ID5059 - Knowledge Discovery & Data Mining
Coursework 2

Predicting Genuine/Fraudulent Transactions

7 April 2023

Report Outline
Introduction
Methodology

1. Data Exploration
    a. Explanatory Variables
    b. Target Variable
2. Data Wrangling
    a. Imputation
    b. Outliers
    c. Balancing Data: Random Undersampling, SMOTE
    d. Data Scaling
3. Model Fitting & Selection
    a. Undersampled Data Model
4. Predictions for Test Data

Results

1. Feature Importance
2. Imputation and Imputation Model
3. Undersampled Data Confusion Matrices
4. Test Data Submission

Discussion

University of
St Andrews

Introduction

The aim of this study is to predict genuine and fraudulent credit card transactions from given attributes (binary classification problem). Most of the dataset used in this study has been processed using Principal Component Analysis (PCA), a technique that transforms the original data into a set of uncorrelated variables in a lower-dimensional space. As a result, all other numerical explanatory variables but “Time” and “Amount” were opaque. Due to confidentiality reasons, the data do not provide original features or any more background information, meaning the report will not be able to provide precise suggestions, but only recommendations at where to look for cues. The study employed statistical methods and several machine learning models to assess the influence of predictors on transaction genuineness, and to obtain predictions. When predicting fraudulent and genuine transaction classes, avoiding any false negatives is crucial. A false negative occurs when a fraudulent transaction is passed off as genuine, which is detrimental. Vice-versa, a false positive, i.e. a genuine transaction classed as fraudulent, would not cause such great harm, however is still costly.

Methodology

Explanatory Variables

The dataset contains 219,129 instances (rows) and 32 features (columns), with 28 principal components, “Time”, “Amount”, “ID”, and prediction variable “Class”. As PCA was performed, and supposedly only the most important components were chosen to explain around 90% of data variance, the resulting data set was still extensive, the underlying data set proves to be complex. The variable "Time" represents the duration, in seconds, between each transaction and the initial transaction recorded in the dataset, while "Amount" refers to the transaction amount. Both variables exhibit non-normal distributions, which call for scaling to improve the performance of machine learning models (see **Fig. 1**).

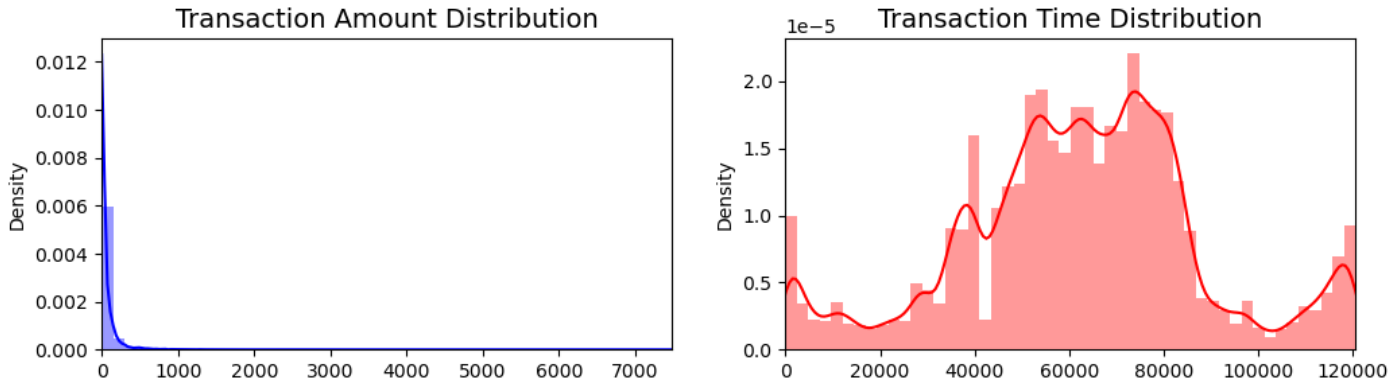


Figure 1: Explanatory variable “Time” and “Amount” Density Distributions

Target Variable

The data exhibits an extreme class imbalance, where only 0.21% of the cases are classified as fraudulent and the remaining 99.79% are non-fraudulent (see **Fig. 2**). Addressing class imbalances is crucial for fraud detection, as models trained on imbalanced data may assume that most transactions are genuine, leading to biassed predictions and incorrect correlations between variables. Random Undersampling method and Synthetic Minority Over-sampling Technique (SMOTE) were used to balance the data. This will allow us to identify patterns accurately.

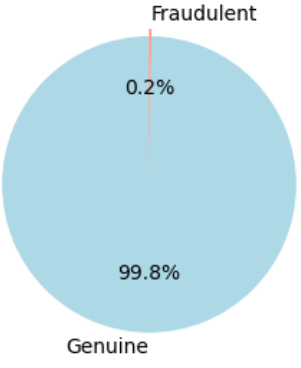


Figure 2: Distribution of Class in Original Train Data

Imputation

To assess the three imputation methods, firstly a copy of the training dataset was created and a small proportion of the data (5%) was randomly selected and removed. The Mean, K-Nearest Neighbours (KNN), and Iterative imputation methods were selected for this analysis.

To accurately assess the effectiveness of the imputation methods, the imputed values were compared with the original using two metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The RMSE is a metric that is more sensitive to outliers in the data, whereas the MAE is less so. Therefore, using both metrics resulted in more comprehensive and robust results.To further validate which method of imputation is the best, the study trains the best performing model, evaluated later in the analysis, with the training data from each respective imputation method (after further data cleaning). The receiver operating characteristic curve (ROC) scores of the model are then measured using each imputation data set and calculate which model performs the closest to the original.

Outliers

Handling outliers is generally crucial, as they may impact the models performance, and affect the relationships between variables. As such, Tukey's method for identifying outliers in data variables involves calculating the interquartile range (*IQR*), which is the difference between the third (*Q3*) and first (*Q1*) quartiles (Tayyip Saka, 2019). Then, the lower and upper bounds are computed as *Q1 - factorIQR and Q3 + factorIQR*, respectively. All data points that fall outside these lower and upper bounds are considered outliers.

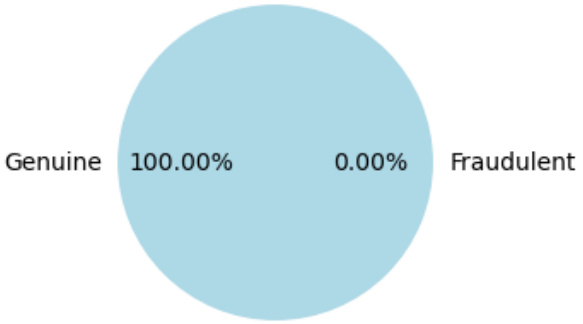


Figure 3: Distribution of Class in Train Data Without Outliers

After the outliers were removed, the data only contained genuine transactions (see ***Fig. 3***). This suggested that the outliers were linked to fraudulent cases, as a result, the outliers were kept in the training data.

Balancing Data: Random Undersampling, SMOTE

When training a machine learning model on an imbalanced dataset with limited instances of the fraudulent class, the resulting performance may be inadequate. To address this issue, naive Random Undersampling is employed, which randomly deletes entries from the majority (non-fraudulent) class (Amy, 2022). The drawback of this method is that the potential information loss may lead to decreased prediction performance. Another method was tested - Synthetic Minority Oversampling Technique (SMOTE), which up-samples the minority class (Amy, 2022). It does so by generating a subset of new synthetic examples that are close to the other same class points. In the end, the SMOTE dataset indicated overfitting (evidenced by ROC scores of 99%+), therefore, the Random Undersampled data set was used for the rest of the study.

Feature Importance

After computing the correlation matrix for the resulting undersampled dataset, some variables showed higher correlation (highest at 0.51), but not concerning enough to indicate multicollinearity. As the principal components supposedly explain around 90% of data variation, all variables were retained in the dataset, to be used in modelling and predictions. Furthermore, the Gradient Boosting algorithm was used to quantify feature importances, providing insights into each variable's contribution to the model's predictive power.

Data Scaling

The variables 'Amount' and 'Time' have not been scaled and should be to match the rest of the data.

However, even though the data have already been PCA transformed, it is not known if scaling has been employed prior to the transformation. As such, two methods were tested: scaling just the two variables, and scaling the entire dataset. Scaling the whole dataset performed the best in model evaluation, as such this was done. The boxplots and summary of the data from the exploratory data analysis section have displayed outliers, especially in the "Amount" variable, thus, the robust scaler was selected as it ignores the outliers from the calculation of the mean and standard deviation (scikit-learn, n.d.).

Model Fitting & Selection

Undersampled Data Model

A total of 5 classifiers were chosen to be tested on the random undersampled balanced and scaled dataset constructed as described in the previous section: Logistic Regression, Decision Tree, Random Forest, Support Machine and K Nearest Neighbours. The classifiers were trained and evaluated using 5-fold cross validation. Further grid search of hyperparameters for each classifier was conducted to determine which permutation tested achieved the best fit, using ROC as the scoring method.

By plotting the ROC curves for our 5 classifiers, the Logistic Regression and Support vector machines classifiers (SVC) displayed the best results with ROC scores of 0.7955 and 0.7860 respectively. Due to the close proximity of these scores, respective Recall, Precision, F1 and Accuracy scores were computed for each classifier. In the context of fraudulent credit card transactions, the study put greater significance on the Recall score over Precision score, as not identifying a fraudulent transaction would have a greater impact on the client than misidentifying a genuine transaction as fraudulent. The Logistic Regression classifier again performed slightly better than the SVC classifier in regards to the recall scores which are, 0.74 and 0.73 respectively. For this reason, the study proceeded with the Logistic Regression classifier.

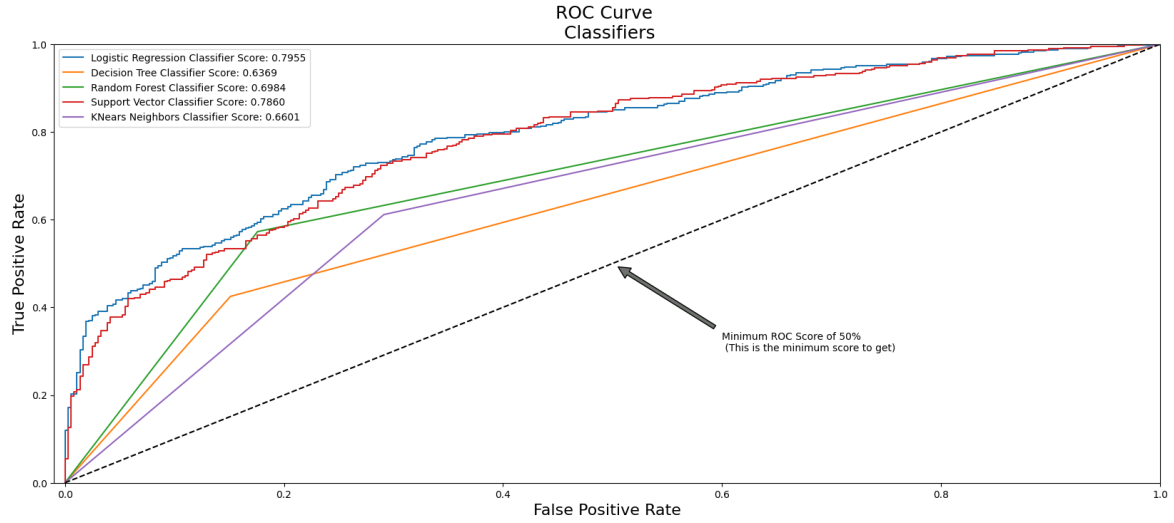


Figure 4: ROC Curves for our 5 classifiers

Predictions for Test Data

The best Logistic Regression model was fitted using scaled random undersampled data with the best hyperparameters found (solver = 'liblinear', C=0.1, penalty='l1'). Then, predictions of probabilities were generated for the test data provided by Kaggle and saved as test_submission_FinalLR.csv.

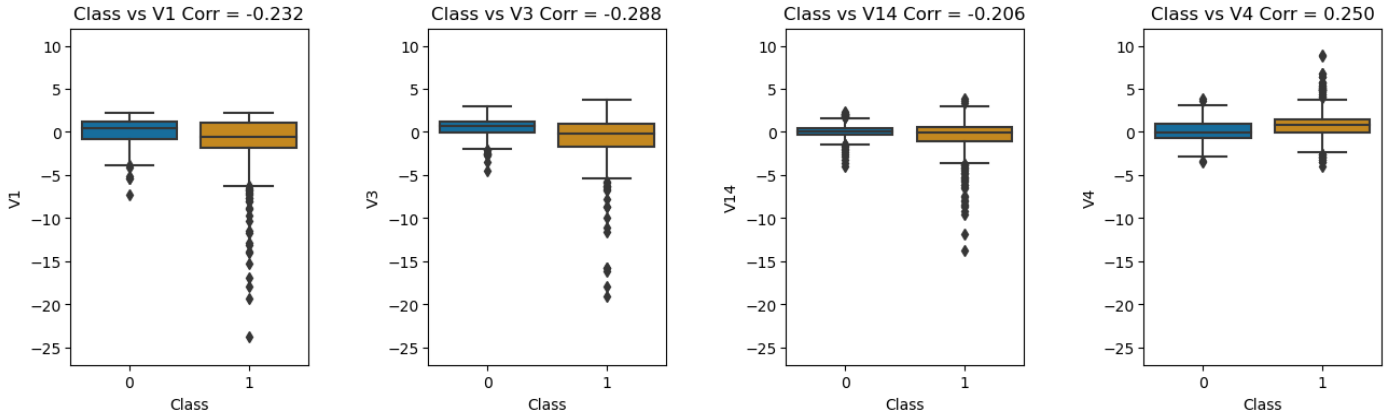
Results

Feature Importance

It is difficult to interpret the covariates as they are principal components and not the original features. Nevertheless, the correlation matrix computed using Random Undersampled Data revealed four important

variables (which happened to be principal components) that exhibited stronger correlations with the "Class" prediction compared to other variables, see **Fig. 5**. The boxplots here differ from genuine class (0) and fraudulent class (1) in each case, especially in the outliers. This is not surprising, as the outliers in the data have defined the fraudulent cases, as discussed in the *Outliers* section in *Methodology*. However, correlation only captures linear relationships between the variables.

To expand on the feature importance results Gradient Boosting algorithm was used. This algorithm aims to identify feature importance by generating a series of decision trees and combining them to make a good classifier. Feature importance scores reflect the value of each feature in constructing the model's boosted decision trees, with higher scores indicating greater importance of attributes used to make key



decisions (Brownlee, 2020). The top four features are displayed in **Fig. 6**.

Figure 5: Principal Components that had High Correlation with the Prediction variable "Class"

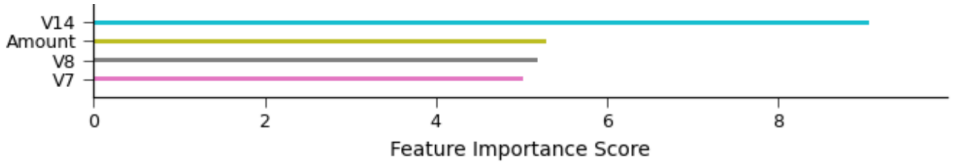


Figure 6: Feature Importances from Gradient Boost Algorithm, Four Top Features

Imputation and Imputation Model

As seen in **Table 1**, the Iterative imputation performed the best, with Mean imputation performing second best. The KNN method performed the worst, it was also the most computationally expensive due to the nature of the process. In order to give the KNN imputation optimal parameters to succeed, various k values were tested, 6 was chosen which minimised Mean Squared Error (MSE), however, the process still performed the worst.

Table 1 – Imputation Method Results

Imputation Method	***RMSE***	***MAE***
Iterative Imputer	2885.80	2415.47
Mean Imputer	2885.86	2415.53
KNN Imputer	3261.15	2654.12

The difference between the Iterative and Mean imputation methods was small, this could justify the sole usage of Mean imputation due to its simplicity, and computational efficiency whilst returning similarly close values to the more complex iterative method. However, if the missing values were not randomly sampled, the Mean imputation method could have been biased and therefore performed worse. Due to the high dimensionality of this data, the iterative approach would be most robust in all scenarios due to the iterative nature of the process and its ability to model non-linear relationships.

Table 2 – Model ROC Scores using Imputed Data

Imputation Method	***ROC***	***ROC Difference from Original (%)***
Iterative Imputer	0.80	0.40
Mean Imputer	0.79	-1.78
KNN Imputer	0.81	0.99

The imputation methods were compared by difference in ROC scores when using a Logistic Regression model. The Iterative Imputer once again performed the best (see **Table 2**), having the lowest percentage difference from the original train data ROC. This metric KNN seemed to perform better, this may be due to it assigning closer values for each class, which would not show up in the previous two metrics used. Overall, considering the results and computational cost the Iterative Imputer was the preferred method.

Undersampled Data Confusion Matrices

For the undersampled dataset that has been split into training and testing sets, the predictive power of classifiers was evaluated by plotting the confusion matrices on this test set (see **Fig. 7**).

Confusion matrices show the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by a classifier, allowing for a better understanding of its performance. Additionally, the desired confusion matrix is plotted to see how the classifiers compare. The best 2 classifiers, Logistic Regression and SVC, perform the best on this testing set in regards to accurately identifying true positive fraudulent transactions. Out of the 83 latent fraudulent transactions, these two classifiers were able to successfully identify 56 of them (67.5%). Although other classifiers have better performance in identifying true genuine transactions, this comes with the trade-off of lessened accuracy predicting fraudulent transactions. In the context of this study (detecting credit card fraud), the priority is to accurately identify fraudulent transactions, as the consequences of not identifying fraud are more significant than misidentifying genuine transactions.

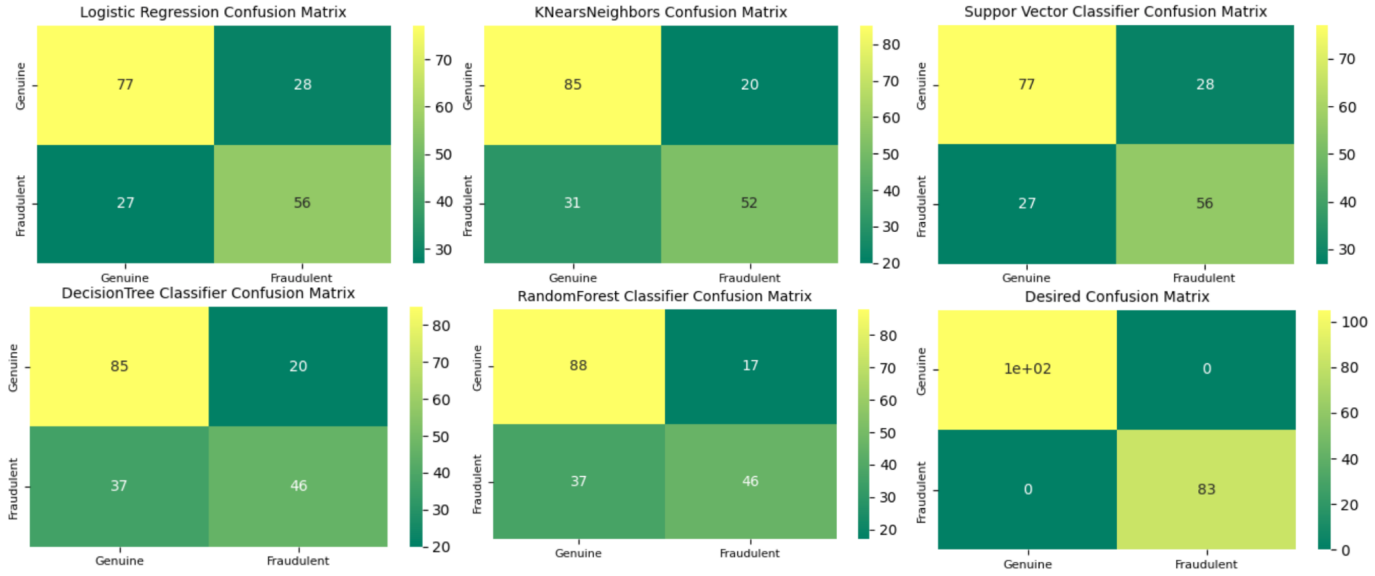


Figure 7: Confusion Matrices of 5 classifiers on the Undersampled Training Set

Test Data

The best performing Logistic Regression model (with best hyperparameters) using scaled random undersampled data has given the scores seen in **Fig. 8**. The same was done using Imputed Data using Iterative method, results seen in **Fig. 9**.

Figure 8: Predictions on Test Data using LR Model with Undersampled Data

Figure 9: Predictions on Test Data Using LR Model with Imputed Data

Discussion

The explanatory variable correlation with the target variable "Class" indicates that: in case of high positive correlation - feature tends to increase the likelihood of the transaction being fraudulent; high negative correlation - feature tends to decrease the likelihood of a transaction being fraudulent. Recommendation for the client is to look into negative principal components 1, 3, 14 and positive ones 4, 19, and explore the underlying composition using the original unclassified data if possible (PennState Eberly College of Science, n.d.). However, the Gradient Boosting algorithm, which identifies the non-linear relationships between the features and the target variable, recognized variables V14, “Amount”, V8, V7 as the most important (negative or positive direction is not defined). Interpretation of the components will be based on finding which variables are strongly correlated with each component (which of the numbers are large in absolute value). Essentially, these important variables should be prioritised for any further data collection, any explanation of the process, f.e. information on data scaling would be beneficial.

As the data was imbalanced, a further suggestion would be to explore a combination of undersampling and oversampling techniques, as this method might outperform the two methods evaluated in this study (Brownlee, 2021). Further combinations of model hyperparameters could be explored to yield better results.

References

1. Tayyip Saka, A. (2019, September 12). *Practical Guide to Outlier Detection Methods*. Towards Data Science. Retrieved April 6, 2023, from https://towardsdatascience.com/practical-guide-to-outlier-detection-methods-6b9f947a161e
2. Amy. (2022, February 19). *Four Oversampling and Under-Sampling Methods for Imbalanced Classification Using Python*. Towards Data Science. Retrieved April 6, 2023, from https://medium.com/grabngoinfo/four-oversampling-and-under-sampling-methods-for-imbalanced-classification-using-python-7304aedf9037
3. scikit-learn. (n.d.). *sklearn.preprocessing.RobustScaler*. scikit-learn. Retrieved April 6, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html
4. PennState Eberly College of Science. (n.d.). *11.4 - Interpretation of the Principal Components*. stat.psu. Retrieved April 6, 2023, from https://online.stat.psu.edu/stat505/lesson/11/11.4
5. Brownlee, J. (2020, August 27). *Feature Importance and Feature Selection With XGBoost in Python*. Machine Learning Mastery. Retrieved April 6, 2023, from https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/
6. Brownlee, J. (2021, May 11). *How to Combine Oversampling and Undersampling for Imbalanced Classification*. Machine Learning Mastery. Retrieved April 7, 2023, from https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/