



# ID5059: Knowledge Discovery and Datamining

## Assignment 2

Ian Liu

Jingjing Lu

Alexander Ross

Ulrike Wachter

Zhihao Zhong

### Detecting EU Credit Card Fraud

## Introduction

Credit card fraud is a growing problem in Europe. The total value of transactions using cards issued in the Single Euro Payments Area amounted to €5.16 trillion in 2019, of which 0.04% was fraudulent (European Central Bank, 2019). This highlights the urgent need for our client to focus on effective detection of fraud and prevention measures. As banks play a vital role in the economy, and fraud undermines public trust in financial systems, understanding bank fraud is essential for promoting economic security.

This report uses various binary classification models to predict fraudulent transactions and aims to present our client with actionable insights and recommendations for addressing this problem.

## Part 1: Methods

### Task 1.1: Exploring and visualizing the data

The data has 32 variables with training set containing 219,129 observations and the test set 146,087, i.e., a 60-40 split (Kaggle, 2023). Before any models can be created or trained, it is vital to visualize the data to understand the variables' distributions, correlations, and any other aspects such as missing values or outliers.

Looking first at the distribution of the fraud class variable in Figure 1, it is clear that the dataset is highly imbalanced, with fraud cases only representing about 0.17% of the dataset. This could lead to model bias if not taken into consideration.

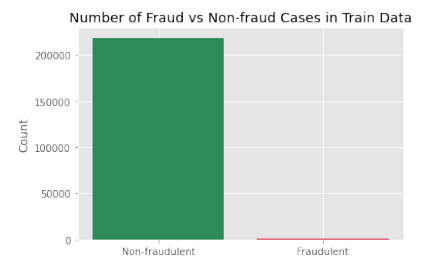


Fig. 1: highly imbalanced class dataset

Next, looking at the amount in Euros spent per transaction we can see that it is a highly skewed variable, with 70% of transactions being below €50, meaning that consumers overwhelmingly purchased smaller day-to-day items. This distribution is visualised in Figure 2 below.

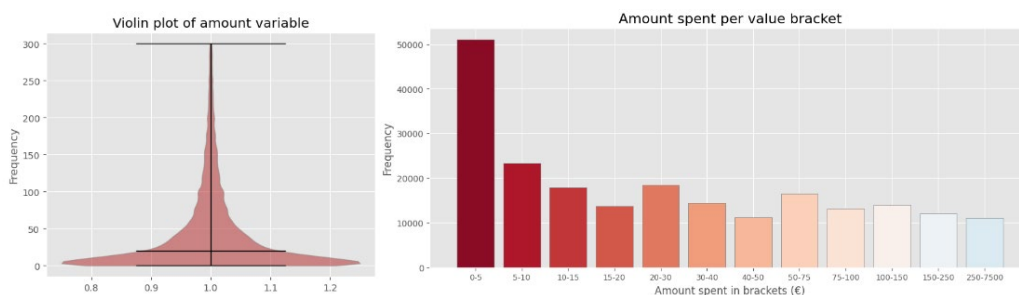


Fig. 2: Amount distribution (left) and amount spent per value bracket (right)

The time variable shown in Figure 3 is more equally distributed. All the transactions take place over a 34-hour window, and each observation contains the seconds elapsed between it and the first transaction in the data set.

The variables 'Class', 'Time', and 'Amount' are the only ones for which a name is provided; all others are standardised and are labelled V1 to V28. Therefore, we cannot assess these variables' meanings but can only infer them from the general distributions shown in their histograms. The majority of the attributes' distributions show close approximation to Normality.

Plotting amount against time, we can see that most transactions occurred in the middle of the period. Although we don't know the hour of the day, we can assume that  $t = 0$  is around midnight, as transactions drop to their lowest around 3-4am. We can also see that the highest fraud case was around €3,000.

Looking closer at the fraudulent cases specifically, it of interest to see if behaviour changes over time for any of the variables. The plots below provide some insight into fraudulent behaviour. Most interestingly, V12 shows that there are two dips around the very early hours of each day, suggesting that V12 is related to the time variable.

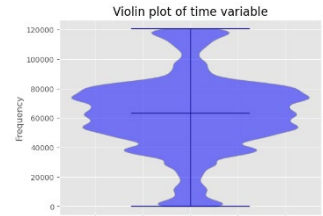


Fig. 3: Distribution of time variable

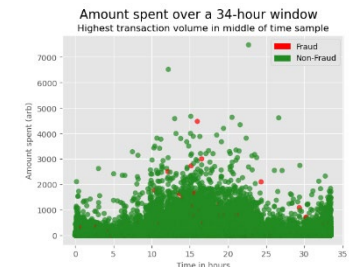


Fig. 4: Amount spent over time

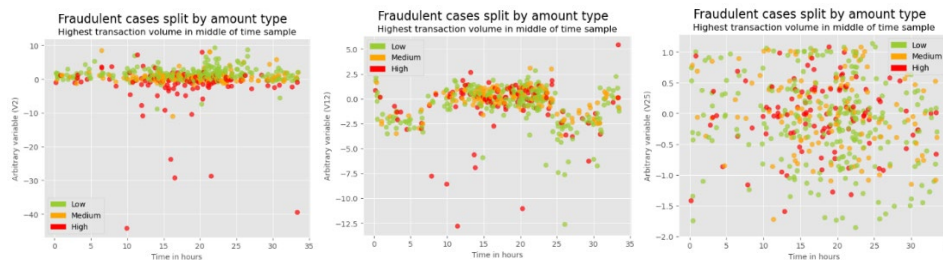


Fig. 5: plotting the 3 variables V2 (left), V12 (middle), V25 (right) for fraudulent cases

Finally, we discover from examining the correlation between different variables that none of the numerical attributes is very correlated with the Class attribute, as all their values lie close to 0 (Figure 6).

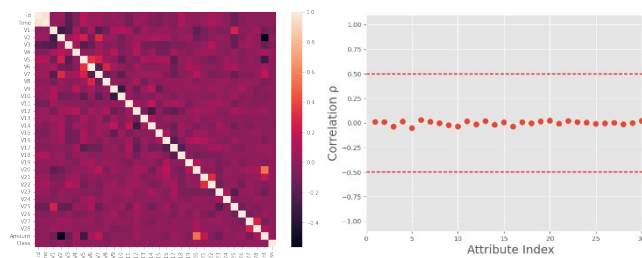


Fig. 6: correlation matrix for all variables (left); correlations between each explanatory variable and Class

### Task 1.2: Preparing the data for analysis

Following our exploratory visualisations, three steps were taken to prepare the data for downstream analysis. Firstly, the variables “Time” and “Amount” were standardised to reduce differences across variables and avoid bias. Secondly, the data was shuffled to avoid potential ordering of the dataset. Thirdly, we considered PCA to reduce model complexity, however, decided to retain all explanatory variables due to the low level of correlation present in the data.

### Task 1.3: Imputing missing data

To address the potential issue of missing data, three methods of imputation were tested: ‘mean’, ‘median’ and ‘most frequent’. 1% of data was randomly removed and a logistic regression model was tested on each method. We found that the ‘most frequent’ imputation method gave the best ROC-AUC score, representing the best-performing model.

	Method	Description	ROC-AUC score
1	Mean	Replaces missing values with the mean of the non-missing continuous values in the same feature	0.504
2	Median	Replace missing values with the median of the non-missing continuous values; less sensitive to outliers	0.499
3	Most frequent	Replace missing values with the most frequent value appearing in that column; best for discrete	0.516

### Task 1.4: Selecting models, training, and fine-tuning them

Our model selection process was dependent on two factors - the models need to be appropriate for a binary classification task and they must also be robust when dealing with unbalanced class data.

These criteria led to a shortlist of seven models: logistic regression, Stochastic Gradient Descent (SGD) classification, gradient boosting classification, decision tree, random forest, k-nearest neighbours (KNN), and naïve Bayes. To ensure class-imbalance robustness for the models, we tested two methods on all of them - resampling using oversampling, which synthesises new data to increase the proportion of the minority class (SMOTE algorithm) to equal that of the majority class (Brownlee, 2020), and undersampling, which removes observations from the majority class until it is of the same size as the minority class. Each method results in an evenly split dataset. The limitations of both models are described in the limitations section below.

## Part 2: Results

Due to our highly imbalanced data, the evaluation metrics we chose were of particular importance. In this case accuracy is not a relevant evaluation metric because it can easily score highly by always predicting the majority class. For this reason, we focused instead on precision, recall, F1 and the area under the receiver operating characteristic curve (ROC-AUC score) to evaluate performance.

The results of our models' performance on the training set are summarised in the table below, in order of best ROC-AUC score, which measures the classifier's ability to distinguish between the positive and negative classes: a perfect classifier will score 1, whereas a purely random classifier will score 0.5. The results are colour-coded by the models' performance on each of the three data sets: **red** is the **original**, **blue** is **oversampled**, and **purple** is **undersampled**.

	PRECISION	RECALL	F1 SCORE	ROC-AUC
NAÏVE BAYES	0.060	0.089	0.072	0.543
	0.026	0.207	0.067	0.787
	0.024	0.171	0.066	0.572
GRADIENT BOOSTING CLASSIFIER	0.079	0.013	0.022	0.506
	0.029	0.542	0.019	0.774
	0.027	0.667	0.013	0.741
RANDOM FOREST	0.250	0.002	0.004	0.501
	0.006	0.174	0.017	0.544
	0.015	0.689	0.013	0.732
DECISION TREE	0.364	0.009	0.017	0.504
	0.006	0.426	0.018	0.697
	0.008	0.489	0.015	0.658
LOGISTIC REGRESSION	0.156	0.011	0.019	0.505
	0.007	0.888	0.006	0.590
	0.003	1.000	0.004	0.544
K-NEAREST NEIGHBOURS	0.000	0.000	0.000	0.499
	0.002	0.360	0.002	0.383
	0.002	0.355	0.002	0.536
SGD CLASSIFIER	0.461	0.572	0.510	0.451
	0.000	0.000	0.000	0.000
	0.003	0.800	0.003	0.501

Table 1: each model's performance on training set using the original, oversampled and undersampled data sets



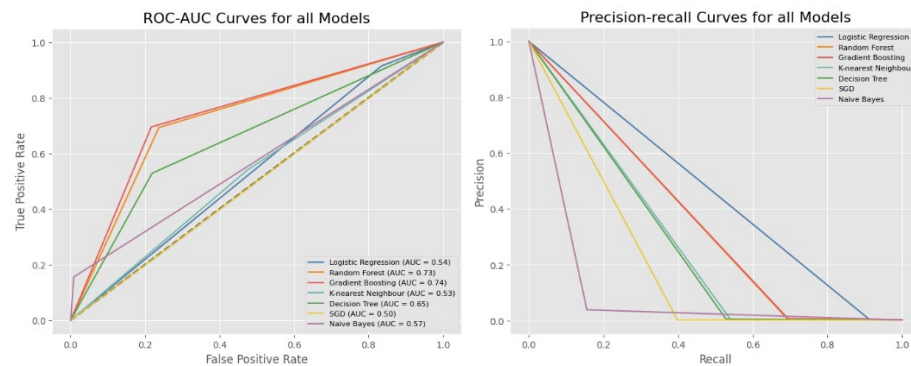


Figure 7: Model results for undersampling

In summary, the gradient boosting classifier has the highest undersampling ROC-AUC score of 0.74, although it has only the second highest F1 score. Logistic regression has the highest F1 score but a very low ROC-AUC score, which can be seen in Figure 7. 1s and 0s were used as classes instead of probabilities. ROC-AUC measures the ability of a model to distinguish between the positive and negative classes across all possible classification thresholds, and is particularly useful when the cost of false positives and false negatives is similar. F1 score is a measure of the balance between precision and recall. It is particularly useful when the cost of false positives and false negatives is different, and when we care more about correctly classifying the positive instances, as in our case.

Interestingly, from an inspection of the feature importance for the best model, we can see that the variable V14 was by far the most important in the predictions made by the classifier, with 36% importance (Figure 8). We recommend the client to prioritize collecting data from V14 to optimize future models.

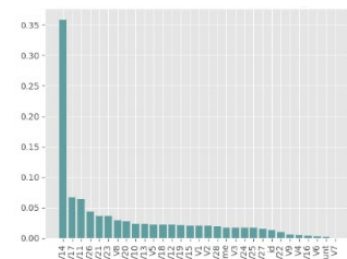


Figure 8: feature importance for the gradient boosting classifier

### Part 3: Discussion and Limitations

Limitations	Potential solution
Oversampling creates synthetic data that have not in reality been observed, so if the synthetic data is too close to the original data the model may not generalise as well due to overfitting. It also increases computational complexity.	We could try other methods to deal with class imbalance such as weighted loss function, more ensemble learning methods, anomaly detection or feature engineering
Undersampling removes a large amount of observed information, reducing the amount	We could try other methods to deal with class imbalance such as weighted loss

of real data the model is exposed to, and possibly its performance. Reducing the data set also reduces variability which may lead to overfitting.	function, more ensemble learning methods, anomaly detection or feature engineering
No experiments with attribute combinations were done since no variable names were provided	Given a dataset with variable names, experts in the credit card fraud space could help to combine sensible attributes
Hyperparameter tuning only implemented for 2 models (gradient boosting classifier and KNN). Tuning for certain models is extremely time-consuming. The notebook only has the implementation for parameter tuning for KNN but not its results due to limited computing resources.	Either invest in paid cloud computing solutions or simplify the grid search and try fewer combinations of parameters. However, for this data set, any meaningful hyperparameter tuning, even with minimum combinations, is still computationally expensive.

## Part 4: Conclusion

Various binary classification models were tested and evaluated in order to help our client predict credit card fraud. The best model was the naïve Bayes classifier using oversampling, with a final ROC-AUC score of 0.747. The second highest score was the gradient boosting classifier with a score of 0.725. This result indicates that the classifier is performing better than a random guess but the performance may still be suboptimal due to the high cost of false negative predictions. Each instance of fraud can impact our client's customers immensely, as well as reflect badly on our client's safety measures. Before using these models in their software systems, further analysis needs to be done. Specifically, more hyperparameter tuning will be necessary and use of more advanced classification algorithms (like CatBoost, a gradient boosted decision tree classifier) will further increase the performance. Notwithstanding, our overall analysis represents an important first step for our client in their pursuit of reducing the impact of credit card fraud on its clients, and further close collaboration will ensure continuous progression to this end.

## Part 5: References

- Wongvorachan T., He S., Bulut O. (2023). *A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining*. 14(1):54. Available at: <https://doi.org/10.3390/info14010054>
- Brownlee, J., (2021). *Imbalanced Classification: SMOTE for Imbalanced Classification with Python*, MachineLearningMastery.com. Available at: <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/> (Accessed: April 5, 2023).
- European Central Bank, (2019). *Seventh Report on Card Fraud in SEPA*. Available at: <https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport202110~cac4c418e8.en.html#toc11>
- Kaggle, (2023). *Playground Series – Season 3, Episode 4*. From Kaggle Competitions. Available at: <https://www.kaggle.com/competitions/playground-series-s3e4/data>

# Kaggle results – screenshot of two top models

**Playground Series - Season 3, Episode 4**  
Tabular Classification with a Credit Card Fraud Dataset  
Kaggle · 641 teams · 2 months ago

Overview Data Code Discussion Leaderboard Rules Team Submissions Late Submission



### Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

Submissions evaluated for final score

All Successful Selected Errors Recent

Submission and Description	Private Score ⓘ	Public Score ⓘ	Selected
 <b>the miners nb.csv</b> Complete (after deadline) · now	<b>0.74702</b>	<b>0.76316</b>	<input type="checkbox"/>
 <b>the miners gb.csv</b> Complete (after deadline) · 14s ago	<b>0.72547</b>	<b>0.77129</b>	<input type="checkbox"/>