P1: PREDICTING BOSTON HOUSING PRICES

ANDY PEREZ

1. Statistical Analysis and Data Exploration

- Number of data points (houses)?
  506
- Number of features?
  13
- Minimum and maximum housing prices?
  All housing prices are within the range [5.0, 50.0]
- Mean and median housing prices?
  Mean housing price is 22.53
  Median housing price is 21.20
- Standard deviation of the housing prices?
  The standard deviation of the housing prices is 9.19

2. Evaluating Model Performance

- Which measure of model performance is best to use for predictiong Boston housing data and analyzing the errors? Why do you think this measurement most appropriate? Why might the other measurements not be appropriate here?

  I decided to settle on the MSE (mean squared error) because it invovles both the variance and the bias of our estimator, ensuring that both are minimized. The other options available are median absolute error, mean absolute error, R^2 score, and explained variance. The median absolute error produces a metric that is rather noisy, or at least when applied to this project. It is indeed robust in the face of outliers, but the data range is relatively reasonable, so it isn't neccesary. I can't really justify the use of MSE over mean absolute error to be honest, it's difficult to discern their differences, besides the fact that the former (MSE) greater emphasizes large errors, and that MSE is used internally in determining splits for the decision tree regression. R^2 and explained variance error evaluations are limited; They're not very useful for determining bias, for example. Hence, I'd rather stick with either MSE or the mean absolute error, of which the former was chosen.
- Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?

  As the depth of the decision tree regression gets larger and larger, the bias

tends towards zero at quite a fast pace, at least compared to the scale of typical prediction error you would have on data outside of the training set. Hence, if you did not split the data set, you would ridiculously overfit the data while thinking you have a near-perfect model that in reality, suffers from huge variance and is of low predicative value.

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

  Grid search is the evaluation of the performance of a parameterized series of models through an exhaustive search throughout a given subset of hyperparameters. In our case, we want to choose an optimal decision tree maximum depth.

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

  Cross-validation is extremely useful. It helps make the grid search less prone to randomness in the choice of testing set, since every point of data gets to, at one point, be part of a testing set. Furthermore, since error is averaged over each train/test split performed in cross-validation, it is a more reliable measure of model performance, especially when data sets are small; It's almost as if you had more data!

3. Analyzing Model Performance

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

  Training error increases as training size increases, at a slower rate the higher the depth of the decision tree. Test error decreases with respect to training size before plateauing.

- 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 the model with a max depth of 10, the bias is nearly nonexistent, the test error is thus nearly all variance, which tends to go down but seems to plateau to a RMSE (root mean squared error) of about 4.5. This model is overfitted. For our model with a max depth of 1, there is a very high bias of about 6.3, which accounts for most of the error of about 7.4, it is underfitted.

- 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?

  On my comlexity graph, the test error seems to drop until somewhere in the vicinity of a max depth of about 5 or 6. There is too much noise to discern any trend afterwards, the RMSE continues to hover somewhere around 5.4, while the training error goes to zero as expected. Thus, I would choose 6 as best generalizing the data. My graph has an actual global minimum there (although the noisiness of the graph is such that it's debatable whether it actually is a minimum). I wouldn't want a depth less than four, and for models with higher max depths than neccesary, the test error would be completely variance, which isn't desirable.

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/resonable price/model complexity.

  There's a moderate amount of randomness in the choice of tree returned by grid search, but after eighty runs, the clear favorite was a regression decision tree of depth 4, with a prediction of 21.63; This tree was chosen 51 times in total, with trees of length 5-9 chosen about four times on average, never one of depth less than four.

- Compare prediction to earlier statistics and make a case if you think it is a valid model

  The noise in my error estimations makes me hesitant to apply the results from the complexity curve. The result of a tree of length four being optimal is, however, consistent with the curve, which demanded at least a depth of 4. The prediced value of the given home for a tree of depth 4 is 21.6297, which passes the sanity test; it is close to the mean and median of our home price data.