Statistical Analysis and Data Exploration:

1. Number of data points: 506
2. Number of features: 13
3. Minimum housing price: $5000
4. Maximum housing price: $50000
5. Mean Boston housing price: $22532.81
6. Median Boston housing price: $ 21200.00
7. Standard deviation: $9188.012

Evaluating Model Performance:

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:
   I believe that mean squared error is the most appropriate metric to measure model performance. We are using Decision Tree Regressor which uses Linear Regression to create an approximate sine curve. Since Linear Regression itself is based on Root mean square error, I find mean squared error a natural way of measuring the error. It also penalizes larger errors more than mean absolute error/median absolute error, hence I find it appropriate.

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:
   We have to build a model that generalizes the best, and to verify if our model generalizes well, it has to be tested on unseen data. Hence, some data is kept aside as the test set.
   If we do not keep a separate unseen test data set, then we will not be able to verify the accuracy of our model. It may also result in overfitting as the model will memorize the data rather than learn from it.

3. What does grid search do and why might you want to use it?
   Answer:
   Grid search performs an exhaustive search over the parameters of our learning algorithm that are not learned by the algorithm (the hyperparameters). In this case: max depth of the decision tree.
   We do not know the best depth for our decision tree, hence we set it as a tweaking parameter in our code. Instead of checking for each parameter value ourselves, we can ask sklearn gridsearch to do an exhaustive search over the parameter space (in this case: max depth of the tree). It takes our learner, the scorer function and returns the best fit model.

4. Why is cross validation useful and why might we use it with grid search?
   Answer:
   Cross-validation provides a better way of testing our model since it runs over different training and testing set, multiple times. In k-fold cross validation, the algorithm is run k times and the error is averaged from those k results. This is a more reliable way of evaluating a learning algorithm's performance.
   Coupling CV with grid search provides us a mechanism to test our hyperparameters values over different pairs of training and testing set, thereby making the model robust.

Analyzing Model Performance:

1. Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?
   Answer:
   As the training size increases, the training error increases since. I guess it happens because it sees new kinds of feature vectors. The test error has a downward trend with the increase in training size. I think this happens because the model learns to generalize better. As the model complexity increases, training error tends to go to 0.

2. 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?
   Answer:
   As the depth of the tree increases, the training error decreases and the model tries to overfit the data. When the max_depth = 1, model is too simple, and it has high bias/underfitting, since both the training and testing error are high. And when the model is fully trained (at max depth=10), it suffers from high variance/overfitting, since training error is almost zero but the test error is not coming low.

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:
   Training error reduces with increase in model complexity. Testing error reduces until a point and then it starts increasing and then behaves randomly.
   Based on this graph, the best model is at max-depth=4 because usually after max_depth>4 test error starts increasing. Hence, if we increase the depth, model becomes too complex and overfits.

Model Prediction:

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:
   After running the program several times, the most common price that I got for the given feature row was $21629.74 at max_depth=4.

2. Compare prediction to earlier statistics and make a case if you think it is a valid model.
   Answer:
   The price that I got falls within 1 standard deviation of the median. I believe this is a valid model, since it is giving quite consistent results.