**CIS 508**

**Machine Learning in Business**

**In-Class Assignment 5**

**Done By:**

**Ishika Gupta – 1229311176**

This is a binary classification problem where we must predict whether a customer is going to churn or not. So, there are 2 unique classes: 0 and 1.

The dataset is an imbalanced dataset – which means that we do not have equal information about the 2 classes. We have significantly less information about class 1.

Thus, when a dataset is imbalanced, we cannot rely on accuracy as a performance measure. We must look at other metrics such as precision, recall and f1 score.

Before building these models, the following data preprocessing steps were performed:

1. One-Hot encoding to convert the categorical columns to numerical columns.
2. SMOTE Oversampling. This was used to balance the dataset.

The training dataset used to build these models are now all numeric values, and the number of records belonging to class 0 and class 1 are now equal.

All these models were created with default parameters.

MLP

A screenshot of a computer

Description automatically generated

For the MLP model, we have an accuracy of 54% only. It does not perform very well. It has a relatively low F1 score for both classes which means that either the precision or the recall is low in each of the classes. For class 0, there is high recall and low precision. This means that this model is not being picky about classifying records as class 0. It’s following a ‘better be safe than sorry’ approach. Its flagging as many records as class 0 as possible, even if they are actually not. It covers most of the actual class 0 records, but at the cost of a high false positive rate. For class 1, there is a high precision and low recall which means that the model is being picky about classifying records as class 1. It won’t classify a record as class 1 unless and until it has overwhelming evidence that the record belongs to class 1. Because of this, it is unable to identify most of the actual class 1 records. But, when it does classify a record as class 1, we should trust that the model is correct in classifying the record as class 1.

SVC

A number of numbers on a white background

Description automatically generated

SVC has a low accuracy as well. (only 64%). It has a higher accuracy than MLP. Its precision, recall, and f1 score interpretations are the same as MLP.

KNN

A number of numbers on a white background

Description automatically generated

KNN performed really well. It has a high accuracy (99%) and it has high f1 scores for both classes. This means that the precision and recall values for both classes are high. This means that this model is able to capture most of the records belonging to that class with a high precision.

Decision Tree – without hyperparameter tuning.

A screenshot of a computer program

Description automatically generated

A screenshot of a graph

Description automatically generated

Hyperparameter tuning decision tree:

It performed well. It has high accuracy and high f1 score values for both classes. It did not perform overfitting since the training and testing accuracy are both high, and there is no significant difference between them. This means that this model generalizes well. It created a tree with 19 leaf nodes. It has a depth of 7. Its not a huge tree.

A screenshot of a computer program

Description automatically generated

The best parameters given after performing hyperparameter tuning are:

A tree with 20 leaf nodes and a depth of 8. The gini criterion should be used for performing the best split.

A screenshot of a computer program

Description automatically generated

Even after performing hyperparameter tuning, the results are the same as the DT model without hyperparameter tuning. This could be because the ranges of values given for the parameters in hyperparameter tuning are close to what the ideal values are supposed to be.

Top 5 features according to hyperparameter tuned decision tree:

A screenshot of a computer

Description automatically generated

CUSTMOS, MOUPMO, TOTCHNG, INCOME, MOUCH3M are the top 5 features according to the hyperparameter tuned decision tree model.

Random forest stacked model:

A screenshot of a computer program

Description automatically generated

This model also performed well. This could be because the base algorithms used for stacking are all appropriate for this dataset and problem statement.

In individual models, KNN and DT gave the best performance.

There was no difference in the performance of the DT model with and without hyperparameter tuning.

The stacked random forest model gave a better performance than MLP and SVC. It gave the same performance as KNN and DT.

For MLP, and SVC, just the default model is not enough to get a good performance. We need to perform hyperparameter tuning to get good results.

In general, to improve the performance of models, we can perform normalization so that all columns have the same range and equal impact on making a prediction. We can also detect and remove outliers to get more accurate or representative results.