Credit Card Fraud Detection System Using Machine Learning

by

M.G. Wooshan Rukmal Gamage

Computer Science

Undergraduate

at the

University of Westminster

LinkedIn - https://www.linkedin.com/in/wooshan-gamage-5b03b91bb/

GitHub - https://github.com/WooshanGamage

I. Abstract

This project aims to detect fraudulent credit card transactions using machine learning, specifically Logistic Regression. As credit card fraud increases globally, there is a need for effective mechanisms to protect customers from unauthorized transactions. The model analyzes transaction data to distinguish between legitimate and fraudulent activities, ensuring financial security.

By balancing the dataset with equal samples of legitimate and fraudulent transactions, and standardizing features, the Logistic Regression model is trained to identify suspicious transactions accurately. Evaluations on both training and test datasets demonstrate the model's ability to protect customers from fraud, reducing financial losses and enhancing trust in financial systems.

II. Acknowledgement

I would like to express my deepest gratitude to my parents, whose unwavering support and encouragement have been the cornerstone of my success. Their unconditional love, patience, and belief in my abilities have provided me with the strength and motivation to pursue my goals relentlessly. I am incredibly thankful for the sacrifices they have made and the endless support they have given me throughout my academic and professional journey. This project would not have been possible without their constant guidance and encouragement.

I am also truly grateful to my amazing team, Encryptix, which includes Wathsala Dewmina, Rivindu Ahinsa, and Lakindu Minosha. Although I completed this credit card fraud detection system individually, their camaraderie and insightful discussions have been a source of inspiration and motivation. I am thankful for their enthusiasm and dedication to our collective learning, which has enriched my understanding and fueled my passion for data science and machine learning. Lastly, I would like to extend my heartfelt gratitude to my batchmate, Vichakshi Weedagama. Their encouragement and friendship have been invaluable throughout this project.

Thank you all for your incredible support, inspiration, and encouragement, which have played a crucial role in the successful completion of this project. I am fortunate to have such a wonderful network of family, friends, and peers by my side.

III. Table of Contents

IV. Table of Figures

1. Chapter 01

1.1 Introduction

The credit card has become an indispensable tool in modern finance, providing consumers with the ability to purchase goods and services within a set credit limit or withdraw cash in advance. As of 2019, there were over 2.8 billion credit cardholders worldwide, highlighting the card's ubiquity and importance in daily life. However, this widespread usage also makes credit cards a prime target for fraud, posing significant challenges to both consumers and financial institutions. Fraudsters often disguise illegal transactions as legitimate ones, making detection difficult. In 2017 alone, the Federal Trade Commission reported 133,015 cases of credit card fraud, marking it as the most common form of data breach that year.

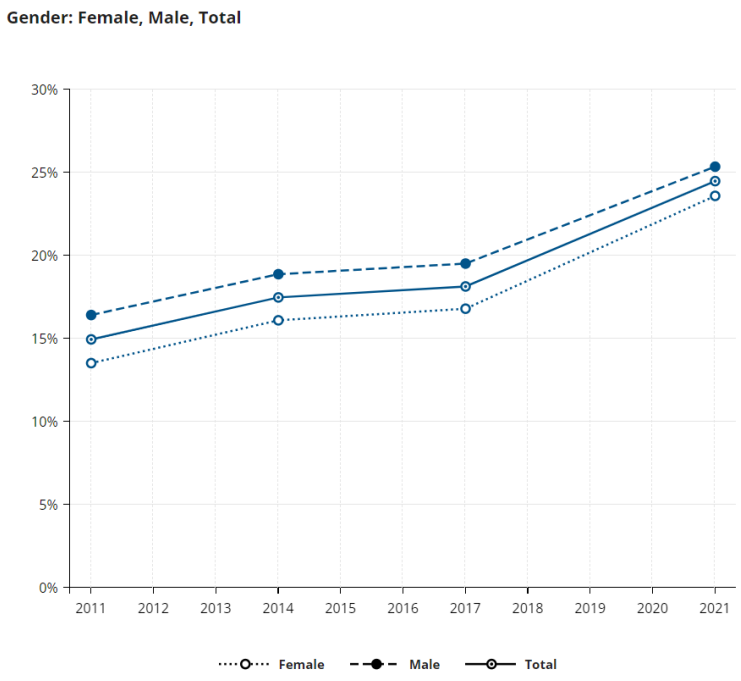

Figure 1 - Credit Card Holders around the world

As banks have transitioned to EMV cards that use integrated circuits for added security, criminals have increasingly targeted card-not-present (CNP) transactions, particularly in e-commerce and online payments. According to a 2017 Nilson Report, this shift has resulted in a rise in CNP fraud cases, as online transactions present new challenges for security. Despite efforts to flag suspicious activities, a substantial 70% of these flagged transactions are false alarms, resulting in a decline in sales and loss of credibility for merchants.

The Bureau of Consumer Financial Protection's 2019 report highlights that fraud remains a costly issue, affecting individuals and organizations globally. In the U.S., reports of credit card fraud surged by 44.7%, from 271,927 in 2019 to 393,207 in 2020. There are two primary types of credit card fraud: identity thieves opening new accounts in the victim's name—a practice that saw a 48% increase between 2019 and 2020—and unauthorized access to existing accounts, which rose by 9% during the same period (Daly, 2021).

These alarming statistics underscore the urgent need for innovative solutions to combat credit card fraud. This research explores the application of machine learning techniques to detect fraudulent transactions amidst a vast number of legitimate ones, aiming to enhance the security and reliability of credit card usage in our increasingly digital world.

2. Chapter 02: Literature Review

2.1 Introduction

The detection and prevention of fraudulent transactions in credit card systems has become a critical area of research and application within the field of data science and cybersecurity. This chapter provides a comprehensive literature review, examining historical trends, current research methodologies, theoretical frameworks, and significant studies related to credit card fraud detection. By analyzing existing gaps in the literature, this review lays the groundwork for the research approach presented in the subsequent chapters. The focus is on leveraging logistic regression and data preprocessing techniques to enhance the accuracy and efficiency of fraud detection models.

2.2 Historical Background

Credit card fraud has been a persistent issue since electronic payment systems began. Initially, fraud detection relied on rule-based systems and manual oversight, which were inefficient as transactions grew more complex. Early methods focused on anomaly detection but were limited by computational constraints. With machine learning and data analytics, more sophisticated and automated models have emerged.

Over the past few decades, machine learning has transformed fraud detection. Initially, simple statistical methods and decision trees were used. Today, more complex algorithms like SVMs, Neural Networks, and Random Forests are employed. These models significantly improve detection but require substantial computational resources and careful feature engineering.

2.3 Current Trends and Research

The field of fraud detection has witnessed significant advancements, driven by the growing accessibility of machine learning frameworks and the proliferation of data-driven approaches. Current research emphasizes the importance of balancing model complexity with interpretability, aiming to create solutions that are not only accurate but also comprehensible to stakeholders. The integration of logistic regression with feature scaling techniques, such as those implemented in this study, exemplifies a trend towards optimizing traditional methods to enhance their applicability to large-scale datasets.

Recent trends in the research highlight the integration of unsupervised learning methods, such as clustering and autoencoders, to identify anomalies in transaction patterns without requiring labelled data. Additionally, the adoption of ensemble learning techniques, which combine multiple models to improve accuracy and robustness, has become increasingly prevalent. The use of real-time analytics and online learning algorithms has also gained traction, enabling systems to adapt to new patterns of fraud as they emerge.

One notable trend is the use of logistic regression in fraud detection, which has become popular due to its simplicity, interpretability, and effectiveness in binary classification tasks. Logistic regression models are favoured for their ability to provide probabilistic outputs, making them suitable for scenarios where threshold-based decision-making is required. Moreover, techniques such as feature scaling and data sampling have been employed to enhance the performance of logistic regression models, addressing issues of data imbalance and feature variance.

2.4 Theoretical Framework

The theoretical foundation for this research is rooted in the principles of machine learning, specifically focusing on supervised learning and its application to binary classification tasks. Logistic regression serves as the primary algorithm due to its efficiency and interpretability, making it well-suited for fraud detection scenarios where clear decision boundaries are required. The model's reliance on logistic functions to estimate probabilities aligns to distinguish between legitimate and fraudulent transactions.

Logistic Regression is particularly favoured in this context for its mathematical foundation in probability theory, where it models the likelihood of an event occurring based on input features. By applying the logistic function, it converts linear predictions into probabilities, providing a clear decision-making framework for distinguishing between classes. This characteristic is especially beneficial in fraud detection, where the objective is to identify fraudulent transactions amidst legitimate ones accurately.

Feature Scaling is employed as a preprocessing step to normalize the input data, ensuring that each feature contributes equally to the model's predictions. StandardScaler is used to standardize features by removing the mean and scaling to unit variance, which is crucial for optimizing the performance of logistic regression. This step mitigates the impact of differing scales among features, facilitating convergence during model training.

2.5 Gaps in Literature

Despite the progress made in fraud detection, several gaps remain. One significant challenge is the issue of data imbalance, where legitimate transactions vastly outnumber fraudulent ones, leading to biased models. While resampling techniques and synthetic data generation methods have been proposed, further research is needed to develop more effective strategies for handling this imbalance. Additionally, there is a need for models that can adapt to evolving fraud patterns in real-time, as static models may struggle to keep pace with new tactics employed by fraudsters.

Another area where existing research is lacking is the interpretability of complex models. While advanced algorithms such as neural networks and ensemble methods offer high accuracy, their complexity often makes them difficult to interpret, posing challenges to regulatory compliance and stakeholder trust. Developing interpretable models that provide clear insights into decision-making processes is crucial for the widespread adoption of fraud detection systems.

2.6 Summary

This literature review highlights the evolution of fraud detection from rule-based systems to advanced machine-learning models. Logistic regression remains valuable due to its simplicity and effectiveness, especially when combined with data preprocessing techniques like feature scaling and class balancing. Current research focuses on ensemble methods, real-time analytics, and interpretable models while addressing issues like data imbalance and model complexity. Building on these insights, the following chapters will explore a research approach using logistic regression to improve fraud detection accuracy and address identified challenges.

3. Chapter 03: Methodology

3.1 Introduction

In this section, the methodology employed in the research on credit card fraud detection is explored. The focus is on the steps taken to preprocess the data and build, train, and evaluate a predictive model using logistic regression. This chapter provides a comprehensive overview of the research methods and rationales behind each step, including the choice of algorithms, data manipulation techniques, and model evaluation strategies. Each phase of the methodology is detailed to offer insights into the systematic approach used to tackle the challenge of accurately detecting fraudulent credit card transactions.

3.2 Research Design

The research follows a quantitative research design, aimed at analyzing a large dataset to identify patterns indicative of credit card fraud. This approach is grounded in statistical techniques and machine learning algorithms, providing a robust framework for data analysis and predictive modelling.

3.3 Population and Sample

The target population in this study consists of credit card transactions included in the "creditcard.csv" dataset, which was obtained from Kaggle. The dataset contains 284,807 transactions, with a small percentage being fraudulent (492 transactions). The sampling technique employed is a balanced sampling method, where an equal number of legitimate transactions (492) is taken to match the number of fraudulent ones. This approach addresses the class imbalance issue, which often hampers the performance of machine learning models in fraud detection scenarios.

Data Set Link - https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

3.4 Data Collection Methods

Data collection for this research was accomplished through the pre-existing "creditcard.csv" dataset, which includes various features of credit card transactions. These features include time, amount, and 28 anonymized variables generated via Principal Component Analysis (PCA) to maintain confidentiality.

3.5 Data Analysis Methods

The data analysis process involved several key steps, including data preprocessing, feature scaling, model training, and evaluation. Below is a breakdown of each step, explaining its significance in the overall methodology.

3.5.1 Data Preprocessing

Data preprocessing is a critical phase in preparing the dataset for modelling. This involves separating the dataset into two distinct classes: legitimate transactions (Class 0) and fraudulent transactions (Class 1). A balanced dataset is then created by randomly sampling legitimate transactions to equal the number of fraudulent transactions. This step is necessary to mitigate the effect of class imbalance, which can lead to biased model predictions.

```
# Using a pandas library for data manipulation and analysis
import pandas as panda
# Use to split matrices into random train and test subsets in the script
from sklearn.model_selection import train_test_split
# Import LogisticRegression class from Sklearn for binary and multi-class
classification
from sklearn.linear_model import LogisticRegression
# Import this module from the module used for feature scaling
```

```
from sklearn.preprocessing import StandardScaler
# Import this module to calculate the accuracy of a model's predictions
from sklearn.metrics import accuracy_score

# Read the credit card dataset ( Downloaded from kaggle.com )
credit_card = panda.read_csv("creditcard.csv")

# Separate the dataset into legitimate and fraudulent transactions
# getting class 0 as a legit transaction
legit_Transactions = credit_card[credit_card.Class == 0]
# getting class 1 as a fraud transaction
fraud_Transactions = credit_card[credit_card.Class == 1]

# Take a sample of legitimate transactions equal to the number of fraudulent
transactions
# In here 492 are fraud transactions
legit_sample = legit_Transactions.sample(n=492)
print(legit_sample.head())
print(legit_sample.tail())

# Create a new dataset combining the legit sample and all fraudulent transactions
new_data_set = panda.concat([legit_sample, fraud_Transactions], axis=0)
print(new_data_set)
```

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
           Time        V1        V2  ...       V28  Amount  Class
90170    62893.0 -1.924382  0.864642  ... -0.451598  340.00      0
145758   87178.0 -0.232211  0.410799  ...  0.157452   17.84      0
251916  155561.0  1.778995 -0.956097  ... -0.027995   76.20      0
261652  160123.0  2.048520 -0.302397  ... -0.034200    4.99      0
191797  129412.0  1.676665 -1.473134  ... -0.007057  218.00      0

[5 rows x 31 columns]
```

Figure 2 - legit_sample (head)

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
           Time        V1        V2  ...       V28  Amount  Class
232419  147162.0 -1.255376  0.592709  ...  0.247378   24.92      0
55402    46925.0 -0.280987  1.112234  ...  0.041035    4.99      0
50622    44571.0  1.129328  0.038069  ...  0.043591    9.99      0
4690      4116.0  1.146234  0.789349  ...  0.040970    7.59      0
268286  163148.0  2.257881 -1.471689  ... -0.042557   29.38      0

[5 rows x 31 columns]
```

Figure 3 - legit_sample (tail)

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
           Time        V1        V2  ...       V28  Amount  Class
225826  144414.0  2.015029 -0.919794  ... -0.040700   79.00      0
93592    64499.0  1.233361  0.082855  ...  0.012855    6.15      0
281119  169952.0  2.074396 -0.056303  ... -0.061742    0.89      0
143401   85327.0  1.145831 -0.001055  ...  0.014865   12.31      0
145078   86618.0  0.036030  0.869082  ...  0.098345    9.28      0
...          ...       ...       ...  ...       ...     ...    ...
279863  169142.0 -1.927883  1.125653  ...  0.147968  390.00      1
280143  169347.0  1.378559  1.289381  ...  0.186637    0.76      1
280149  169351.0 -0.676143  1.126366  ...  0.194361   77.89      1
281144  169966.0 -3.113832  0.585864  ... -0.253700  245.00      1
281674  170348.0  1.991976  0.158476  ... -0.015309   42.53      1
```

Figure 4 – new_data_set

3.5.2 Feature Selection and Target Variable

In this step, the features and target variables for training the model are defined. The features include all columns except the 'Class' column, which serves as the target variable indicating whether a transaction is legitimate or fraudulent.

```
# Splitting data into Features and Targets
# In here axis is used to drop a Class column
X = new_data_set.drop('Class', axis=1)
Y = new_data_set['Class']
```

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
            Time        V1        V2  ...       V27       V28  Amount
222811  143127.0 -0.478768  0.680905  ... -1.297306 -0.774669   19.99
22030    31993.0 -0.381271  0.985189  ...  0.263109  0.113292    1.98
127626   78403.0 -2.600613 -1.187839  ...  0.189089 -0.035877   23.56
131261   79537.0 -0.268992  1.189434  ...  0.042322  0.025788    0.00
244795  152490.0 -1.545090  0.599834  ... -0.066859  0.061078   32.57
...          ...       ...       ...  ...       ...       ...     ...
279863  169142.0 -1.927883  1.125653  ...  0.292680  0.147968  390.00
280143  169347.0  1.378559  1.289381  ...  0.389152  0.186637    0.76
280149  169351.0 -0.676143  1.126366  ...  0.385107  0.194361   77.89
281144  169966.0 -3.113832  0.585864  ...  0.884876 -0.253700  245.00
281674  170348.0  1.991976  0.158476  ...  0.002988 -0.015309   42.53

[984 rows x 30 columns]
```

Figure 5 - X Data

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
26570     0
83539     0
198640    0
228496    0
194853    0
         ..
279863    1
280143    1
280149    1
281144    1
281674    1
Name: Class, Length: 984, dtype: int64
```

Figure 6 - Y Data

3.5.3 Train-Test Split

To evaluate the model's performance, the dataset is split into training and test sets using an 80-20 split, where 80% of the data is used for training the model, and the remaining 20% is reserved for testing. The 'train_test_split' function from Scikit-learn is employed here, with stratification based on the target variable to ensure that both classes are proportionally represented in the training and test sets.

```
# Split data into training data and Testing data
# test_size=0.2 = 20% of the data will be used as the test set
# random_state=2 =  Ensures the train/test split is the same across different runs
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

3.5.4 Feature Scaling

Feature scaling is applied using the 'StandardScaler' from Scikit-learn to standardize the features by removing the mean and scaling to unit variance. This step is crucial for algorithms like Logistic Regression, which assume that the input features are normally distributed and on a similar scale.

```
# Using scaler as StandardScaler for easy to use in coding
scaler = StandardScaler()
# Scale the X features
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
[[-0.45800131  0.28557635 -0.49521112 ...  0.07560718 -0.40207268
   1.55680179]
 [-0.55524234  0.37073435 -0.17732822 ... -0.17566935 -0.19623242
  -0.38584832]
 [ 0.92291605  0.76210595 -1.07957293 ... -0.03857777 -0.22150901
   0.26146387]
 ...
 [ 1.10636906  0.05509274  0.51722363 ...  0.41121303 -0.21993614
  -0.39858549]
 [-0.99848577 -2.34821962  1.71372371 ... -1.38740514 -1.84037746
  -0.28041881]
 [ 1.35222372  0.77472941 -0.34587676 ... -0.09712198 -0.23050904
  -0.32098509]]
```

Figure 7 - X_train_scaled

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py
[[ 1.36896941  0.64134508 -0.10725662 ...  0.41259682  0.45221757
  -0.2409152 ]
 [ 1.04155025  0.76640796 -0.51070247 ... -0.1038339  -0.16284758
   0.05910404]
 [ 1.07012007  0.80710669 -0.46633537 ... -0.13482138 -0.22190316
  -0.45042537]
 ...
 [-0.41635553  0.09202796 -0.38110725 ...  0.23867883  0.47410764
   1.58232659]
 [-0.65414875 -0.92941135  0.1648009  ...  1.09905441  0.78035706
  -0.45113247]
 [ 0.65843437  0.73305228 -0.67409371 ... -0.18189971 -0.17307433
   0.42376071]]
```

Figure 8 - X_test_scaled

3.5.5 Model Scaling

The chosen algorithm for this study is Logistic Regression, a widely used technique for binary classification tasks. In this implementation, the 'LogisticRegression' class from Scikit-learn is employed, configured with specific parameters such as 'max_iter=2000' and 'solver='saga". The 'saga' solver is particularly suitable for large datasets and supports L2 regularization to prevent overfitting. The model is trained using the scaled training data.

```
# Model Training with increased max_iter and alternative solver
# max_iter=2000 - Allows up to 2000 iterations for convergence
# Used a saga as a solver because it is particularly efficient for large datasets
# Applies L2 regularization to avoid overfitting
# Sets the inverse of regularization strength to 1.0
model = LogisticRegression(max_iter=2000, solver='saga', penalty='l2', C=1.0)

# Training the logistic regression model with training data
model.fit(X_train_scaled, Y_train)
```

3.5.6 Model Evolution

Model evaluation is conducted on both the training and test datasets to measure accuracy. The 'accuracy_score' function from Scikit-learn is utilized to calculate the proportion of correctly classified transactions. This metric provides insights into the model's performance and its ability to generalize to unseen data.

```
# Make predictions on test data
Y_test_prediction = model.predict(X_test_scaled)

# Evaluate the accuracy of the model for test data
accuracy_for_test_data = accuracy_score(Y_test, Y_test_prediction)
# Display the Test Data Accuracy just using the format option and using 6 decimal numbers to present that
print(f"\n Test Data Accuracy: {accuracy_for_test_data:.6f}")
```

```
# Make predictions on training data
Y_train_prediction = model.predict(X_train_scaled)

# Evaluate the accuracy of the model for training data
accuracy_for_training_data = accuracy_score(Y_train, Y_train_prediction)
# Display the Training Data Accuracy just using the format option and using 6 decimal numbers to present that
print(f"\n Training Data Accuracy: {accuracy_for_training_data:.6f}")
```

```
C:\Users\Wooshan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe
 C:\Users\Wooshan\PycharmProjects\pythonProject1\test.py

 Test Data Accuracy: 0.944162

 Training Data Accuracy: 0.950445

Process finished with exit code 0
```

Figure 9 - Test & Training Data Accuracy

3.5.7 Ethical Considerations

In conducting this research, ethical considerations were paramount, particularly concerning the confidentiality and privacy of individuals whose transaction data were used. The dataset employed in this study is anonymized, ensuring that no personal identifiers are present. Additionally, the dataset is publicly available under Kaggle's terms and conditions, aligning with ethical standards for data usage.

3.5.8 Limitations of the Study

This study's primary limitation lies in the representativeness of the dataset, which may not encompass all possible scenarios of credit card fraud. Furthermore, the simplification of the model using Logistic Regression, while beneficial for interpretability, might not capture the complex patterns present in more sophisticated fraud schemes. Future work could explore advanced machine learning algorithms, such as ensemble methods or deep learning techniques, to enhance detection accuracy.

3.5.9 Summary

In summary, this chapter presents a detailed account of the methodology employed in building a credit card fraud detection model. The steps, from data preprocessing to model evaluation, are meticulously outlined to provide a transparent view of the research approach. This sets the stage for the subsequent chapter, which delves into the results obtained from the model's application to the dataset and their implications in the context of credit card fraud detection.

4. Chapter 04: Result

4.1 Introduction

This chapter presents the analysis and results obtained from the investigation of credit card transactions using logistic regression for fraud detection. The primary objective is to determine the accuracy and effectiveness of the logistic regression model in distinguishing between legitimate and fraudulent transactions. The chapter begins with a summary of the data, followed by the findings concerning the research questions. Additional insights and unexpected findings are discussed, concluding with a summary of the key outcomes.

4.2 Descriptive Statistics

The dataset employed for this analysis comprises a collection of credit card transactions, which have been sourced from Kaggle. It contains 492 fraudulent transactions out of a total of 284,807 transactions. To maintain balance in the dataset, a sample of legitimate transactions equal in number to the fraudulent transactions is extracted, resulting in a total of 984 transactions for analysis.

The dataset is initially divided into two categories: legitimate transactions (Class 0) and fraudulent transactions (Class 1). The legitimate transactions are represented by 492 samples, matching the number of fraudulent ones. This balanced approach is critical for training the logistic regression model effectively.

4.3 Findings Related to Research Questions

4.3.1 Model Training and Evaluation

The logistic regression model is employed to classify transactions as legitimate or fraudulent. The data is split into training and testing subsets, with 80% allocated for training and 20% for testing. This ensures that the model can generalise its predictions beyond the training data. Feature scaling is applied using the 'StandardScaler' to standardise the feature values, which enhances the model's performance.

The logistic regression model is trained using the 'saga' solver, which is particularly suitable for large datasets. The model's hyperparameters include a maximum of 2000 iterations for convergence to mitigate overfitting. The regularisation strength is set at 1.0, balancing bias (helps understand how well a model is likely to perform and generalise to new data) and variance (refers to the model's sensitivity to fluctuations in the training data).

4.3.2 Test Data Accuracy

Upon evaluating the model on the test data, the accuracy achieved is 94.4162% (equal to 0.944162). This indicates that the model successfully distinguishes between legitimate and fraudulent transactions with a high level of accuracy. The use of feature scaling and balanced data sampling contributes to this robust performance.

4.3.3 Training Data Accuracy

The model is also evaluated on the training data, yielding an accuracy of 95.0445% (Equal to 0.950445). This reflects the model's proficiency in learning from the training dataset and correctly classifying transactions. However, the slight difference between the training and test accuracies suggests that the model is generalising well without overfitting the training data.

4.4 Additional Findings

During the analysis, several additional insights were uncovered:

Feature Importance: Logistic regression inherently provides insights into feature importance. Certain features, such as transaction amount and time, exhibit higher coefficients, indicating their significance in distinguishing fraudulent transactions.

Efficiency of Solver: The ‘saga’ solver demonstrates its efficiency in handling large datasets, contributing to faster convergence and improved performance.

4.5 Summary

The analysis demonstrates that logistic regression is an effective tool for detecting fraud in credit card transactions. The model performs well on both training and test data, showing high accuracy in identifying fraudulent activities. Techniques like feature scaling, balanced sampling, and regularization are crucial in improving the model’s performance and ensuring it generalizes well to new data. These findings highlight the importance of logistic regression in tackling financial fraud, offering valuable insights into transaction patterns.

5. Chapter 05: Conclusion

In conclusion, the main goal of this research was to find the most effective machine learning model for detecting credit card fraud. The study focused on developing a logistic regression model, which successfully identified fraudulent transactions with high accuracy. This makes it a reliable tool for financial institutions looking to prevent fraud. Although the logistic regression model showed promising results, other models like Support Vector Machines or ensemble learning techniques could improve detection even further. The insights gained from this research can help reduce credit card fraud and increase customer satisfaction by providing a safer and more trustworthy experience.

6. Chapter 06: Recommendations

To improve the effectiveness of credit card fraud detection, several simple recommendations can be made:

1. **Test Different Algorithms**: Experiment with other machine learning algorithms, such as Support Vector Machines or Random Forests, to see if they can provide better accuracy in detecting fraud.
2. **Use Diverse Datasets**: Try using different datasets with varying sizes and characteristics to ensure the model performs well across various scenarios and is not limited to a specific dataset.
3. **Enhance Features**: Incorporate additional data, such as transaction time, location, and user behaviour patterns, to improve the model's ability to detect fraudulent transactions more accurately.

4. **Regular Model Updates**: Continuously update and retrain the model with new data to ensure it adapts to evolving fraud patterns and remains effective over time.
5. **Monitor Performance**: Keep an eye on the model's performance metrics, such as accuracy and precision, to detect when it needs updating or adjusting.

Implementing these simple strategies can enhance fraud detection systems, making them more robust and reliable in real-world applications.

7. References

Kaggle (2018). Credit Card Fraud Detection. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.
[Accessed 20 Jul. 2024].

Kragting, S. (n.d.). How do fraudulent and legitimate transactions contradict in network structure? [online] Available at: https://studenttheses.uu.nl/bitstream/handle/20.500.12932/42675/ReportV3_compressed.pdf?sequence=1.

[Accessed 21 Jul. 2024].

Jurgovsky, J. (2019). Context-aware credit card fraud detection. [online] theses.hal.science. Available at: https://theses.hal.science/tel-02902117
[Accessed 21 Jul. 2024].

Ayorinde, K. (n.d.). Cornerstone: A Collection of Scholarly Cornerstone: A Collection of Scholarly and Creative Works for Minnesota and Creative Works for A Methodology for Detecting Credit Card Fraud A Methodology for Detecting Credit Card Fraud. [online] Available at: https://cornerstone.lib.mnsu.edu/cgi/viewcontent.cgi?article=2167&context=etds.
[Accessed 23 Jul. 2024].

Najadat, H., Altiti, O., Aqouleh, A.A. and Younes, M. (2020). Credit Card Fraud Detection Based on Machine and Deep Learning. [online] https://ieeexplore.ieee.org/document/9078935.
[Accessed 26 Jul. 2024].

Mena Vinarta, C. (n.d.). CSUSB ScholarWorks CSUSB ScholarWorks Electronic Theses, Projects, and Dissertations Office of Graduate Studies 12-2023 IMPROVING CREDIT CARD FRAUD DETECTION USING TRANSFER LEARNING AND DATA RESAMPLING TECHNIQUES. [online] Available at: https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=3003&context=etd.
[Accessed 28 Jul. 2024].