1) Statistical Analysis and Data Exploration

Number of data points (houses)?

* 506

Number of features?

* 12

Minimum and maximum housing prices?

* 5 and 50

Mean and median Boston housing prices?

* 22.5 and 21.2

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?
Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?
What does grid search do and why might you want to use it?
Why is cross validation useful and why might we use it with grid search?

Firstly, this is a regression problem, so that narrows down the applicable measures. In this case, we want a model the predicts a continuous variable and the best measure of both training and testing performance is the deviation between the predicted value and the actual value for each observation. Because the model can have both negative and positive deviations from the actual labels, we need a measure that accounts for both positive and negative deviations. The best way to day this is to square the deviations and then average them to get the Mean Squared Error (MSE).

There is no special reason to split the Boston housing data into training and testing data, it's just a good practice for getting a sense of how the model will perform in the real world. Without doing this, you reduce the odds that your model will be general enough to be accurate when making predictions on a new set of features.

Grid search is a method of finding a good parameter iteratively. In this case, we use grid search to find the depth of the regression tree that produces the lowest bias and lowest variance. In other models, we might use grid search to find the optimal value for the tuning parameter, etc. It comes with the caveat that you may "skip" over the optimal parameter depending on how "granular" the grid is or you could end up at a local minimum. Cross validation is useful for the same reason as previously describe for splitting the data into training and testing data. In this particular case, K-fold Cross Validation was used as the default setting for sklearn's GridSearchCV() function. K-fold Cross Validation is useful because we reduce the dependency of the model parameters on how the data was split up into testing and training data for Cross Validation without k-folds.

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?
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?
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?

In general for most of the depth iterations, the test error decreases as the training sample increases in size. The training error is generally increasing.

The optimal point in the bias versus variance tradeoff seems to be around max depth =2 (without using k-fold Cross Validation). At max depth = 1, the model seems to be ok in terms of performance, but the sample size has to increase quite significantly to get a convergence between the test error and the training error. At max depth = 2, the test error and training error converge quite quickly and there is an overall decrease in the error (i.e. reduction in both variance and bias). For depth >= 3, signs of overfitting start to occur where you do get a decrease in error but the variance increases with each increase in depth. At depth =10, the variance as implied by the difference between the test error and traiing error is quite massive. This suggests the reduction in bias is really the result of overfitting rather than a genuine increase in the predictive power of the model.

But, k-fold cross validation suggests something slightly different. The default setting for the GridSearchCV() function in sklearn is 3 folds. When using the default setting, it shows that both bias and variance decrease up to max depth=3. After that point, the mean error stays flat but variance continues to increase. Both conventional cross validation and k-fold cross validation are saying something very similar (the the optimal max depth is either 2 or 3), but in this case, I would take the k-fold cross validation implied optimal depth =3 as the best model because it should be a more robust validation technique in general but especially because this is not a small data set (i.e. I would trust conventional cross validation if this were a small data set of less than ~50 observations).

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.
Compare prediction to earlier statistics and make a case if you think it is a valid model.
This model is valid and has moderate predictive power. Our best model has an MSE of 35.06. This implies that for any given prediction, we could expect an error in the absolute prediction of the house price to be +/- 5.9. One quick comparison would be to compare the model against a model that simply uses the mean house price of 21.2 for all predictions. This hypothetical model would be with 9.2 (the standard deviation) 67% of the time. Comparison against this hypothetical model helps put the error of the model we made in perspective and you can see that it does a fairly decent job. It's also important to keep in mind that there is no model that will be able to make predictions that are 100% correct because house sales involve an element of randomness. In addition, there are many more factors that go into house prices that would be impractical to include in the dataset. For example, there could be one outlier that sold for a price much higher than the predicted amount because it has historical significance. Even though that could be one case in the data set, outliers can have a big impact on the model parameters and it wouldn't neccessarily be practical to include a dummy variable for historical significance just to try and reduce the error. In the end, it comes down to, "what's this model going to be used for?"!