**HW4: Fraud Analytics**

**-Rijul Sherathia**

**Exploratory Table**

**Calendar

Description automatically generated**

**Calendar

Description automatically generated**

**Boxplot**

**Chart

Description automatically generated**

**Performance vs Complexity plot**

Random Forest

**Chart, line chart

Description automatically generated**

When the depth of the trees is increased, the trees become more complex and are able to capture more information from the data. However, this increased complexity leads to overfitting as the trees may become too specific to the training data and fail to generalize well to new data. Similarly, when the minimum number of samples per leaf is decreased, the trees are allowed to grow deeper and can become more specific to the training data. This also leads to overfitting, as the trees may become too tailored to the training data and fail to generalize well to new data. Overfitting can also lead to a decrease in out-of-time (OOT) error because the model has become so specialized to the training data that it can accurately predict the outcome of those specific instances. This leads to a false sense of confidence in the model's ability to generalize to new data. I have increased the depth from 7 to 27 and decreased the number of samples per leaf 20 to 1 to make the model overfit.

Decision TreeChart, line chart

Description automatically generated

The decision tree classifier starts to overfit when we increase the depth because increasing the depth of the tree allows it to create more complex and detailed decision boundaries, which can lead to a better fit of the training data. However, as the depth increases, the model can start to capture the noise and idiosyncrasies of the training data, which may not generalize well to unseen data. This means that the model may start to memorize the training data rather than learning the underlying patterns and relationships, leading to poor performance on new data. Therefore, a deeper tree can fit the training data more accurately but may result in poor generalization performance.

For Decision Tree, I have increased the depth from 7 to 40 to make the model overfit and ran 5 iterations with gradual increase in the depth to see the overfit trend. As a result, the gap between training and testing accuracy widens, and the out-of-time accuracy may decrease, indicating that the model is not able to generalize well to new data.

XG Boost

Chart, line chart

Description automatically generated

Overfitting in an XGBoost model can occur due to various reasons. Some of the parameters that I changed which lead to overfitting in an XGBoost model are:

* min\_child\_weight: Increasing min\_child\_weight can also lead to overfitting since it reduces the number of samples in the leaves, making the model more prone to overfitting.
* n\_estimators: The number of trees or rounds of boosting. Increasing the number of trees can lead to overfitting since the model becomes too complex and fits to noise in the training data.

After the point with n\_estimators around 50 and min\_child\_weight approximately 4 it started to overfit the model and lead to gap between the training and testing accuracy.

LGBM Classifier

Chart, line chart

Description automatically generated

Increasing the number of leaves in LGBM leads to overfitting because it allows the model to capture more complex and specific interactions between features in the training data, which may not generalize well to new, unseen data. Similarly, increasing the maximum depth of the trees in LGBM also leads to overfitting because it allows the model to capture more intricate relationships between features in the training data, which may also not generalize well to new, unseen data.

For LGBM model, I have increased the depth from 3 to 23 and also increase num of leaves from 40 to 200 and increased the n\_estimators from 80 to 200 in the loop of 10 iterations leading to overfitting of the model and increasing the gap between training and the testing accuracy.

Neural Network Classifier

Chart, line chart

Description automatically generated

When experimenting with a neural network model, we increased the size of the hidden layer gradually from 1 to 81, with a step size of 20. This led to an increase in model complexity. However, we observed that beyond a certain point (in this case, 41 hidden layers), the model began to overfit the training data, as evidenced by a widening gap between the training and testing accuracy. As the overfitting worsened, the out-of-time accuracy also decreased, indicating that the model was no longer able to generalize well to unseen data. These results suggest that increasing the size of the hidden layer beyond a certain point may not necessarily improve model performance, and could in fact lead to overfitting.