1) Statistical Analysis and Data Exploration

Size of data (number of houses): 506

Number of features: 13

Minimum price: 5.0

Maximum price: 50.0

Mean price: 22.5328063241

Median price: 21.2

Standard deviation of house price: 9.18801154528

2) Evaluating Model Performance

Question 1: 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?

Answer 1: The Sum of the Squared Error(SSE) is best measure of the model performance for predicting Boston housing data and analyzing the errors.

Sum of Absolute Error(SAE) is another measurement which can be used to measure the model performance. However, there are problems with this measurement. If we use the Sum of Absolute Error then there is a possibility that the negative values may be canceled out by the positive values resulting in a low error which might not be true. This does not happen in SSE as all the values are made positive by squaring. Also, SSE makes small errors(decimal values) smaller and large errors larger by squaring them and making the error for the model large.

Question 2: Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?

Answer 2: It is important to split the Boston housing data into training and testing data sets because we need to check if the model we are working with gives reliable results. The training and testing give an estimate of performance we can expect form the model on an independent data set. It also helps to identify if the model suffers from overfitting or underfitting.

Question 3: What does grid search do and why might you want to use it?

Answer 3: Grid search is used to find which parameter is best for making a model. Grid search makes combination of parameters tunes and cross validation to determines which parameter tune gives the best performance.

Question 4: Why is cross validation useful and why might we use it with grid search?

Answer 4: For making a good predictive model we need data points for training and testing. If we use more data points for training then there is less data available for testing and if more data is used for testing less is available for training. What cross validation does is that it splits the data into k sets. Then it runs k separate learning experiments and picks one of the k sets as a testing set and the rest k-1 sets as data sets. After running k times it takes the average of the test results.

If cross validation is run with grid search the we can get the best parameters for the dataset by running it k times, which is useful for building a good predictive model.

3) Analyzing Model Performance

Question 1: Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?

Answer 1: As the depth of the decision tree increases we can observe that the training data has less number of errors. At max depth of 10 the training data has close to 0 errors(at 350 samples) and the testing data has about 30 errors(at 350 samples).

Question 2: Look at the learning curves for the decision tree regressors 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?

Answer 2: If we look at the learning curve of max depth 1 we can see that the Sum of Squared Error(SSE) high for the training data which suggests that there is high bias. For the learning curve of max depth 10 the SSE of training is very close to 0 and the testing data which has high error suggests that we have overfit the data.

Question 3: 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?

Answer 3: As the model complexity increases the the training data does well and there is less error. However, the model does not do a good job of predicting the testing data as there are are more errors in it. This occurs due to overfitting where the model does a really good job on the training data(by moving through all the training data points) but it does badly on the testing data. Looking at the model complexity graph and the data obtained from best parameter estimator(running the model multiple times and checking reg.best_params_ values) we can conclude that the best fit occurs at maxdepth between and including 4 and 7.

4) Model Prediction

Question 1: 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.

```
Answer 1: GridSearchCV(cv=None, error_score='raise',
           estimator=DecisionTreeRegressor(criterion='mse',
max_depth=None, max_features=None,
                  max_leaf_nodes=None, min_samples_leaf=1,
min_samples_split=2,
           min_weight_fraction_leaf=0.0, random_state=None,
           splitter='best'),
       fit_params={}, iid=True, loss_func=None, n_jobs=1,
       param_grid={'max_depth': (1, 2, 3, 4, 5, 6, 7, 8, 9,
10)},
                     pre_dispatch='2*n_jobs',   refit=True,
score_func=None,
                    scoring=make_scorer(performance_metric,
greater_is_better=False),
       verbose=0)
Best parameter: {'max_depth': 6}
House: [11.95, 0.0, 18.1, 0, 0.659, 5.609, 90.0, 1.385, 24,
680.0, 20.2, 332.09, 12.13]
Prediction: [ 20.76598639]
```

Question 2: Compare prediction to earlier statistics and make a case if you think it is a valid model.

Answer 2: For the features House: [11.95, 0.0, 18.1, 0, 0.659, 5.609, 90.0, 1.385, 24, 680.0, 20.2, 332.09, 12.13] the prediction is Prediction: [20.76598639]. This prediction is close to the median house price and lies close to the mean. Therefore, we can conclude that this is a reliable model.