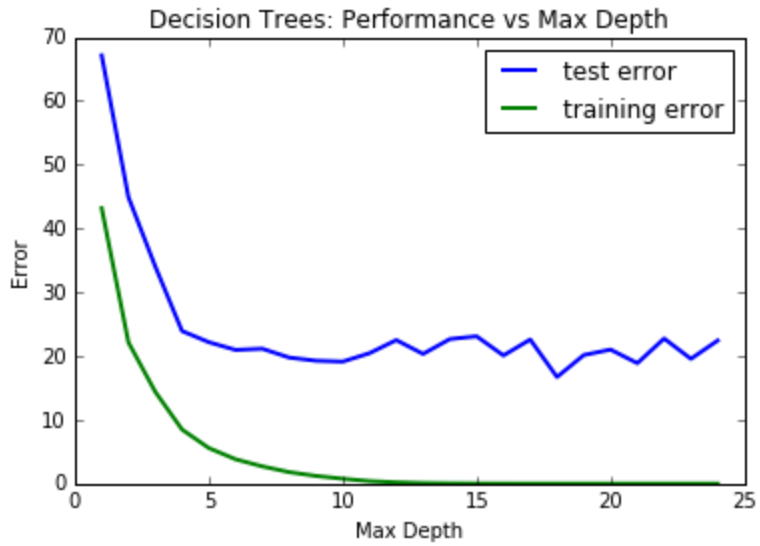
Q: Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?

Depth 1	Depth 10
	

A: For **Depth 1**, the testing and training errors get pretty close to each other in several places and tend to stabilize as the training size increases. Both errors are quite high and fairly close to each other which suggests **Bias and Underfitting**.

For a **Depth of 10**, the training error is several magnitudes smaller than the testing error at all times which suggests that we can easily train the model to fit the data but it fails to achieve the same low error when making predictions on the test set. This suggests **Variance and Overfitting**.

Q: 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?



A: The training error seems decrease asymptotically to 0 as the max depth increases. The test error also decreases rapidly until around 5 or 6 when it more or less oscillates around 20. This suggests that max depths of over 5 all give roughly the same error but I'd pick **5** to be most optimal (since it'll require the least computing power without sacrificing better accuracy.)

Section 4: Model Prediction

Q: 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.

A: I ran my code several times and got the following depths and predictions:

Depth	Prediction
4	21.62974359
4	20.96776316
4	18.81666667
4	21.62974359
4	21.62974359
4	19.99746835
4	20.76598639
4	19.32727273
4	19.99746835
5	20.96776316
6	21.62974359
6	21.62974359
6	20.76598639
6	21.62974359

6	20.76598639
7	20.96776316
7	21.62974359
9	21.62974359
10	21.62974359
10	19.99746835

Several of these predictions have repeated which suggests that there are multiple sets of parameters in the final model which minimizes the testing error. If every estimate were different, I would be inclined to average them but since I have several distinct estimates that have repeated themselves, then I'd be more inclined to pick the most common estimate: **21.630**. Consider also that in Section 3, we found the optimal learning depth to be 5+ and most depths over 5 yielded that estimate compared to 1/3 of them at a depth of only 4.

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

A: The mean is 22.5328 and the median is 21.2 plus the standard deviation is 9.188. This estimate clearly falls within 1 standard deviation of the mean which is the most common for data in a normal distribution. Although this doesn't guarantee its accuracy, it certainly doesn't raise any red flags the way an outlier would.