How can we increase the r2 value

Feature Engineering: Create new features or transform existing ones to better capture the underlying patterns in the data.

Hyperparameter Tuning: Optimize the hyperparameters of the DecisionTreeRegressor.

Model Selection: Try different models or ensemble methods.

Data Cleaning: Ensure the data is clean and free of outliers or errors.

Cross-Validation: Use cross-validation to ensure the model generalizes well.

However, we can further explore the impact of regularization by adjusting these parameters more finely or by using techniques like pruning.

Pruning involves removing branches of the tree that do not provide significant information gain, while regularization involves constraining the tree's growth to prevent it from becoming overly complex.

<https://medium.com/@deryl.baharudin/why-we-need-to-do-regularization-in-decision-tree-machine-learning-70e77ac48b79>

We can prune our decision tree by using information gain in both post-pruning and pre-pruning. In pre-pruning, we check whether information gain at a particular node is greater than minimum gain. In post-pruning, we prune the subtrees with the least information gain until we reach a desired number of leaves.

What is cost complexity pruning

<https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html>

The DecisionTreeClassifier provides parameters such as min\_samples\_leaf and max\_depth to prevent a tree from overfiting. Cost complexity pruning provides another option to control the size of a tree. In DecisionTreeClassifier, this pruning technique is parameterized by the cost complexity parameter, ccp\_alpha. Greater values of ccp\_alpha increase the number of nodes pruned. Here we only show the effect of ccp\_alpha on regularizing the trees and how to choose a ccp\_alpha based on validation scores.

Don’t know if this is good

Pre-pruning (Early Stopping):

Maximum Depth (max\_depth): Limit the depth of the tree. Nodes beyond this depth will not be split.

Minimum Samples per Leaf (min\_samples\_leaf): Ensure that each leaf has at least a minimum number of samples.

Minimum Samples per Split (min\_samples\_split): Require a minimum number of samples to split a node.

Maximum Features (max\_features): Limit the number of features to consider when looking for the best split.

Post-pruning (Cost Complexity Pruning):

Reduced Error Pruning: Remove nodes if it does not degrade the performance on a validation set. This is usually done bottom-up (from leaf nodes to the root).

Cost Complexity Pruning (CCP): Also known as weakest link pruning. This involves pruning the nodes of the tree in a way that reduces the cost complexity criterion, which is a trade-off between the tree's size and its error rate.