# Cardiovascular Death Rate Capstone: In-Depth Analysis

Zach Wimpee

November 8, 2020

## 1 Brief Review

- **Goal** - Build a model to predict annual change in a county's cardiovascular death rate
- **Data** - a data set has been constructed and stored in a pandas DataFrame. Each row corresponds to an individual county for a single year between 2011-2018. The columns store values for a set of Census variables as well as the cardiovascular death rate.
- **Statistical Insights** - if the data is partitioned by the resulting sign change in cardiovascular death rate, then a statistically significant difference can be found between the feature distributions for the members of the partition.

Expanding on this third bullet point, **the data appears to lend itself to a binary classification model**.

## 2 Model Building

The aim of this project is not only to build a predictive model for changes in cardiovascular death rate, but also to recommend a course of action from its conclusions. The Scikit-learn **DecisionTreeClassifier** seems like a natural choice, since such a model would provide a greater level of interpretability than, say, a SVM classifier. By looking at the splitting criterion of the internal nodes, it should be easy to see what areas could be addressed to prevent increases in cardiovascular death rates. However, overfitting is a known limitation of Decision Tree classifiers. If issues with overfitting are encountered then an ensemble learner will be built from the Scikit-learn **RandomForestClassifier**.

## 3 Decision Tree

### 3.1 Parameter Tuning

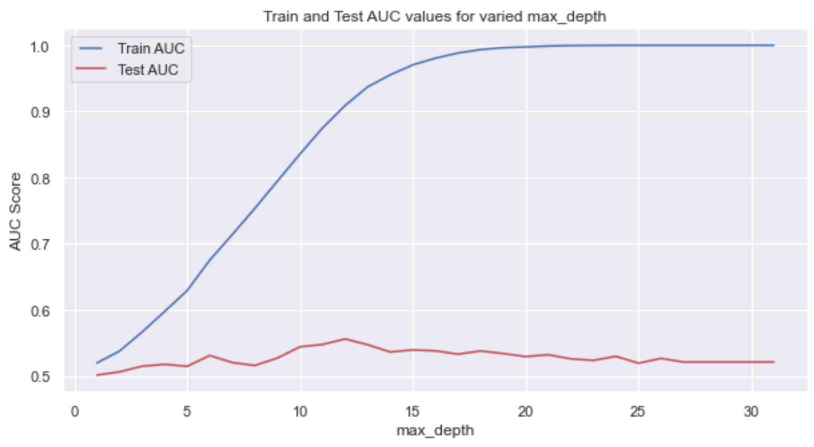
The Scikit-learn **DecisionTreeClassifier** has several parameters that can have a significant impact on model performance:

- max_depth
- min_samples_split/min_samples_leaf
- max_features
- max_leaf_nodes

To begin tuning these parameters, they will first be considered separately. A decision tree will be trained and tested over a range of values for each parameter. The tree's performance for each parameter value will be measured using the Area Under the Receiver Operating Characteristic curve (AUROC) for both the training and testing sets. This metric is chosen because it is a good indicator of how well the model can distinguish between the two classes. The train and test set AUROC scores will then be plotted together versus the corresponding parameter values. These figures will then be used to determine the range of values for each parameter to use in a grid search cross-validation of parameter configurations.

### 3.1.1 max_depth

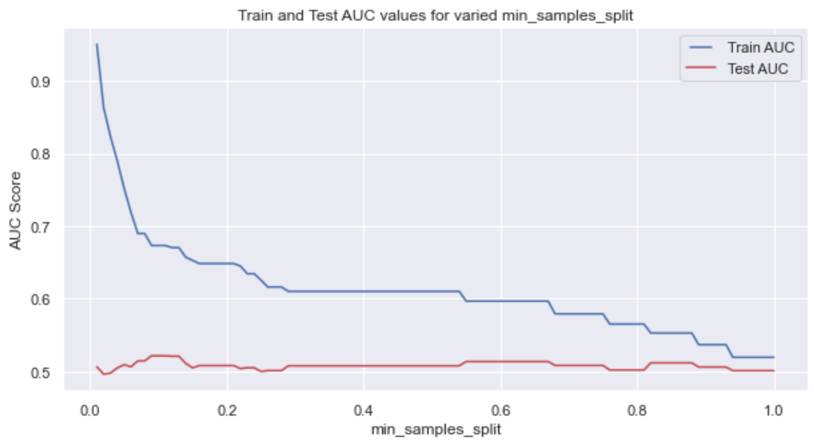
As its name suggests, this parameter controls the maximum depth for which to grow the tree. For example, setting $max\_depth = 1$ would result in a tree with a single split into two leaf nodes. Too high a value for this parameter can lead to high variance and will overfit the training set. Too low a value can lead to high bias and will underfit the training set.



Unfortunately, generalization to unseen data is poor across the board regardless of the value chosen for this parameter. Even more unfortunately, **this issue will continue to be seen throughout the parameter tuning procedures, and can likely be attributed to poor predictive power of the constructed feature set.** However, for the purposes of this project we will work with what we have and simply keep in mind what may have gone wrong or what should have been done differently.

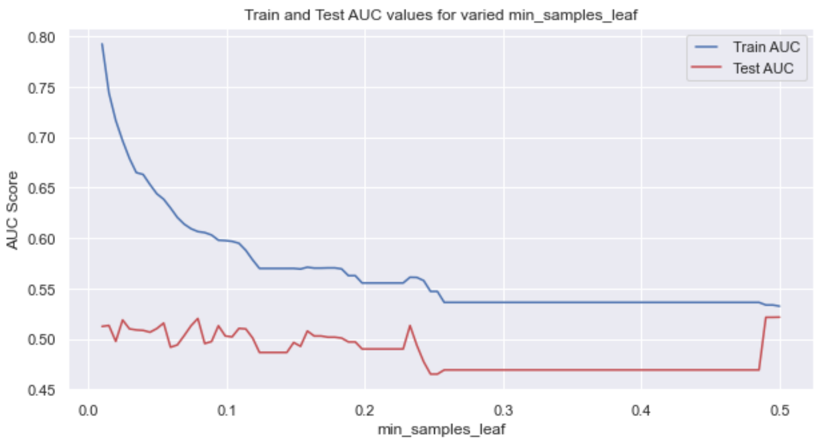
### 3.1.2 min_samples_split

This parameter designates the minimum number of samples required to split an internal node. It can be either an integer value greater than 1, or a floating point value for a fraction of the total number of samples.



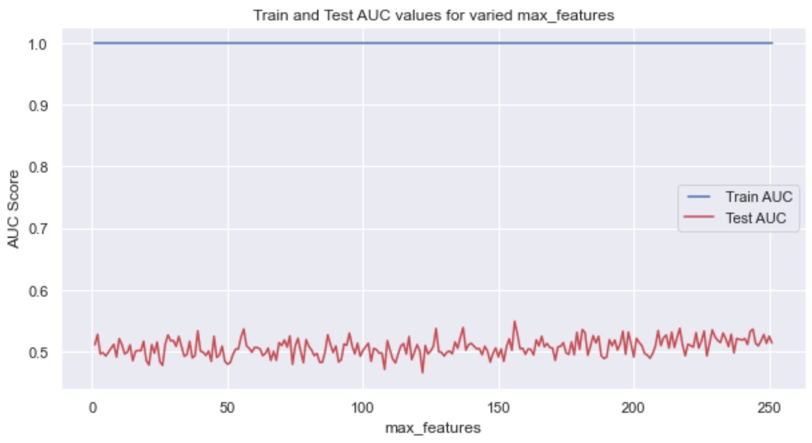
### 3.1.3 min_samples_leaf

This parameter sets the minimum number of samples that must be at each leaf node. This means that an internal node can only be split if each resulting leaf node contains at least this number of samples. This parameter can be either an integer greater than 0, or a float value.



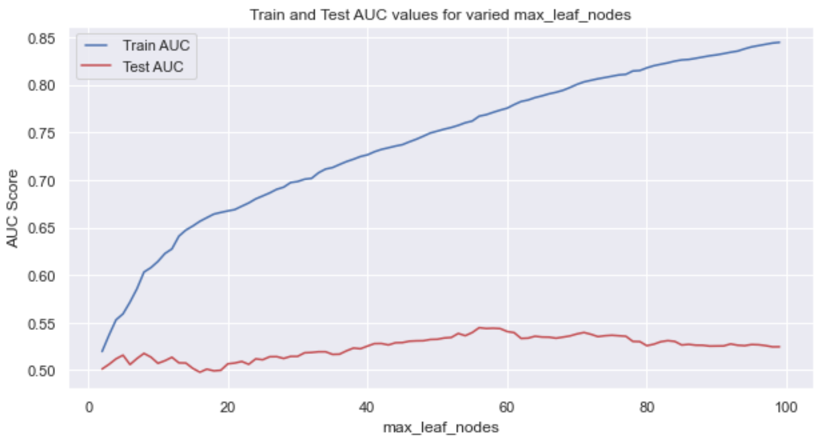
### 3.1.4 max_features

This parameter controls the number of features to consider at each node when looking for the best split. It can be an integer value, a float value, or one of three possible string values corresponding to an operation on the total number of features. If $max\_features = None$, then the value is set to the total number of features.



### 3.1.5 max_leaf_nodes

This parameter sets the maximum allowed number of leaf nodes.



## 3.2 Grid Search Cross Validation

Based on the above plots, the following set of parameter values will be searched over to find the optimal parameter configuration.

- max_depth: 10, 11, 12, 13, 14
- min_samples_split: 10 values between 0.10-0.15
- min_samples_leaf: 10 values between 0.05-0.10
- max_features: None, 'auto', 'sqrt', 'log2'
- max_leaf_nodes: 55, 56, 57, 58, 59

This grid of parameter values was searched for the best configuration using *roc_auc* as the scoring criteria. As mentioned previously, the plots suggested poor generalization to unseen data, so unsurprisingly even the optimal parameter configuration resulted in an *roc_auc* score of 0.495 on the test set. Now it needs to be determined if this poor performance is due to the tendency of decision trees to overfit, or if the feature set we have constructed is lacking any predictive power.

## 3.3 Feature Selection and Decision Tree Results

As a first step, the feature importance values of the tuned model can be examined. Using the ranked feature importance values to train a new decision tree on a reduced feature set might provide a clue to where the issue truly lies. This procedure resulted in an *roc_auc* score of 0.526 on the test set, a rather insignificant improvement.

The data will now be used to attempt to train a random forest classifier. If this also results in poor performance, it seems reasonable to suspect that the issue lies within the data set itself, and new features should be acquired in order to build a model with any predictive power.

# 4 Model Comparison and Results

## 4.1 Results

Repeating the above procedures again for a random forest classifier resulted in a test set *roc_auc* score of 0.527 on the full feature set, and 0.523 on the reduced feature set. This is essentially a worst case scenario.

## 4.2 Conclusion

The goal of this project was to build a predictive model for county-level changes in annual cardiovascular death rates, and then use the model to provide recommendations for preventing increases in this rate.

The first attempt to train a model with a decision tree failed due to overfitting, with minimal improvements seen even by reducing model complexity. A second attempt to train a model using an ensemble learner also failed. This suggests an inherent issue with the data itself, particularly that the features have little to no predictive power regarding changes in cardiovascular death rate.