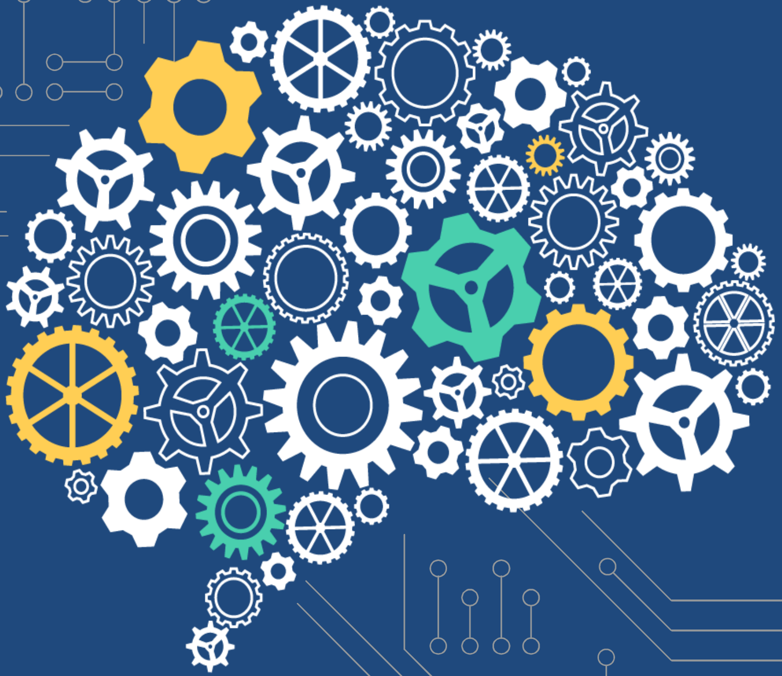


Credit Card Fraud Detection Using Machine Learning Algorithms

Project Presentation

Homayoun Gerami

Agenda:

- **Introduction**
- **Data Overview**
- **Exploratory Data Analysis (EDA)**
- **Data Engineering:**
  - The amount feature transformation
  - Final Dataset for the ML modeling
- **Models Building and Testing:**
  - Model Performance Measures for Imbalanced Data
  - Models Performance Comparison
- **Observations & Conclusion**

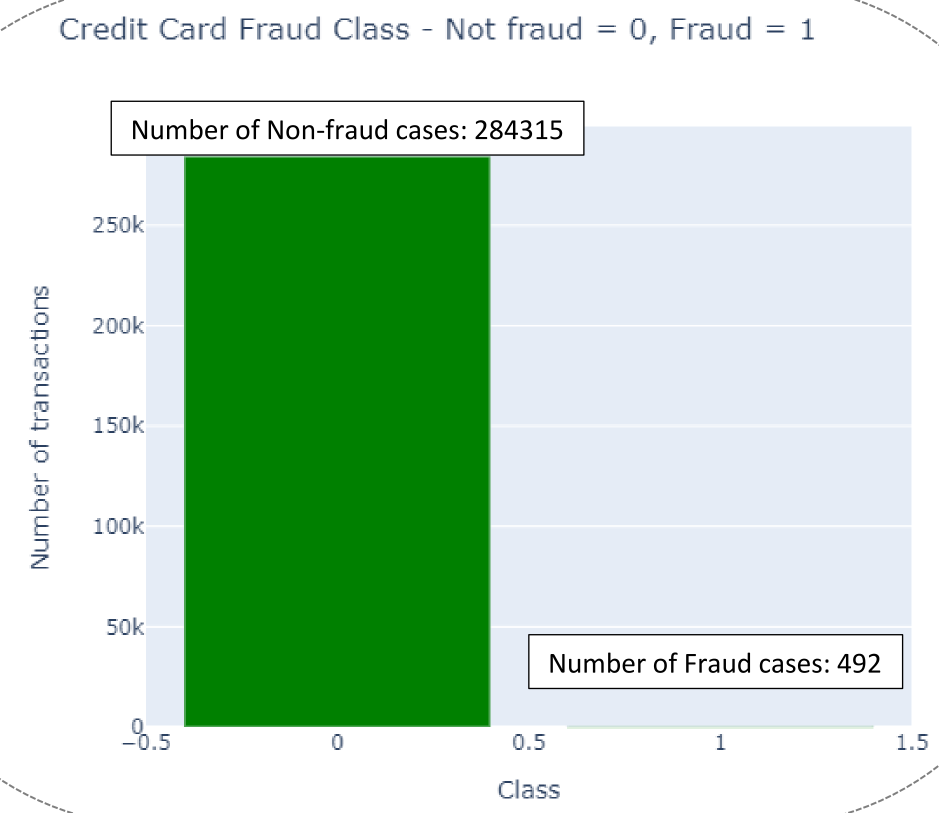
Introduction

- Credit cards are used to a large extent all over the world, and in our day to day transactions. While credit card usage is constantly increasing, online fraud transactions is also a developing threat and detection of such frauds is a serious matter.
- The credit card fraud detection is a challenging task, since the fraudulent users actively attempt to resemble the legitimate ones.
- In this study, three machine learning (ML) methods were used to accurately classify credit card transactions as fraudulent or not.

Data Overview:

- The credit card data comes from Kaggle (https://www.kaggle.com/mlg-ulb/creditcardfraud) and contains transactions made by European credit cardholders in two days of September 2013.
- The dataset contains 30 features, with two features Time and Amount in their original format.
- The Time variable represents the time in seconds since the first transaction over the 48 hour observation period.
- The remaining 28 features are numerical and were derived from a principal component analysis (PCA). This was done to protect user identities and sensitive features.

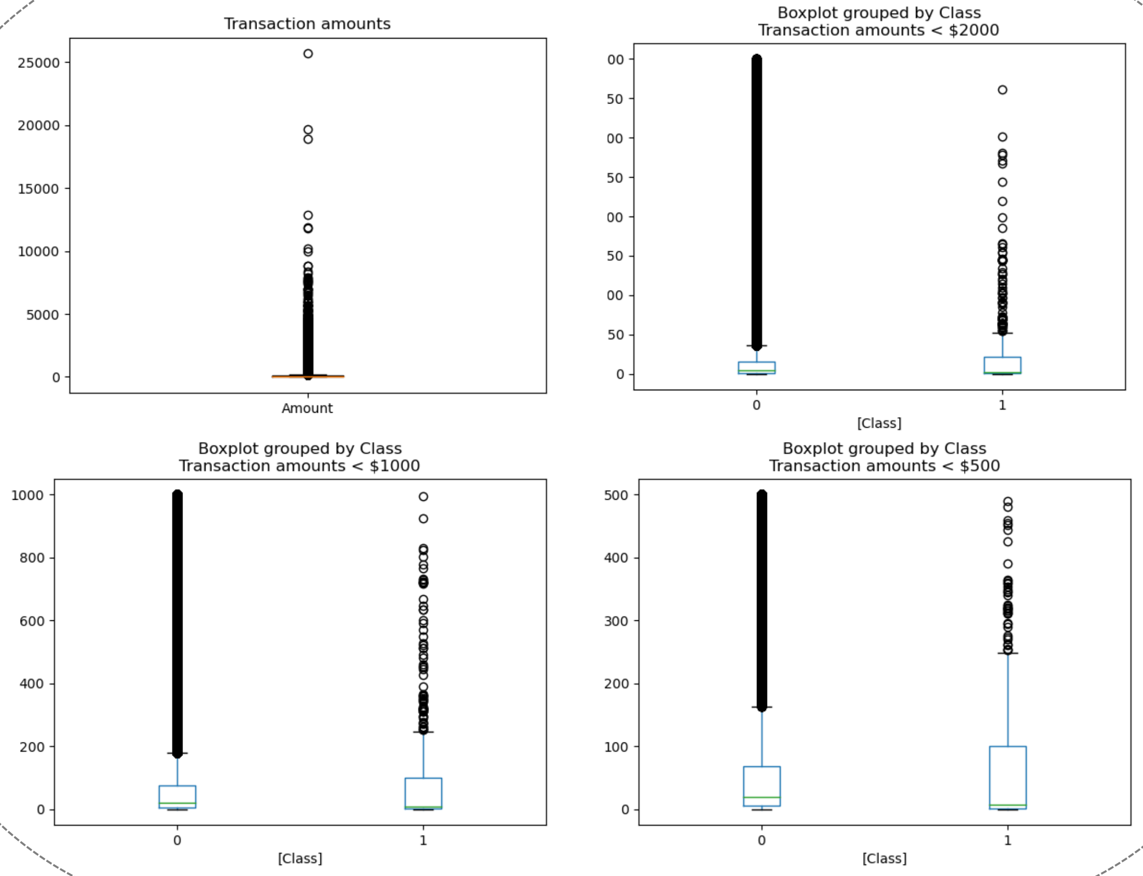
EDA, Fraud and Non-Fraud cases in the dataset:



The dataset is highly imbalanced, with the Minority cases accounting for only 0.17% of all transactions, and that is the main reason doing under-sampling for further model designs.

EDA, Distribution of transaction amounts:

- Class 0 : Non Fraud, Class 1 : Fraud


Transaction amounts

Boxplot grouped by Class
Transaction amounts < $2000

Boxplot grouped by Class
Transaction amounts < $1000

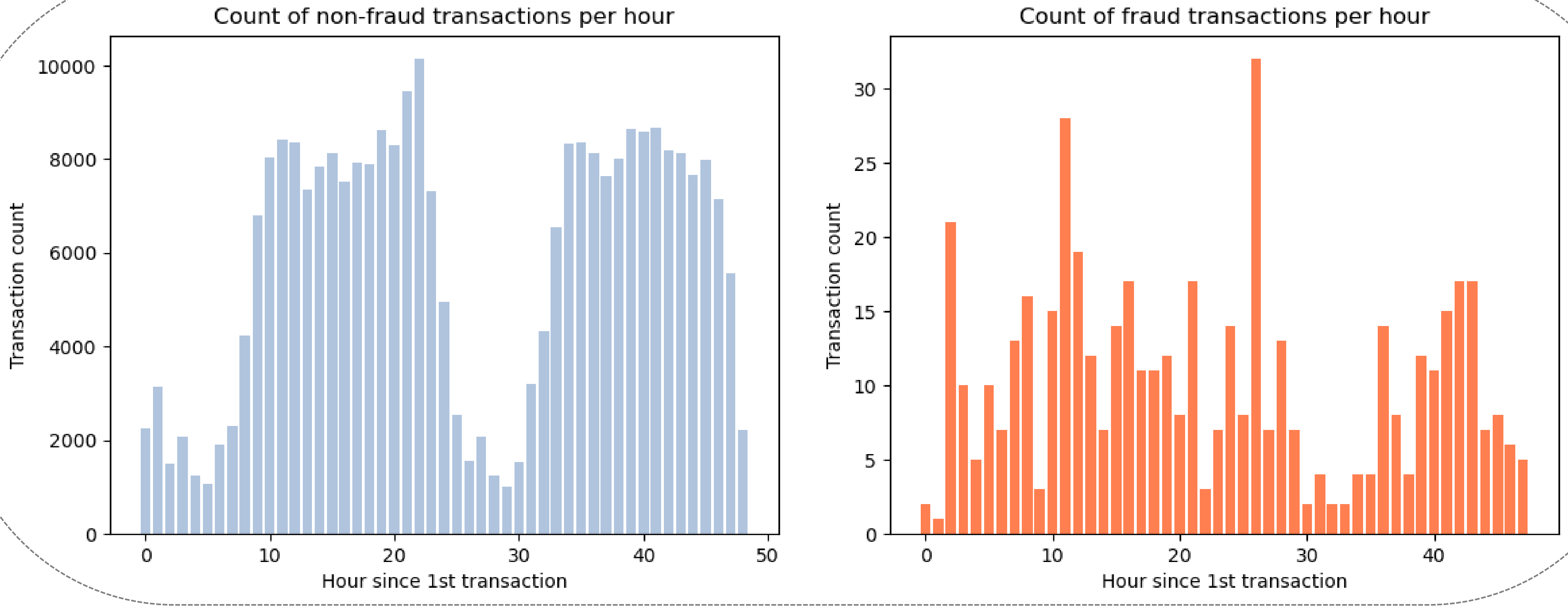
Boxplot grouped by Class
Transaction amounts < $500

- 3 Non-Fraud with amount greater than $1500 (top left)
- all of the fraud transactions smaller than $2000 (top right panel), with most of the transactions below $100.
- Fraudulent transactions had a higher Q3 than non-fraudulent transactions (bottom right).

EDA, Density plot of transaction amounts over time

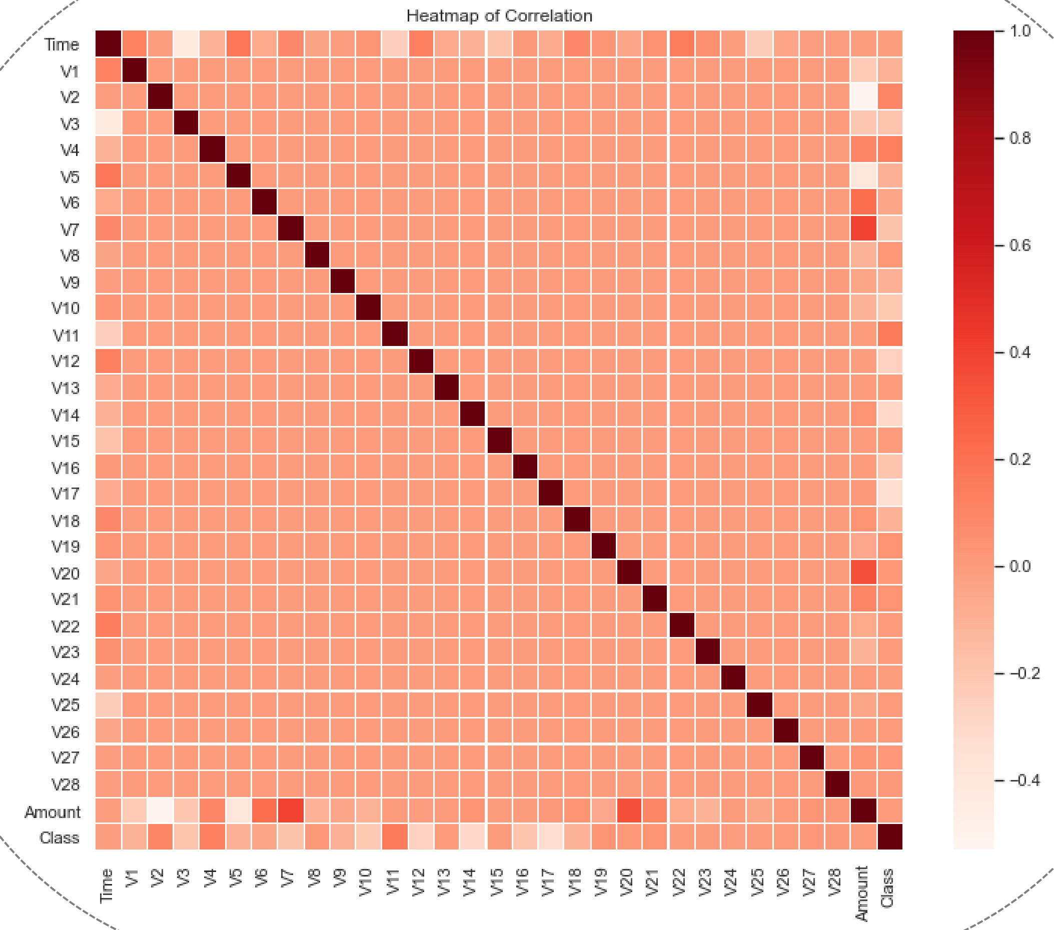
Left: Non-Fraud, Right: Fraud

The x-axis represents the hour since the 1st transaction



- The results for the fraudulent transactions (left panel) show two clear bi-modal distributions, which most likely corresponds with daytime transactions for each of the two days.
- For the fraudulent transactions, we do not see a clear pattern or distribution in the number of transaction counts. The highest number of fraudulent transactions was 32, which occurred in the 28th hour.

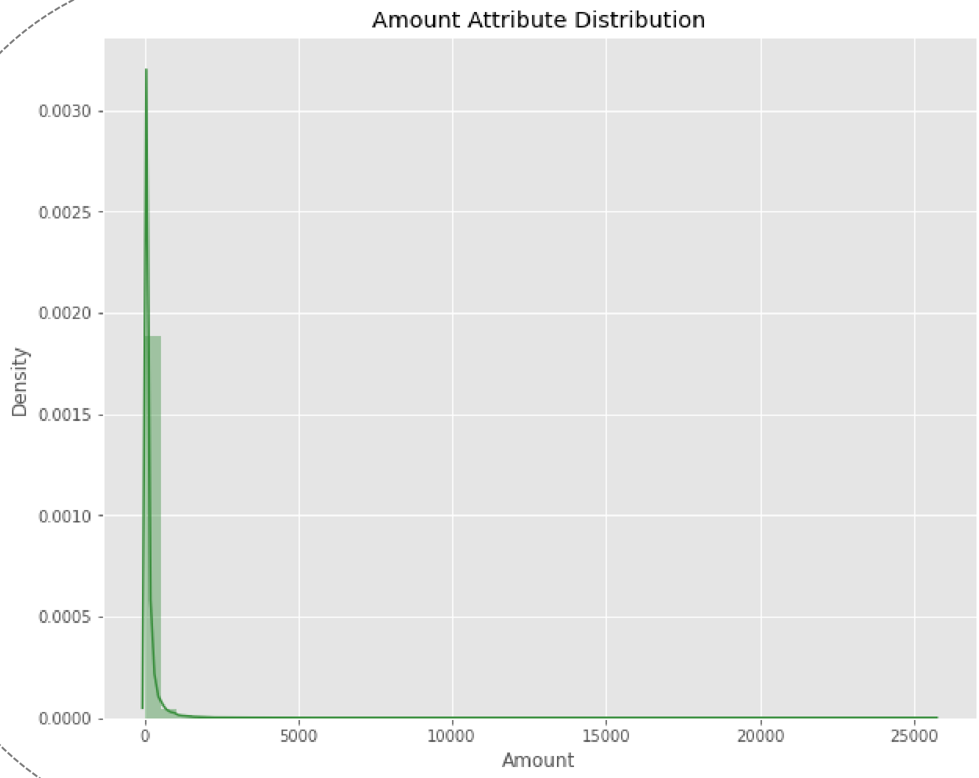
EDA, Correlation Analysis

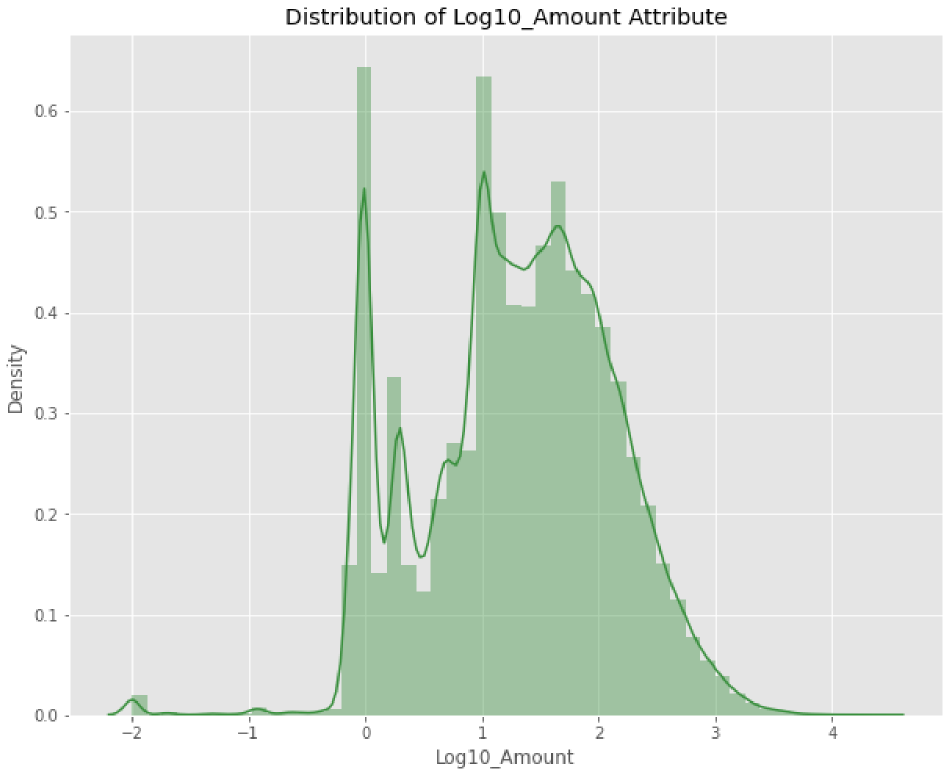


- Features with high negative correlation (<0.5) with response variable: V12, V14, V17

- Features with high positive correlation (>0.5) with response variable: V2, V4, V11

Data Engineering , Amount variable transformation





Because the range of values in the Amount feature was wide ($1{$25; 000), we transformed this feature using log base 10.

Data Engineering , Final Data Input to the models:

- The best results was obtained using all principal components in the final dataset (V1, V2, …, V28)
- Because the range of values in the Amount feature was wide ($1-25000), I transformed this feature using log base 10.
- I dropped the original Amount feature and included the log 10 version in the final dataset.
- I also decided to drop the Time since first transaction variable because it was not informative

Data Modeling Approach:

- To classify credit card fraud, we evaluated the accuracy of three ML models: Logistic Regression, Support Vector Machine, and Random Forest Regression
- Repeated Stratified KFold sampling with k = 5 and 3 repeats to split the data into training and testing datasets. For the Kth training set, we did synthetic random sampling to balance the data using the RandomUnderSampler method from the imblearn python library.
- To better assess the effect of under-sampling, we under-sampled 15%, 10%, and 2% of the majority cases. It is important to note, that the sampling is only performed on the training dataset, the dataset used by an algorithm to learn a model. It is not performed on the holdout test or validation dataset. The reason is to evaluate the resulting model on data that is both real and representative of the target problem domain.
- After balancing the training datasets, we then used the trained models on the original (unbalanced) test datasets. To evaluate the models, we considered several metrics that are appropriate for imbalanced data and that emphasize the accuracy of predictions in the Minority class.

Model Performance Evaluation Metric for Imbalanced Datasets

PR AUC versus ROC AUC

- ROC AUC is the area under the curve where x is false positive rate (FPR), or fallout, and y is true positive rate (TPR), or recall.
- PR AUC is the area under the curve where x is TPR, or recall, and y is precision.
- For the imbalanced datasets, we should avoid using ROC AUC measure and instead use PR AUC Often, the Accuracy or ROC-AUC score is used to measure the performance of ML algorithms. But the Accuracy score can be a poor measure for imbalanced classifications, because the larger number of Majority cases can easily overwhelm the number of cases in the Minority class.

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

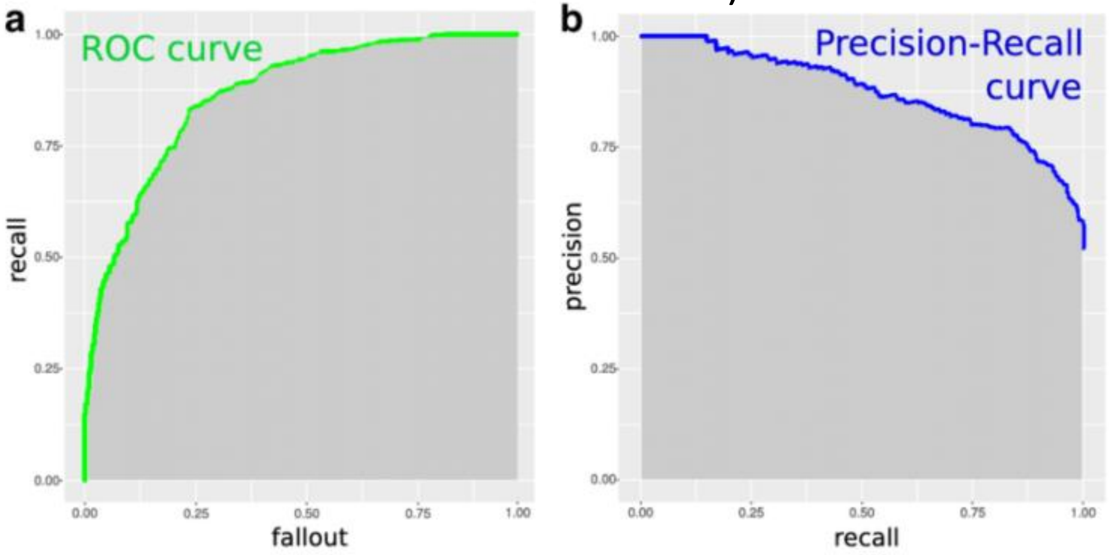
$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

a) ROC AUC

b) PR AUC



Model Performance evaluation Metric for Imbalanced Datasets

Precision Recall and F1 Scores

- The Precision, Recall, F1 scores, and AUPRC are recommended as alternatives to the Accuracy and AUC-ROC scores for imbalanced data.
- These metrics are important because they calculate the accuracy for the Minority class, that is, the ability
- of the ML models to correctly classify the small number of fraudulent transactions in the dataset.

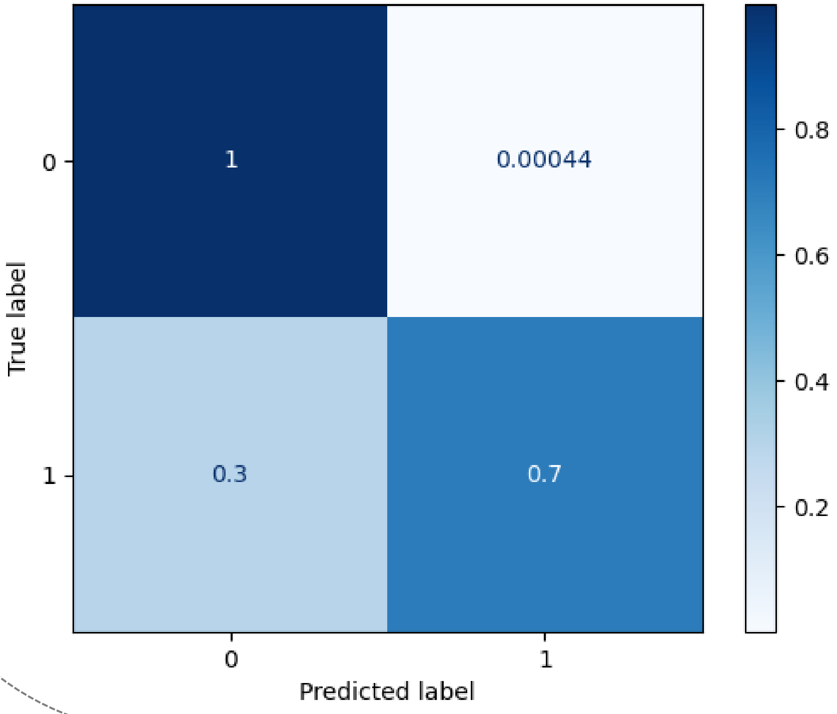
$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

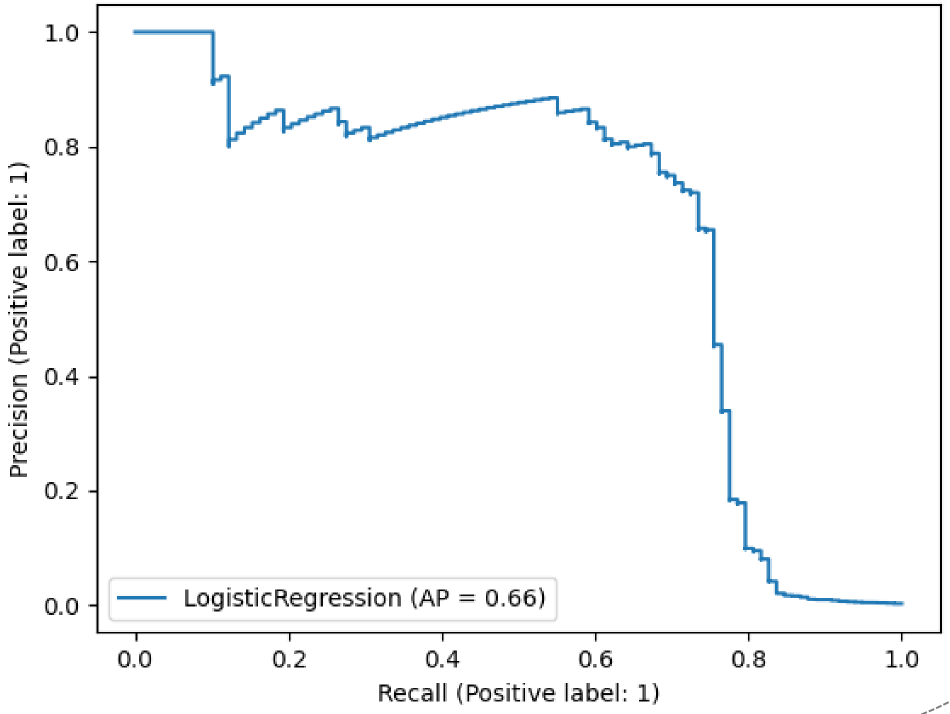
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}.$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

$$\boldsymbol{F_\beta} = \frac{(\mathbf{1} + \boldsymbol{\beta}^{\mathbf{2}}) \cdot (\textbf{precision} \cdot \textbf{recall})}{(\boldsymbol{\beta}^{\mathbf{2}} \cdot \textbf{precision} + \textbf{recall})}$$

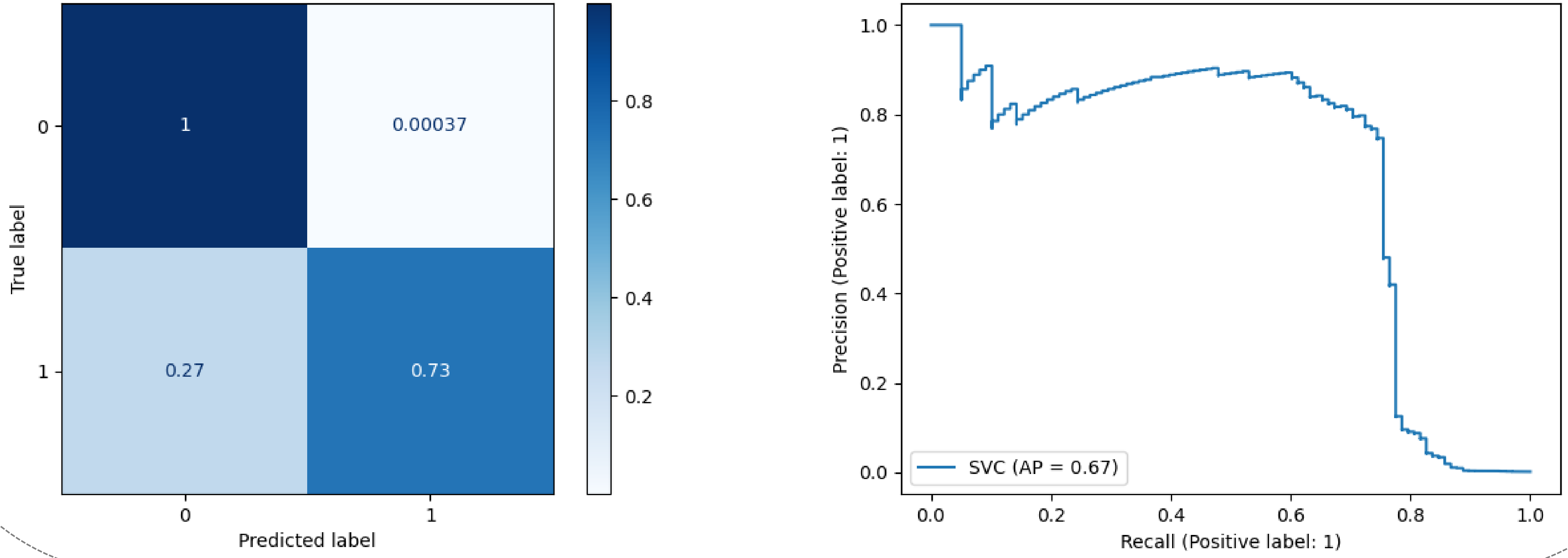
Logistic Regression Model, 2% under-sampling:





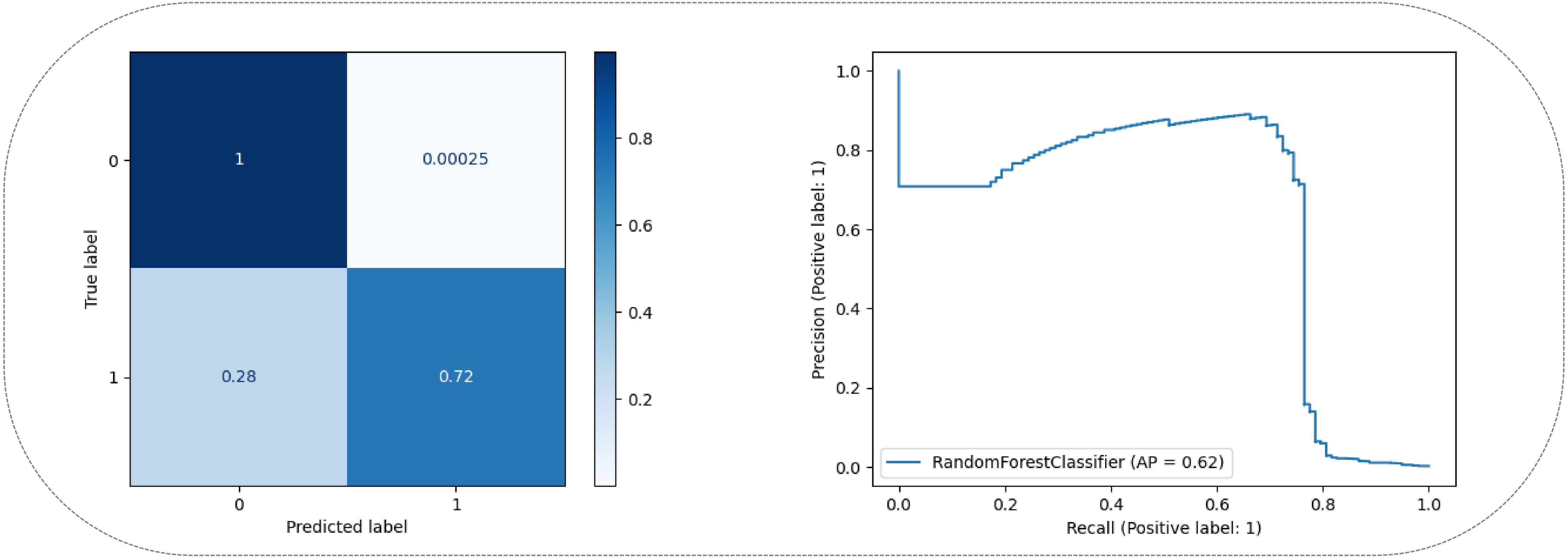
Note that these results are from a single fold rather than an average over all 5 folds.

Support Vector Machine Model, 2% under-sampling:



Note that these results are from a single fold rather than an average over all 5 folds.

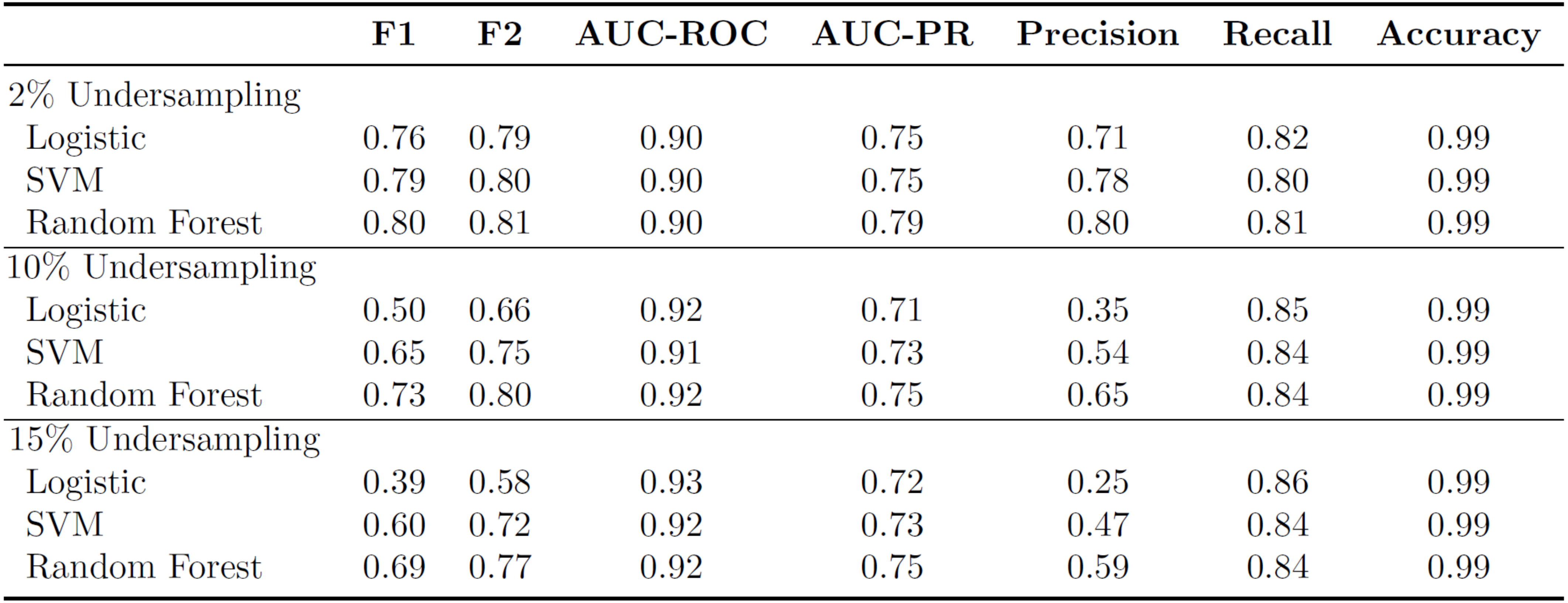
Random Forrest Model, 2% under-sampling:



Note that these results are from a single fold rather than an average over all 5 folds.

Performance of the three ML algorithms

2%, 10% and 15% under-sampling



	F1	F2	AUC-ROC	AUC-PR	Precision	Recall	Accuracy
2% Undersampling							
Logistic	0.76	0.79	0.90	0.75	0.71	0.82	0.99
SVM	0.79	0.80	0.90	0.75	0.78	0.80	0.99
Random Forest	0.80	0.81	0.90	0.79	0.80	0.81	0.99
10% Undersampling							
Logistic	0.50	0.66	0.92	0.71	0.35	0.85	0.99
SVM	0.65	0.75	0.91	0.73	0.54	0.84	0.99
Random Forest	0.73	0.80	0.92	0.75	0.65	0.84	0.99
15% Undersampling							
Logistic	0.39	0.58	0.93	0.72	0.25	0.86	0.99
SVM	0.60	0.72	0.92	0.73	0.47	0.84	0.99
Random Forest	0.69	0.77	0.92	0.75	0.59	0.84	0.99

Observations and Conclusions:

- The three ML methods that were evaluated for Fraud Detection in this project were Logistic Regression, Support Vector Machine, and Random Forest classier
- The Amount variable was transformed using base log 10 because of the wide range of values. The Time since first transaction variable was dropped because it was not informative.
- The Accuracy or ROC-AUC score often is used to measure the performance of ML model. But the Accuracy score can be a poor measure for imbalanced classifications, because the larger number of Majority cases can easily overwhelm the number of cases in the Minority class, such that unskilled model can achieve Accuracy scores of 99% or higher.
- The Precision, Recall, and F1 scores are recommended as alternatives to the Accuracy and AUC-ROC scores for imbalanced data.
- The results show a relatively good performance for the Random Forest classier, with F2 scores of 0.81, 0.80, and 0.77 across the 2%, 10%, and 15% under-sampled datasets, respectively. SVM was the next best classier based on the F2 scores. Overall, Precision was good for the 2% under-sampled dataset, but got worse with higher under-sampling.

References:

- Patrick Whatman, Credit card statistics 2021: 65+ facts for Europe, UK, and US. https://blog.spendesk.com/en/credit-card-statistics
- A Delmaire and J Pointon, Credit card fraud and detection techniques : a review Credit card fraud and detection techniques, Technical Report 2, 2009.
- Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. Credit card fraud detection-machine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), pages 1-5. IEEE.
- Raphael Pierre, Detecting Financial Fraud Using Machine Learning: Winning the War Against Imbalanced Data. https://rpubs.com/chidungkt/448728
- Jason Brownlee, Random Oversampling and Undersampling for Imbalanced Classification. https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset
- Davide Chicoo, Ten quick tips for machine learning in computational biology https://www.researchgate.net/figure/a-Example-of-Precision-Recall-curve-with-the-precision-score-on-the-y-axis-and-the_fig1_321672019