Data Preprocessing – Feature Selection

Features that were used were:

A screenshot of a computer

Description automatically generated

One-Hot Encoded Variables:

"BorrowerState", "BusinessAgeDescription", "BusinessType", "Industry", "Veteran", "Term\_cat"

Label Encoded Variables:

"ProcessingMethod", "LoanStatus", "HubzoneIndicator", "LMIIndicator", "RuralUrbanIndicator"

Decision Tree Base Model Evaluation

Using Normal Decision Tree:

Classification Report and Confusion Matrix:

A screenshot of a computer

Description automatically generated A blue square with white text

Description automatically generated

Hyperparameter Tuning to obtain the best parameters for Decision Tree with the use of Optuna:

After 200 trials:

A screenshot of a computer

Description automatically generated A blue square with white text

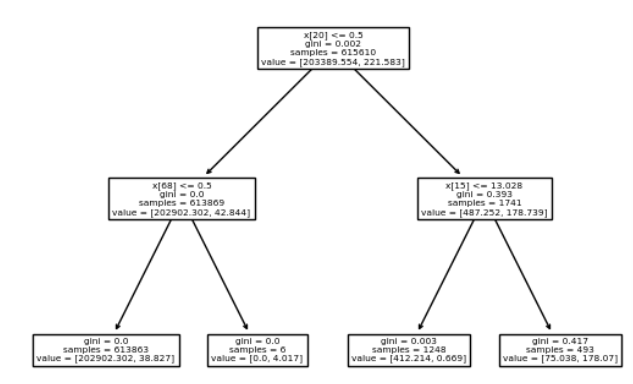
Description automatically generated

Recall Score significantly improved from 0.465 to 0.806

Class Weight (Class 0 : Class 1) as a ratio: 0.3305647591315227

Max Depth: 2

The Gini Index has been calculated by splitting of features and the following features were found to have larger Gini Index.



Printing out the list of features to see which features are being split:A screenshot of a computer screen

Description automatically generated

From the features, it is noticed that the tree takes feature 20 (payroll\_over\_other) as the parent node and the subnodes include feature 68 (BorrowerState\_TX) and feature 15 (log\_CurrentApprovalAmount).

Finding the best method to address imbalanced data through trial and error:

Oversampling with ADASYN:

Use of ADASYN: ADASYN will focus on the samples which are difficult to classify with a nearest-neighbours rule while regular [SMOTE](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html#imblearn.over_sampling.SMOTE) will not make any distinction

(<https://imbalanced-learn.org/stable/auto_examples/over-sampling/plot_comparison_over_sampling.html>)

A screenshot of a computer

Description automatically generated A blue and white squares with black text

Description automatically generated

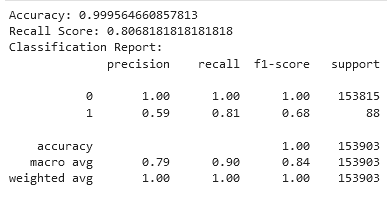
The recall is very high of 0.93, but the precision of model is very bad as it may predict non fraudulent cases as fraudulent which may prompt unnecessary investigations.

Undersampling with TomekLinks:

## Use of TomekLinks: This technique is the modified version of CNN (Condensed Nearest Neighbor

) in which the redundant examples get selected randomly for deletion from the majority class. (reason why TomekLinks is used because there is a shorter runtime compared to CNN)

(<https://www.analyticsvidhya.com/blog/2022/05/handling-imbalanced-data-with-imbalance-learn-in-python/>)

 A blue square with white text

Description automatically generated

It appears that undersampling with TomekLinks made no difference to the dataset with the same recall score as that of decision tree model. However, for consistency, we will continue to use the TomekLink Undersampling method for AdaBoost

AdaBoost

Although AdaBoost may result in overfitting of data, we have random forest model as a bagging methodology that Song Xi will provide to compare differences.

A screenshot of a computer

Description automatically generated A blue square with white text

Description automatically generated

It appears that boosting has resulted in a significant drop in recall and slight drop in precision.

After 200 trials of hyperparameter tuning with Optuna:

There is a slight improvement to the results with the following considered as the best hyperparameters:

'n\_estimators': 158

'learning\_rate': 0.9926477871357113

A screenshot of a computer

Description automatically generated A blue square with white text

Description automatically generated