Project 1 : Predicting Boston House Prices

Yoonyoung Cho

February 14 2016

1 Statistical Analysis and Data Exploration

The **506** Data Points that each represent the mapping of a house to its price comprises of **13** different features; Minimum housing price was **5.0** and the maximum was **50.0**. The mean was **21.53** and median was **21.2**. Standard deviation was **9.19**.

1.1 Relevant Features

The following graph demonstrates how the respective features relate to the final price:

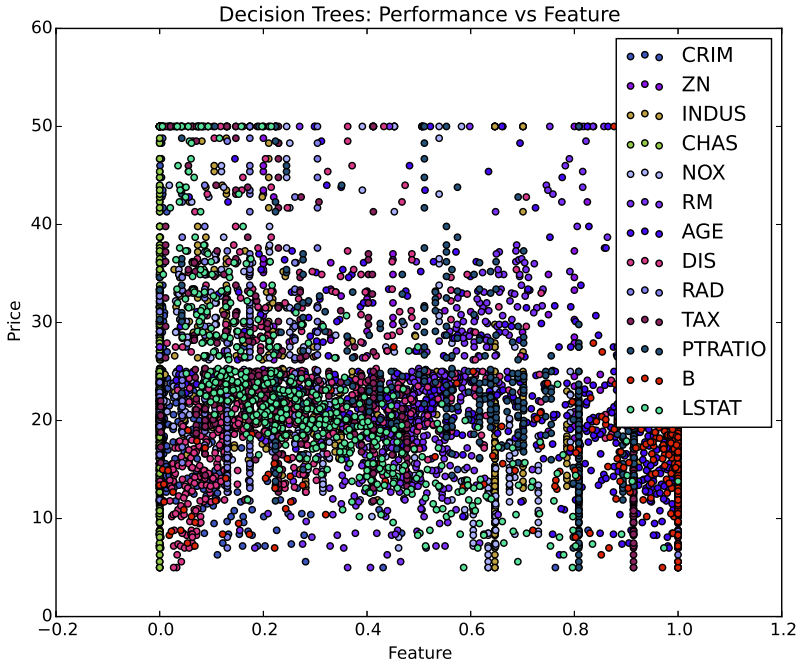
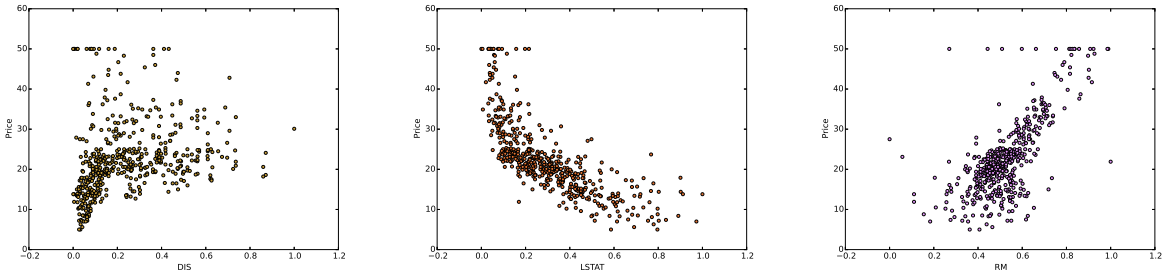


Figure 1: Performance Versus Features(normalized)

In isolation, the features look as follows:



(a) Distances to Employment Centers (b) Lower Status of Population (c) Avg. Rooms per Dwelling

Figure 2: Three Features that, among others, display a strong correlation to price.

The three features that seem most significant are the Lower Status of Population, the average number of rooms per dwelling, and distances to employment centers; each of these features are critical to the quality of living, and is consistent with intuition: The shorter the distance is to the workplace, the less the population of *lower status*, and the more number of rooms there are per dwelling, the price of the house would increase[1].

2 Evaluating Model Performance

2.1 Metrics

The best metrics for evaluating the performance of the model is `mean_squared_error`, as the model is that of regression: the price value outputs are continuous rather than discrete. Because there are no set labels, metrics such as `accuracy_score` which necessitates an exact match of labels will perform poorly for this type of data set. Furthermore, unlike `mean_absolute_error`, `metrics_squared_error` penalizes the data on an instance of a large error; this is due to the fact that each error is further weighted by itself(squared). A Single big error would count more negatively in the scoring than multiple small errors[2].

2.2 Testing

It is critical to split the data into training and testing data, in order to avoid overfitting. To evaluate the model without excessive variance – in which the model becomes overly complicated in order to adapt to the given data but fails to generalize for other cases – having a set of testing data allows to validate the model against *general* cases.

2.3 Optimization

It is useful to employ Grid search in fitting the model, as it searches for the optimal combinations of values for a number of different parameters; in this case, it was used for only one parameter, which is the maximum depth of the decision tree.

2.4 Cross-Validation

Cross validation is setting aside another set of data to evaluate the performance, and minimizing the error accordingly. For grid search, this is useful since we can set aside a set of data to test how the model performs as some of the parameters(maximum depth of the decision tree) are adjusted.

[1]In terms of the LSTAT parameter, it may as well be that the price of the house in the neighborhood is responsible for the greater proportion of wealthier people.

[2]However, in this case, the two metrics(squared and absolute) produced similar predictions.

3 Analyzing Model Performance

Figure 3: Graph of Error v. Size of Training Data at max depth of 2.

3.1 Training Size

Generally, as training size increases, training error increases[3] and testing error decreases.

3.2 Max Depth

For the depth of 1 (least complex model), neither the testing data nor the training data displays a significant reduction in error, which is an evidence of underfitting (the model cannot adequately represent the complexity behind the real mechanics of mapping from the input to the output)

For the depth of 10 (most complex model), the model fits the training data very well, but doesn't necessarily perform well on testing data. This is an evidence of overfitting (the model represents the training data very well, but in doing so fails to generalize for the testing data, not performing as well or even worse).

[3]but the rate of increase decreases, which indicates learning.

3.3 Model Complexity

Figure 4: Graph of Error v. Maximum Depth of the Decision Tree

According to the model complexity graph, the testing error and the training error begins to diverge at a maximum depth of approximately 2, at which point the tree adapts itself to the training data. However, it is natural that the model would perform better against the training data; looking at the test data, the error consistently decreases until the max depth of about 6, after which it displays erratic behavior. Thus, it seems that overfitting occurs after the depth of 6, and the model best generalizes the dataset at the depth of 6.

4 Model Prediction

4.1 Optimal Max Depth

Grid search identified the max depth of 6 as most optimal for the model. Whereas I initially thought the point at which the divergence occurs for testing and training error(2) was the point of minimum error, the actual overfitting seems to start a bit later, thus reaching the midpoint between underfitting and overfitting at the depth of approximately 6.

4.2 Prediction

Assuming "best" means most consistent with market consensus, the model predicted the best selling price for the given features to be $21630. Verifying the results with K-nearest neighbor regressor, the prediction reports $23500, similar to that obtained via the decision-tree network.

4.3 Application

While the results are indeed similar, given that their difference amounts to $1900 (quite significant), I would take the learned output into account when determining the price but not rely entirely on the result.