Predicting Boston Housing Prices

1. Statistical Analysis and Data Exploration

Here are some basic statistics and features description that describe the dataset analyzed in this project.

Number of records	**506**
Number of features	**13**
Minimum housing price	**5** ($5,000.00)
Maximum housing price	**50** ($50,000.00)
Mean housing price	**22.532**($22,532.81)
Median housing price	**21.2** ($21,200.00)
Standard deviation	**9.18**($9,180.01.00)

Feature description:

CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq. ft.
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centers
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

2. Evaluating Model Performance

1. **Which measure of model performance is best to use for regression and predicting Boston housing data? Why is this measurement most appropriate? Why might the other measurements not be appropriate here?**

   In regression type of problems we are dealing with model that make predictions on continuous data. In this case we care how close the prediction is. Since Boston housing is regression problem for evaluating the model's performance, we should consider performance metrics for regression (mean absolute error, mean squared error, median absolute error, explained variation or the coefficient of determination).

   Mean squared error is used as a model performance metric among the available score metrics for regression. Mean squared error is a common metric for regression problems, it converts all the errors to positives (the difference between predicted and the true value is squared). Mean

squared error gives more weight to bigger error values (outliers), which based on my assumption might be common in any housing data.

Another alternative to MSE is the mean absolute error which is more robust to outliers, so I think that MSE is more appropriate for the task.

Using classification metric for regression problem will not make a sense.

2. **Why is it important to split the data into training and testing data? What happens if you do not do this?**

   We separate training and testing sets so we can get a better idea whether the model can generalize to unseen data rather than fit to the data just seen. In order to properly evaluate the model, the data must be split into two sets: a training set and a testing set. The training set's is used to train the model, while the testing set is used for evaluating the trained model's performance in data that it has never seen before. If we don't split the data, we risk having a model that can only make good predictions with the training data set and, so we would end up with an over fit model.

3. **What does grid search do and why might you want to use it?**

   GridSearchCV is a way of systematically working through multiple combinations of parameter tunes and cross-validating as it goes to determine which tune gives the best performance. Grid search is used to find and evaluate model for best tuning parameters. Testing against 10 depth parameters with grid search helps us to find the best parameter that optimize training and test errors.

4. **Why is cross validation useful and why might we use it with grid search?**

   Cross-validation is a way of measuring the predictive performance of a statistical model. The method of statistical sampling to get a more accurate measurements is called cross-validation. The aim in cross-validation is to ensure that every example from the original dataset has the same chance of appearing in the training and testing set. Cross validation allows us to avoid the problem of overfitting.

   Cross validation can be easily done using grid search as it can take the train and test data separately as arguments and automatically check for best model parameters using CV parameter.

   When a dataset is limited in size, as in our case, cross validation becomes really useful. K-fold cross-validation is used for model validation (3-fold cross-validation), this method provides maximum accuracy.

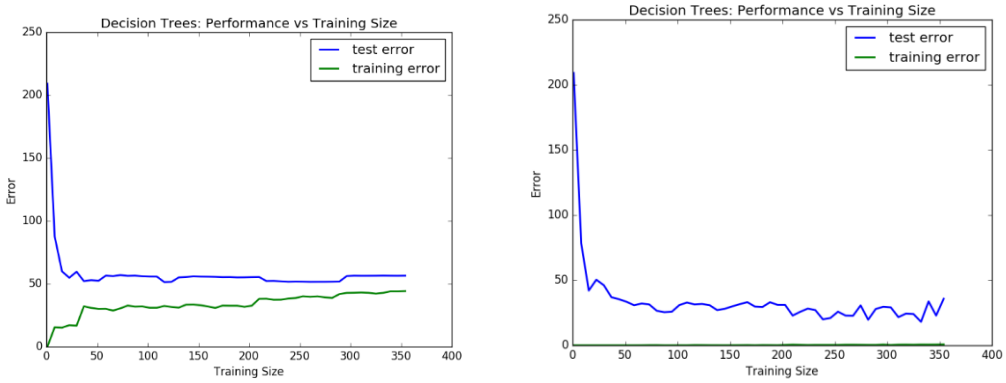
3. Analyzing Model Performance

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

   Learning curves helps us to understand how better the model gets at predicting the target as we increase the number of instances used to train it.

   When we look at the relationship between the training data size and error we should generally see how as training point's size increases training error slightly increases and testing error falls. Learning curve graphs show that training and testing errors converge with increase of the training size. This should make sense since we are trying to build models that learn from experience.

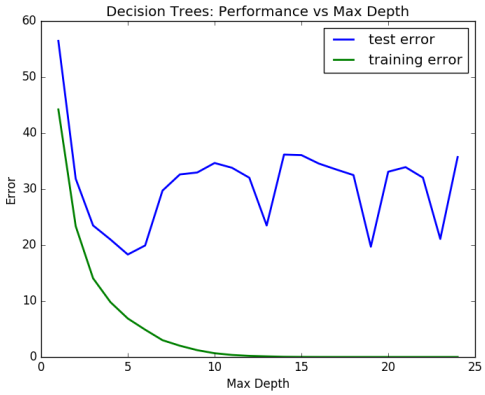
2. **Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?**



   On the first learning curve graph (max_depth = 1) the training and testing errors converge and are relatively high which means that the model suffers from high bias / under fitting. On the last graph (max_depth = 10) the training error is almost zero, which implies a perfect fit, and the training and testing errors have a gap in between, that leads me to conclude that when the model is fully trained it suffers from high variance/overfitting.

   Ideally, both training and test errors should become relatively low.

3. **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?**



Model complexity graph shows that with increasing model complexity the training error reduces exponentially but after a certain point training error flattens out. Test error decreases exponentially but after a certain point test error increases slightly or flattens out. Max depth bigger than 10 does not seem to affect the training error and only increases variance as the test error grows with increase of max depth.

Since the preferred model complexity is usually the one that minimizes test error. The best generalization gives the model with max depth 4, since the testing error is the smallest, after max depth 4 the testing error is increasing as a result of the overfitting.

4. Model Prediction

1. **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 generalization is confirmed by calling the best_params_ member of GridSearchCV, which gives values 4 and 6 with high frequency. This is due to the random sampling done by cross-validation. For this model I'll go with max_depth 4 since it was the most frequent value.

   - Most common max_depth = 4
   - Prediction = 21.62974359 ($ 21,629.74),

2. **Compare prediction to earlier statistics and make a case if you think it is a valid model.**
   - The price of **21.62974 ($ 21,629.74**) is reasonable since it is within the standard deviation range (9.188) of the mean 22.532($22,532.81) and it lies within the min/max range of [**5** ($5,000.00) - **50** ($50,000.00)].