1) Statistical Analysis and Data Exploration

- Number of data points (houses)?
  - 506
- Number of features?
  - 13
- Minimum and maximum housing prices?
  - Min 5.0
  - Max 50.0
- Mean and median Boston housing prices?
  - Mean 22.532
  - Median 21.2
- Standard deviation?
  - House price STD 9.188

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?

  I used one of the regression metrics because predicting Boston housing is a regression problem. Under regression metrics category I chose **mean_squared_error(MSE)** from **sklearn.metrics.**

  I read a lot about the differences between MSE and MAE. It depends on the case, for example MSE is really good for getting rid of negative errors. When researching, I found **"**In cases where you want to emphasize the spread of your errors, basically you want to penalize the errors that are farther away from the mean**".**

  I did some research about, **median_absolute_error**. It seems this is often used with highly skewed data but Boston housing data is not skewed data.

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

  For most of the supervised learning techniques we need two data sets, training and testing data.

  We use training data to learn or build our model. We use testing data to measure our model to see how good it is predicting new instances (unseen data points).

  If we don't split it, we don't have any measure to evaluate the model's performance with unseen data. So we cannot test how good it is predicting when it sees new data points that it has not seen before.

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

  To create the best model for predicting, you need to find and choose the best parameters for the learning algorithm. In order to choose the best parameters you need to test a range of values for each parameter and test the algorithm's performance (output or prediction) with a cost or loss functions. Grid search learns the data with a range of values for each parameter and tests it with the cost or loss functions to find the best value for the algorithm

  Whichever parameter values lead us to the best output or lowest cost is the best tune for the learning algorithm. Grid search is useful when you want test your algorithm's performance with different data.

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

  The concept of cross validation is testing and evaluating the model with unseen data points. **K-fold** and **Leave-one-out** are two types of cross validation. In **K-fold,** we split the data set to N parts and we use N-1 parts for training and one part for testing. Each time we use a different part for testing, until we train and test all of N parts.

  With cross validation we train and test all parts of the data set. Cross validation is useful because it gives a chance for each part of the data set to be learned and tested.

  Leave-one-out has the same idea as k-fold but each part is one data point.

  We might use cross validation in our grid search to make sure the new parameters for the model are performing well with each part of the data set because sometimes our model with the new parameter is doing well only with some parts of the data set but not all parts.

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?

  As we increase training size, we are making better predictions which leads to lower testing errors. It also leads to higher training errors. But training errors are highly related to maximum depth, because in depth 10 training errors slope is really low.

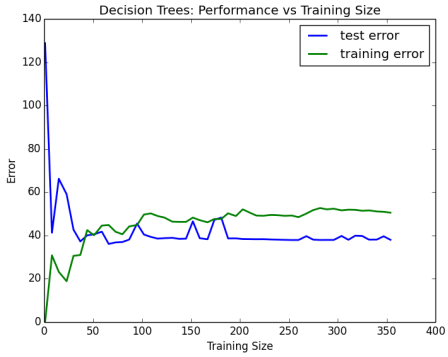

Figure 1 - Learning curve Depth = 1

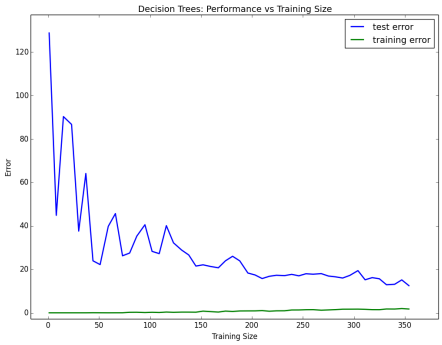

Figure 2 - Learning curve Depth = 10

- 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?

  It does not suffer from either high bias or high variance. Overfitting occurs when our hypothesis fits well with the training set but fails to generalize new samples. In depth 10 learning curve graph, we are getting better (lower) training and testing errors.

  After depth 5 we are only getting better at training errors but not testing errors while testing errors are also not getting worse. That's why I said we are not suffering from over fitting, because you asked about the first and last learning curve graphs.

  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 this graph it is defiantly obvious that we are suffering from over fitting, because test errors going up after 4 (it is sometimes 5, because 4 is a local minimum and 5 becomes global minimum like figure_4) and we are getting worse result in the test data and it shows the model is getting fitter with training data which is not good at

generalizing or unseen data point any more. That's why I think 4 is the best depth for this model.

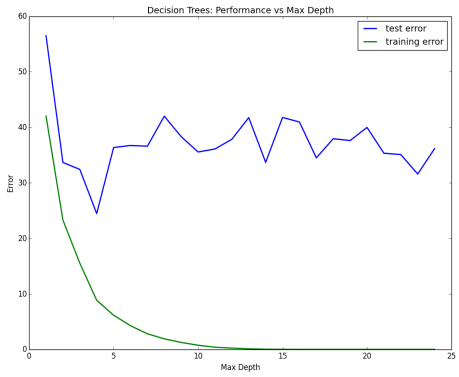

Figure 3- Testing errors go up after depth 4

Figure 4- showing local minimum and global minimum

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.

  When I was using **reg.best_params_** to return best depth. It was different between 4 and 5. It means my best depth is 4 or 5 and housing price was changing from Prediction: [21.62974359] or [20.96776316]

  Also I was using **verbose=3** which shows more detail about GridSearchCV and **Score**.

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

  I did not understand your question especially from (**earlier statistics**). Could you please explain which statistic and then I will explain it. Thanks