Statistical Analysis and Data Exploration

This question is integrated into the project notebook output.
Using the NumPy library, calculate a few meaningful statistics about the dataset:

> *Boston Housing dataset statistics (in $1000's):*
>
> *Total number of houses: 506*
> *Total number of features: 13*
> *Minimum house price: 5.0*
> *Maximum house price: 50.0*
> *Mean house price: 22.533*
> *Median house price: 21.2*
> *Standard deviation of house price: 9.188*

1) Of the available features for a given home, choose three you feel are significant and give a brief description for each of what they measure.

> *CRIM: This measures the crime rate per person by town. As crime can be have a significant effect on housing price, this rate is important to consider.*
>
> *TAX: This measures the property tax per $10,000. Many people use this rate to compare trade-offs of other values.*
>
> *PTRATIO: The is the ratio of pupils to teachers by town. For someone with kids, this statistic may be very important to consider.*

2) Using your client's feature set `CLIENT_FEATURES` in the template code, which values correspond to the chosen features?

> *CRIM: 11.95*
>
> *TAX: 680.0*
>
> *PTRATIO: 20.2*

Evaluating Model Performance

3) Why do we split the data into training and testing subsets?

> *Splitting the data allows us to train our algorithm with a specified percentage of the data set and then do cross validation with the performance of that training with the remainder. This ensures we don't get false positive results by using the same data for both purposes. Of course the model will work if you use the same set of data for testing as you used for training. The training subset teaches the model how to see the data and the testing subset is for turning a trained algorithm loose on the data. The more you dedicate to training, the better the model, but the less you have to test against. Finding the right mixture is essential to building a good machine learning model.*

4) Which performance metric below is most appropriate for predicting housing prices and analyzing error? Why?

> *I chose Mean Absolute Error because we are working on a continuous data set and this is a*

regression problem, not one of classification. I like MAE because all values are positive, preventing errors in opposite directions from canceling each other out. Additionally, the errors are not as intensified as they would be in a Mean Squared Error approach, which would place increasingly heavier weight as errors size was larger.

5) What is the grid search algorithm and when is it applicable?

The grid search algorithm exhaustively searches a parameter space for the best score for cross validation. This is very useful when you are trying to tune a specific parameter. In the case above, we are tuning the max_depth parameter by testing every instance between 1 and 10 to find the best score, which is provided by the make_scorer function.

6) What is cross-validation and how is it performed on a model? Why would cross-validation be helpful when using grid search?

One method of cross validation is k-fold cross validation. In this method, the data set is split into k number of equal sized parts and then a single "fold" is held back as testing data while the rest is used for training data. This is repeated for each of the folds so that by the time it is finished, all observations have been used as both training an testing data. This allows an observer to obtain an average metric of how the machine learning algorithm will perform. In a limited data set, this can provide leverage to report results as if it were a much larger data set. Compared to a static training and testing split, the k-fold approach leverages the same amount of data to produce a far more optimized algorithm because it is much less susceptible to overfitting or accidental bias.

For grid search, it is useful because it allows each point in the parameter space to be tested against several different splits in the data and averaged into a performance number, thereby providing a more robust output as to which parameter produces better results.

Analyzing Model Performance

7) Choose one of the learning curve graphs your code creates. What is the max depth for the model? As the size of the training set increases, what happens to the training error? Describe what happens to the testing error.

I chose the graph for max_depth of 3 to answer this question. The errors in the training set increased as the number of data points in the set did. There was a bump around 150 data points, but it mostly flattened out after about 100.

For the testing set, the errors jumped up and down quite a bit as the number of data points increased, but the error rate generally trended downward.

8) Look at the learning curve graphs for the model with a max depth of 1 and a max depth of 10. When the model is using the full training set, does it suffer from high bias or high variance when the max depth is 1? What about when the max depth is 10?

When the model has a max_depth of 1, it suffers from high bias which means that the error rate between the testing and training data sets are very close. In the max_depth of 1 graph, we can see that the error rate hovers around 5-6 and there is no significant difference in error rate as the number of data points increase.

When the model has a max_depth of 10 it suffers from high variance. Although the training

errors are fairly low in this model, the testing errors are all over the graph and far higher than the training set. This indicates that the model may have been overfitted to background noise or incorrect data associations.

We can possibly fix the variance with more data points and training further, reducing some of the overfitting and making the model generalize better. However, the bias likely will not turn into a better fit no matter how many more data points we throw at it because the bias indicates that the algorithm has made assumptions in the training set that only work well with the training set and do not translate well outside that limited view. Therefore, the usefulness of the algorithm on as-yet-unseen data is limited if it is useful at all.

9) From the model complexity graph, describe the training and testing errors as the max depth increases. Based on your interpretation of the graph, which max depth results in a model that best generalizes the dataset? Why?

The training errors decrease as the complexity increases. However, the testing errors only decrease to a certain point at which time the error rate stabilizes and holds fairly constant regardless of additional complexity.

A depth of 7 appears to be the optimal generalization because the errors for both training and testing sets are relatively close and the errors for testing are at the lowest point. Training errors do go lower, but the variance increases because testing errors do not. Any additional complexity beyond 7 appears to increase overfitting.

Model Prediction

To answer the following questions, it is recommended that you run your notebook several times and use the median or mean value as your result.

10) Using grid search, what is the optimal max depth for your model? How does this result compare to your initial intuition?

The optimal max_depth is close to what I concluded based on the graph of errors versus depth. I had originally determined a depth of 7 to be the most efficient but the system determined that 6 was more optimal. My best guess on the difference is that with a depth of 6 the training errors were near lowest point and the variance was quite a bit less compared to a depth of 7.

11) With your parameter-tuned model, what is the best selling price for your client's home? How does this selling price compare to the statistics you calculated on the dataset?

The best price for my client's home is 20.766. My client's home is slightly below both the mean and median calculated against the data set earlier.

12) In a few sentences, discuss whether you would use this model or not to predict the selling price of future clients' homes in the Boston area.

If I had more data, I think this model would be very useful for calculating housing prices. The data would need to be up-to-date because the housing market reacts to far more than just raw prices and features. But those things at a point in time are very indicative of the going price for a house in a specific market. I think this would be interesting to validate against a larger set of data to see how well my model holds up. As this model could continuously learn from a database, I would predict this model getting better and better over time.