Credit Card Fraud Classification

Data Mining Assessment of Learning Report

Class: LA02

Group 8:
2440007226 - Eduardo Brilliandy
2440026585 - Felix Museng
2440007062 - Gian Reinfred Athevan
2440012390 - Hizkia Christian Purnomo

INTRODUCTION

Credit card fraud is one of the most frequent crimes that happens in today's society. There are a lot of ways people can be susceptible to credit card fraud, for example from stolen personal information, stolen cards, or credit card skimming. This experiment attempts to identify fraudulent credit card transactions so customers won't be charged for items they did not purchase using data mining techniques and machine learning algorithms.

By using data mining and machine learning, this experiment attempts to detect which transaction is fraudulent or a real transaction done by the card holder. This experiment uses 3 different tree based classification algorithms to analyze, compare, and determine which algorithm is most suitable. The algorithms used include Decision Tree, Random Forest, and XGBoost Classification.

Imbalanced data occurs when there is an imbalanced distribution of the target class. Imbalanced data occurs frequently in the real world. In the case of fraud detection, the amount of genuine transactions is much larger than the amount of fraud transactions. This experiment also attempts to determine the best strategy to deal with a heavily imbalanced dataset. These strategies include using class weights, undersampling, and oversampling.

DATA

The dataset that is used is a Credit Card Fraud Detection dataset from Kaggle. The dataset contains Anonymized credit card transactions labeled as fraud or genuine transactions.

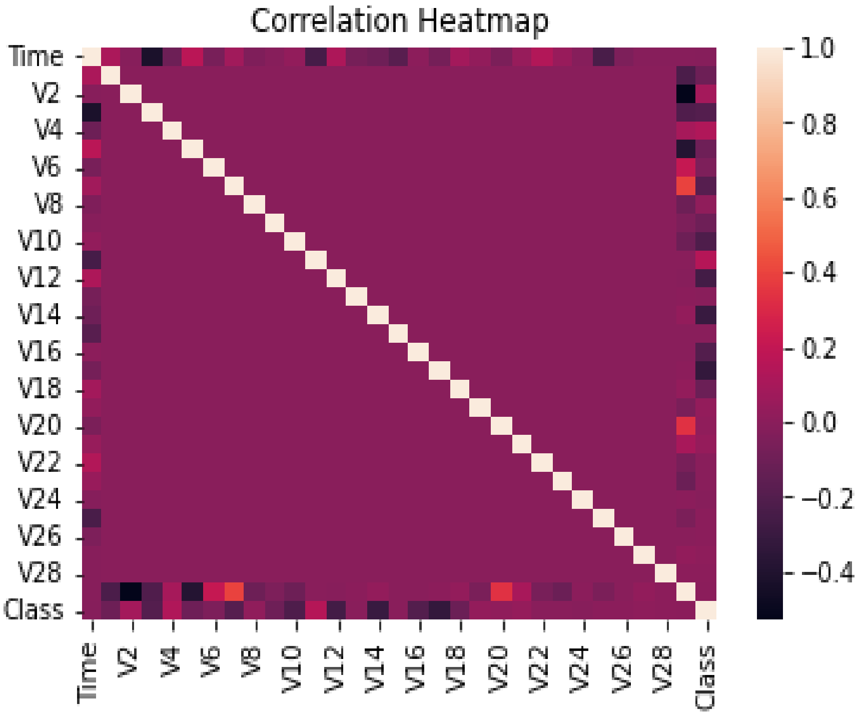
The dataset contains 284807 rows of data and 30 features. The dataset contains no missing values and all of the data are already numerical values, therefore Data Cleaning is no longer needed.

The features in the dataset have been processed using PCA (Principal component analysis) Dimensionality reduction, except for the Time and Transaction Amount Features. Since PCA requires the data to be scaled first, only the Time and Amount features need to be scaled. The Scaling of the data is done using Standard Scaler. Standar Scaler standardizes each of features by rescaling the value so that the mean is 0 and the standard deviation is 1, centering the data.

To further reduce the dimension of the data, feature selection will be done. Feature Selection chooses the important parameters, based on the problem that is going to be solved, to avoid getting unimportant patterns and get rid of noise from the data. In this experiment, the features will be selected using the Select K Best method. This method selects the features that contribute most to the target class based on the variation of the features.

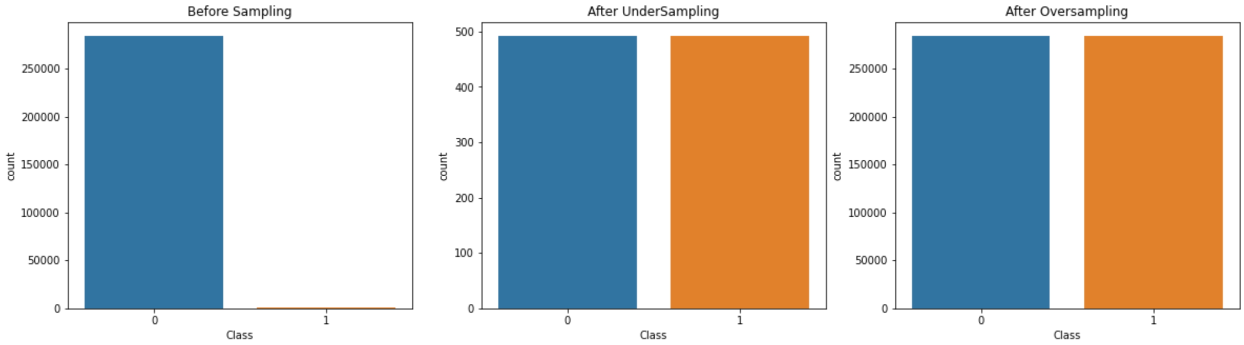
DATASET PROBLEM SOLVING

Fig. 1 Correlation Heatmap



The correlation between the features is shown in the correlation heatmap in Figure 1 above. It is found that features V1 to V28 have no correlation with each other. This is expected because those features are the result of PCA.

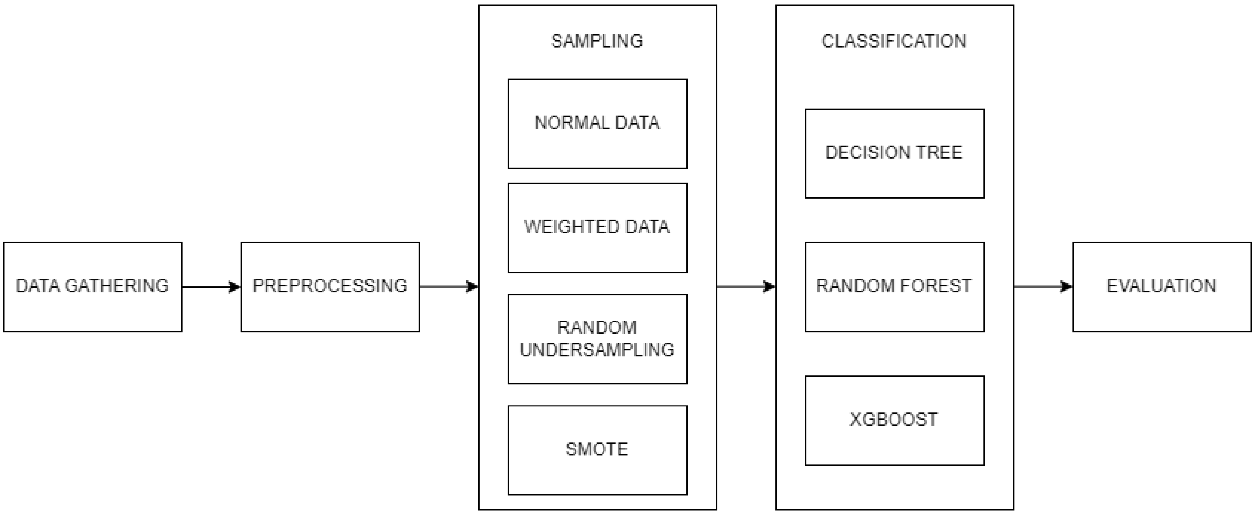
Fig. 2 Class distributions



The dataset is highly unbalanced, the positive class (fraud) accounts for only 0.172% of all transactions. Therefore, undersampling and oversampling will be applied to the dataset. To undersample, Random Undersampling is used. To oversample, SMOTE (Synthetic Minority Oversampling Technique) is used. Figure 2 above shows the class distribution from before sampling, after undersampling, and after oversampling respectively. (0: not fraud, 1: fraud)

METHOD

Fig. 3 Methodology flowchart



In this experiment, 3 tree-based classification algorithms will be used and compared to determine which algorithm performs the best. The algorithms used are Decision Tree, Random Forest, and XGBoost. Below are the description for each algorithm:

- Decision Tree:

  Decision tree creates a tree-like model of decisions based on features provided by the data. The algorithm starts by identifying the feature that best splits the data into classes, and then it splits the data based on that feature. It continues to repeat this process until all the data is classified or until a stopping criterion is reached.

- Random Forest:

  Random forest uses a large number of individual decision trees that operate as an ensemble, having each tree produce a class prediction, then the class with the most votes will become the model's prediction. It also uses a classifier so that it could handle missing values and prevent the model from overfitting when there are more uncorrelated trees created.

- XGBoost:

  XGBoost or Extreme Gradient Boosting is a machine learning algorithm that creates a prediction model by combining the predictions of many weak models. XGBoost is an efficient implementation of gradient boosting that uses decision trees as the weak models. It is based on the idea of gradient descent, which involves minimizing a loss function by making small adjustments to the model's parameters.

Additionally, each model will be trained using the normal dataset, weighted dataset, undersampled dataset, and oversampled dataset. For the weighted data, the weight will be determined using the ratio of the target class to make the weights for each class equal. For the undersampled data, the random undersampling method is used. This method samples the number of the majority class to be equal to the minority class. For the oversampled data, the SMOTE (Synthetic Minority Oversampling Technique) method is used. This method synthesizes the minority class to increase the number of the minority class until it is equal to the number of the majority class.

Therefore, there will be a total of 12 models to compare. In this experiment, each model will use default parameters.

EVALUATION

To evaluate each model, k-fold cross validation will be used with k = 5. This means that the data will be split into 5 subset, one subset will be used as the Testing data to test the model while the other subsets will be used as the Training data to fit the model. This process will be iterated for each subset. The scores for each iteration will then be averaged.

Because of the heavily imbalanced dataset, accuracy cannot be used as a valid evaluation metric for this experiment. This is because if the model labels all of the test data as genuine transactions, the accuracy will still be above 99%. Instead, the metrics that will be used to evaluate the models are Precision, Recall, F1 Score, and ROC AUC. These metrics will provide more useful information than the default accuracy.

Below are the definitions for each metric used:

- Precision

  TP/(TP+FP)

  Precision is a measure of fraud transactions predicted against all transactions that are predicted as fraud. A low precision means the model predicts a lot of genuine transactions as fraud.

- Recall

  TP/(TP+FN)

  Recall is a measure of fraud transactions predicted against all fraud transactions in the dataset. A low recall means the model predicts a lot of fraud transactions as genuine. Because it is more important to predict fraud transactions correctly, recall is a more important measure than precision in this experiment.

- F1 Score

  2*(Precision*Recall)/(Precision+Recall)

  F1 Score is a measure that takes both Precision and Recall. Usually, the value of the F1 score is between the Precision and Recall score. If either Precision or Recall is low, then the F1 score will also be low.

- ROC AUC

  ROC AUC (Receiver Operating Characteristic Area Under Curve) is the Area under the ROC curve. The ROC curve is the graph between the True Positive Rate and the False Positive Rate. AUC ranges between 0 to 1, where 0 means the predictions of the model are 100% wrong, and 1 means the predictions of the model are 100% correct.

RESULT & ANALYSIS

Table 1 Decision Tree Results

	Normal	Weighted	Undersampled	Oversampled
Precision	0.8766	0.8795	0.5078	0.6792
Recall	0.8880	0.8687	0.9141	0.8901
F1	0.8814	0.8740	0.4890	0.7452
ROC AUC	0.8880	0.8687	0.9141	0.8901

Table 2 Random Forest Results

	Normal	Weighted	Undersampled	Oversampled
Precision	0.9763	0.9780	0.5249	0.9267
Recall	0.8923	0.8811	0.9382	0.9095
F1	0.9301	0.9239	0.5397	0.9177
ROC AUC	0.9448	0.9470	0.9797	0.9724

Table 3 XGBoost Results

	Normal	Weighted	Undersampled	Oversampled
Precision	0.9604	0.9604	0.5221	0.5498
Recall	0.8902	0.8902	0.9401	0.9462
F1	0.9221	0.9221	0.5336	0.5863
ROC AUC	0.9799	0.9799	0.9807	0.9817

From the results above, The highest Recall was achieved by the XGBoost model using the Oversampled data at 94.62%. It also achieved the highest ROC AUC scores at 0.9817. However, the precision of this model is very low at 54.98%. This means that this model prioritizes more on labeling the data as fraud. This model performs the best if correctly predicting genuine transactions is less important or less expensive than correctly predicting fraud transactions.

However, if correctly predicting both genuine and fraud transactions are of equal importance, then the Random Forest model using the Normal data performs the best with a precision of 97.63% and an F1 score of 0.9301. The model also maintains a recall of 89.23%.

Next, it is found that weighting the target class in all 3 models had little to no effects at all. Therefore, using the normal dataset would be preferable compared to weighing it since it takes less computational time. It is also found that undersampling and oversampling could increase the correct prediction of fraud transactions, but could perform worse in correctly predicting genuine transactions. Overall, oversampling is found to yield better results compared to undersampling where it performed slightly better for the Decision Tree and Random Forest models, and performed way better for the Random Forest model.

Lastly, it is found that both Random Forest and XGBoost models performed better than the Decision Tree model. This result is expected since both Random Forest and XGBoost are algorithms that were made to improve on the Decision Tree algorithm.

CONCLUSION

In conclusion, The XGBoost model using Oversampled data performed the best if correctly labeling fraud transactions is the most important and the Random Forest model using Normal data performed the best if correctly labeling genuine and fraud transactions are equally important. Overall, the Decision Tree model performed the worst out of the 3 models.

For the strategies to deal with an imbalanced dataset, weighing the target class had little to no impact. Undersampling and Oversampling helped in correctly predicting fraud transactions but made less correct genuine transaction predictions.

IMPLICATION

Machine learning can be used as the first layer of detecting credit card scam and warns people when their credit card is compromised. The findings in this experiment can help improve future fraud detection systems in detecting fraudulent transactions. This research has shown that both Random Forest and XGBoost can be reliable ways of detecting credit card fraud in banks that provide credit cards.

Further studies could test other classification algorithms to see if higher scores could be achieved. Further studies could be done to test the effects of sampling on an imbalanced dataset by testing different ratios of sampling.

REFERENCES

ULB, M. L. G.-. (2018, March 23). *Credit Card Fraud Detection*. Kaggle. Retrieved December 1, 2022, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. *Procedia Computer Science, 165*, 631–641. https://doi.org/10.1016/j.procs.2020.01.057

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit Card Fraud Detection - machine learning methods. *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. https://doi.org/10.1109/infoteh.2019.8717766

Lee, K. C. (2021, March 19). *The evolution of trees-based classification models*. Medium. Retrieved December 10, 2022, from https://towardsdatascience.com/the-evolution-of-trees-based-classification-models-cb40912c8b35

Pandey, R. (2021, May 16). *How to deal with an imbalanced dataset*. Medium. Retrieved December 12, 2022, from https://medium.com/analytics-vidhya/how-to-deal-with-an-imbalanced-dataset-47c8ce98c459

Brownlee, J. (2021, January 4). *Random oversampling and undersampling for imbalanced classification*. MachineLearningMastery.com. Retrieved December 12, 2022, from https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

Gupta, P. (2017, June 5). *Cross-validation in machine learning*. Medium. Retrieved December 13, 2022, from https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

Google. (n.d.). *Classification: Roc curve and AUC | machine learning | google developers*. Google. Retrieved December 13, 2022, from https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Google. (n.d.). *Classification: Precision and recall | machine learning | google developers*. Google. Retrieved December 13, 2022, from https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall