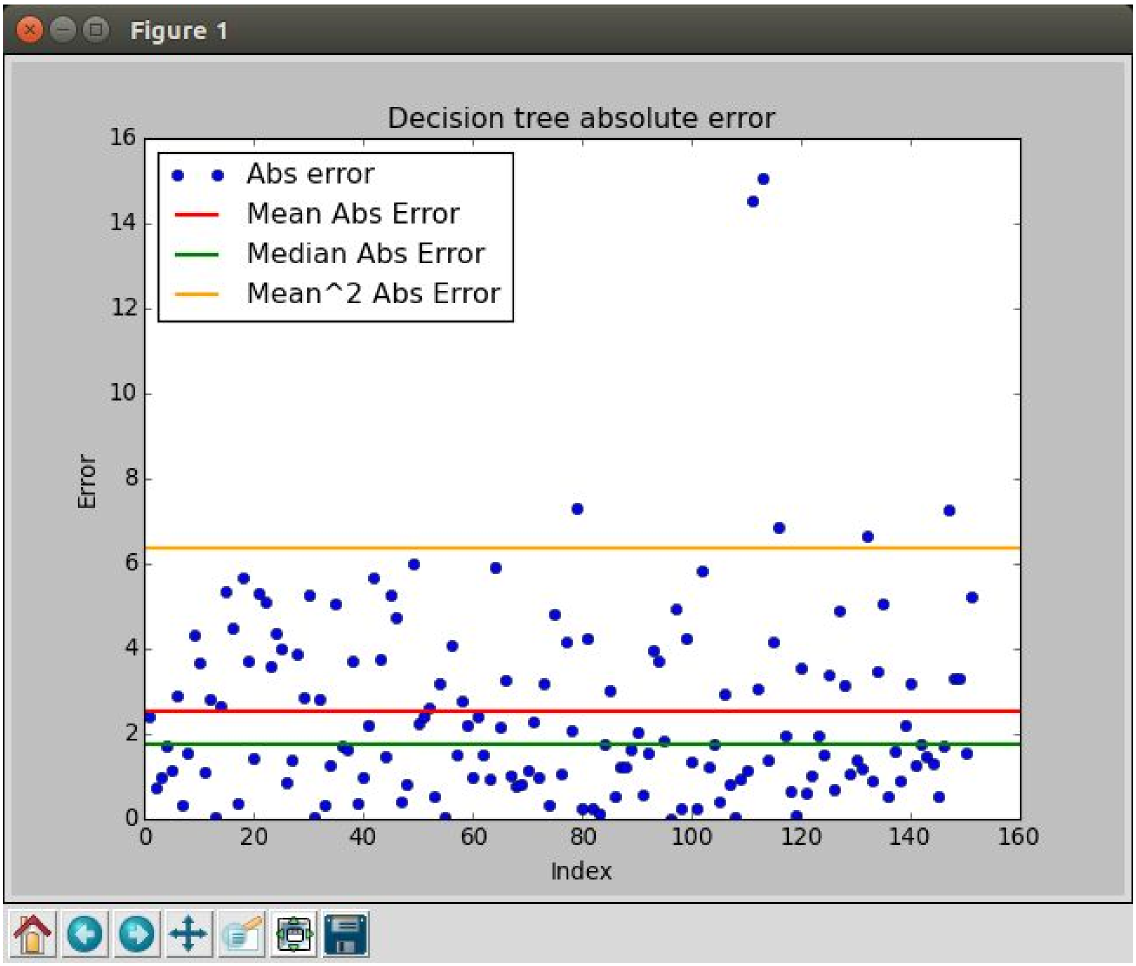
Questions and Report Structure r1

1) Statistical Analysis and Data Exploration

- Size of data (number of houses) : 506
- Number of features : 13
- Minimum price : 5
- Maximum price : 50
- Calculate mean price : 22.533
- Calculate median price : 21.200
- Calculate standard deviation : 9.188

2) Evaluating Model Performance

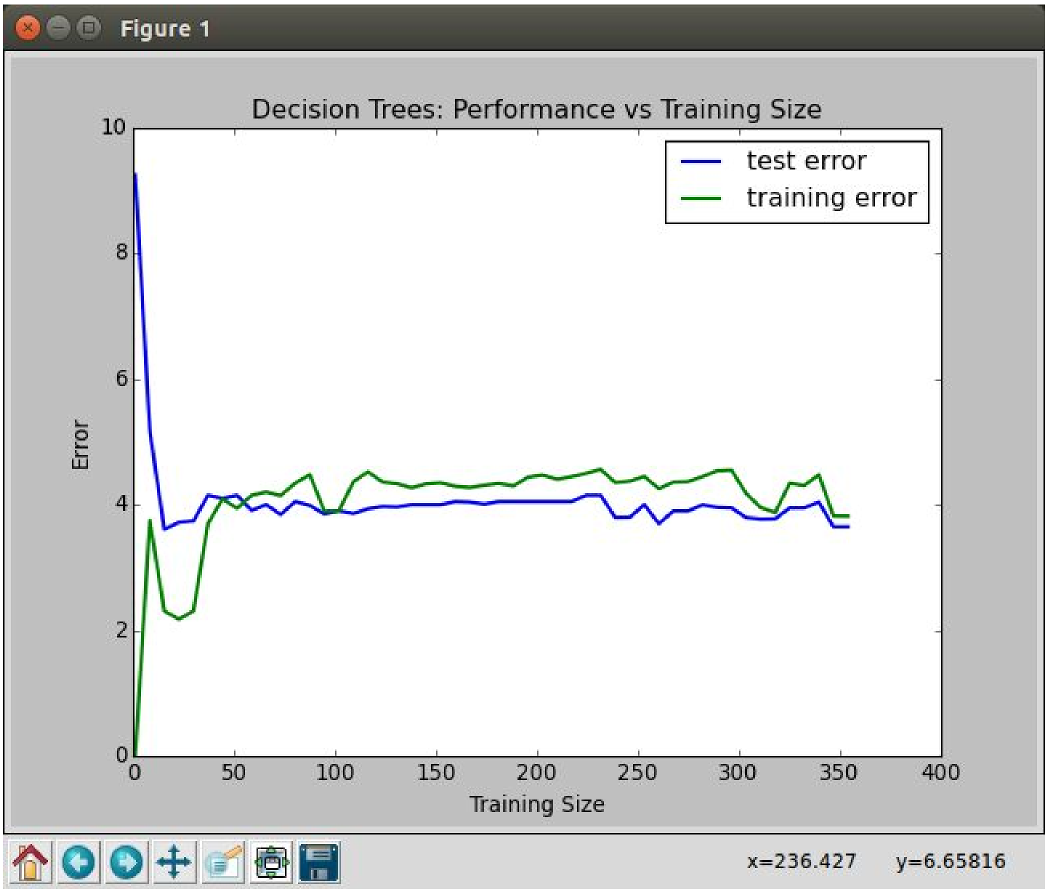
- **Which measure of model performance is best to use for predicting Boston housing data and analyzing the errors? Why do you think this measurement most appropriate? Why might the other measurements not be appropriate here?**
  - I believe the “median_absolute_error” is the best measure for model performance.
  - This is because there are a small number of outliers with large errors in the absolute error results, which pulls the mean higher.
  - Whereas by using the median absolute error, we can disregard the extent of the outliers which gives a lower error.
  - This graph shows the absolute error of each test prediction with the mean, median and mean squared of the set for comparisons of model performance. This was done with a model set to max_depth of 5.
  - 

  - Other metrics based on the mean will suffer similar problems such as using the mean squared error.

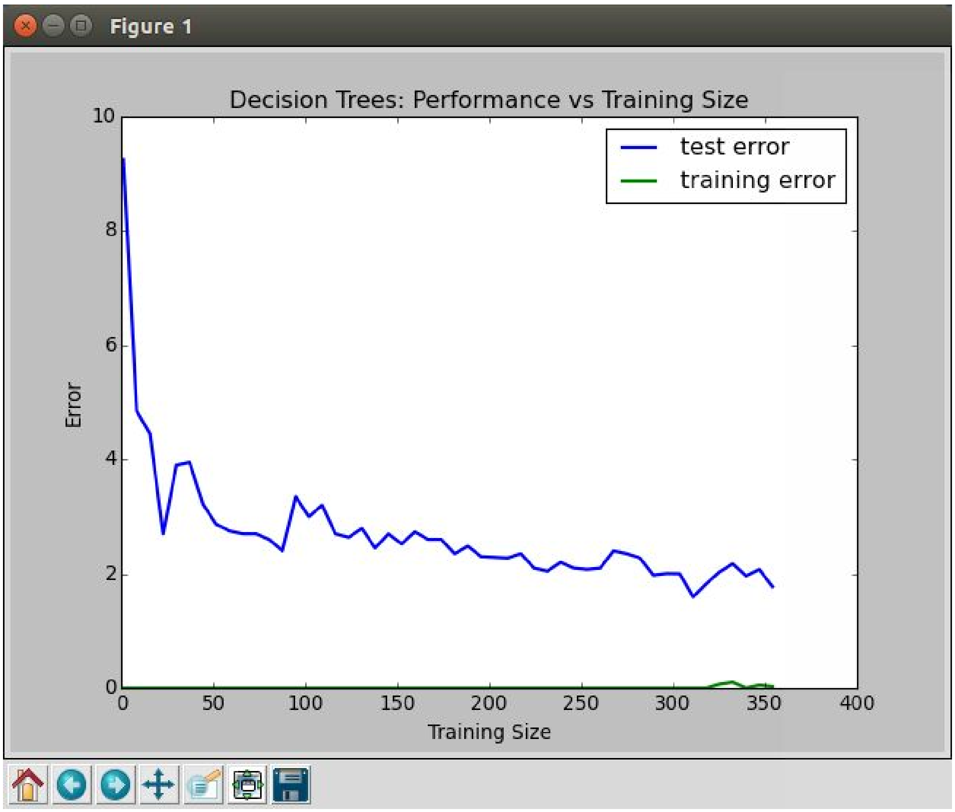
- **Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?**
    - It is important to split data into testing and training parts in order to get a true picture of how the model performs with unseen data.
    - This is because when we train a model, it uses the data in the training set to base future predictions on, and so all data used for training will give biased results.
    - So if we keep some data separate from the training set, we can get an indication of how our model will perform with new unseen data, and if not we will not have a clear indication.
- **What does grid search do and why might you want to use it?**
    - The grid search allows us to iterate over parameters that can be used to tune a model and return the combination that achieved the most accurate results.
    - It allows us to automatically try many parameters quickly and efficiently.
- **Why is cross validation useful and why might we use it with grid search?**
    - We can use cross validation to compare different models using the same testing and training data.
    - The most basic form is where we split the data into 2 sections only, one for training and one for testing. This is good for large data sets.
    - A more advanced cross validation involves splitting the data into multiple sections, and then we test and train using a different section as the testing set each time. The results are then averaged. This is good way to test models if we have less data.
    - We can employ grid search with cross validation to compare the same model type with different parameters, and then select the best model.

3) Analyzing Model Performance

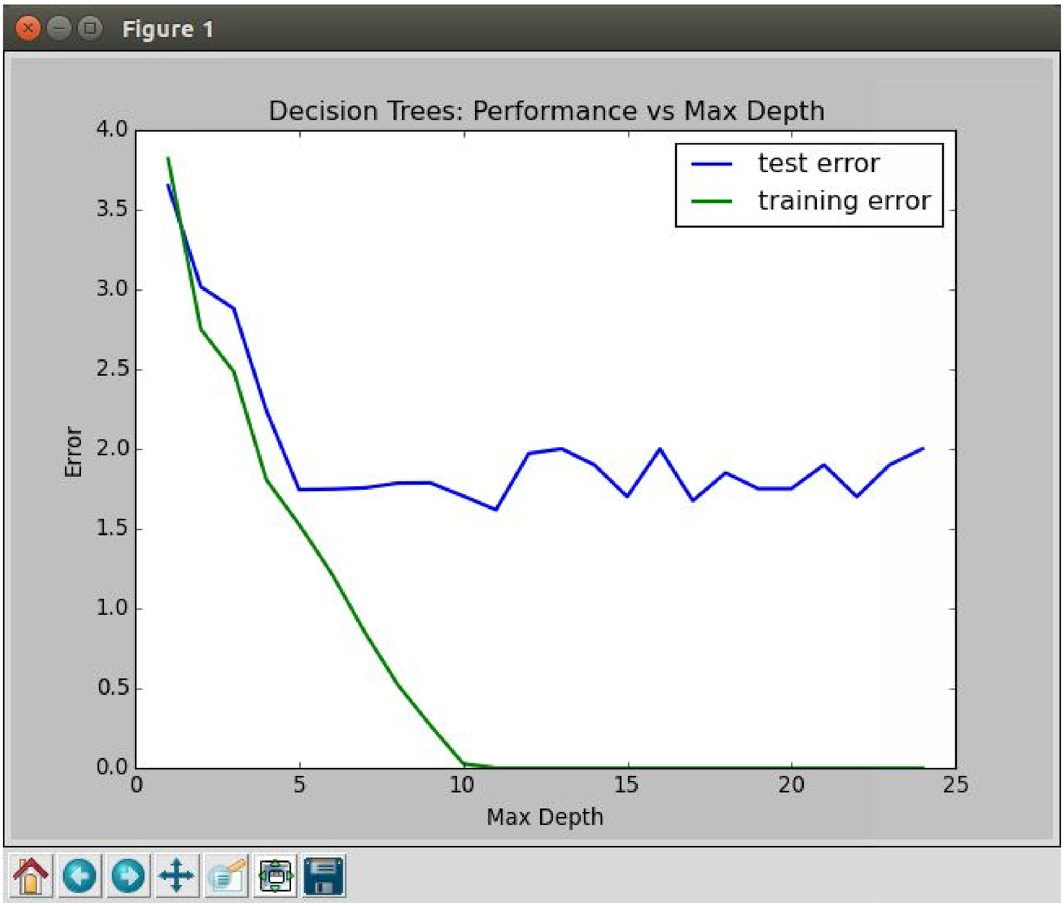
- **Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?**
    - As the sample size increases the test error decreases
    - As the sample size increases the training error increases
- **Look at the learning curves for the decision 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 at depth 1 the training error is very high.
    - This shows that model suffers from high bias or underfitting and the model is too simple.
    - This graph shows error vs training size at a depth of 1.



-
- Once the model is trained to level to 10, the training error is nearly zero.
- This shows that the model fits nearly perfectly to the training data, which is a sign of high variance or overfitting.
- The model trained to a depth of 10 is shown below.



- **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?**
  - Initially the testing and training data are both very high and drop rapidly until a depth of 4.

  - After this the training error continues to drop rapidly until it reaches 0 error at around a depth of 11.
  - The test error platoes at a depth of 5 and the error stays constant at about 2.25.
  - The graph below indicates this.
  - 

  - I believe the graph is underfitting when the depth is less than 4, as there is a high level of error, which would indicate that the model is too simple.
  - And the graph is overfitting from a depth of 6, as the increase of complexity has no effect on the test error, but shows higher accuracy on the training error.
  - I would say that a depth of 5 is the “sweet spot” for this model, where we can minimise underfitting and overfitting.

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.**
  - Results from GridSearch
    - max_depth=5

```
GridSearchCV Best Model Parameters
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, 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: [ 20.76598639]
```

- **Compare prediction to earlier statistics and make a case if you think it is a valid model.**
  - The results show that this house is cheaper than the mean and median and is within 1 standard deviation of the mean. Z-score −0.192
  - I think the model is valid as it fits the test data consistently with a low error at a depth of 5.