P1: Predicting Boston Housing Prices

Reported by Yuki Wakatsuki (yuki.oyabu@gmail.com)

1. Statistical Analysis and Data Exploration
   1.1. Number of data points (houses): **506**
   1.2. Number of features: **13**
   1.3. Minimum housing prices: **5**
   1.4. Maximum housing prices: **50**
   1.5. Mean housing prices: **22.53**
   1.6. Median housing prices: **21.2** (Q1: **17.03**, Q3: **25.0**, IQR: **7.98**)
   1.7. Standard deviation: **9.19**

2. Evaluating Model Performance
   2.1. 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 think that Mean Squared Error, as known as MSE, is best to use for predicting Boston housing data and analyzing the errors. Since DecisionTreeRegressor defined in scikit learn library can only use MSE[1] to measure the quality of a split and train a model from training data, MSE should also be used to measure performance of a model. For the same reason, other measurements[2] like R^2 Error and Mean Absolute Error as known as MAE, are less appropriate.

   2.2. Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?
      - To estimate performance to an independent dataset and check if a model overfits or not, it is important to split the Boston housing data into training and testing data. If we do not split the data, we cannot evaluate model's performance to an unseen data set and cannot avoid making models overfits to a training data set.

   2.3. What does grid search do and why might you want to use it?
      - Grid search is a way of systematically working through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance. If we have candidate parameters but don't know which ones should be used, it makes sense to use grid search to estimate best performed parameters.

   2.4. Why is cross validation useful and why might we use it with grid search?
      - Cross validation helps us to measure average performance to an unseen data set (a testing data set), and this measurement is almost essential for models to avoid overfitting. If we use it with grid search, we can select best performed hyper parameters based on average performance to unseen data, which means models with selected hyper parameters will well-perform to unseen data.

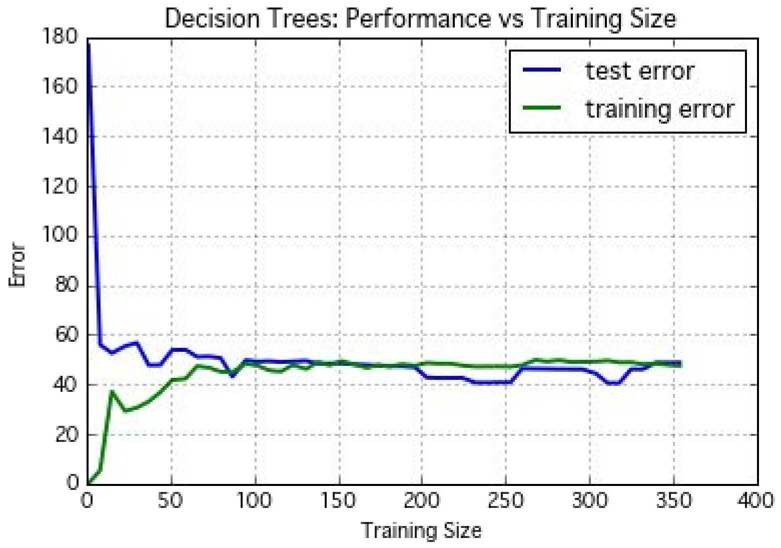
3. Analyzing Model Performance
   3.1. Look at all learning curve graphs provided. What is the general trend of training and testing error as training size (n) increases?
      - $TrainingError_i \leq TrainingError_j, \ (0 \leq i < j \leq n)$
      - $|TrainingError_i - TrainingError_{i-1}| \geq |TrainingError_j - TrainingError_{j-1}|, \ (0 < i < j < n)$
      - $TestError_i \geq TestError_j, \ (0 \leq i < j \leq n)$
      - $|TestError_i - TestError_{i-1}| \geq |TestError_j - TestError_{j-1}|, \ (0 < i < j < n)$

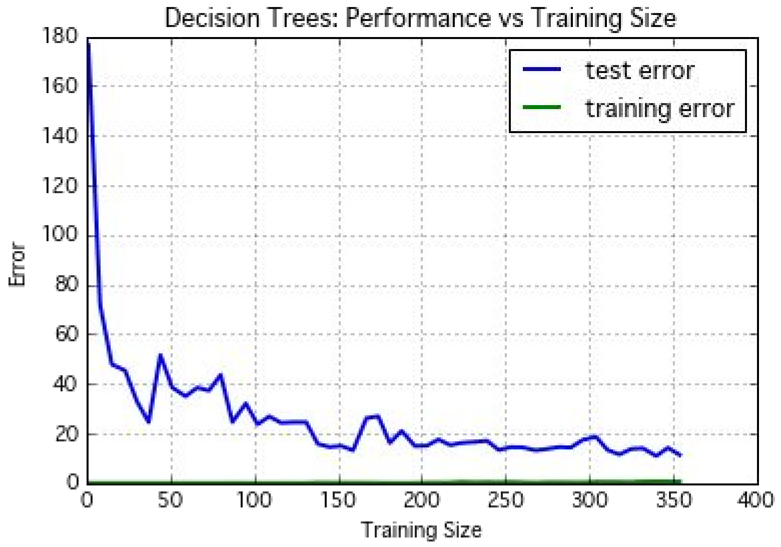
[1] http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html, 01/02/2016
[2] http://scikit-learn.org/stable/modules/model_evaluation.html, 01/02/2016

- $TestError_i - TrainingError_i \geq TestError_j - TrainingError_j, \quad (0 \leq i < j \leq n)$

3.2. 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?

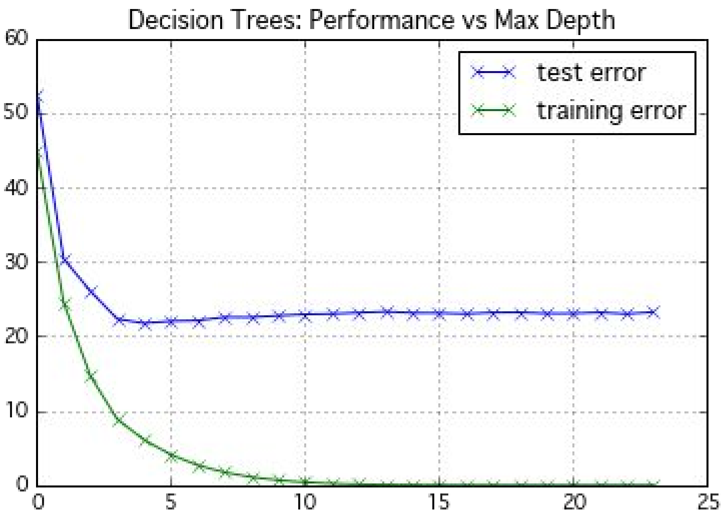
- The regressor with max depth 1 suffers from high bias/underfitting because both test and training error are much higher than one with max depth 10. On the other hand, the regressor with max depth 10 suffers from high variance/overfitting because test error fluctuates and test error is much higher than training error even if training data size is bigger . Please see figures shown below. The first figure shows the learning curves with max depth 1 and the second figure with max depth 10.





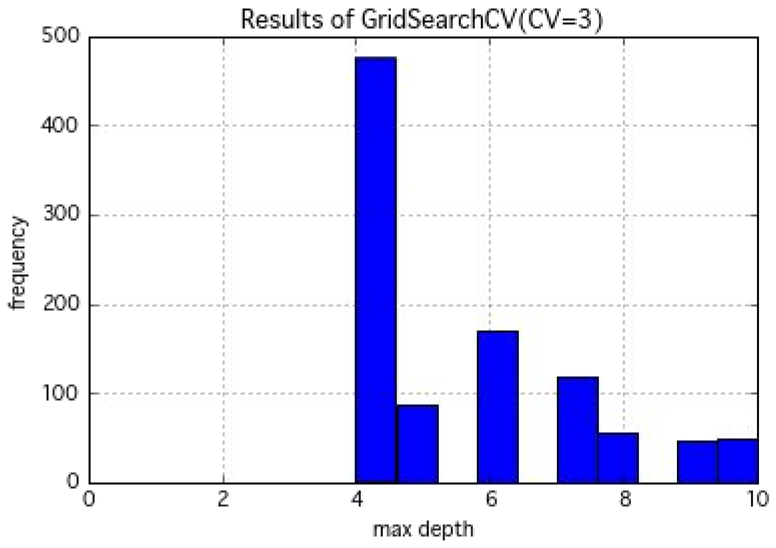
3.3. Look at the model complexity graph. How do the training and test error relate to increasing model complexity? Based on this relationship, which model (max depth) best generalizes the dataset and why?

- The figure shown below plots average training and test error from 1000 trials. In the figure, x-axis and y-axis each represent model complexity and error score. There are 3 important points you can see from the figure.
    - Training error continues to decrease as model complexity increases. In the figure training error becomes almost 0 in models with over max depth 10.
    - Test error stops to decrease as model complexity increases. In the figure models with over max depth 4 do not improve its performance against test data.
    - Difference between test and training error increase as model complexity increases. Which means models more overfit as model complexity increases.
- I conclude that models with max depth 4 best generalizes the dataset. In theory, best generalized models can well-perform against test data and keep complexity as lower as possible. By above observations, we found that models with max depth 4 have almost best performance compared with more complex models, and have obviously better performance compared with less complex models.



4. Model Prediction
    4.1. 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.
        - I ran the program 1000 times to identify the most common/reasonable price/model complexity based on distribution of price/model complexity. The figure shown below is histogram about model complexity (max depth) selected by grid searches. It is obvious that max depth 4 is most frequently selected as the best model parameters. As described in 3.3, models with higher max depth more overfit against test data, so it is reasonable to use the models with max depth 4, which is the lowest max depth among candidates, as a

common/reasonable model complexity and housing price predicted by the models.



- Model with detailed model parameters reported by grid search:

```
DecisionTreeRegressor(criterion='mse', max_depth=4, 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')
```

- Predicted housing price: **21.68**

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

- Since predicted value is not outlier compared with the earlier statistics, I think it is a valid model. According to the earlier statistics, range of non-outlier prices is between 5.06 and 36.96. Since predicted value (21.68) is obviously within this range, predicted value is not outlier. Also, the predicted value is between Q1 and Q3, which means the predicted value is more of average housing price than outlier price.