December 2015
Carl Gosselin

Project 1 – Predicting Boston Housing Prices

1) Statistical Analysis and Data Exploration

- Number of data points (houses): 506

- Number of features: 13

- Minimum and maximum housing prices:
  - Minimum: 5.0
  - Maximum: 50.0

- Mean and median Boston housing prices:
  - Mean: 22.53
  - Median: 21.1

- Standard deviation: 9.188

2) Evaluating Model Performance

- Which measure of model performance is best to use for predicting Boston housing data and analyzing the errors?

  I chose the **mean absolute error** as the most appropriate measure for this exercise.

  - Why do you think this measurement most appropriate?

    The **mean absolute error** is more robust to outliers/ extremes. I wanted minimal influence from outlying housing prices (outliers).

  - Why might the other measurements not be appropriate here?

    The **mean squared error** emphasizes the outliers and extremes. I think the pricing score is better predicted when outliers are not emphasized.

    The **median absolute error** is less appropriate as I don't think the value that is halfway through an ordered data set would give the most appropriate prediction.

    The **r2_score** and **explained variance score** are acceptable metrics if all is needed is a quick look at the performance of the model but it doesn't provide any more information, especially for analyzing the errors. The score for both of these metrics are on a scale of 0 to 1. 1 being the best possible score.

December 2015
Carl Gosselin

Project 1 – Predicting Boston Housing Prices

- Why is it important to split the Boston housing data into training and testing data?

  The data split serves as a check on overfitting. It's also a way to test the model on unseen (independent) data. In other words, the testing partition is used to test the performance of the model after it has learned from the data in the training partition.

- What happens if you do not do this?

  If there is no split in the data, the model will be tested on data that has been used during the training process. This would create a desirable test score. However, the predictive performance of the model on new/unseen data will still be unknown.

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

  Grid search does an exhaustive search through a list of parameters using a chosen estimator and returns the optimal parameter (e.g. depth). Grid search is guided by a performance metric that is chosen by the data analyst which is deemed most appropriate for the data at hand. Grid search is used to find the best, or optimal, parameters to a model.

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

  Cross validation is used to validate the performance of training prior to being tested with the final test dataset. By default, GridSearchCV has a 3-fold splitting strategy. Instead of holding out yet another set of data to validate training, different segments of the data within the training partition is used for each validation fold. Cross validation minimizes waste of data. In a sense, it helps you do more with less data.

December 2015
Carl Gosselin

Project 1 – Predicting Boston Housing Prices

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?

  As the size of training size increases, testing error seems to increase while the training errors decrease. This is a case of overfitting the data. Too much attention is paid during the training phase. The model does not generalize well when new data is introduced.

- 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 model suffers from high variance/ overfitting at depth 10 - the training line nears to zero errors and the testing line continues to display a high level of test errors.

- 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 training and test errors run in parallel form until depth 5. I would say that depth 5 best generalizes the dataset. After depth 5, the two error lines begin to move away from each and display different behaviour - the training error line continues to descend smoothly down to virtually no errors where the test error line continues to fluctuate. After depth 5, the dataset starts to show signs of overfitting (much higher errors on test than training).

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 model chose the best parameter as max depth 5 with a house price prediction of 20.765986

In comparing the prediction to earlier statistics, the model places the house price slightly below the mean and median price for the Boston dataset, 22.53 and 21.1 respectively. This looks to be a valid model when looking at the features of the house and comparing them to the features/price relationship of the houses in the Boston dataset.