Credit Card Fraud Detection

MATH 509 Final Project Report

https://github.com/Saira55/509-project

Saira Faiz, Harshilkumar Pareshbhai Kansara, & Colby Jamieson

12/8/2022

Introduction

Credit card fraud is a pervasive problem, affecting 24 per cent of Canadians at some point in their life (Forunly, 2022). The ways in which fraudsters gain access to an individual's credit card information is quickly becoming more sophisticated and difficult to avoid. Credit card fraud can occur when someone:

- Gains credit card information through phishing emails or telephone calls
- Hacks a website containing credit card information
- Find physical credit card information
- Steals someone else's identity

To mitigate the growing threat of credit card fraud, we use statistical modeling to predict whether a transaction is likely to be fraudulent, or non-fraudulent, so these transactions can be flagged and investigated before they are processed. Various types of models are used, their prediction power compared, and then a final recommendation is made.

Problem

Individuals can protect themselves by becoming more educated and checking their banking records regularly; however, on a large scale it may not be realistic to expect the public to be this diligent. Furthermore, relying on credit card holders to check their balances and report suspicious charges is cumbersome.

Statistical models can help predict which transactions are most likely to be fraudulent, and potentially provide credit providers a way to stop these transactions from being processed. In theory, a credit card transaction should have key characteristics that hint at its classification of either legitimate, or fraudulent. A statistical model would have to pick up on these characteristics, and make accurate predictions to be considered a useful, preventative tool.

Data

The data used in this project (Appendix A) is from Kaggle, a website where professionals and hobbyists compete in data science related challenges. The dataset consists of 284,807 credit card transactions with 31 columns per transaction.

Of the 284,807 entries, only 492 are fraudulent, making up 0.17% of the dataset. This proportion may reflect the real-world prevalence of fraudulent credit card transactions; however, a more balanced dataset is preferable to train the model. If the data remains unbalanced, the model may tend to always predict the majority class (Brownlee, 2020). To balance the data, samples were taken from the original data merged into a single dataset, resulting in an even split of fraudulent to non-fraudulent transactions. Once the model is trained, it will be tested on the original dataset, which is closer to the type of data that would be encountered if the model was deployed.

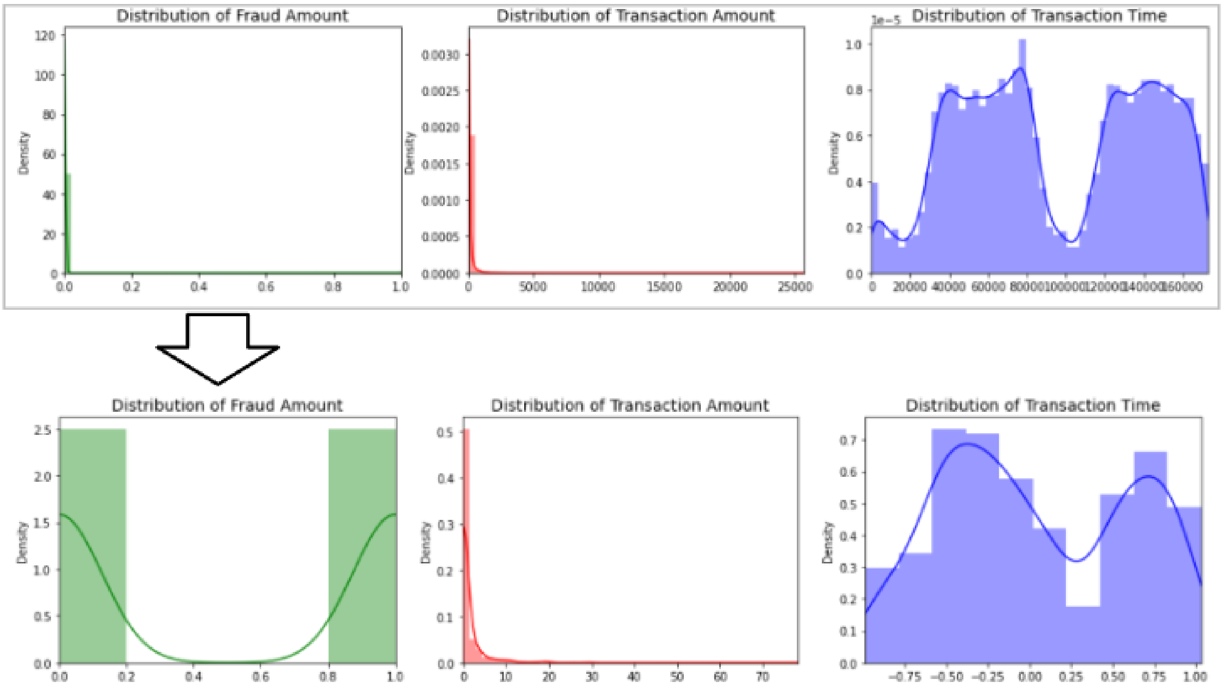


Figure 1: Original unbalanced dataset is transformed to a balanced distribution of fraudulent to non-fraudulent transactions while distributions of transaction amount and transaction time remain similar to original data.

The dataset also had a few remaining columns that were unscaled. Scaled data ensures that the values in each respective column is in a similar range to other columns. Making sure data is scaled helps many statistical models learn and recognize patterns in the data (Verma). To prepare the dataset further, three additional columns were scaled using a robust scaler[1].

The last step in preparing the data was to transform the distributions of each column. In the original dataset, the distribution of the data in each column is skewed. Using a power transformer[2] the data was transformed to resemble a normal distribution. Once these data are transformed, they are more symetric, and should work better with a wide variety of models (Malato).

[1] Robust scaler from the Sci-Kit Learn python library scales data using a quantile range while removing the median (“sklearn.preprocessing.RobustScaler — scikit-learn 1.1.3 documentation”).

[2] Power transformer from the Sci-Kit Learn python library transforms data to that it more closely resembles a normal distribution (“sklearn.preprocessing.PowerTransformer — scikit-learn 1.1.3 documentation”).

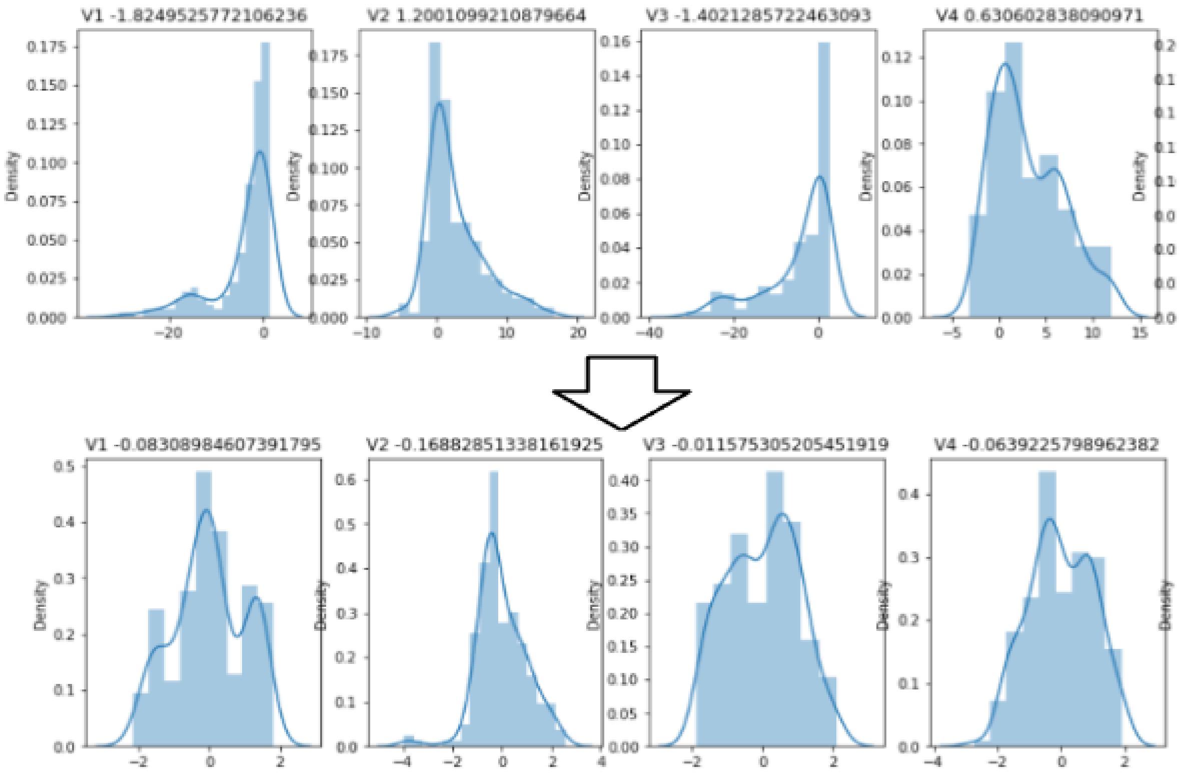


Figure 2: A subset of original column data is transformed from being skewed to resembling a normal distribution.

With the data now balanced, scaled, and symmetric (Appendix B), it is ready to use for training the selected models.

Model Formulation

Variables & Parameters

The 31 columns in the data set contain three explicitly named parameters:

- Time of transaction
- Amount of transaction
- Transaction class (legitimate or fraudulent)

The 28 additional columns have been given arbitrary titles in order to limit the chance of a contributor being able to be identified.

The data has been preprocessed by the uploader to ensure there are no null or erroneous values. This makes using the data for statistical modeling a lot less complex. The columns have also been reduced from a larger set using a process called principal component analysis (PCA)[3].

Assumptions

To use the data provided to make meaningful predictions, some assumptions need to be made. Firstly, the fundamental assumption for the model to work is that fraudulent credit card transactions have distinct patterns discernible by a statistical model.

Secondly, since the data is a reduced form of a dataset with more columns, the model relies on the effectiveness of the PCA methodology to preserve the informational integrity of the original dataset. If too much information is lost in the process of dimension reduction, the model may make bad predictions or be unable to make useful predictions on transaction data outside of our test data.

Restrictions

Since most of the data has been reduced and renamed, feature analysis is somewhat limited. If columns kept their original name, it may be possible to make further reasonable assumptions about the relationship between different independent variables, which could improve the model.

Problem Solution

Methodology

Five models were trained on the balanced dataset and tested on the original data:

- Artificial Neural Network (ANN)
- XGBoost
- Logistic Regression with Markov Chain Monte Carlo (MCMC)
- Random Forest
- Decision Tree

Each model was fitted to the data and tested on their predictive power on the test data. To compare the models, five scores were used:

- Receiver operating characteristic curve (ROC)
- F1 score
- Specificity
- Sensitivity
- Accuracy

[3] Principal component analysis reduces a dataset to be more manageable while preserving informational integrity (Jaadi, 2022).

Interpretation

Summary of Findings

All but one model made reasonably useful predictions as to which transactions were fraudulent. The MCMC model had the highest accuracy rating on the test data at 0.98, meaning that 98% of the predictions made from the model were correct. The ANN model fared the worst, with an accuracy score of 0.55.

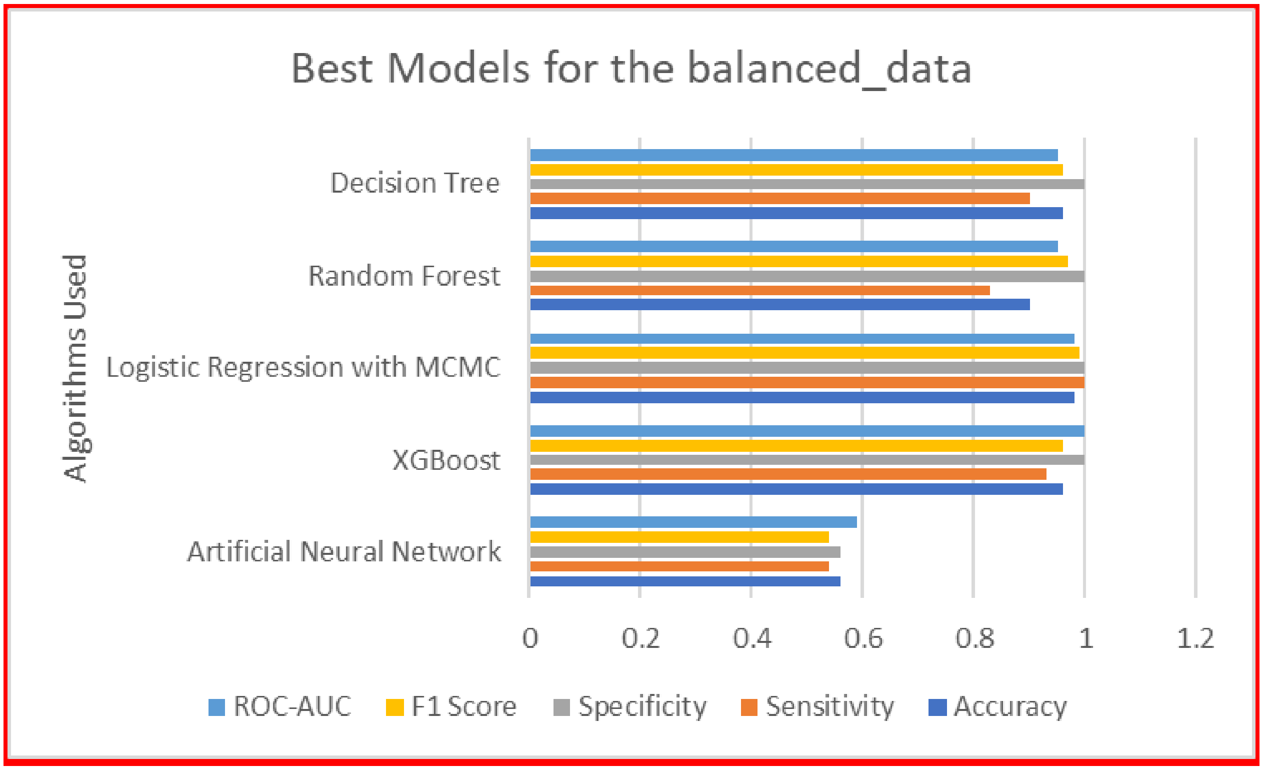


Figure 3: All models except the ANN performed well on various measures.

The ROC score is a measure of the model’s precision and recall scores. Precision is how good the model is at predicting a specific category. Recall tells you how many times the model was able to detect a specific category. For example, high precision means that the model is good at identifying fraudulent transactions relative to falsely labeling them as legitimate. Recall refers to the model’s ability to correctly predict fraudulent transactions relative to the number of legitimate transactions labelled fraudulent.

The precision-recall tradeoff is a very challenging problem we have to solve when working with imbalanced data and some use cases should prioritize precision while others should prioritize recall, there is no universal right or wrong answer. In our use case of fraudulent credit card transactions we give a little more weight to precision than most models do in this case because the consequences of false positives are understated.

When scaling a model to real-world applications, the ROC score must be monitored to ensure performance is generally maintained. It may be that the more computationally expensive models perform better at scale in this regard.

Model Critique

Limitations

Although the model was trained on a relatively large dataset, it is still only a tiny fraction of the amount of data that it would need to process in a real world implementation. There may be characteristics of the data set used for model training that are unique to its source.

Future Improvement

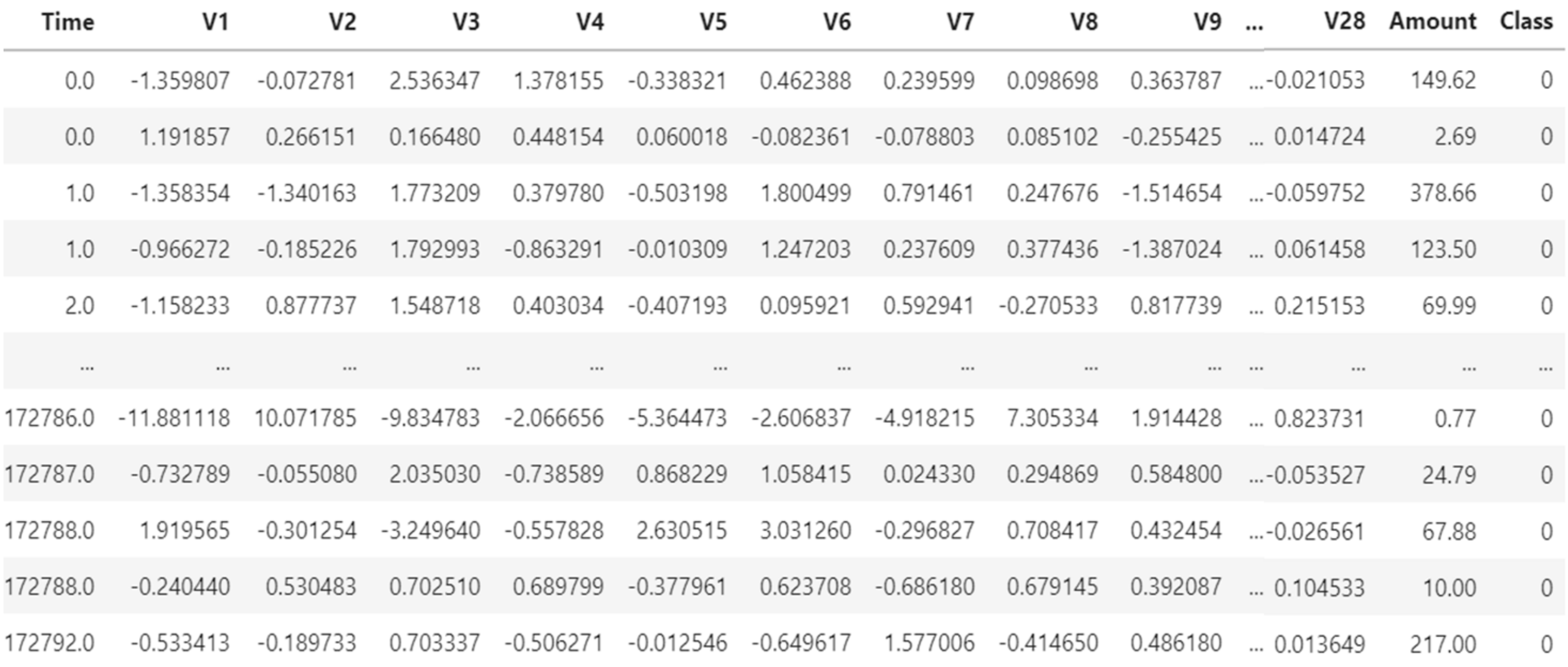
To prepare models for implementation, they should be tested on larger, and more diverse datasets. If performance wanes as it is introduced to more data, a process for continual monitoring and improvement with short evaluation cycles should be developped.

There are various models and methods not used in this report that are also available for classification problems:

- Convolutional Neural Networks
- Recurrent Neural Networks
- K-nearest neighbor
- Naive Bayes
- Support Vector Machines
- Applying MCMC techniques to other algorithms

Future work could experiment with these other techniques and compare their performance.

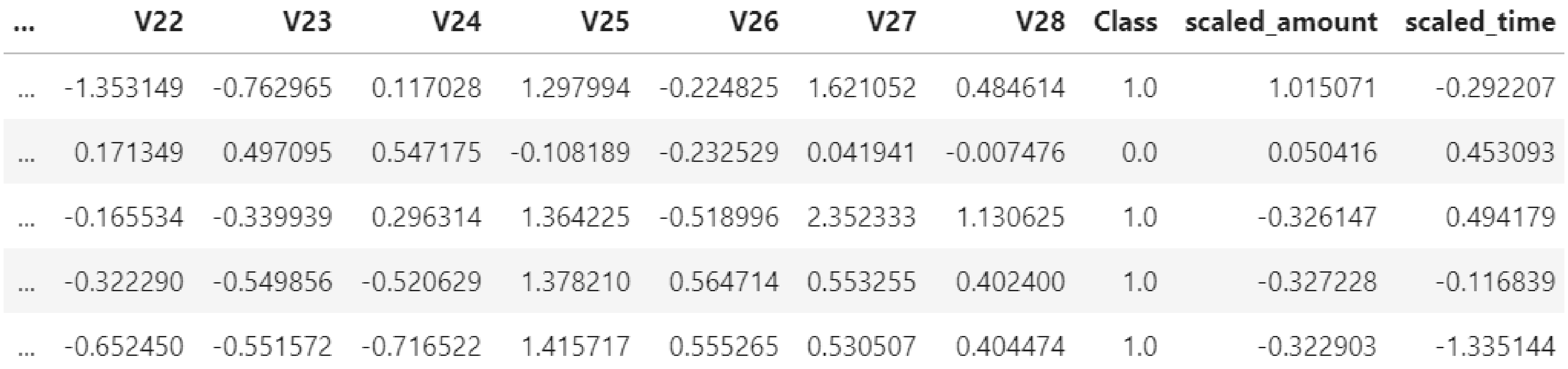
Appendix A: Original Dataset



Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V28	Amount	Class
0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.021053	149.62	0
0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	0.014724	2.69	0
1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	-0.059752	378.66	0
1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	0.061458	123.50	0
2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	0.215153	69.99	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.823731	0.77	0
172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	-0.053527	24.79	0
172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	-0.026561	67.88	0
172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.104533	10.00	0
172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.013649	217.00	0

Figure 4: Original dataset from kaggle.com.

Appendix B: Transformed Dataset



...	V22	V23	V24	V25	V26	V27	V28	Class	scaled_amount	scaled_time
...	-1.353149	-0.762965	0.117028	1.297994	-0.224825	1.621052	0.484614	1.0	1.015071	-0.292207
...	0.171349	0.497095	0.547175	-0.108189	-0.232529	0.041941	-0.007476	0.0	0.050416	0.453093
...	-0.165534	-0.339939	0.296314	1.364225	-0.518996	2.352333	1.130625	1.0	-0.326147	0.494179
...	-0.322290	-0.549856	-0.520629	1.378210	0.564714	0.553255	0.402400	1.0	-0.327228	-0.116839
...	-0.652450	-0.551572	-0.716522	1.415717	0.555265	0.530507	0.404474	1.0	-0.322903	-1.335144

Figure 5: Subset of transformed dataset

Works Cited

Brownlee, Jason. “Why Is Imbalanced Classification Difficult? - MachineLearningMastery.com.” *Machine Learning Mastery*, 17 February 2020, https://machinelearningmastery.com/imbalanced-classification-is-hard/. Accessed 6 Dec 2022.

“15 Worrying Credit Card Fraud Statistics In Canada.” *Forunly*, 23 August 2022, https://fortunly.com/ca/statistics/credit-card-fraud-statistics-canada/#gref. Accessed 6 December 2022.

Jaadi, Zakaria. “Principal Component Analysis (PCA) Explained.” *Built In*, 8 August 2022, https://builtin.com/data-science/step-step-explanation-principal-component-analysis. Accessed 6 December 2022.

Malato, Gianluca. “When and how to use power transform in machine learning.” *Your Data Teacher*, 21 April 2021, https://www.yourdatateacher.com/2021/04/21/when-and-how-to-use-power-transform-in-machine-learning/. Accessed 6 December 2022.

“sklearn.preprocessing.PowerTransformer — scikit-learn 1.1.3 documentation.” *Scikit-learn*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html. Accessed 6 December 2022.

“sklearn.preprocessing.RobustScaler — scikit-learn 1.1.3 documentation.” *Scikit-learn*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html. Accessed 6 December 2022.

Verma, Yugesh. “Why Data Scaling is important in Machine Learning & How to effectively do it.” *Analytics India Magazine*, 29 August 2021, https://analyticsindiamag.com/why-data-scaling-is-important-in-machine-learning-how-to-effectively-do-it/. Accessed 6 December 2022.