Machine Learning Techniques for Electricity and Gas Fraud Detection

Group 42: Choi Yat Long, Alvis Low Yue Han, Wong Pui Lun, Le Diep Bao Tran

Introduction

Fraud detection is a critical concern for businesses across various industries. Basic utilities such as electricity and gas companies, are commonly suspectable to customer fraud. The consequences of fraud for organizations can be severe, resulting in substantial financial harm, lasting harm to reputation, possible legal consequences, and a decrease in employee confidence and trust. These outcomes can ultimately threaten the organization's existence[1].

In this report, we aim to address the problem of fraud detection within a pool of customers for an Electricity and Gas Company using machine learning techniques. Identifying fraud using customer base data is challenging, particularly when dealing with complex, high-dimensional datasets with imbalanced class distributions. To tackle this issue, we plan to utilize innovative techniques in handling data imbalances and summarizing variable transaction information for each client, primarily focusing on four machine learning algorithms—Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron.

Recent guides on rectifying imbalanced classification[2] concluded that under-sampling can be useful. In the context of customer fraud, fraudulent occurrences are rare compared to legitimate ones. As such, the majority class is considerably larger than the minority class which may introduce biases in our models. With the focus on balancing class distribution, we aim to reduce the number of observations from the majority.

Logistic Regression
○ Utilizes the nature of the multivariate customer dataset, to accurately predict a binary outcome. ○ Is robust to noises
Random Forest
○ Handles complex data with non-linear relationships. ○ Reduces overfitting with multiple decision trees to predict outcome.
Naive Bayes
○ Assumes multivariate independence. ○ Suitable for solving multi-class prediction problems.
Multilayer Perceptron
○ Handles complex data with non-linear relationships. ○ Generalize effectively to new, previously unknown data when properly trained.

Table I: Advantages of ML algorithms used.

Recent Works

1. In 2017, Mubarek Aji and Prof Eşref Adali from Istanbul Technical University proposed their work titled "Multilayer Perceptron Neural Network Technique for Fraud Detection". They conducted intrusion detection based on the multilayer perceptron algorithm, Naïve Bayes, and decision tree algorithm. With a result of 99.47% using Multilayer perceptron with nine selected features.[3] However, the authors conducted the study based on the NSL-KDD data set where the size of both the training and testing datasets is adequate, allowing the study to be conducted on the entire data set. Although this eliminates the necessity of random selection of the data set, it doesn't indicate the model's generalization to unseen data.

   In real-world scenarios, models trained on one dataset may encounter different data distributions or patterns. In this study, the data used would lead to variations in performance. Therefore, while consistent and comparable results are desirable for comparisons, relying solely on experiments conducted on the complete dataset may not fully reflect the model's performance in diverse situations.
2. In 2023, a study from the LAUTECH Journal of Engineering and Technology proposed a credit card fraud detection model that employs a multilayer perceptron algorithm for training and testing the dataset.[4] In the study, a large sized dataset was used. However, the authors did not specify any explicit mention of data cleaning, validation, or handling of outliers.

   The absence of explicit mention of data cleaning, validation, or handling of outliers in the study poses several risks to the viability and versatility of the proposed model. Without proper data cleaning procedures, the dataset may contain errors, inconsistencies, or missing values that could adversely impact the model's performance, ultimately undermining the reliability of the model in real-world scenarios.

Dataset

The two CSV files: "client.csv" and "invoice.csv" are large-sized datasets. The "client.csv" file contains client information for over 21,000 individuals, while the "invoice.csv" file contains information on invoices issued to these clients. The common column between the two files is the "id" column, which serves as the unique identifier for each client.

The initial phase involved meticulous data preparation, including importing the datasets into Pandas Data Frames and partitioning the "client.csv" data into training and testing sets using an 80-20 split. To address the challenge of class imbalance, we implemented under-sampling techniques on the training dataset, to minimize the majority class. The representation of fraudulent and non-fraudulent clients.

Furthermore, we streamlined the data by implementing an "invoice_to_client" function to the existing dataset, facilitating a more efficient analysis. Our findings underscored the significance of feature engineering and undersampling in enhancing fraud detection capabilities, laying the groundwork for future modeling endeavors and the exploration of advanced machine learning techniques.

Attribute	Definition
consummation_reach	Maximum sum of all consummation level among all invoices reached.
range_year_client	Year Range of the gas and electricity invoices of a client respectively
mean_year_client	Mean year of the gas and electricity invoices of a client respectively
sd_year_client	Standard deviation: year of the gas and electricity invoices of a client respectively
prop_counter_statue	Proportion of each counter statue amongst all invoices
reading	Median and range of the reading
unique_counter_num	Number of unique counter number of a client
unique_tarif_type	Number of unique tariff types
unique_counter_code	The number of unique counter code
counter_coeff_have1	Binary variable which is 1 if at least 1 invoice consist a couner_coeff of 1
consommation_sum	Mean and standard deviation of the
mean_num_invoice	Mean number of gas and electricity invoice respectively
sd_num_invoice	Standard deviation number of gas and electricity invoice respectively
mean_oldVSnew	Mean difference of old and new invoices
range_month_num	Range of month_num amongst the invoices of each client.

Table II: Features added, definition.

Things to Note:
Numerical data: Mean was used to measure the concentration of the data and standard deviation to measure the dispersion of data.
Ordinal data: A unique number of ordinal data, or its proportions was used to measure the concentration while using the range to measure the dispersion.
Categorical data: One-hot encoding is a technique used to convert categorical variables into a numerical format.

Finally, we standardize the numerical data, normalize the ordinal data, and one-hot encode the categorical data respectively to ensure the variables are appropriately scaled and represented in our machine-learning model.

Methods

I. ML Models

A. Logistic Regression

We choose Logistic Regression due to its robustness. Despite its simplicity, logistic regression is expected to perform well in practice if we assume that the features are relatively linearly related to the target variable. Logistic Regression is also less prone to overfitting compared to more complex models, making it a robust choice for datasets with limited samples (especially when used with Undersampling). Additionally, Logistic Regression can handle both numerical and categorical input variables, which can be useful for our specific dataset.

B. Random Forest

Random Forest is an ensemble of Decision Trees. It is a good example of bootstrap aggregation, an ensemble learning method that combines many weak learners with high variance to generate a strong learner with low variance. It works by creating many decision trees at training time and outputting the class which is the mode of the classes.
We chose Random Forest due to its simplicity and ability to capture underlying relationships of the features.

C. Naive Bayes

Naive Bayes works with the assumption of feature independence, which can be considered as the most different approach compared to other models.

D. Multilayer Perceptron (MLP)

We hope to capture the more subtle and non-linear patterns in the dataset by implementing a more complex model.

II. Fine Tuning Parameters

A. Principal Component Analysis (PCA)

PCA identifies the directions, or principal components, along which the data varies the most. It then retains only the top principal components, thus transforming high-dimensional data into a lower-dimensional space while minimizing information loss. However, stepwise regression assumed the linearity between the variables, which may not be true in most of the cases.

B. Stepwise Regression

Stepwise Regression obtains the subset of features that best explain the variation in the target. For our customized Stepwise Regression, we add features into our desired subset one at a time. For each remaining feature, the algorithm fits an ordinary least squares regression model using the current subset. It then extracts the p-value associated with that feature from the model's summary statistics. If the minimum p-value is below the threshold (0.05 in this case), the corresponding feature is added to the desired subset.

C. Hyperparameter Importance Analysis (HIA)

Hyperparameter importance analysis involves assessing the impact of different hyperparameters on the performance of a machine-learning model. Hyperparameters are parameters that are set prior to model training and control aspects such as model complexity, regularization, and optimization strategy. Hyperparameter importance analysis typically involves systematically varying the values of different hyperparameters and evaluating their effect on model performance.

We first use HIA to obtain an importance rate for each of the features. Then, we determine a threshold and select the features which have the importance rate greater than that threshold for model training. Thereafter, a graph of threshold against accuracy, true positive rate (TPR) and true negative rate (TNR) is plotted to determine the optimal threshold. We then proceed to extract the highest accuracy and its corresponding TPR and TNR to be used as metrics for further evaluation.

D. K-fold Cross Validation

Due to the time consumed for running the K-fold Cross Validation would be 60 mins longer as our method of implementing the X_train, X_test is more different from the ordinary data split implemented by scikitlearn, we decided not to proceed with K-fold Cross Validation and refine by using other methods, such as the aforementioned Stepwise Regression and Hyperparameter Importance Analysis instead.

Results

We evaluate the model's performance based on the model's True Positive Rate and True Negative Rate

Fig 1: Primary Implementation of the Four Models.

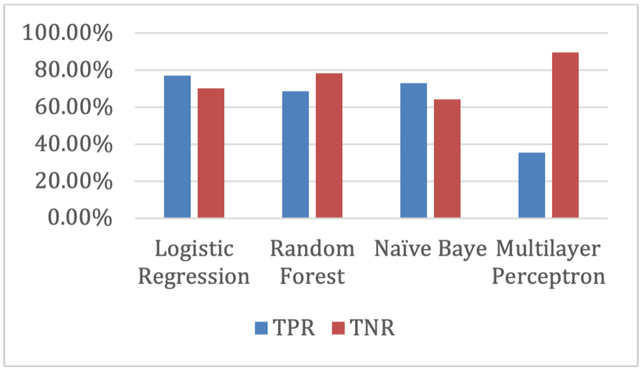


Fig 2: After PCA analysis

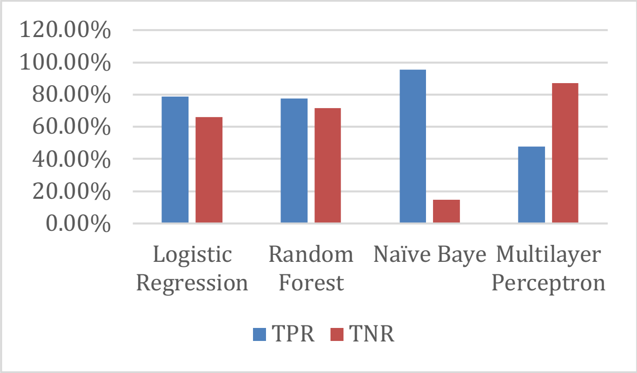


Fig 3: After Stepwise Regression

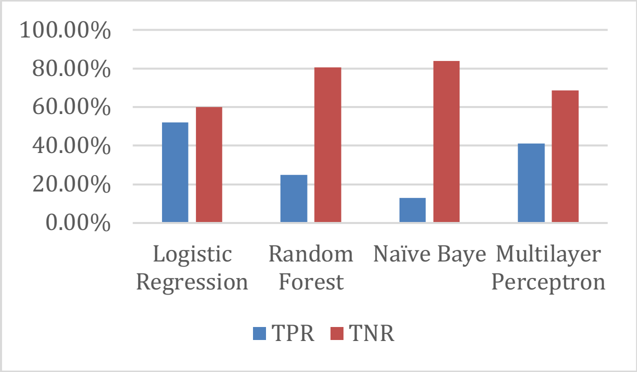
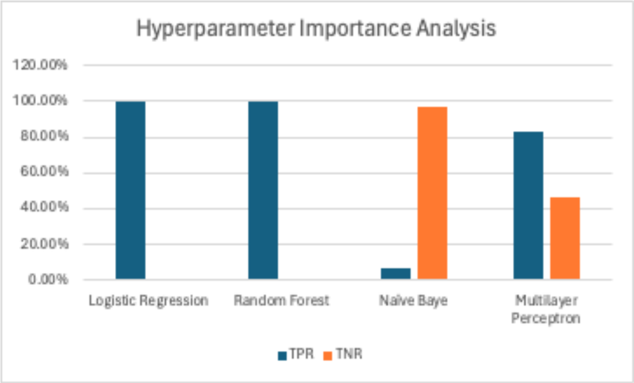


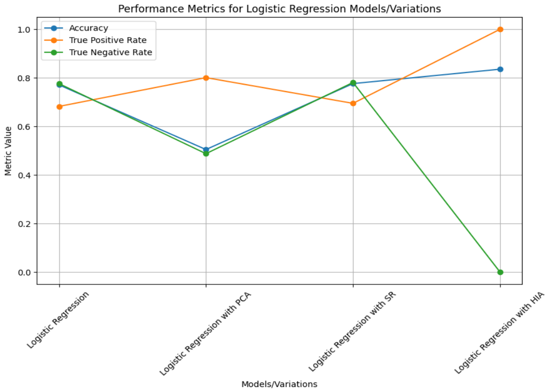
Fig 4: After Hyperparameter Importance Analysis



Conclusion

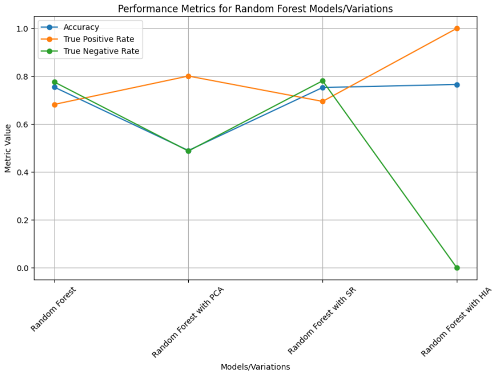
In our comparative analysis, we assessed the accuracy, TPR and TNR of each model alongside their respective variations to ascertain the most effective parameter fine-tuning method for each model. Our priority lies in selecting models with a high TPR to minimise false negative, when a fraudulent client is incorrectly classified as non-fraudulent.

Fig 5: Logistic Regression



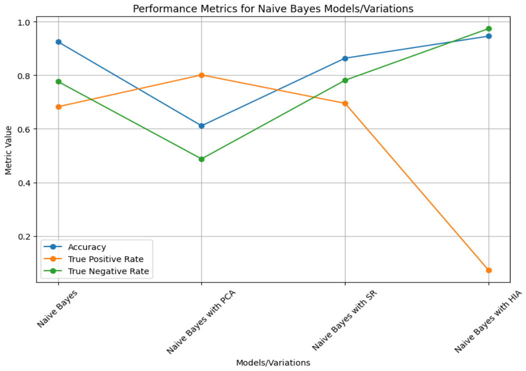
In the case of Logistic Regression, PCA emerges as a favourable option. While HIA yields the highest TPR, it concurrently leads to a 0% TNR, undermining its effectiveness for identifying fraud.

Fig 6: Random Forest



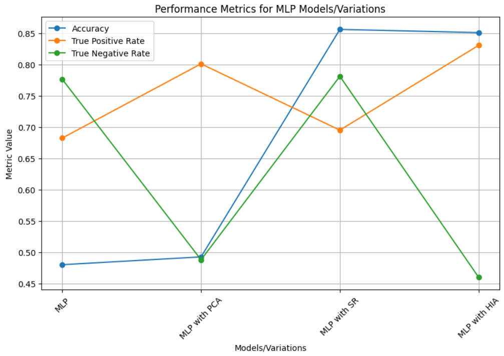
Similar to Logistic Regression, Random Forest exhibits optimal performance when combined with PCA. Nonetheless, the marginal increase in TPR does not justify the large drop in both TNR and accuracy. We assert that Random Forest operates sufficiently independently, or alternatively, we can employ SR to enhance all metrics by a small extent.

Fig 7: Naïve Bayes



PCA resulted in an improved TPR when paired with Naive Bayes. However, the small improvement in TPR comes with a considerable reduction in both accuracy and TNR. We conclude that Naive Bayes is best used without parameter fine-tuning for optimal performance.

Fig 8: MLP



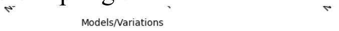
It is quite clear from the graph that pairing MLP with SR results in the most desirable outcome.

In our pursuit of fraud detection, our focus lies not in identifying the optimal model alone, but rather in deploying a comprehensive array of models along with their respective parameter fine-tuning techniques. This multi-pronged approach ensures a robust evaluation, enhancing our confidence in the classification outcomes.

References

[1] "Fraud Detection." Fraud.com, https://www.fraud.com/post/fraud-detection#:~:text=Fraud%20detection%20is%20crucial%20for,safeguards%20assets%20and%20sensitive%20information.

[2] "Undersampling." Masters in Data Science, https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/.



[3] Tan, Chia Wei, and Nur'Aini Abdul Rashid. "Multilayer perceptron neural network technique for fraud detection." ResearchGate, https://www.researchgate.net/publication/320829520_Multilayer_perceptron_neural_network_technique_for_fraud_detection.

[4] Elhaj, Fathi, and Jomana Al-Nuaimi. "DEVELOPMENT OF CREDIT CARDS FRAUD DETECTION MODEL." ResearchGate, https://www.researchgate.net/publication/378742518_DEVELOPMENT_OF_CREDIT_CARDS_FRAUD_DETECTION_MODEL.