Boston housing report

1) Statistical Analysis and Data Exploration

- Number of data points (houses)?

The dataset contains 506 data points

- Number of features?

The dataset contains 13 features describing the houses

- Minimum and maximum housing prices?

The cheaper house in the dataset is worth 5.0
The most expensive house in the dataset is worth 50.0

- Mean and median Boston housing prices?

The average price of a house in the dataset is 22.53
The median price of a house in the dataset is 21.2

- Standard deviation?

The standard deviation of prices of the dataset's houses is 9.19

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?

We are here trying to predict a continuous outcome. Hence we should use a metric that allow us to estimate how far our prediction is from our target value. I have chosen mean squared error in order to accentuate the penalty when the prediction is far away from target.

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

It is important to split the data between training and testing because it enables to train our predictive model on one set of data and to assess the performance of the model on “unseen data”. Assess the predictions made on the data we hold out gives us a good evaluation of the ability of the model to perform well on new data.

If we only evaluate the model on data the model has seen in the past, we are unable to evaluate correctly how it will behave in the future when using it on new data points. Which is really what we expect from our algorithm.

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

Grid Search allows to fit the model with different parameters. It saves a lot of time to tune the algorithm parameters.

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

Cross validation is different from a mere train / test split.

We partition the data in several folds. We use one fold for testing, the others for training. We then iterate by turning a different fold into the new test fold, we fit our model on the new train fold and compute our error metric. Finally we average the results of the selected error metric to evaluate the model performance.

There are several benefits to it:

- It allows to use the whole dataset to estimate our error
- It allows to use the whole dataset to fit our model

It is helpful to use it with grid search because if you don’t do this there is a chance you will pick up a parameter of your model as a result of the randomness involved in splitting your data in two parts.

Increasing the number of folds will yield more accurate results but requires more computational resources. Hence a balance has to be found.

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?

Testing errors decreases sharply until training size reaches 50. Pattern is not obvious after but testing errors seem to decrease slightly.

Training errors increases with size, specially when the model is simple.

In all cases, training errors remain inferior to test errors. When increasing training size training and testing errors are getting closer.

- 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?

With max depth 1, training errors suggest that the model suffers from high bias. The model is not powerful enough to capture the underlying structure of our data.

With max depth 10, the model fits perfectly the training examples. The gap between training and testing data indicates that our model is specialized in the training data and suffers from overfitting.

- 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?

Somewhere between max depth equals 5 and 7, testing errors seems to reach a plateau. All other things being equal we would prefer a simple model.

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.

Price Predicted : 20.96

Corresponding max depth : 5

Predicted price is close to 21, almost the median of our dataset, close to the mean, a good sign that our model works well.