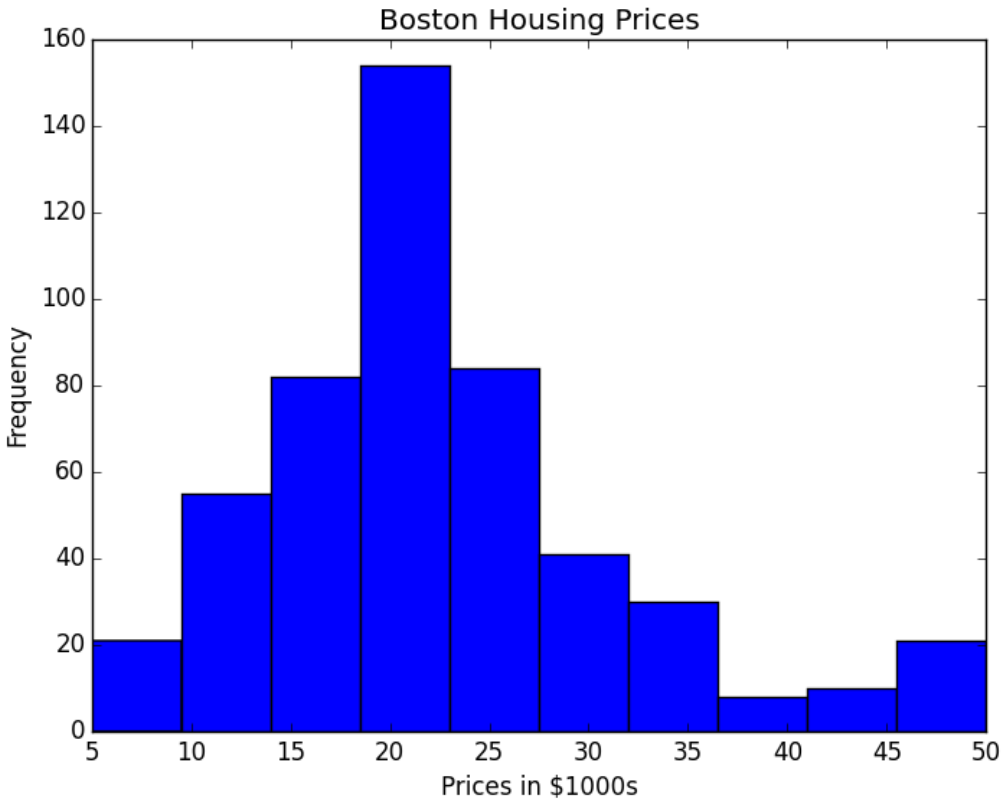
Questions and Report Structure

1) Statistical Analysis and Data Exploration

- Number of data points (houses)?
  - `The number of houses is 506`
- Number of features?
  - `The number of features is 13`
- Minimum and maximum housing prices?
  - `The minimum price of a house is 5.0`
  - `The maximum price of a house is 50.0`
- Mean and median Boston housing prices?
  - `The average price of a house is 22.5`
  - `The median price of a house is 21.2`
- Standard deviation?
  - `The standard deviation of housing prices is 9.2`



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?
    - Mean Absolute Error (MAE) is most appropriate for the non-Gaussian nature of the housing prices. Instead of using Mean Squared Error (MSE), I opted for MAE because it is more robust to outliers since the median plus the IQR revealed some outliers and a test for normality revealed an infinitesimal p-value. On the other hand, using MSE produced the same end result for the final housing price. There are other regression scoring metrics for determining how well the model fits the data, like Explained Variance and R2, but these do not measure the error so much as they measure variability in predictions.
    - Other metrics, specifically Classification metrics (ie. accuracy, precision, recall, etc.) would not make sense for the continuous output that is the price of a house.
- Why is it important to split the Boston housing data into training and testing data?
    - It is important to split the Boston housing data into training and testing sets so the model's accuracy can be measured against new information. Splitting the data also means you do not have to search for new data after the model is trained.
- What happens if you do not do this?
    - If the data is not split and the model is fit to the entire set, then there is no reliable way of predicting how the model will perform when it encounters new information.
- What does grid search do and why might you want to use it?
    - Grid search allows the model to be validated over a set of parameters to find the parameters with the lowest cross-validation error. Essentially, grid search trains a model for each parameter in the set and measures that model against the cross-validation set. This exploration of parameters is to compare the accuracy of each model on an independent data set. Cross-Validation is used in conjunction with Grid Search because if the end model was based on tuning to the test set, then the performance of the model on

the test set would no longer be valid because of the potential to choose the model that only performs well on the test set and not further new information aka overfiiting.

- Why is cross validation useful and why might we use it with grid search?
    - Cross-Validation helps to prevent the two main causes of error in a model: bias and variance. A model built on training data can be measured against and tuned to new data that is not in the testing set. CV allows for the possibility of increased model complexity without over-fitting. Cross-Validation can be done in a variety of ways, but basically splits the data from your training set into a smaller training set and a validation set. This allows grid search to test a trained model against data it did not train without the risk of overfitting on the test set. It also expresses how robust a model might be to new information. You would not use the Cross-Validation set to express the effectiveness of your model because it is still possible to overfit the data with more iterations of parameter testing on the same Cross-Validation set.

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?
    - The testing error decreases as training size increases until leveling out. On the other hand, the training error increases with increasing training size as the model looses its ability to compensate for the natural variation in the data. For the fully trained models, as the max depth of the tree increases, both the training and testing error decrease until there is insufficient data to train away the high variance.
- 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 with a max depth of 1 suffers from high bias. This is indicated in the graph when the training line and the testing line converge at a high error. The test error will not improve beyond that of the training error. Seeing as models with deeper depths have a lower training and test error, there is much room for improvement. Simply put, the model with max depth 1 is underfitting the data.

- On the other hand, the model with a max depth of 10 has high variance. This is not a terrible situation as a model with high variance can be improved by training on more data. For this particular situation, however, the model with a max depth of 10 is overfitting the training data. In the graph this is shown by a very low training error while the test error remains somewhat high.

- 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?
    - While the training error will decrease to 0 with increasing complexity, the testing error levels off after roughly five increases in max depth. Using Occam's razor as a guiding principle, I would say that the model with a max depth of 4 is the simplest and most accurate model.
    - As an additional note, though it is not apparent from the complexity graph below, the test error will start to rise after reaching a minimum. Essentially, though the test error looks like it is leveling off, it will actually rise as the complexity of the model is more capable of overfitting the training data.


Decision Trees: Performance vs Max Depth
test error
training error
Error
Max Depth
0
1
2
3
4
5
6
0
5
10
15
20
25

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.
    - `The best parameter according to GridSearchCV is a max_depth of 5.`
    - `House: [11.95, 0.0, 18.1, 0, 0.659, 5.609, 90.0, 1.385, 24, 680.0, 20.2, 332.09, 12.13]`
    - `Prediction: [ 20.96776316]`
- Compare prediction to earlier statistics and make a case if you think it is a valid model.
    - The prediction seems in line with the earlier statistics given that it does not produce an outlier or anything at extreme ends of the IQR. In fact, we have a price roughly in the center of the housing price spectrum. On the other hand, our earlier statistics did not take into consideration the conditional dependencies that the model did in the regression tree. I would still consider the model more valid than a strict average of the prices given that its mean absolute error for the validation sets used in Grid Search was roughly $3.7 thousand with its own standard deviation of 0.729. Comparing that standard deviation with the standard deviation of the housing prices, which was 9.2, I believe the model adds value in the form of a more accurate price. To verify that even further, when the model is measured against the test set, the mean absolute error is roughly $2,000.