Questions and Report Structure

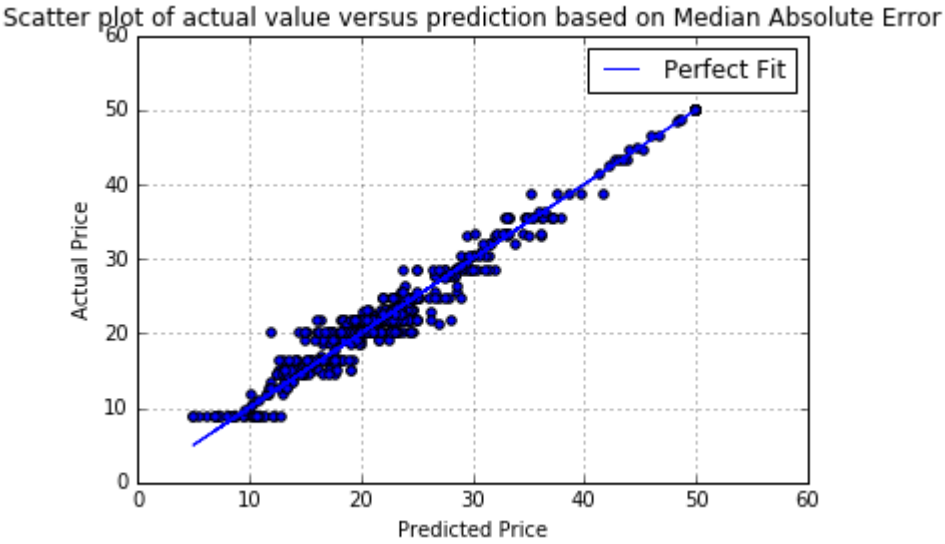
1) Statistical Analysis and Data Exploration

- Number of data points (houses)? **506**
- Number of features? **13**
- Minimum and maximum housing prices? **5.0** and **50.0** respectively
- Mean and median Boston housing prices? **22.533** and **21.2** respectively
- Standard deviation? **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?

  We want to determine the gap between the **predicted** outcome and the **actual** value and then go with the measure that **minimizes** this **gap** in a **consistent** manner. **Median Absolute Error** had the lowest error values and is reflected quite well (IMO) in the chart below.



  Other performance metrics I tried included:

  - Explained Variance Score
  - R2 Score

    - Mean Absolute Error
    - Mean Squared Error (as well as it's root)

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

  Breaking the dataset into two separate datasets means that we can avoid **overfitting** our model on the initial data used to **train** our understanding of patterns within the data of interest and then have the opportunity to **test** how well this model **generalizes (predicts)** to data that the model hasn't seen before.

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

  Enables you to test multiple sets of parameters using K-fold cross validation.

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

  Cross validation reduces the **variance** in outcomes that might occur by relying on a single run of train/test split, since it ensures that the data capture in both the test and the training set reflect the proportional composition of the full data set (stratification).

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?

  The test and training error tends to move towards each other and then reach a level of stabilisation.

- 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?

  Overfitting. At depth 1, the training error settles at about 6.5 and the test error at 7. But by depth 10, the training error has gone to 0.5. While the training error fluctuates around 5. So the gap in training error and test error has increased about nine times, from 0.5 to 4.5. This points to overfitting of the algorithm to the training data and leads to the predictions not being able to generalize well to new data.

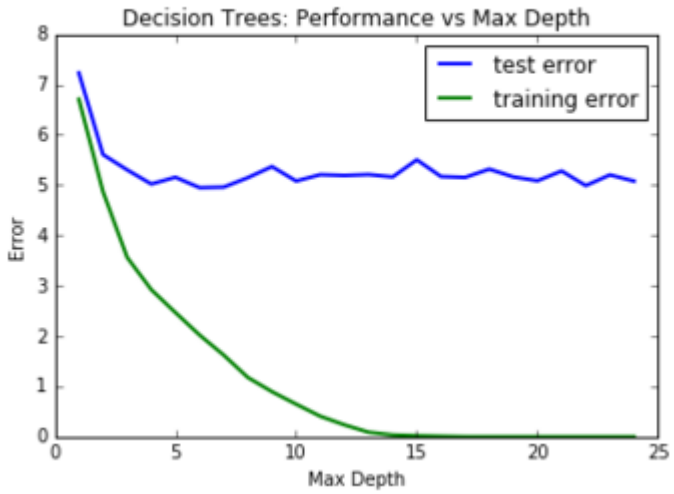
```
Decision Tree with Max Depth:
1
```


```
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?

  About at depth 4 (through visual inspection or 5 according to the best_score_ function), since by then both the test- and the training error has seen a dramatic drop and the test error stabilises at this point so this means that the algorithm won't improve its ability to generalize beyond this point.

```
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?

```
DecisionTreeRegressor(criterion='mse', max_depth=5, max_features=None,
           max_leaf_nodes=None, min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, presort=False, random_state=None,
           splitter='best')
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: [ 20.96776316]
```

Yes, it looks like a valid model. After several submissions the prediction ranged between 19 and 22, all values within the minimum value of 5 and maximum of 50.