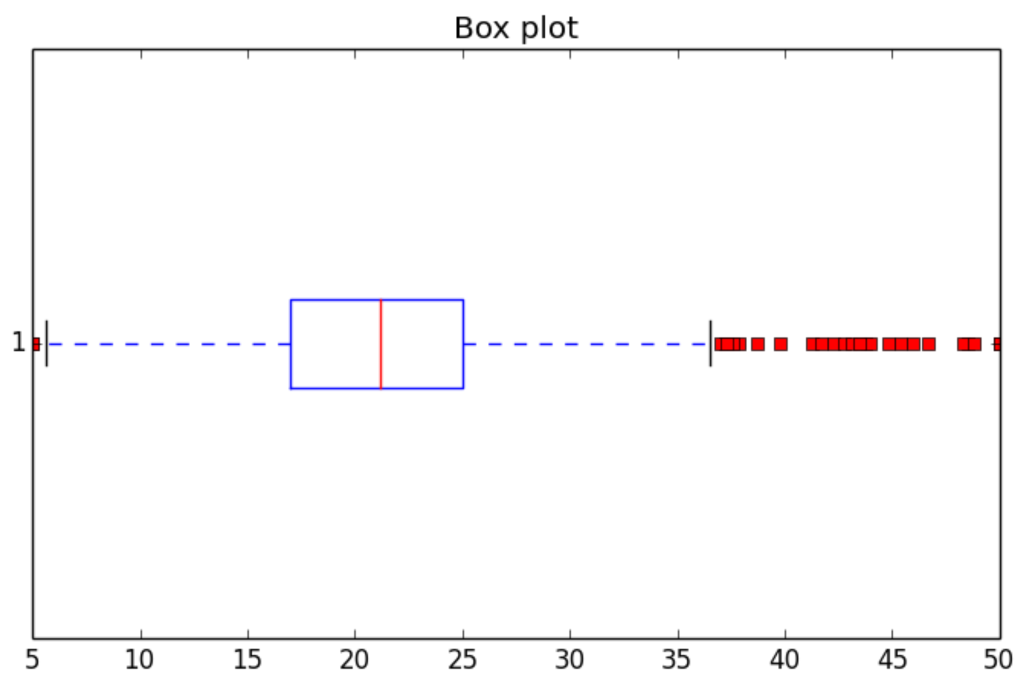
Predicting Boston Housing Prices

Questions

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: 22.532806324
    - Median: 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?
    - For this type of data and regression I believe the mean squared error to be the most appropriate measure of the error.
- Why do you think this measurement most appropriate?
    - Mean squared error is best suited for discrete data like the one we are using here.[1]
    - Mean squared error is more “strict” by assigning little weight to small errors (the square of a very small number is an even smaller number) and assigning a large weight to large errors.
    - Similar to standard deviations, by squaring, negative errors are added instead of subtracted.
- Why might the other measurements not be appropriate here?
    - Because we are dealing with a regression, we can discard all the Classification, Ranking and Clustering metrics.
    - Inside the regression metrics we can choose from:
        - Mean Absolute Error
            - It is better suited for continuous variables[2]
            - Has no “penalization” for large errors
            - Despite this, it is a viable metric, just not the ideal.
        - R2 Score
            - It does incorporate a squaring that deals with negative values and large error penalization.
            - It is a measure of goodness of fit, not of error. We are looking specifically for an error metric.
- Why is it important to split the Boston housing data into training and testing data?
    - By splitting the data, we can train the model and then test it, allowing us to examine if we are running into a high bias or high variance situation with our model. This way we can ensure the model is not overfitting or underfitting.
- What happens if you do not do this?
    - The model could end up with high variance or high bias.
- What does grid search do and why might you want to use it?

---

1 http://worldofpiggy.com/squared-or-absolute-how-different-error-can-be/

2 http://www.eumetcal.org/resources/ukmeteocal/verification/www/english/msg/ver_cont_var/uos3/uos3_ko1.htm

  - Grid search takes care of testing different combinations of parameters in order to exhaustively determine what is the best combination of parameters for the model based on the provided performance metric.[3]
  - In this case we are looking for the optimal model depth.
- Why is cross validation useful and why might we use it with grid search?
  - Cross validation is useful because it allows us to test how well are we modeling the data, we can see if we are overfitting or underfitting. Just one split is maybe the simplest for of cross validation and is appropriate when there is plenty of data[4]. K-Fold Cross Validation splits the data into K chunks of training and testing data. Running each round with each chunk of training and testing data and averaging the result of all the rounds in order to get an average error. This removes any inconsistencies or patterns that might form if we only split once and allows us to maximize the available data assessing the real potential of the algorithm.
  - Grid search combined with cross validation will test and find the parameters that fit the data best. It will have all the advantages of running the cross validation discussed above and will check all the combinations of parameters against this average error run through multiple rounds, maximizing the potential of the available data.

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?
  - At the start, as the training size increases the train error decreases.
  - On the training side, more training data means an increase in training error.
  - Both trends seem to "slow down" until both training error and test error converge into a value (a different value for training and test).
  - After this convergence, increasing the training size has no meaningful effect.
- 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?
  - Depth 1:
    - There is evidence of underfitting as there is a high error for both training and test. This indicates the model does not really generalize the data.

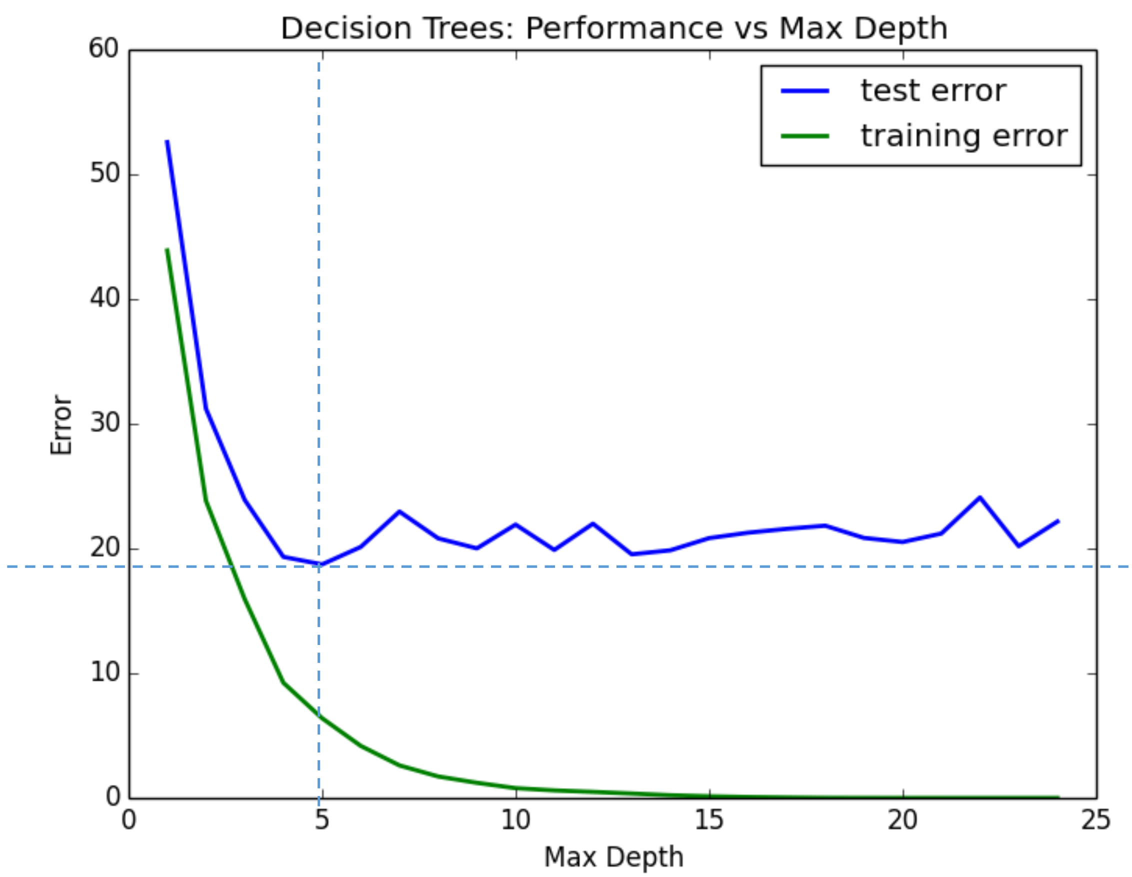
---

3 https://en.wikipedia.org/wiki/Hyperparameter_optimization

4 http://www.anc.ed.ac.uk/rbf/intro/node16.html

- Depth 10:
  - It is very clear that the model is “overfitting”. We can see this because there is practically zero error with the training data, but as soon as we go to test data, we have a much higher error measurement.[5]

- Look at the model complexity graph. How do the training and test error relate to increasing model complexity?



- As the complexity increases, the test error decreases until we reach the ideal. Then the test error actually seems to increase as the depth increases, showing that with a

---

5 http://www.autonlab.org/tutorials/overfit10.pdf

higher complexity depth the model is overfitting the data. Before the “sweet spot” the model has not generalized the data enough.

- Based on this relationship, which model (max depth) best generalizes the dataset and why?
  - I have added 2 lines to the graph at the local minimum (in the case of this run, it was actually the 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.
  - Run 1: max_depth=4 -> Prediction: [21.62974359]
  - Run 2: max_depth=4 -> Prediction: [21.62974359]
  - Run 3: max_depth=6 -> Prediction: [20.76598639]
  - Run 4: max_depth=4 -> Prediction: [21.62974359]
  - Run 5: max_depth=5 -> Prediction: [20.96776316]
  - Run 6: max_depth=4 -> Prediction: [21.62974359]
  - After multiple runs it seems clear that the final results are:
    - Max Depth: 4
    - Predicted Price: 21.62974359
- Compare prediction to earlier statistics and make a case if you think it is a valid model.
  - 21.62974359 is less than one standard deviation from the mean and median.
  - 21.62974359 is within the minimum (5) and maximum (50) range of the data.