1) Statistical Analysis and Data Exploration

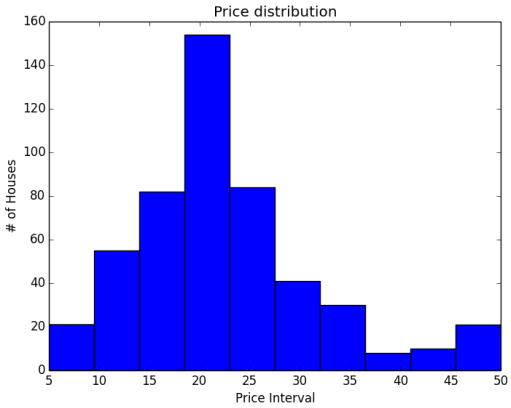
- Number of data points (houses)? 506
- Number of features? 13
- Minimum and maximum housing prices? Minimum Price: 5.0 Maximum Price: 50.0
- Mean and median Boston housing prices? Mean Price: 22.5228 Median Price: 21.2
- Standard deviation? 9.18801

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?

  Predicting the Boston housing data is a regression problem, so measures like accuracy, precision and recall are not appropriate (based on the fact that they work on classification problems). In this case we should consider Mean Absolute Error (MAE) and Mean Squared Error (MSE) based on the fact that we would like the classifier to predict house prices as closely as the real values.

  In this data set we can see that our prices ranges from 5 to 50 while mean and median prices are 22.5 and 21.2 respectably. There seams to be a normal distribution in house with prices between 5 and 40 and then we can see another pick between 40 and 50. I would argue that houses with prices over 40 are not outliers, but rather an important subgroup to watch.



Graph 1: Number of house on price interval.

In this case I would choose MSE since it weights large errors relatively higher than MAE. In this sample the data is heavily clustered at the 20-25 price range and I would value having a good prediction at the 40+ prices range.

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

  Splitting the data into training and testing sample is vital to train and evaluate the performance of our models. Training samples are used to train the Machine Learning algorithms and test samples are used to evaluate the performance of the model on 'new' data points. With the test sample we are able to compare the predicted label vs the real label so we can check how did the model perform on data points that we not used in the learning process. This process is called cross validation.

  The main goal of cross validation to evaluate the predicted power of a model and to avoid over fitting to the training sample. We may be able to get high accuracy on training samples but that doesn't imply high accuracy on new unknown data points. In this sense, splitting the data and using a particular sample to allows us to evaluate the performance of the model on new data points. If cross validation is not used, training errors might lead us to wrong conclusions.

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

  Each Machine Learning algorithm has its own set of parameters that helps us tune their learning process. For example, in decision trees we can set up the maximum depth, the minimum size for splits or the minimum size for a leaf. In order to find the optimal set of parameters we use grid search. This process consists in training the model several times with different parameters and check with which configuration we achieved the best performance.

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

  As stated before, cross validation is useful to evaluate the performance of a particular model with new data points (not used in the learning process) and to avoid overfitting. Both objectives should be considered by grid search in order to determinate which parameter configuration is the optimal.

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?

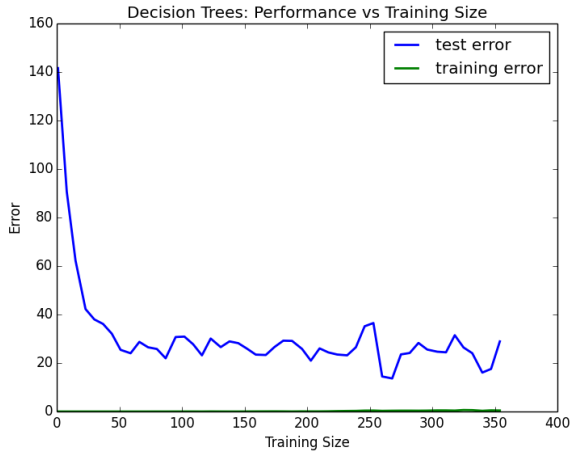
  The general trend is that at higher training size the test error goes down and the train error goes up. We see a big fall in the test error at very small training sizes ([0-50]) and then the test error error goes down at a much slower rate.

- 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?

  Looking at the regressor with max depth 1 we can see that there is a very small variance because both the test and training errors seam to converge at a similar error. This errors are higher than the errors from the regressor with max depth 10 (the max depth 1 error is around 50 and max depth 10 is around 20) so I would argue that comparatively there is bias on this regressor although not a high one.

Graph 2: Learning Curve for the Decision Tree Regressor at max depth 1.

On the other hand, the regressor with max depth 10 clearly shows high variance/overfitting. The training error is very close to 0 at high training sizes while still far from the training error (training error close to 0 and test error around 20). I would argue that we have a clear case of overfitting on the training data.

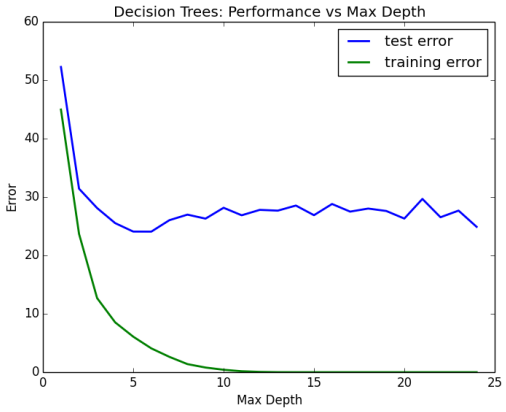


Graph 3: Learning Curve for the Decision Tree Regressor at max depth 10.

- 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?

  As the max depths increases, we see that the training errors tends to 0 (at around max depth 10) were overfitting seams to appear. The test errors goes down from around 50 at max depth 1 to around 25 at max depth 5 where it seams to stabilize through higger max depths.

  Based on the model complexity graph, I would argue that the best model sits around the max depth of 5. At this max depth we have the global minimum for the test error and the training error is not as close as 0 as higher max depths. The test error seams to go up at higher max depths which may be because of a training overfitting.



Graph 4: Model Complexity Graph.

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.

  The most common price prediction is around 21.6. Through several runs of the model, we can see that the optimal max depth sits around 4-6 which is consistent with the analysis from the model complexity graph.

  The predicted house price is in a normal range of values (most houses are in the 20-25 price range) and is in normal ranges with the mean and median.

  Further studies should be done to understand the regression function and check if results are consistent with the house's attributes.