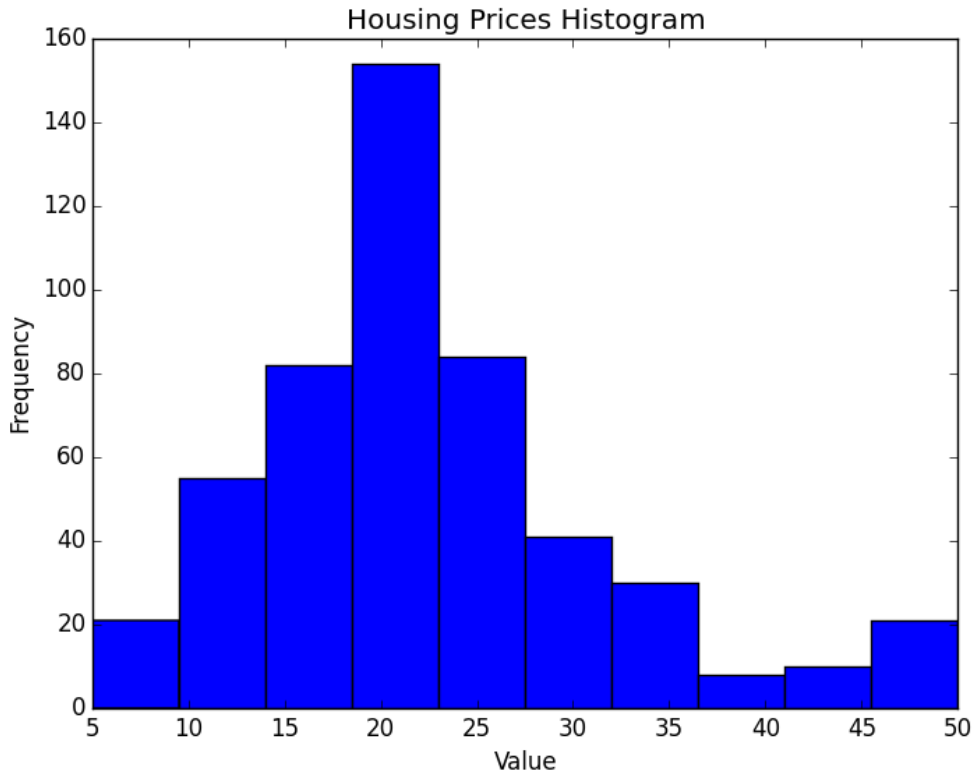
Boston Housing Price Prediction Report

1) Statistical Analysis and Data Exploration

- There are 506 examples in housing data set
- There are 13 number of features given to predict the housing price
- Minimum and maximum housing prices in boston are 5.00 and 50.00 respectively. Looks strange. It seems, as if maximum is set by some authority. There are 16 houses priced at 50.00.
- Mean Housing price is at 22.53, while median housing price is 21.20. Data is slightly rightly skewed, which is expected. Because housing price has set minimum of 0, but maximum can shoot to any value. Mean would have been much higher, had highest housing price not capped at 50.
- Standard deviation of the housing prices is 9.19.



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?

Ans:- Housing prices are continuous values and for which regression metrics like mean squared error and mean absolute error are more appropriate. I chose mean-squared-error over mean-absolute-error because squaring emphasizes large differences and model minimizing total error would have narrower confidence interval. Disadvantage of using mean-squared error model using mean-squared-error is prone to outliers error. If there are few outliers, model will be skewed towards them.

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

  Ans:- Idea behind splitting data into training and test datasets is to see, how well the model is performing. We do that by training model on training set and evaluating its performance on test set(an independent set) to get the goodness-of-fit measure for the model. It also serves as a check on over-fitting. Also, we can plot the learning curves to know does training on more data is of any use. And similarly, plot the model complexity curve to see if increasing the complexity helps in making model better.

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

  Ans: Grid-Search takes a number of parameters and their possible values and generates a grid from them. For a function then it return the parameter tuple on the grid, which optimize the value of the function.

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

  Ans. While finding the model for the data, we have lot of parameters like depth of a decision tree and (kernel, gamma) for support vector machine. Cross-validation is an iterative procedure that breaks the data into say k equal bins and in each iteration randomly chooses one of the bins to test data and rest of the data to train the model. It takes the average of error found in each iteration to be final error of the model. There are few advantage of using k-fold cross-validation:- all the data takes part in training as well as in testing making the process more robust

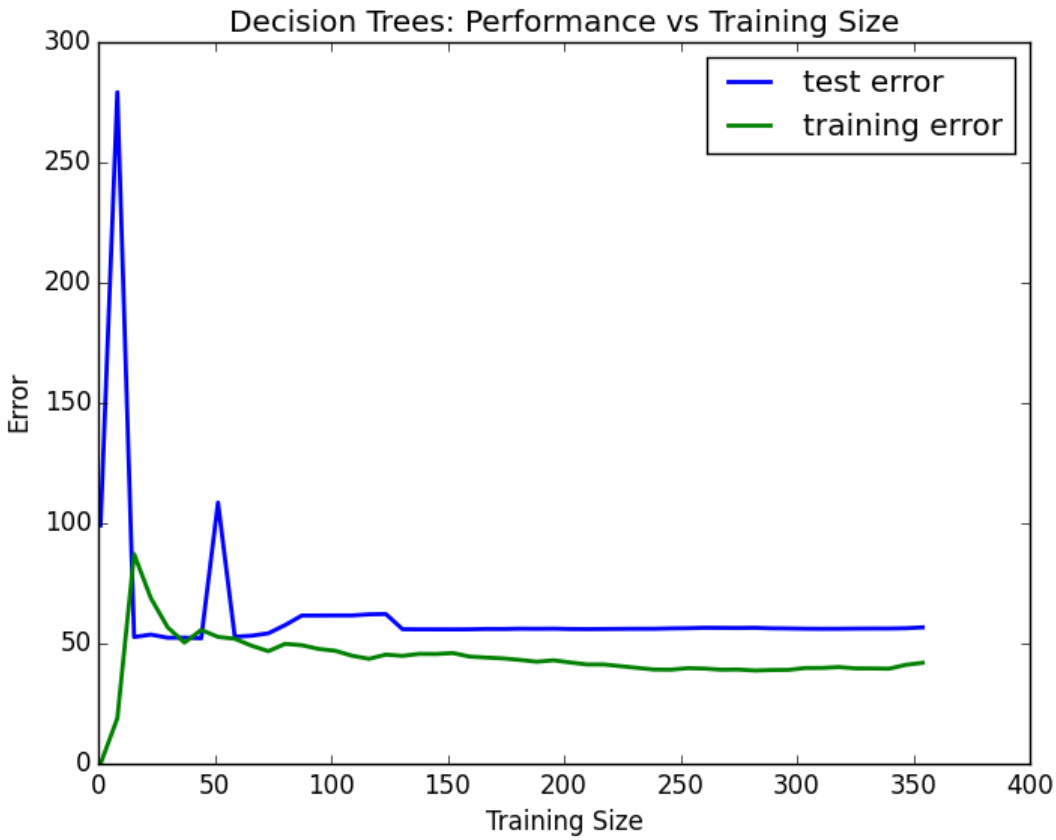
  To find optimal parameter tuple on the parameter space, we can use grid search.

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?

Ans:- As training size increases, training error increases while testing error decreases. This is because it is easier to fit the training data completely on a smaller training size. As training data size increase error increase and stabilizes eventually. Model learned on smaller training size has poor prediction power because of overfitting and end up getting high testing error. As training size increases, model does better and testing error starts to decrease.

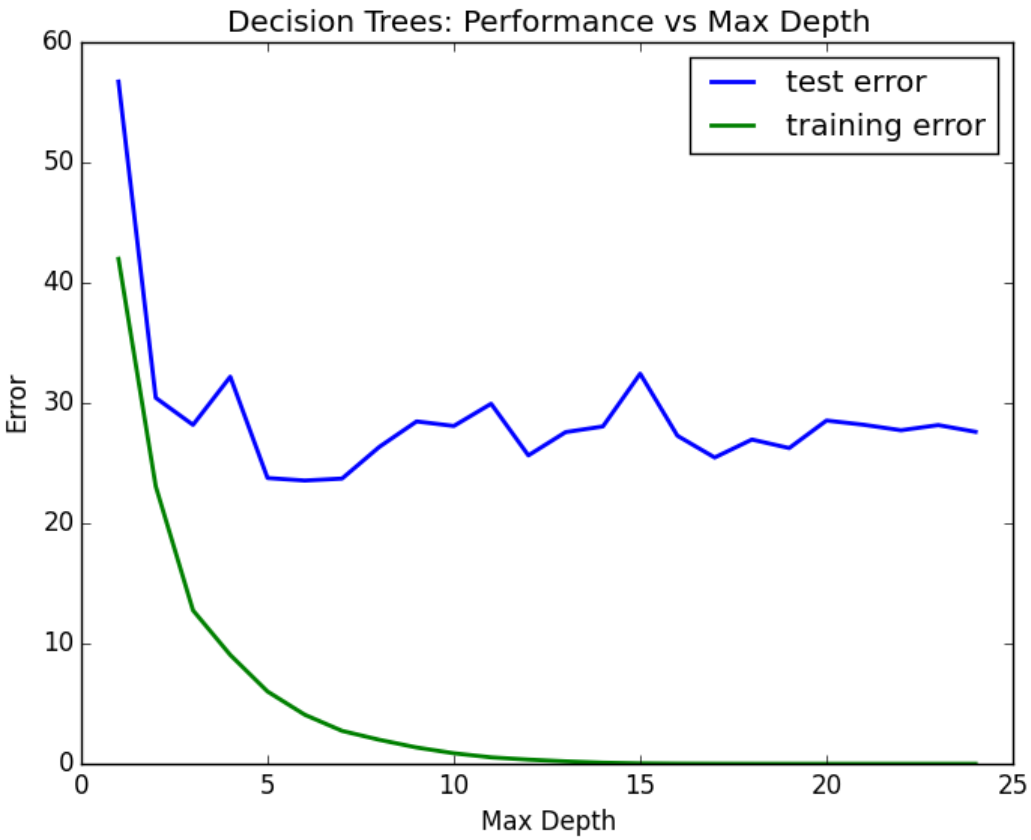
- Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?



First model(max_depth: 1) suffers from high bias because it has high training error and training error and testing error converge with increase in data size.

Second model (max_depth: 10) suffers from high variance because it has low training error and there is high gap between eventual training error and testing error.

- 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?



Looking at the above graph, as the model complexity increases training error drops to zero, while testing error stabilizes.

Increasing complexity of the model makes sense as long as in decreases the testing error. After max_depth =5, testing error stops decreasing. Till max_depth =3, testing error decrease drastically. So, max_depth between 3 and 5 should be good. Following occam's razor principle, I would feel comfortable with max_depth =3.

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.

Housing price for the given feature vector comes out to be 20.76, which

-Lies in 1-std deviation interval of mean i.e. [22.53-9.19, 22.53+9.19].

-