IT1244 Report:
Fraud in Electricity and Gas Consumption Dataset

Lim Yu Long, Lee Zi Yang, Quek Zong Jin, Wong Song Ming

Introduction

Before the advent of AI/ML, the methods used in fraud detection ranged from manual inspection of measuring devices to statistical analysis of certain metrics. This report highlights the team's exploration and use of AI/ML techniques to analyse data and to detect fraud in electricity and gas consumption. We will also delve into the challenges faced with dataset processing and evaluate the performance of the models created.

Problem Significance

In modern societies which have a high level of dependence on electricity and gas, ensuring the accuracy and authenticity of consumption data is a major concern. Consumers may engage in fraudulent behaviors ranging from meter tampering to data manipulation to illegally reduce their utility bills. On a large scale, this not only causes revenue losses for utility providers but also distorts the energy consumption data used in policy-making and sustainable energy management. The effects of this issue thus range from the individual level all the way to the national level.

Literature Review

From our research, we found 2 existing research papers into electricity fraud which sought to use machine learning techniques to detect fraud. The first titled "Fraud Detection in Energy Consumption : A Supervised Approach" used data from Spanish utilities companies to implement multiple supervised learning algorithms such as K-Nearest Neighbour and decision trees, and they carried out feature selection and data aggregation. The second paper titled "Electricity Theft and Energy Fraud Detection" similarly used data from utilities companies to implement decision tree, random forest and support vector machine algorithms.

The first paper prioritized the area under the curve (AUC) as its main evaluation metric, which varied from 0.6 - 0.84 in their results. The second prioritised F1 score which ranged from 0.66 - 0.83. Our team plans to take a similar approach to these 2 papers with some adaptations to suit our dataset which may be smaller / less detailed than theirs. The issue of having insufficient time to conduct the papers was raised by the author of the second paper. In our efforts to enhance their work, we placed a stronger emphasis on fine-tuning hyperparameters rather than the extensive search for models to use. This shift in focus stems from the recognition that the data yielded positive results from their established models.

Project Work

To tackle this project, we began with a thorough analysis and exploration of the client and invoice data. This was followed by feature engineering to prepare the dataset for modeling. Next, we created a few baseline models which were taught in this module, namely linear regression and K-nearest neighbour. In addition, we explored more advanced machine learning techniques such as XGBoost to compare against the base models.

After verifying that our models work, we shifted our focus to optimization. We placed our efforts into tuning and improving our models with methods such as oversampling to address the class imbalance as well as hyper parameter tuning.

Issues faced

Our team faced 2 main issues with the dataset. The first issue was the extreme imbalance between classes and the second was the usability of IDs.

Data Imbalance

From our initial exploration, we found that the invoice file contained 127927 rows of data with a negative (no fraud) label and a mere 7566 rows of data with a positive (fraud) label. This is an extreme case of class imbalance with a near 17:1 imbalance. This imbalance in the data results in several issues.

Firstly, there is a high bias in the data towards the majority class (no fraud). This could result in models always predicting the majority class to reduce the error rate which would render our model useless. Secondly, there is an unusually low amount of the minority class which is positive fraud. This may hinder the models from learning enough about the underlying patterns to accurately predict fraud. Hence, we decided that traditional validation

methods may not be useful. As an example, having a model always predict no fraud might lead to high accuracy due to the rarity of fraud. However, such a model would obviously be pointless for identifying fraud.

Data Quality

At first glance, the large number of unique IDs in the dataset suggests that there will be enough data to create an accurate model. However, upon further inspection, we found that many of the IDs in the client csv do not correspond to any entry in the invoice csv.

After filtering out the IDs which did not have any corresponding invoices, the number of unique clients was cut from 135493 to a mere 31603 with the label distribution being 1757 positive and 29846 negative. This is a 76.6% reduction in the size of the dataset, leading to high information loss which could affect the performance of our models.

Mitigation Strategies

At first glance, the large number of unique IDs in the dataset suggests that there will be enough data to create an accurate model. However, upon further inspection, we found that many of the IDs in the client csv do not correspond to any entry in the invoice csv.

After filtering out the IDs which did not have any corresponding invoices, the number of unique clients was cut from 135493 to a mere 31603 with the label distribution being 1757 positive and 29846 negative. This is a 76.6% reduction in the size of the dataset, leading to high information loss which could affect the performance of our models. To address these challenges, we considered several strategies.

Resampling Techniques:

To combat the imbalance, we considered oversampling the minority class and undersampling the majority class. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) were also explored to generate synthetic samples for the minority class.

Alternative Evaluation Metrics:

Rather than relying solely on accuracy, we included other metrics such as precision, recall, F1 score. These metrics offer a more holistic view of the model's performance, especially in the context of imbalanced datasets.

Pre-processing and Feature Engineering

After our exploration, we realised that the original dataset had to be processed before it could be used for modeling. First, we had to convert several categorical features in the invoice CSV into a numerical format. For dates, we used the datetime module to convert the data into numerical months and years. For counter type, we used integer encoding to convert it into a binary format.

Some features were aggregated or binned to create new features. In particular, region was binned into a new feature called region group. We also used the new / old index and the 4 consommation levels to create 2 new features which represent the change in index and the total consumption. These features were then used to check whether the change in index was consistent with the total consumption for each invoice.

Next, we used ID to aggregate the invoices for each client and calculated various metrics such as median, standard deviation ect. Lastly to reduce dimensionality, we dropped features such as ID and date (raw) which were unnecessary for modeling. Some features which were created from the above aggregation were also dropped as they were mostly 0.

AI/ML Models Explored

To address the fraud detection problem outlined in the Introduction, we employed a variety of AI/ML techniques, each with its unique strengths. Our aim was to assess the performance of both classical and contemporary models to determine the best fit for our dataset.

The models we've chosen to explore and test on this dataset are as follows:

- K-Nearest Neighbours
- Simple Linear Model
- Multi-Layer Perceptron
- XGBoost

K-Nearest Neighbours (KNN)

Formulation

KNN is an algorithm which makes predictions by searching for the 'k' nearest training samples by euclidean distance. It then returns the output value (class) that has the highest frequency among the 'k' nearest data points.

Justification

KNN is a non-parametric and instance-based learning algorithm, making it suitable for datasets where relationships between variables may be complex and non-linear. Its simplicity and ability to adapt quickly to changes make it a good starting point for our analysis.

Linear Model

Formulation

A linear model makes a prediction by computing a weighted sum of the input features, plus a bias.

$\Theta 0 + \Theta 1 x1 + \Theta 2\ x2 + ... + \Theta n\text{-}1 xn\text{-}1 + \Theta n\ xn = 0$

Justification

Even though our problem might involve non-linear patterns, starting with a simple linear model still helps establish a baseline for more advanced models. Linear

models are also computationally less intensive and provide insights into the linear relationships within the dataset.

Multi-Layer Perceptron (MLP)

Formulation

MLP, a class of feedforward neural network, consists of at least three layers of nodes: an input layer, one or more hidden layers, and an output layer. Each node in a layer is connected to every node in the previous and subsequent layers, with associated weights.

Justification

MLP can model complex non-linear relationships. Given its capacity to learn intricate patterns through its hidden layers, it's particularly useful when linear models fall short. Additionally, it can be fine-tuned further by adjusting the number of hidden layers and nodes.

Extreme Gradient Boosting (XGBoost)

Formulation

XGBoost is an optimized distributed gradient boosting library that employs the gradient boosting framework. At its core, it constructs an additive model using decision trees in a forward stage-wise manner.

Justification

XGBoost is known for its performance and computational speed. It has built-in capabilities to handle imbalanced datasets, making it especially relevant for our problem. Moreover, its ability to compute feature importance provides valuable insights.

Evaluation of Models

Testing Methodology

To evaluate our models, we did a 9:1 train-test split of the original dataset, whilst ensuring that the ratio of the labels are kept consistent by taking 10% of both positive and negative cases. This is to ensure that the test data mimics the real world data as closely as possible.

Evaluation Metrics

In total, we used 5 metrics to assess the performance of our model. These are accuracy (Acc), precision (prec), recall, F1 Score and area under curve (AUC). We decided to prioritise F1 score for the results shown below as we wanted to strike a balance between precision and recall. Note that the results shown below may differ from the results shown in our code as we selected the best results from multiple runs using different parameters.

Results

Model	Metrics				
	Acc	Prec	Recall	F1	AUC
Linear	0.812	0.161	0.563	0.250	0.695
KNN	0.945	0.560	0.080	0.139	0.538
MLP	0.879	0.221	0.466	0.300	0.685
XG	0.944	0.489	0.125	0.199	0.559

Table 1: Baseline metrics

As can be seen in Table 1, MLP had the highest F1 score which is likely due to its ability to model complex relationships. KNN was the worst performing model with by far the lowest F1 score which was likely due to the high dimensionality of our dataset. Overall, the use of AI/ML did not yield the results our team expected with even our best performing model still having a low F1 score and AOC. We experimented with several methods to improve our results which we will discuss in the following sections.

Hyperparameter Tuning

For each of our models, we experimented with tuning various hyperparameters to improve our results. For linear model, we adjusted the learning rate and experimented with different loss functions. For KNN, we experimented with different values of K. For MLP, in addition to the learning rate and loss function, we also tried using different hidden layers as well as varying the number of hidden layers. To find the optimal parameters, especially for XGboost which has many hyperparameters, we used GridSearchCV.

Resampling of Dataset

Oversampling

Model	Metrics				
	Acc	Prec	Recall	F1	AUC
Linear	0.799	0.153	0.580	0.243	0.696
KNN	0.841	0.147	0.386	0.213	0.627
MLP	0.874	0.203	0.432	0.276	0.666
XG	0.889	0.260	0.545	0.353	0.727

Table 2: Metrics with random oversampling

Random Oversampling or upsampling is a technique used to correct class imbalance by randomly replicating instances of the minority class (fraud). In theory, this should lead to an improvement in the performance of our models as they would no longer favour the majority class (no fraud).

As can be seen in table 2, random oversampling had mixed results. For linear model and MLP, oversampling had a slightly negative impact on the performance of the model. However, it greatly improved the performance of KNN and XGBoost with the latter becoming our best performing model overall. This is likely because the performance of KNN and XGBoost are significantly impacted by imbalanced data which is mitigated by random oversampling.

Downsampling

Downsampling is another technique used to correct class imbalance. Unlike oversampling, it involves removing instances of the majority class (no fraud) from the dataset to balance the classes. However, we quickly realised that this technique is not suitable for our dataset.

Because of the extreme 16.9:1 ratio between no fraud and fraud in our dataset, we would have to remove almost 90% of the data to achieve a 1:1 ratio. This paired with the poor data quality would leave us with too little data to work with. Hence, we ultimately chose not to experiment with this technique.

Synthetic Minority Over-sampling Technique (SMOTE)

Model	Metrics				
	Acc	Prec	Recall	F1	AUC
Linear	0.544	0.089	0.778	0.160	0.654
KNN	0.775	0.127	0.517	0.204	0.654
MLP	0.724	0.107	0.540	0.179	0.637
XG	0.918	0.291	0.324	0.306	0.638

Table 3: Metrics with SMOTE

SMOTE is an advanced oversampling method that combats class imbalance by generating synthetic samples between instances of the minority class (fraud). Interestingly, when we applied SMOTE to our dataset, we observed a decline in every model's performance when compared to random oversampling. We have 2 possible explanations for why this is the case.

Categorical Features:

Our dataset is rich in categorical features, such as regions, districts and types of invoices. SMOTE, by its nature, creates interpolated data points which might not make logical sense for categorical variables.

For example, creating a synthetic point between two distinct regions or invoice types might result in a data point that doesn't represent a realistic scenario thus negatively affecting the performance of our models.

Inherent Data Characteristics:

The electricity and gas usage patterns between two minority samples might not necessarily be indicative of fraud. As SMOTE generates synthetic points by interpolating between these minority instances, it could inadvertently produce data that doesn't truly capture fraudulent behavior.

In conclusion, while SMOTE is potent, its effectiveness hinges on the nature of the data and it is unfortunately poorly suited to our current dataset.

Further Improvements

There are several improvements that we could make. For example, we could further refine the parameters used for our models particularly for linear model and MLP. In addition, we could further reduce the dimensionality of our data through techniques such as principal component analysis. Lastly, we could experiment with additional machine learning algorithms such as logistic regression and random forest which may perform better than our existing models.

Conclusion

Overall, the performance of our models was lacking which is partially due to the poor quality of the dataset. Still, we managed to improve the performance of our models through various techniques such as oversampling. Our team learned a lot about machine learning from this project including some things that were not covered in this module such as SMOTE. The hands-on experience may also prove useful for future modules and projects.

References

Coma-Puig, B., Carmona, J., Gavalda, R., Alcoverro, S., & Martin, V. (2016). Fraud detection in energy consumption: A supervised approach. https://www.cs.upc.edu/~gavalda/papers/dsaa2016.pdf

Rajab, A. J. (2023, April 5). Electricity theft and Energy Fraud Detection - Rochester Institute of ... https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=12630&context=theses