Credit Card Fraud: Model Selection and Undersampling

Brian Zschau
Department of Computer & Data Science
Wentworth Institute of Technology
550 Huntington Ave, Boston, MA 02115
contact@zschaub.com

I. Problem Introduction

In 2021 the United States alone lost 32.34 billion dollars in credit card fraud and that number is only expected to get higher every year [1]. Credit card fraud detection is a problem that people face every day without realizing it. Whenever someone makes a transaction online or at the store with a card their bank uses a model to try and predict if that purchase is you or is fraudulent. This process has to be near instantaneous. If there are too many false positives then the customer will have their card decline and the credit card company will lose customers. If there are too many false negatives then the customer loses money and if the credit card company can't regain the loss they have to pay for it.

Credit card fraud detection has many difficult aspects that has to be overcome. [2] First there is a massive amount of data that is generated every day and this data has a lot of features. The second is the data is unbalanced [3] with many more non-fraudulent transactions then fraudulent transactions. Finally the data is nonstationarity [4] because people's spending habits change and the way fraudsters try to defraud people evolve over time.

II. Literature Review

[5] goes into details on why the credit card fraud detection is an interesting problem and how it is unique when implemented. Some of the points it makes is that it is extremely unbalanced with vastly more real transactions then fraudulent transactions. You also gain access to new labeled data every day as people mark transactions fraudulent or not as they use their cards allowing you to evolve your model. The unbalanced nature of credit card data is looked at in [3] and goes over effectively undersample a dataset to train a model better using unbalanced data. [4] goes over more difficulties with credit card fraud focusing the issues of unbalanced, nonstationarity, and the lack of data due to privacy concerns. [6] goes over all of these issues and how to get around them. This is a much more in depth paper then the rest.

There are two types of transactions, face-to-face which is when the buyer buys directly from someone and e-commerce where the buyer does not have to interact with another person. Solutions usually focus on e-commerce transactions for detecting fraud. [7] looks at transfer learning by transferring a model trained on e-commerce transactions and applying it to face-to-face transactions.

Not only is credit card fraud a problem you have to solve almost instantaneously but there is also a lot of data so [8] created SCARFF which is a Spark framework to handle the massive amount of data a credit card company would have to handle and process for credit card fraud processing.

[9] uses a hybrid approach combining supervised and unsupervised learning algorithms to improve their model's performance over using just supervised learning algorithms. The reason they want to be able to use unlabeled data is because of changing customer behaviors and fraudsters' ability to invent novel fraud patterns.

III. Project Details

I will not be using raw credit card data in this project due to the privacy concerns that would bring. The data I will be using is from Kaggle [10] and is a dataset of credit card transactions labeled fraudulent or not. The actual information of the credit cards have been run through Principal Component Analysis (PCA) transformation resulting in 28 features plus amount and time that we can work with. This allows us to work with credit card data without compromising anyone's personal information. The dataset is highly unbalanced, the fraudulent transactions account for 0.172% of all transactions in the dataset.

This project is split into two parts. The goal of part one is to compare the performance of multiple models. I will be comparing the rate of false negatives, the time to get a prediction, and the overall accuracy. The reason we are interested in the number of false negatives is this is the number we want to be zero. False positives are inconvenient to the customer but can be easily fixed but if there is a false negative then money can be lost.

The models I will be comparing are Decision Tree, Logistic Regression, Random Forest, and a Neural Network. I choose these models because they all have their strengths and weaknesses and could all work for this problem.

Part two of the project I will undersample the dataset and run it through the same four models. I will compare the results I get with the undersampled data and see if I can get an less false negatives. Undersampling data can be a useful tool where you remove the majority class in a unbalanced dataset giving

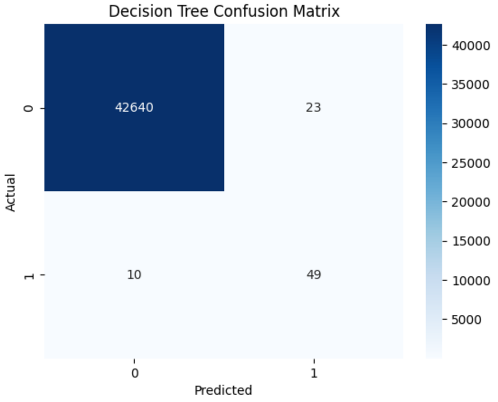

Fig. 1. Decision Tree Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

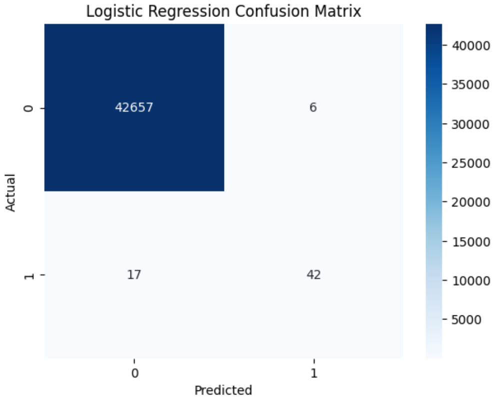

Fig. 2. Logistic Regression Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

you less data to train on but allowing your data to be more balanced.

IV. Results and Evaluation

All results can be found here [11]. I will be using a 70% train, 15% test, and 15% train split for my data. In part 2 the training set will be undersampled. There are a total of 42,722 transactions in the test set. When talking about prediction speed it is the speed to predict all 42,722 transactions not a single transaction.

A. Part One: Model Comparison

1) Decision Tree: The first model we are looking at is the Decision Tree. The decision tree model got us an overall accuracy of 99.92%, a prediction speed of 0.0055 seconds, and 10 false negatives. Fig. 1 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

2) Logistic Regression: The second model we are looking at is Logistic Regression. The logistic regression model got an overall accuracy of 99.95%, prediction time of 0.0040 seconds, and 17 false negatives. Fig. 2 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

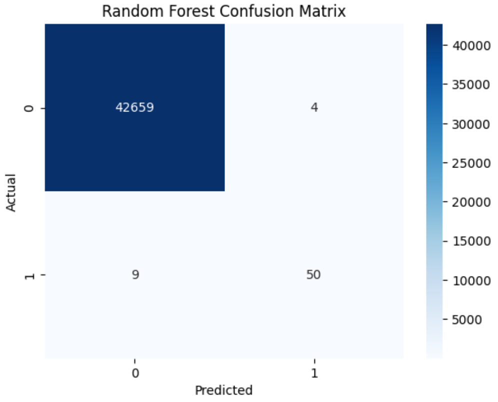

Fig. 3. Random Forest Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

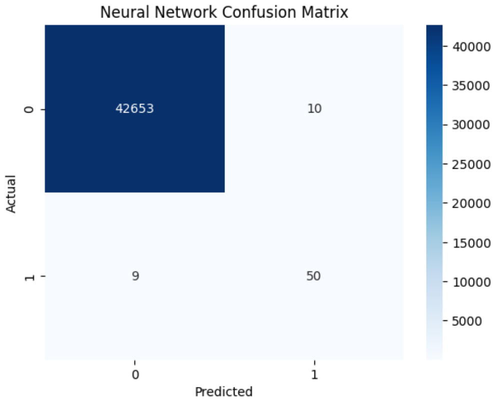

Fig. 4. Neural Network Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

3) Random Forest: The third model we are looking at is Random Forest. The random forest model got an overall accuracy of 99.97%, prediction time of 0.2252 seconds, and 9 false negatives. Fig. 3 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

4) Neural Network: The last model we are looking at is a Neural Network. The neural network model got an overall accuracy of 99.96%, prediction time of 1.1201 seconds, and 9 false negatives. Fig. 4 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

5) Comparing Models: Table I compares the four different models. We can see that decision trees and logistic regression have the fastest prediction times however logistic regressions have the most false negatives out of the four models. In terms

Model	**Accuracy**	**Prediction Time**	**FN**
Decision Tree	99.92%	0.0055 seconds	10
Logistic Regression	99.95%	0.0040 seconds	17
Random Forest	99.97%	0.2252 seconds	9
Neural Network	99.96%	1.1201 seconds	9

TABLE I
Performance Metrics of Different Models. FN = False Negative

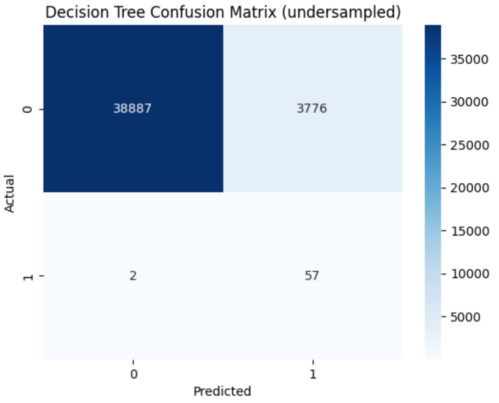

Fig. 5. Decision Tree Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

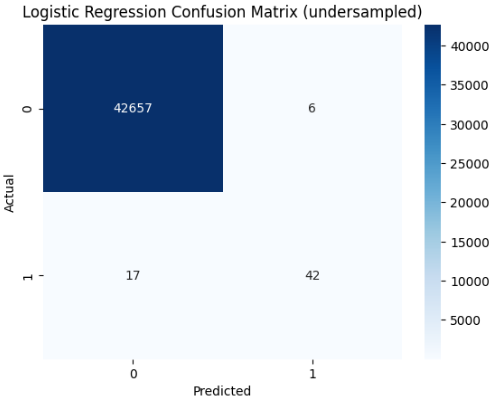

Fig. 6. Logistic Regression Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

of overall accuracy random forest and neural networks have the best accuracy however they are all very close because of how unbalanced the data is. If I was to deploy one of these models I would go with either a decision tree for its prediction time and false negatives or random forest for its accuracy and false negatives.

B. Part Two: Undersampling

We are now going to undersample the data using imbalanced-learn [12] and seeing if we can improve model performance in terms of false negatives and overall accuracy. We will be undersampling the training data but all other data is staying the same.

1) Decision Tree: The decision tree model with undersampled data had an accuracy of 91.16%, prediction time of 0.0060 seconds, and 2 false negatives. There was 8 less false negatives using undersampled data then using the original data. Fig. 5 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

2) Logistic Regression: The logistic regression model with undersampled data had an accuracy of 91.16%, prediction time of 0.0037 seconds, and 17 false negatives. There was no change in the number of false negatives when using undersampled data then using the original data. Fig. 6 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

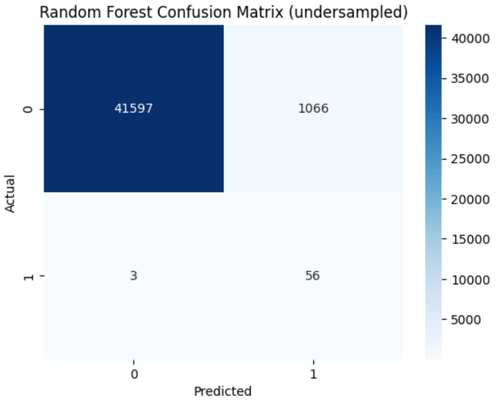

Fig. 7. Random Forest Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

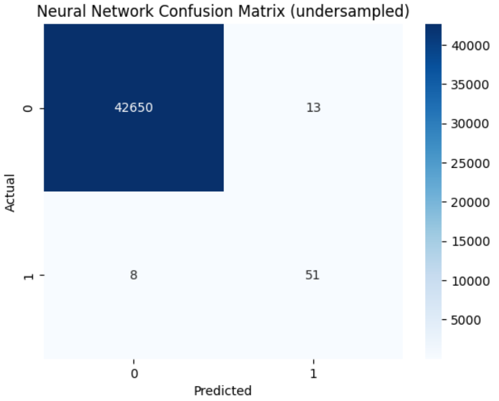

Fig. 8. Neural Network Confusion Matrix, 0 is non-fraudulent 1 is fraudulent

3) Random Forest: The random forest model with undersampled data had an accuracy of 97.50%, prediction time of 0.1842 seconds, and 3 false negatives. There was 6 less false negatives using undersampled data then using the original data. Fig. 7 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

4) Neural Network: The neural network model with undersampled data had an accuracy of 99.95%, prediction time of 1.1566 seconds, and 8 false negatives. There was 1 less false negatives using undersampled data then using the original data. Fig. 8 is the confusion matrix with 0 being non-fraudulent transactions and 1 being fraudulent transactions.

Model (undersampled)	**Accuracy**	**Prediction Time**	**FN**
Decision Tree	91.16%	0.0060 seconds	2
Logistic Regression	91.16%	0.0037 seconds	17
Random Forest	97.50%	0.1842 seconds	3
Neural Network	99.95%	1.1566 seconds	8

TABLE II
PERFORMANCE METRICS OF DIFFERENT MODELS WITH UNDERSAMPLING. FN = FALSE NEGATIVE

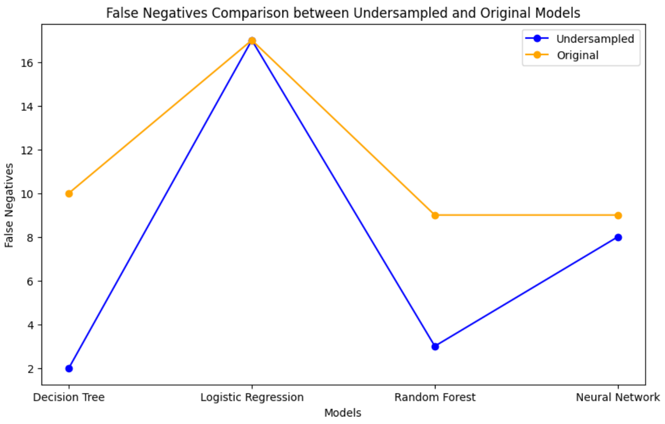

Fig. 9. False Negatives Model Comparison Undersampled vs Original

Model	FN (Original)	FN (Undersampled)	Change
Decision Tree	10	2	-8
Logistic Regression	17	17	0
Random Forest	9	3	-6
Neural Network	9	8	-1

TABLE III
Performance Metrics of Different Models with Undersampling. FN = False Negative

5) Comparing Models: Table III compares the four different models. Prediction time did not change from using undersampled data. Accuracy went down overall however false positives also went down. Table III and fig. 9 compares the false negatives and the change that using undersampled data had. Decision tree and random forest had the most change and logistic regression had no change. If I were to deploy a model I would deploy a random forest model with undersampled data because of its high accuracy and low false negatives.

V. Conclusion and Future Scope

In conclusion, this project has provided valuable insights into the effectiveness of various machine learning models in the context of credit card fraud detection. The initial evaluation of Decision Trees, Logistic Regression, Random Forest, and Neural Networks on the unbalanced dataset revealed impressive overall accuracies and low false negatives. However, it became evident that class imbalance posed a challenge, particularly with Logistic Regression, where the number of false negatives remained unchanged after undersampling. Random Forest and Neural Networks demonstrated notable improvements, showcasing the potential benefits of addressing class imbalance.

With more time, several avenues could be explored to further enhance the project's depth and impact. Firstly, a thorough hyperparameter tuning process could be conducted to optimize each model's performance. Additionally, more advanced techniques for handling class imbalance, such as oversampling, synthetic data generation, or the utilization of ensemble methods specifically designed for imbalanced datasets, could be implemented. I would also like to look into SCARFF [8] because credit card fraud is also a big data problem.

References

[1] J. Egan, "Credit card fraud statistics."

[2] Y.-A. Le Borgne, W. Siblini, B. Lebichot, and G. Bontempi, *Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook*. Université Libre de Bruxelles.

[3] A. Dal Pozzolo, O. Caelen, R. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification,"

[4] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," vol. 41, pp. 4915–4928.

[5] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," vol. PP, pp. 1–14.

[6] A. D. Pozzolo, "Adaptive machine learning for credit card fraud detection,"

[7] B. Lebichot, Y.-A. Le Borgne, L. He, F. Oblé, and G. Bontempi, "Deep-learning domain adaptation techniques for credit cards fraud detection," pp. 78–88.

[8] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF : a scalable framework for streaming credit card fraud detection with spark," vol. 41.

[9] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection,"

[10] "Credit card fraud detection."

[11] B. Zschau, "zschaub/data-mining-credit-fraud."

[12] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.