Combination of Machine Learning Techniques and Oversampling Ratios for Credit Card Fraud Detection: A Comparative Analysis

Zami Nizam Uddin

Student ID: 221057490 **Submission Date: 25/08/2023**

MSc Computing and Information system final project

Queen Mary university of London, 327 Mile End Rd, Bethnal Green, London E1 4NS

Abstract:

This research paper enters upon a comparative analysis of distinct machine learning algorithms: Logistic regression, Random Forest, Gradient Boosting, Gaussian Naïve Bayes and K-Nearest Neighbors (k-NN) for credit card fraud detection. After preprocessing a dataset of, each algorithm was trained and evaluated for both training and test sets, to assess for potential overfitting. A major aspect of this study is the use of "Synthetic Minority Over-Sampling Technique" or SMOTE with varying oversampling ratios (1.0, 0.7, 0.5, 0.3, 0.1) to address the class imbalance between fraudulent and non-fraudulent samples, which is inherent in a fraud detection dataset. Each algorithm's performance was thoroughly validated using 5-fold crass validation technique for each oversampling ratio. The research further enhances understanding of the models through visual integration of confusion matrices, Precision-recall curves and accuracy plots for each algorithm as well as make prediction on unseen data. This helps provide insights into real-world applicability. The goal of this research is to guide the choice of the most effective machine learning algorithm for credit card fraud detection, clarifying a theoretical understanding of primary application in digital financial security.

Table of contents

Page

1. Introduction

1.1. Background:

With the rapid growth of digital transaction worldwide, the integrity and security of financial data has become ever so crucial. Use of credit cards are on a rise as they introduce incredible convenience and benefits, however, comes with possible threats of fraud. Therefore, introduces an imperative need to explore advanced methodologies that are both adaptive and efficient.

Machine learning is a subset of artificial intelligence, it offers promising solutions to such problems. Leveraging different machine learning models to learn patterns in vast data, machine learning models cam potentially identify unusual and potentially fraudulent activities with higher accuracy than traditional systems like Blacklisting, Manual verification, velocity checks etc. which have been foundational in detecting fraud but with the evolving nature of fraud, the system needs to be a more adaptive and sophisticated solution like machine learning. However, every solution comes with its own challenges. One of the main and significant challenge in terms of credit card fraud detection is imbalance of fraudulent and non-fraudulent classes in a dataset as fraudulent cases

usually tend to occur less frequently in comparison to genuine ones. Such imbalances can cause algorithms to perform well in terms of accuracy but perform poorly in recall causing it to likely, miss most fraudulent transaction.

This research dives into this problem by comparing various machine learning techniques in their ability to detect fraudulent transactions. This can be improved by balancing the data for which the use of various Synthetic Minority Over-Sampling Technique (SMOTE) ratios proved to be crucial. Further, use of cross validation for each algorithm on each over sampling ration helps highlight the best combination of oversampling ratio and machine learning algorithm that offers the best accuracy for detecting fraudulent transaction.

In further sections of this research, I will be developing the complexities regarding machine learning models and their significance in their detection accuracy and the importance of handling imbalanced data. By the end I hope to establish a clear understanding of the most potent among the selected algorithms to tackle this problem.

1.2. Objective:

Primary objective of this research is to conduct a thorough, comprehensive and comparative analysis of multiple machine learning techniques naming: Logistic Regression, Random Forest, Gradient Boosting, Gaussian Naïve Bayes and K Nearest Neighbors (K-NN). For this research I am using a simulated dataset I found on "kaggle.com", the aim is to probe into the effects of different SMOTE over sampling ratios on the performance of the algorithms. Through this research my goal is to determine the optimal machine learning technique and over sampling ratio that yields the highest predictive accuracy, precision and recall. Through this investigative research the aim is to offer a dependable and efficient model recommendation for real world application in terms of credit card fraud detection.

1.3. Scope:

The extent of this research covers:

- Dataset preprocessing: For this research I'm using a simulated dataset that represents typical credit card transactions. The dataset being simulated ensures confidentiality while replicating a real-world transactional pattern.
- Analyzed Machine learning Techniques: For this research I selected 5 machine learning models to examine. They are:
    - Logistic Regression
    - Random Forest
    - Gradient Boosting
    - Gaussian Naive Bayes
    - K-Nearest Neighbors (K-NN)
- handling data Imbalance: In this research I will probe into the challenges posed by severely unbalanced datasets, which tends to be a common issue in fraud detection. I am using a technique called "Synthetic Minority Oversampling Technique" in short SMOTE to address the issue. The analysis is done on synthetic oversampling ratios of 1.0, 0.7, 0.5, 0.3, 0.1.
- Evaluation Metrics: Each model is evaluated using metrics such as Precision-Recall, f1-Score, accuracy and confusion matrix. Further each model is evaluated on both training and test sets to detect potential overfitting.
- Visualizations: Outliers, imbalance and evaluation metrics are visualized to give a clearer picture of the models
- Model Validation: To ensure dependability of the algorithms, each algorithm is cross-validated (5-folds) on each oversampling ratio.
- Predictive Capability: Beyond the analysis the models are also used to predict on new unseen data in order to showcase potential real worlds application.

This research does not extend to other machine learning models apart from the ones mentioned previously, datasets other than simulated one used, oversampling technique beyond SMOTE and real-world deployment or integration of the model into a transaction system.

The aim here is to provide a thorough comparison of the selected algorithm's performances in context of credit card fraud detection and on the efficacy of different SMOTE oversampling ratios. The scope ensures a focused exploration of the topic with the potential to lead to actionable insights and recommendations.

2. Literature Review

2.1. Credit Card Fraud Detection:

Credit card fraud detection has always been a pressing challenge in the modern banking and financial sector. Because of the fast digitization of the financial world, opportunities for malicious activities have also increased. Traditional credit card fraud detection system depended on rule-based systems which uses predefined rules based on known fraudulent patterns to flag potential or suspicious transactions. For example, a Cumulative transaction system; where a rule can dictate if a transaction for a particular account exceeds a certain amount within a certain period it would trigger an alert to catch potential fraudulent transaction. But in such a scenario a genuine transaction meeting the same points most likely will also trigger an alert which will in turn can cause problems for a genuine user of a transaction service. However, these systems, while adept at detecting known fraud types, often faltered when encountering novel fraudulent tactics (Sahin, Bulkin, & Duman, 2013)[1].

Realizing the limitations of rule-based systems, researchers are studying more dynamic approaches like machine learning and data driven methods. These techniques can detect patterns within massive datasets, this enhancing the detection's accuracy (Bhattacharyya and others., 2011)[2].

The aspect of data imbalance is a prevailing issue for the literature as it is inherent in a dataset for fraud detection as the amount of fraudulent transaction tends to be much lower than genuine ones. This is a problem because it can introduce potential bias in a model's predictive capability. To deal with such issues researchers have developed various oversampling techniques, one of which is SMOTE, which is represented during the fitting and training of the data (Chawlaand others., 2002)[3].

In summary the field of credit card fraud detection has transformed significantly and for better from static rule-based systems to more dynamic and sophisticated approaches like machine learning. However, the issue with handling data imbalance and real time detection remains areas of active research.

2.2 Machine Learning Techniques for Fraud Detection in this research

1. Logistic regression: It is a statistical method widely utilized for binary classification problems, logistic regression has been applied in the realm of fraud detection to distinguish between fraudulent and non-fraudulent transactions. Despite its simplicity, logistic regression is highly interpretable and provides probabilities that can be easily thresholder to make classifications. Researchers like Sahin and others (2013)[1] have demonstrated the applicability of logistic regression in credit card fraud detection, showcasing its potential when dealing with high dimensional datasets

2. Random Forest: It is An ensemble learning method, the Random Forest algorithm has gained prominence for its ability to handle large data sets with higher dimensionality. By constructing a multitude of decision trees during training and producing the mode of the classes (classification) of the individual trees for predictions, Random Forest can achieve high accuracy rates. Carcillo and others (2018)[4] revealed that Random Forest, due to its inherent ability to manage imbalance in data, was particularly effective in fraud detection scenarios.

3. Gradient Boosting: Gradient boosting trains models sequentially with each one correcting the errors of its predecessor. This mechanism of transforming weak learners into stronger ones makes gradient boosting techniques particularly potent for various fraud detection tasks.

4. Gaussian Naïve Bayes: This algorithm is fundamentally rooted in Bayes theorem, has been adopted for fraud detection for its efficiency and speed. In the Gaussian Naive Bayes variant, it's assumed that continuous features associated with each class are distributed according to a Gaussian or normal distribution. Its probabilistic approach allows for easy interpretation of results, with Zhang (2004)[6] noting its effectiveness in classification tasks, even when the independence assumption is violated. In the context of credit card fraud detection, its ability to quickly adapt to changing data makes it a valuable tool in the arsenal of machine learning techniques.

5. K-Nearest Neighbors: This algorithm works by measuring the distance between input samples and deciding based on the majority class of its 'k' nearest neighbors, K-NN can be particularly effective when data distributions are unknown. A comparative study by Ahmed and others (2016)[5]

highlighted the efficacy of K-NN in detecting credit card fraud, especially when combined with feature engineering techniques

2.3 Handling Imbalance Data with SMOTE

2.3.1 The Challenge posed by data Imbalance:

In many real world classification problems especially in cases of credit card fraud detection, data-sets often tend to be severely imbalanced. In this case this is due to the under representation of the fraudulent transaction classes compared to the legitimate transaction class, as fraudulent transactions tend to be much less than legitimate ones. This inherent asymmetry can lead machine learning algorithms to develop bias towards the majority class and over look the minority class, which in case of fraud detection is the most important class for accurate prediction.

2.3.2 Synthetic Minority Over-sampling Technique (SMOTE):

To address the issue of data imbalance various techniques of resampling the dataset with synthetic data have been developed. SMOTE is one of them. It works by generating synthetic examples in feature space, that is the algorithm selects a few similar instances from the minority class using a distance measure, then modifying an instance, one attribute at a time by a random amount within the difference to the neighboring instances. This process creates synthetic instances that are not mere copies of existing data but rather, blends the features of existing instances.

2.3.3 Significance of SMOTE in this research:

The importance of SMOTE is in it's ability to mitigate the risk of overfitting which is a common occurrence when oversampling minority classes by replication. By generating synthetic data, SMOTE ensures that the model is exposed to a more diverse set of examples during training, promoting a better generalization during testing. However SMOTE can also lead to potential overfitting which is where testing different oversampling ratios become crucial as we will see in this research as this research explores different SMOTE oversampling ratios to identify a more stable fraud.

3. Methodology

3.1 Data Collection:

The dataset used in this research is a simulated set of credit card transactions, spanning a period from 1st January 2019 to 31st December 2020. This simulated data captures the transactional behaviours of 1,000 credit card holders engaging with a pool of 800 merchants. The dataset source can be found at [7]. Although the source provides a separate training and test set for this project I used only the test set to both train and test my models.

Acknowledgements

The invaluable contributions of Brandon Harris warrant special mention, particularly for his stellar efforts in conceptualizing and crafting the Sparkov Data Generation tool. And Kartik Shenoy for his dataset generated using Brandon Harris's tool.

3.2 Data Preprocessing

The dataset for this study was sourced from "fraudTest.csv". After initial exploration, the dataset consisted of multiple features, with the first few rows and last few rows examined for a general overview. The statistical summary showed key statistics of numerical attributes, and performed a check for missing values which returned no missing values were identified.

The "trans_date_trans_time" column was converted into a pandas date-time object, allowing for the extraction of additional time-related features. Certain columns like 'merchant', 'category', 'gender', and 'job' were transformed from categorical to numerical using Label Encoding to ensure compatibility with machine learning models.

For outlier detection, the Isolation Forest algorithm was used, visualized on a scatter plot. This method detects outliers by identifying points that are isolated rapidly when building trees.

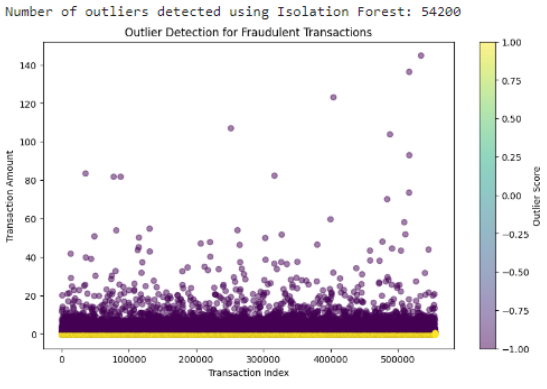

Fig:1.1. Dataset Outliers & Visualisation of Outliers

A significant class imbalance was identified, where fraudulent transactions constituted a tiny percentage of the entire dataset. This imbalance can lead to less than ideal classifier performance,

as the classifier can very easily become biased towards predicting the majority class given the scale of the imbalance.

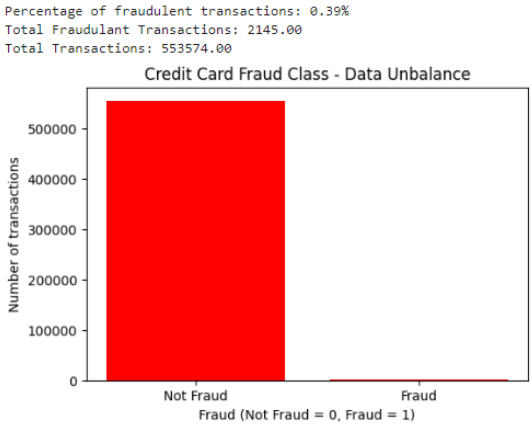

Fig:1.2. Data Imbalance & Visualisation

To counteract this SMOTE was applied on the dataset.
Finally, the pre-processed dataset was split into training and test sets with allocation of 80% and 20% to each sets respectively, to ensure a good balance for both training and evaluating the model.

3.4. Model Development

This research aims to compare the performance of some various learning algorithms. Here's a breakdown of each algorithms used:

1. Logistic regression: The first algorithm used is logistic regression with default parameters, from Sklearn's Logistic Regression model.

2. Random Forest: This algorithm was used with the following hyper parameter tuning:
Number of estimators = 150
Maximum depth of trees = 10
Minimum number of samples required to split an internal node: 100
Random state: 42 for reproducibility.
This was achieved with the help of documentation from sklearn, Random Forest Classifier.

3. Gradient Boosting: It is another ensemble method used in this research with the following parameter:
Number of estimators: 150
Learning rate: 0.05
Maximum depth of the tree: 2
Subsample: 0.8
Random state: 42 for reproducibility
This was achieved with the help of documentation from sklearn Gradient Boosting Classifier.

4. Gaussian Naive Bayes: Gaussian Naive Bayes classifier from sklearn Gaussian Naive Bayes model with default parameters.

5. K-Nearest Neighbors: K-NN was also used with default parameters from sklearn K-Nearest Neighbors Classifier model. The optimal number of 'K' is determined by evaluating the algorithm's performance for "k" values ranging from 1 to 20. For each "k" a knn classifier is trained on the training data and makes predictions. The error rate for each "k" is calculated as the fraction of incorrect predictions and stored in a list. After iterating over all "K" values, the optimal "k" corresponding to the lowest error rate is identified and printed.

3.5. Evaluation Metrics

3.5.1. Classification report:

Scikit leanr's metrics "classification_report" is a utility function that prints out an algorithms precision, recall, f1-score and accuracy. Which are further visualized through plots using "matplotlib" module's "pyplot" function. Further and ROC curve is also plotted for each algorithm to visualize evaluation of each algorithm's performance. As shown in Fig:2.1

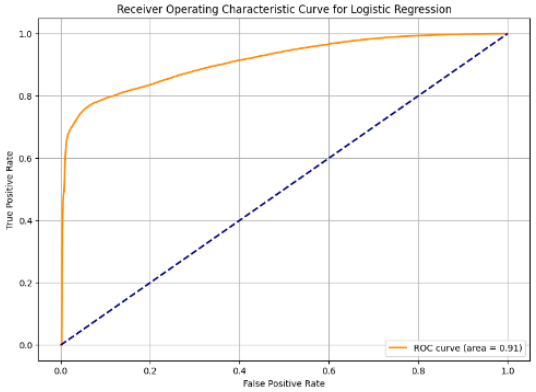

Fig:2.1. ROC Plot for LogReg at os rate .7

3.5.2. Cross validation:

To establish the model's generalization capability, 5-fold cross-validation was applied. This involved partitioning the training dataset into five subsets, iteratively training the model on four and validating on the fifth. The mean accuracy score from these five runs was then computed. This was repeated on each algorithm on each of the applied SMOTE Ratios.

3.5.3. Confusion matrix:
This matrix provides a visualization of the model's prediction capabilities by showcasing true positives, false positives, true negatives, and false negatives. This is also repeated for each algorithm and visualised through plotting a heatmap figure. Shown in Fig:2.2

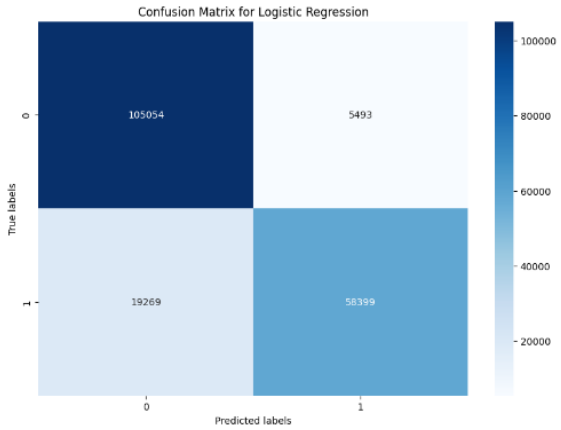

Fig:2.2. Confusion Matrix for LogReg

3.5.4 Predictions on New Data: Each of the trained models was also utilized to predict new data samples, mapping them to their respective classes.

The application of these metrics provided a comprehensive insight into the performance of each model, facilitating a robust comparison.

4. Experiments and Results

All classification scores on test and training data and cross validation scores on test data are recorded in a ".xlsx" file named "ML algorithm - Evaluation and Cross val Scores at different oversampling ratios.xlsx". The file will be uploaded with supporting documents and a link to github repo containing the fill will be provided in references [7].

4.1 Oversampling with SMOTE:
For this research each algorithms have been applied to the model, tested and cross validated at various SMOTE oversampling ratios 1.0, 0.7, 0.5, 0.3, 0.1. To test for the best ratio at which each algorithm provides the best results. Further all algorithms were tested on the training data to look for potential overfitting and all results suggested toward having no overfitting.

1. Logistic regression: For this research we prioritise the F1 score, as for fraud detection of the minority class which are fraudulent transaction. In-case of Logistic regression from all the results recorded. Oversampling ratio of 0.1 gives us a higher accuracy, ratio 1 returns the best F1-Score for the minority class and the rest of the ratios provide a more balance F1-Score across both classes.

2. Radnom Forest: The highest test accuracy is achieved at an oversampling ratio of 0.1 with an accuracy of 0.98, which is also consistent with the highest cross-validation mean score of 0.98.

3. Gradient Boosting: Based on both the test metrics and the cross-validation scores, an oversampling ratio of 0.1 seems to offer the highest performance but 0.3 returns a more realistic over all better score for the Gradient Boosting model for this project. It achieves the highest accuracy, a competitive macro average F1-Score, and the highest mean cross-validation score. However, Its computation time, due to its nature, increased significantly at higher oversampling ratios, making it less desirable for very large datasets.

4. Gaussian Naive Bayes: While the oversampling ratio of 0.1 provides the highest accuracy and cross-validation scores, the significant dip in recall for the minority class at this ratio is a concern. Considering the balance between accuracy and the importance of correctly detecting frauds, the oversampling ratio of 0.3 seems to provide a good trade-off. The accuracy and cross-validation scores are still relatively high, and the recall for the minority class is significantly better. However, it didn't surpass the performance metrics of tree-based models like Random Forest and Gradient Boosting.

5. K-Nearest Neighbours: Based on the test scores and cross-validation scores, the best oversampling ratio for the K-NN model for this project is 1.0. This ratio not only provides the highest accuracy and F1-score values for the minority class on the test dataset but also yields the highest mean CV score, indicating stable and consistent performance across different subsets of the dataset.

4.2. Model Comparison:
In this section, a comparatively analysis of the performance of the five distinct machine learning models is done.
Based on the weighted F1-scores, K-Nearest Neighbours (K-NN) consistently performs the best across all oversampling ratios, achieving a score of 1.0. However, a perfect score is suspect in real-

world scenarios. The Random Forest classifier also performs admirably, especially at the lower oversampling ratios of 0.1 and 0.3. If we go strictly with the results K-NN does seem to be the best choice for fraud detection for this research. However, K-NN returns almost 100% accuracy which suggests possible overfitting and K-NN is known to be very prone to overfitting. So we can disregard K-NN in this comparison.

Between the rest of the algorithms, Random Forest appears to perform the best over all the other algorithms specifically at ratios 0.1 and 0.3. Given the same principal of potential overfitting I would nominate 0.3 to be the best choice for Random Forest.

Random forest Training Data Performance:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	442798
1	0.97	0.88	0.93	132918
accuracy			0.97	575716
macro avg	0.97	0.94	0.95	575716
weighted avg	0.97	0.97	0.97	575716

Classification Report for Random Forest:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	110776
1	0.97	0.88	0.92	33154
accuracy			0.97	143930
macro avg	0.97	0.94	0.95	143930
weighted avg	0.97	0.97	0.97	143930

Fig:3.1.Random Forest Score at 0.3 1

Behind Random Forest should come Gradient boosting at ratio 0.3 followed by logistic regression at ratio 1 and at last Gaussian Naive Bayes at ratio 0.1.

5. Discussion

5.1 Key Findings

Performance Variability: Different oversampling strategies displayed varied performances across machine learning algorithms. Particularly, Random Forest demonstrated robust results across different SMOTE oversampling ratios. However, certain algorithms such as K-NN displayed sensitivity to the oversampling ratios.

Optimal Oversampling: An oversampling ratio of 0.1 and 0.3 consistently yielded promising results across most algorithms, balancing the trade-off between improved performance on the minority class and potential overfitting.

Precision-Recall and Accuracy Trade-offs: Precision-recall curves revealed that while accuracy was high for most models, certain algorithms, especially at lower oversampling ratios, faced challenges in distinguishing between false positives and true positives.

5.2 Implications

Real-world Application: The insights gained from this study can be applied by financial institutions and credit card companies to refine their fraud detection systems. Implementing a machine learning approach instead of the traditional ones, especially with the recommended algorithms and oversampling ratios, can potentially enhance the accuracy and precision of fraud detection mechanisms.

Cost Savings: Early and accurate detection of fraudulent transactions, as demonstrated by the high-performing models in this study, can result in significant savings for institutions by reducing losses related to undetected fraud. If they prove to function as intended

Consumer Trust: An effective fraud detection system enhances consumer trust. As false positives are reduced (avoiding flagging legitimate transactions as fraudulent), users face fewer transactional interruptions, thus improving their overall experience. For this research, the results have not yet reached a point where we can unambiguously assure consumer trust. However, it can provide a foundational basis upon which further advancements can be built.

5.3 Challenges and Limitations

Overfitting Concerns: While SMOTE helps in balancing the dataset, there's a risk of overfitting, especially at higher oversampling ratios. This is because SMOTE generates synthetic samples, which may lead the model to draw overly specific boundaries that don't generalize well to unseen data.

Simulated Data Limitations: The dataset used was simulated, which might not capture the true complexities and hidden patterns of real-world transactional data. While simulated data allows for a controlled environment to test algorithms, the findings need validation on actual transaction data for real-world applicability.

Model Complexity: Some models, especially Random Forest and Gradient Boosting, can become computationally intensive with increasing data volume. As a result, real-time detection might face challenges, and optimization or model simplification might be help with for deployment.

Non-Consideration of Temporal Patterns: This study treated transactions as independent data points. However, in the real world, considering the sequence of transactions might add valuable context, potentially improving the accuracy of fraud detection.

6. Conclusion and Future Work

6.1 Conclusion:

In the evolving world of credit card fraud detection, machine learning techniques have emerged as powerful tools to detect and counteract fraudulent activities. This research undertook a comprehensive comparison of various machine learning techniques, specifically logistic regression, random forest, gradient boosting, Gaussian naive Bayes, and k-NN, evaluated under different SMOTE oversampling strategies.

The experiment's comprehensive nature revealed that while all models demonstrated potential in fraud detection, there were nuanced differences in their performance across varying oversampling ratios. The importance of addressing data imbalance was evident, and the efficacy of the SMOTE technique in enhancing model performance was confirmed. By comparing the algorithms in this structured manner, deeper understanding of their behaviour in the specific context of credit card fraud detection was established.

6.2. Future work:

This research dives into the potential of a credit card fraud detection mechanism exclusively utilizing SMOTE for oversampling. Notwithstanding the results obtained, numerous alternative oversampling methodologies might offer better outcomes and deeper insights. It is my aspiration to investigate these techniques in subsequent research endeavours.

Further I would like to utilise both separate training and test datasets. Since due to the size of the dataset I worked with only the test dataset for both training and testing my models. Which in itself took very long time for models like Random Forest and Gradient Boosting, taking up to 7 minutes to run a single test on higher SMOTE Ratios like 1.0 and 0.7 and over 25 minutes for 5 fold Cross validation. Although utilising just the test dataset, gives an idea of how the model might generalise to unseen data, for the most accurate model, using of separate training and test dataset is crucial.

References

1. Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, 40(15), 5916-5923. https://doi.org/10.1016/j.eswa.2013.05.021

2. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613. https://doi.org/10.1016/j.dss.2010.08.008

3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357. https://doi.org/10.1016/j.eswa.2013.05.021

4. Carcillo, F., Le Borgne, YA., Caelen, O. and others. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. Int J Data Sci Anal 5, 285–300 (2018). https://doi.org/10.1007/s41060-018-0116-z

5. Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2016). An empirical comparison of machine learning models for time series forecasting. Econometric Reviews, 29(5-6), 594-621. https://doi.org/10.1080/07474938.2010.481556

6. Zhang, H., 2004. The optimality of naive Bayes. Aa, 1(2), p.3.

7. GitHub, (2023), MSc-Project---Comparative-analysis-of-ML-Models-for-a-Credit-card-fraud-detection-system. [online] Available at: http://bit.ly/github_repo_MSc_Project_Zami