

Questions and Report Structure

1) Statistical Analysis and Data Exploration

- Number of data points: **506**
- Number of features: **13**
- Minimum housing prices: **5.0**
- Maximum housing prices: **50.0**
- Mean Boston housing price: **22.53**
- Median Boston housing price: **21.2**
- Standard deviation: **9.18**

2) Evaluating Model Performance

- 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?

I think that R^2 is best to use because it is scaled (I could use the explained variance for the same reason, but I have to choose only one). It can help us appreciate the goodness of our model, regardless the order of magnitude of the values.

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

It is important to split the data because it lets you know how your model performs on a sample other than the one it was trained on, thus avoiding over fitting. If you don't do this, you might end up with a model that performs well on the training data but that is unable to generalize on future data sets.

- Which cross validation technique do you think is most appropriate and why?

k-fold cross-validation is most appropriate because it uses every data point both for training and validation, K-1 times for training and 1 time for validation.

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

Given an algorithm and sets of parameters values, grid search builds a model for any parameters values combination using cross validation. It then determines which combination of parameters gives the best performance. We might use it to automatically tune our machine learning algorithm.

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?

As the training size increases, the training performance generally decreases while the testing performance increases and they seem to converge with bigger training sizes and lower max depths.

- 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 max depth 1, we can see that the training and test performance converge and are not good (lower than 0.5). This means that the model suffers from high bias/underfitting.

For max depth 10, the performances are quite high, but there is a gap between the training performance (near 1) and the testing performance (under 0.8). This is a sign of high variance/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?

Training performance increases with model complexity while test performance seems to be more stable.

On the other hand, GridSearchCV suggests a max depth between 4 and 6 over subsequent executions.

Based on this, the max depth of 4 should be the best one since it produces a less complex model. Even if it has a bad training performance, its test performance is near the one of other more complex models. Also, it suffers from less variance/overfitting than the other models, since the gap between training and test performances is lower.

4) Model Prediction

- Model makes predicted housing price: **21.63**
- Detailed model parameters: DecisionTreeRegressor

Parameter	value
criterion	'mse'
max_depth	4
max_features	None
max_leaf_nodes	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
random_state	None
splitter	'best'

- Compare prediction to earlier statistics

The prediction is between the mean and median housing prices so it might be a reasonable price.