INSTITUTE OF BUSINESS ADMINISTRATION

IBA SMCS

School of Mathematics and Computer Science

CSE 602: MACHINE LEARNING-I

SEMESTER PROJECT

CLASS IMBALANCE SOLUTIONS EFFECT ON ML PERFORMANCE

GROUP MEMBERS

AYESHA NOOR KHAN	(ERP: 29460)
MARYAM KHAN	(ERP: 08635)

SUBMISSION TO: DR TARIQ MEHMOOD

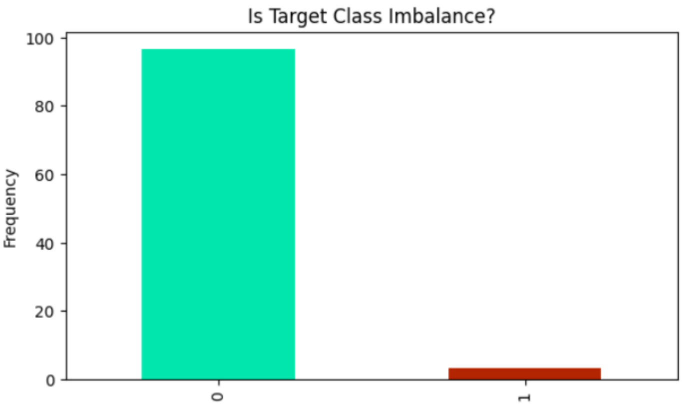
SUBMISSION DATE: 30th MAY 2024

GitHub Repository Link: https://github.com/ayeshank/Machine-Learning-I

Introduction:

This project aims to highlight the impact of Class Imbalance (CI) solution techniques on the machine learning models performance. We will start our demonstration by first explaining the CI techniques (which we have selected for our project) individually as to have a proper idea about how they work and contribute to achieve better results for our ML models.

A brief demonstration of what class imbalance in datasets is shown below:



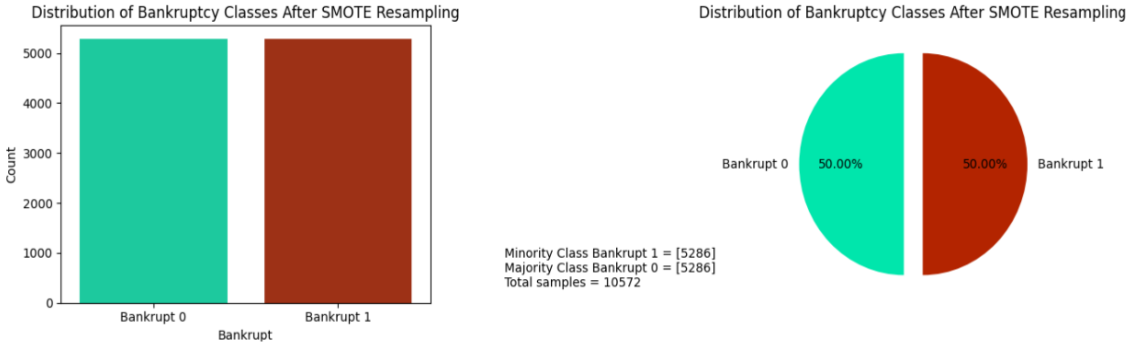
Yes, Dataset Have Imbalanced Class Distribution.

All the datasets have examples consist of majority class target (1) and very few have minority class sample (0). IN this case, the model we executed in baseline is always likely to be overfitted on majority class. We have demonstrated below some CI techniques to overcome this issue in datasets

Class Imbalance Solution Techniques:

1. **SMOTE**:

   A resampling method, SMOTE, stands for **Synthetic Minority Oversampling Technique,** is a class imbalance resolving technique in which we oversample along with synthesizing the examples from the minority class. SMOTE works by simply selecting the examples which are close in the feature space and then drawing a line between the examples in feature space and drawing a new generate sample at a point along that line.
   It selects the examples which are close in feature space with the help of K nearest neighbors of a selected example, hence creates new data points somewhere between them which increase the number of minority class examples and make the dataset balance with majority class.
   For Example:

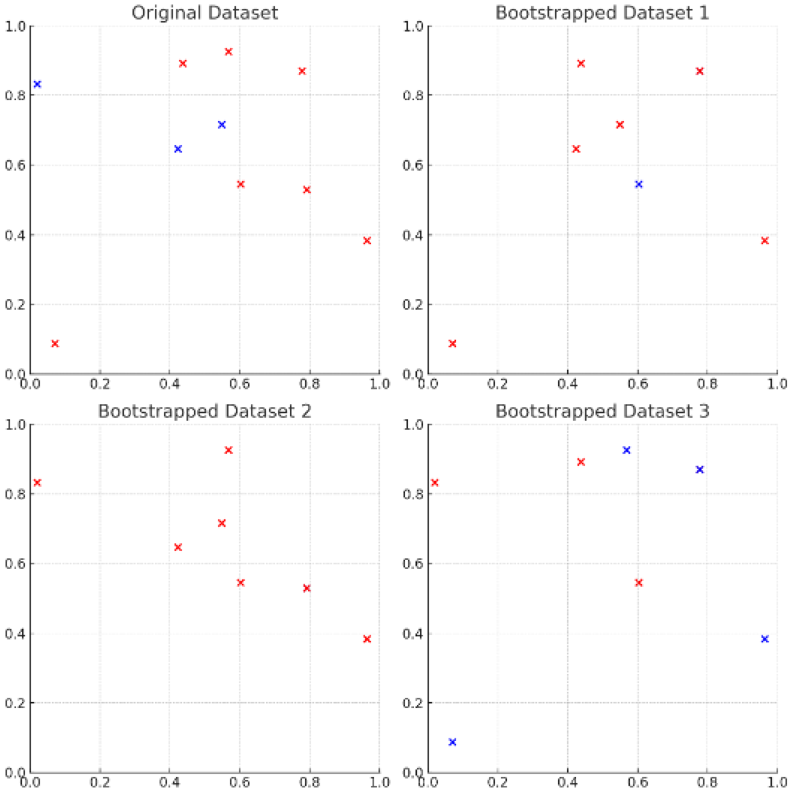


```
Minority Class Bankrupt 1 = [5286]
Majority Class Bankrupt 0 = [5286]
Total samples = 10572
```

2. **BAGGING**:

Bagging is an ensemble method, also known as Bootstrap Aggregating, which works by selecting samples from the original dataset (with replacement) and generate multiple new datasets examples. Note that here sampling with replacement means that some samples may appear more than once which other may not appear in a new subset of original dataset.
After the subset sampling, it trains a model (same model on each subset) on each of the new subsets of original datasets, which is called as base estimator.
After all the models are trained separately on each subset of dataset, it then aggregates their result depending upon nature of dataset, means for regression it do averaging and for classification it do majority voting.
Then we obtain final testing performance metrics for our model original datasets.

Below is the illustration of how bagging creates multiple bootstrapped subset of original imbalanced dataset:



3. **BOOSTING**:

Boosting is another ensemble method in which we train a model on complete original dataset and determine errors. Since we know some part of dataset is not trained well due to precision and recall accuracy imbalance therefore, we increase the weights of wrongly classified instances so that they can be corrected efferently. We again train the model on weighted new data and repeat the process until we get the desired performance metrics. The result will be the aggregation of all the predictions of models by weight.

4. **ONE CLASS LEARNING:**

   One class learning simply means training the model on examples of only one class usually the majority one and the other class is treated as outlier or anomaly. One-Class SVM (Support Vector Machine) is the technique which implements one class learning. Just like the name indicates, One-Class SVM is the type of SVM used for anomaly detection by fitting a boundary around a single class of data points. This define a region that encompasses the normal data, whereas the points which do not lie in this boundary are treated as anomalies or outliers.

Impact Of Each CI Solution on The Classification Performance:

Compared with a imbalanced baseline model, there is a huge impact of precision and recall change in the balanced model as shown below for each CI solution.

MODEL NAME WITH CI	PERFORMANCE METRIC
Baseline Model	All of the baseline models have shown very high accuracy, and even high precision (identified true positive correctly) but their recall (correctly identifying true positive) is very low. It simply indicated that the models got overfitted a lot on the majority class and can't be able to predict minority class correctly. This is the issue which we have to resolve with CI solutions.
SMOTE Model	While using SMOTE resampling method, the accuracy of the model decreases a little bit which is good for our case, but it maintains a good average between precision and recall unlike baseline models. It makes the model more generalized for the testing data and correctly predicted the cases false which are true negative and positive which are true positive, and thus reduces false negatives and false positives.
Bagging	In general, the bagging classifier performed very well with a good balance between all of the performance metrics and makes the model more generalized for testing data. The best advantage is that it is a less complex model as compared to SMOTE models with less parameters as per the rule Occam's Razor.

Boosting	The performance of Boosting is quite similar to Bagging in all of the cases, and it maintains a good balance between precision recall with best auc-roc curve, thus performed really well of imbalanced datasets as compared to baseline models.
One-Class Learning	One-Class Learning with One-Class SVM proves to be a quite complex model it more likely to be overfitted in testing dataset because its accuracy is same as the baseline model, but it somehow decreases the precision and recall of identifying the anomalies points.
Logistic Regression	When executing baseline model with Logistic Regression the accuracy is very high, but precision, recall and auc-roc curve is very low meaning the model just neglected the minority class and overfitted on majority class on testing dataset. After SMOTE, there is a significant drop in accuracy and recall but very high precision. Improved ROC AUC shows better handling of the minority class but at the cost of overall performance. By applying class weight, reduced accuracy but high precision and recall with improved ROC AUC, indicating better balance without major performance loss
XGBoost	Baseline execution of model results in high accuracy, precision, recall, and F1-score with low ROC AUC. After resampling, it reduced accuracy and recall but high precision and improved ROC AUC, indicating a better balance between classes.

Performance Of A CI Solution Get Impacted Significantly by The Choice of Different Algorithms

Yes, when we are using different algorithms, they have performed differently according to their nature and hyperparameter settings.

Decision Trees and Random Forest:

Both the models performed quite well as compared to other models whether in baseline execution or after resampling. They have the tendency to be less likely to be overfitted by the class imbalance issue. In **Bagging**, Random Forest has performed quite well as compared to the decision tree, because decision tree is more complex algorithm as compared to Random Forest which gives best performance metrics in less complex model.

KNN:

K nearest neighbor algorithm performed well than baseline KNN algorithm after doing SMOTE resampling but it’s not much efficient as compared to Decision Trees and Random Forest, because its seems good for datasets which have more grouping, but not performed much well with average f1-score and auc-roc curve almost low.

SVM:

Baseline SVM got a high accuracy and very high precision as compared to recall in majority of datasets where SMOTE SVM performed quite well in terms of performance metrics and identifying TN and TP cases correctly. On the other hand, one class learning method using One-Class SVM is more complex to handle rather than SMOTE SVM which makes the model more difficult to generalize and highly likely to be overfitted. IN short SMOTE SVM quite perform well all over the SVM cases.

Gaussian Naïve Bayes:

Imbalanced or baseline execution of GN Bayes gives almost a random predicting with approximate 50-50 precision, concluding its highly unsuitable for imbalanced datasets. With SMOTE GN Bayes it precision and recall increases a little bit but not much efficient as compare to other models used in the case of class imbalance.

Logistic Regression:

Logistic Regression works well with class weighting with less computational power which may require in SMOTE and other models. Performance of the model significantly decrease while using SMOTE may because of overfitted due to sampling of data or no general trends in data distribution.

XGBoost:

It compromises on the accuracy of dataset but as compared to other model but other performance metrics perform quite well.

Conclusion:

In general, the tree-based algos like RF, DT performed well in case of class imbalance algos even with hyper tuning as compared to linear model like logistic regression, SVM.