**Statistical Analysis and Data Exploration**

* Number of data points (houses)?
* Number of features?
* Minimum and maximum housing prices?
* Mean and median Boston housing prices?
* Standard deviation?

**Answer below**

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 = 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 is the error performance metric used in this project. The reason why this is appropriate it is it will highlight if the predicted values by algorithm has lot of outliers as it is squaring the error of the difference between predicted and actual. This way we know if algorithm is working fine or if it is giving lot of outliers. In this project the outliers are the prices of the house being high. Also from calculus this is differentiable so can be used in minimizing cost function using gradient decent algorithm.

Mean absolute error and Median absolute error doesn’t penalize the outliers that much. Accuracy, precision and recall are classification metrics and since this problem is linear regression this doesn’t work in the above project.

* 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 is to make sure the algorithm is not overfit to the whole housing data and show good results with this particular data but when tried on a new data set it might throw up lot of error as the algorithm is overfit to the data and has high variance.

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

Grid search will take the learning algorithm that we want to use along with parameters like max depth in this project and any cross validation and scoring function to calculate the values of scoring function for each parameter and cross validated data and averages the scoring value for n fold cross validation. This will help in identifying the best fit parameter for a particular algorithm and data.

* Why is cross validation useful and why might we use it with grid search?
  + With cross validation we can us the whole data available for both testing and training data sets as cross validation divides the data into bins and each bin is used as a testing set and remaining bins are used as training sets. With Gridsearch the best parameters for an algorithm are assessed by validating the parameters on cross validation data and picking the parameter that has best average score for n fold cross validation.

**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?
  + Training error increases as training size increases
  + Testing error decreases 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?

When the model is fully trained it suffers from high variance/ over fitting as you can see in below graph for depth = 10 training error is pretty low but testing error is high and there is a big gap between testing and training error.

When maxdepth = 1 the model is highbias/underfitting as seen from below graph as initially with training size is small training error is low and testing error is high and as we keep increasing testing size the testing error will increase and training error decreases but they converge with error being high and this shows model is biased and even increasing of training size will not help when the model is biased.

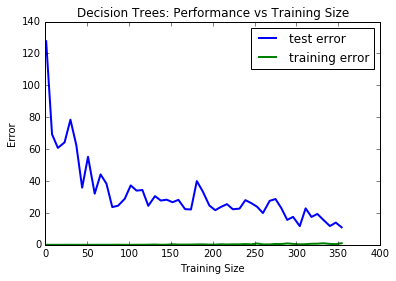
Decision Tree with Max Depth:

1



Decision Tree with Max Depth:

10



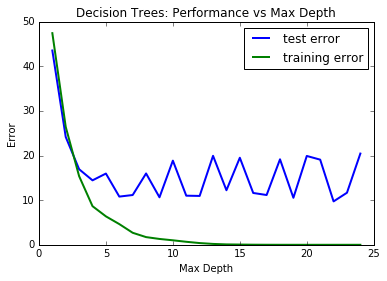
* Look at the model complexity graph. How do the training and test error relate to increasing model complexity? Based on this relationship, which model (max depth) best generalizes the dataset and why?

As shown in below model complexity graph as model complexity is increasing

1) training error is decreasing as the data fit increases with model complexity because of overfitting.

2) testing error decreases in the beginning as the model complexity increases since the model bias decreases until it reaches global minimum and then afterwards it will start increasing because of over fitting of the model.

Max depth 4 best generalizes the data as training and testing errors merge (almost the same) and error value is low. After Max depth 4 we can see the gap between training error and testing error increasing as the model is slowly becoming overfitting and the model is suffering from high variance/overfitting problem.

Model Complexity:

**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.

Because of small data set and randomization of the code the best fit parameter is changing every time we ran, so what I did in the code was to run the particular piece of gridsearchcv and fit code for 10 times and save it in a list and find the most occurred best parameter from the list. The best parameter from the list is 4 as shown in the code.

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: [ 21.62974359]

This is a valid model since the above prediction is within one standard deviation of mean from statistics measured in first question