**Normalization**

**Detection and removal of outliers**

**Dataset balancing by performing over sampling**

**Models:**

1. No parameters were changed. Only default parameters were used.

The training accuracy was 98.72%

Testing accuracy was 88.55%.

Even though the testing accuracy was good enough, there is more than a 10% difference between the training and testing accuracies. Training accuracy is much higher than testing accuracy, which means that the model is unable to generalize. This indicates that the model is overfitting.

1. For the second model, instead of using gini impurity as the splitting criterion, I used entropy. The results of this model were almost the same as the previous model. The testing accuracy only slightly increased and overfitting was still taking place.

Training accuracy: 98.72%

Testing accuracy: 88.64%

1. The third model that I created had more parameter changes.

Criterion: Entropy

Entropy focuses on information gain, that is, quantifying the amount of uncertainty related to data points. It thus creates purer leaves, which ensures that our data points are being classified as accurately as possible. Entropy also creates deeper trees which can catch more complex patterns.

The following parameters were used to reduce overfitting. Overfitting can be reduced by not letting the tree grow to its full depth.

Max\_depth: it tells the algorithm how deep the tree can go. When no restrictions were placed, the algo generated a tree of 62 levels and this tree was overfitting. To avoid overfitting, a value much less than 62 would work. After experimenting with different levels of trees ranging from 10 to 50, the ideal value was found to be 20 levels.

Min\_samples\_split: it’s the minimum number of samples required in a node to split it. If a node has lesser than min\_sample\_split number of samples, the algo will not split the node further, thus avoiding noisy splits. Large values for min\_samples\_split for a dataset as large as the one given to us will avoid noisy splits and improve generalization and reduce overfitting. After experimenting with various numbers ranging from 100 to 500, the minimum number of samples a node should have if it needs to split was found to be 117.

Min\_samples\_leaf: it’s the minimum number of samples required in a leaf node. Larger values for this parameter would allow the tree to have larger leaf nodes which makes the tree less sensitive to any noisy data and generalize better and thus reduce overfitting. Numbers ranging from 10 to 50 were randomly tested and the optimum value found was 10.

Training accuracy: 82.36%

Testing accuracy: 81.64%

Since the training and testing accuracies are very close to each other, the model is no longer overfitting.

The training and testing accuracies are also not bad.

This was the final model chosen for making predictions for the Kaggle test set.

**Kaggle score:** 0.68536

**Why is this a hard problem from machine learning point of view:**

* The difficulty with this problem is that the features are anonymous. We do not know what these features mean, so it’s difficult to determine which features are important to make an accurate prediction, and which features can be dropped. Having noisy or irrelevant features in the dataset can cause the ML model to overfit and decrease its performance in general.
* Since we do not know what the columns mean, the entire process becomes a black box. We do not truly understand how the ml algorithm came up with a prediction or how it decided a certain record should belong to a certain class. It is not explainable or transparent.
* Large number of features also increase computational complexity which can increase development time of the model.
* With many anonymous features, it is also very difficult to explore or visualize the data.