BOSTON HOUSING DATA REPORT

STATISTICAL ANALYSIS AND DATA EXPLORATION

Using the NumPy Python library these figures were calculated from the provided Boston housing market data set.

Number of Houses:	506
Number of Features:	13
Minimum Price ($10,000):	5.0
Maximum Price ($10,000):	50.0
Mean Price ($10,000):	22.5328063241
Median Price ($10,000):	21.2
Price Standard Deviation ($10,000):	9.18801154528

NOTE to Reviewer: Please clarify the number of features question for me. I understand why my last answer was incorrect but I'm unsure as to whether "median home price" would be considered a feature. Looking in the forums it appears the community believes it is, but that doesn't seem right to me as the features are causal to the price, not vice-versa.

EVALUATING MODEL PERFORMANCE

Q: 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?

A: Upon analyzing the dataset provided and doing additional research on the two main error metrics in consideration, Mean Absolute Error, and Mean Squared Error, I chose Mean Absolute Error.

My reasoning is two fold. One follows the logic presented in this literature from Duke University[1] which described the selection process between the two as follows:

[1] http://people.duke.edu/~rnau/compare.htm

"The root mean squared error is more sensitive than other measures to the occasional large error: *the squaring process gives disproportionate weight to very large errors. If an occasional large error is not a problem in your decision situation (e.g., if the true cost of an error is roughly proportional to the size of the error, not the square of the error), then the MAE or MAPE may be a more relevant criterion."*

When analyzing and predicting home costs, the "true cost" of an error is not exponential, rather it is linearly proportional to actual cost of the home.

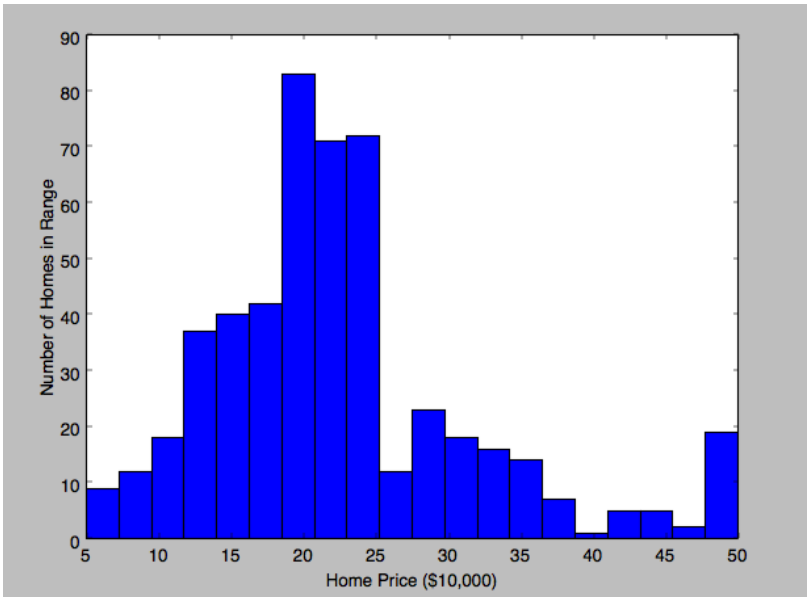

Figure 1: Histogram of Boston Housing Data

My second reason for choosing the linear error metric as opposed to the exponential error metric came from analyzing the histogram of the provided dataset (figure 1). The large number of homes displayed in the 50 range created some doubt as to the models ability to accurately predict the price of houses that were so significantly far from the mean. Assuming that the error on these houses had the potential to be very high, the Mean Squared Error would have been disproportionate to the rest of the dataset.

These outliers being so far from the majority of data points also drew some doubt into being able to effectively select sample sets. This will come up in my thought process again as we discuss selecting the k for cross-validation

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

Splitting data into testing and training sets ensures independent data points which are used to validate the model. If all data points were used to train, the model will have seen all available data and therefore results could hypothetically return a "perfect" model as all points could be fit.

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

Grid search automates selection of model criteria by generating a complete list of all possible variations of parameters. In the case of this problem, using a Decision Tree, it helps automate some of the decisions that go into fitting the model and avoiding over-fitting or under-fitting issues, which in this case is really just the depth of the tree. To do this action manually without grid search would require the model be run for every depth of tree by the operator.

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

Cross validation offers us a way to leverage our data in a more complete way by running multiple test-train splits. In each instance for this model we are using 30% of the data as testing data in each iteration. By using cross validation, we are able to run k-fold iterations of the model with different testing and training sets. This improves the likelihood of an accurate model.

ANALYZING MODEL PERFORMANCE

Q: Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?

For training error, the curve rises sharply from zero until plateauing at some number of training points (approximately 35). As the depth of tree increases the training error curve begins to smooth and flatten back towards zero.

In contrast, the testing error curve declines sharply and plateaus at some number of training points, again approximately 35. Unlike the training error, the testing error does not vary significantly in shape, with the only noticeable change coming above depth 4 where the plateau assumes a slight negative linear trend. This indicates that the model is improving the accuracy of its predictions above depth 4.

Q: 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?

For a decision tree of depth 1, the model displays high bias as the training and testing error both remain almost constant above 35 training data points. This indicates that the algorithm is not improving upon its prediction is biased, therefore underfitting the data. As we step through the

depth of the decision tree, this begins to transition towards high variance until finally at depth 10 we see a training error of nearly zero while the testing data has plateaued at approximately 3.5. This divergence is an indication of overfitting the data.

Q: 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?

Training error declines in a roughly negative exponential curve until it approaches zero, the testing error, however, plateaus after a tree depth of approximately 5. Since the training error is still declining at a depth of 5, this appears to be the best fit for the model as the training error has not yet plateaued and is still declining, while the testing error has flattened.

MODEL PREDICTION

Predicted House Price ($10,000)	20.76598639
Best Parameters	Max Depth: 6, 10-fold CV

Comparing parameters used, based on analysis of the complexity graph, my initial observation was that the testing data error pleateued at a decision tree depth of 5, however the model consistently chose a depth of 6 which is close enough to my estimation that I am comfortable with the model's selection. I increased the cross validation to 10-fold as there was some volatility in the model, both in predictions and in parameters selected, until I increased it to 10.

The predicted house price of 20.77 is a completely rational answer as it falls within 1.76 of the mean and 0.43 of the median. Referencing figure 1, we can see that the largest number of houses fall within this range, making this pricing prediction entirely plausible.

QUESTIONS OF METHODOLOGY AND POTENTIAL IMPROVEMENTS

1) My selection of mean absolute error was based on some research and my general feeling that large errors should be be disproportionately weighted in this example, I do not, however, have complete conviction that this was the correct error metric.
2) I did see quite a bit of variability in my model until, by trial and error, my cross validation *k* was raised to 10
3) The large number of homes clustered in the 50 raise questions in my mind as to if they should be included in the early statistics, doing calculations for outliers would potentially yield a better result. In my estimation removing the large number of homes priced at 50 would pull the mean and median closer to my predicted value.