International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 3, pp 52-66, March-2025

https://doi.org/10.47001/IRJIET/2025.903008

Development and Implementation of a Machine Learning-Based Framework for Credit Card Fraud Detection: A Comparative Study of Random Forest and Logistic Regression Models

Anusiuba, Overcomer Ifeanyi Alex

Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University, Awka, Nigeria
E-mail: oi.anusiuba@unizik.edu.ng

***Abstract* - Credit card fraud remains a pervasive and evolving threat in the digital age, necessitating the development of sophisticated methods for early detection and prevention. This study provides a thorough examination of a machine learning-based credit card fraud detection system, employing two prominent algorithms: Random Forest and Logistic Regression. The research methodology involves preprocessing a diverse and extensive credit card transaction dataset, incorporating various transactional features. Through rigorous feature engineering, the dataset is meticulously prepared for model training and validation. The Random Forest model, an ensemble learning technique, aggregates multiple decision trees to improve predictive accuracy and mitigate the risk of over-fitting. In parallel, Logistic Regression—a classical statistical approach—models the probabilistic relationship between transaction features and the likelihood of fraud. A comparative analysis of these models offers valuable insights into their respective strengths and limitations, guiding the selection of the most suitable model for fraud detection. Model performance is evaluated using critical metrics, including accuracy, precision, recall, and F1-score, with a detailed examination of these indicators across different scenarios to assess each model's ability to distinguish between legitimate and fraudulent transactions. Furthermore, the study explores the practical implications of implementing these models in financial institutions, highlighting their potential to enhance security and reduce financial losses. Ethical considerations, including privacy concerns, model interpretability, and the adaptive nature of fraud patterns, are also discussed, providing a comprehensive perspective on the deployment of machine learning in fraud detection systems. Ultimately, this research contributes to the advancement of financial security, offering a robust analysis of Random Forest and Logistic Regression models and their real-world applications in combating credit card fraud.**

Keywords: Machine learning, Credit Card Fraud Detection, Random Forest, Logistic Regression Models, Credit card fraud, early detection, Prevention system.

I. BACKGROUND OF THE STUDY

Credit card fraud has become an increasingly prevalent issue, presenting significant challenges to financial institutions, consumers, and economies globally. In 2021 alone, losses attributed to credit card fraud exceeded 1.3 billion dollars in the United States (Card Fraud Prevention, 2021). Similarly, in Nigeria, the Nigerian Economic and Financial Crimes Commission reported over 100,000 cases of credit card fraud in the same year, with estimated financial losses surpassing one billion naira. The severity of this issue is compounded by several factors, including inadequate financial regulations, weak enforcement mechanisms, and limited public awareness of fraud prevention strategies. These challenges make credit card fraud particularly pervasive in regions like Nigeria, where financial institutions struggle to deploy effective fraud detection and prevention measures.

Traditional fraud detection methods, including rule-based algorithms and pattern recognition, often produce false positives or negatives and struggle to adapt to evolving fraud tactics. Consequently, there is a growing need for advanced techniques to improve detection accuracy. Machine learning (ML) models, particularly Random Forest and Logistic Regression, have emerged as promising solutions due to their ability to analyze large, complex datasets and improve prediction accuracy. (Anusiuba et al, 2022) Random Forest, an ensemble learning algorithm, has shown strong performance in fraud detection by combining multiple decision trees to enhance classification accuracy and reduce overfitting. It has been successfully applied in hybrid models with techniques like Isolation Forest and Support Vector Machines to address challenges posed by imbalanced datasets (Dornadul & Geetha, 2019). On the other hand, Logistic Regression, while simpler and more interpretable, remains effective in modeling the probability of fraudulent transactions, although it may struggle in imbalanced datasets (Sulaiman et al., 2022).

This study focuses on designing and implementing a credit card fraud detection system using both Random Forest and Logistic Regression models. By comparing their performance on real-world transaction data, the research aims to identify the most effective approach for improving fraud detection in financial systems.

1.1 Statement of the Problem

Credit card fraud has become a pervasive and costly issue for financial institutions and consumers worldwide, resulting in significant financial losses. The increasing volume of credit card transactions, combined with the rising sophistication of fraudulent activities, underscores the urgent need for an effective and secure fraud detection system. Despite the growing implementation of machine learning algorithms to address this challenge, many existing systems still struggle with the inherent complexities of detecting fraudulent transactions in real-time. This is particularly evident in the imbalance between fraudulent and legitimate transactions, which often leads to high false positive rates and low detection accuracy.

Various machine learning techniques, such as Naïve Bayes, Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forest, Genetic Algorithms, J48, and AdaBoost, have been employed in credit card fraud detection. However, each of these models presents specific limitations, particularly in handling the imbalanced nature of fraud datasets and the need for continuous improvement in performance. While Random Forest and Logistic Regression models have shown promise, there is a lack of standard benchmarks and evaluation metrics for accurately comparing their performance and identifying optimal classifiers.

Moreover, the dynamic nature of fraud, coupled with the continuous innovation of fraudsters, poses significant challenges to the detection of novel and complex fraudulent patterns. Existing fraud detection systems must be able to adapt to these evolving threats while maintaining high accuracy and minimizing computational overhead. The goal of this study is to explore and address these challenges by designing and implementing a robust credit card fraud detection system utilizing Random Forest and Logistic Regression models. The research aims to optimize these machine learning models, compare their performance, and develop a reliable system capable of accurately detecting fraudulent transactions. Ultimately, this study seeks to provide a comprehensive solution to the problem of credit card fraud, reducing financial losses for both banks and customers through improved fraud detection capabilities.

1.2 Aim and Objectives of the Study

The aim of this study is to design and implement an effective and secure credit card fraud detection system using machine learning algorithms, specifically Random Forest and Logistic Regression. The study seeks to enhance the accuracy and efficiency of fraud detection systems, minimizing financial losses due to fraudulent credit card transactions. Through this approach, the study aims to provide deeper insights into the patterns of credit card fraud and evaluate the applicability of decision tree-based models in improving fraud detection.

1.3 Objectives of the Study

1. To analyze and explore the dataset of credit card transactions in order to identify patterns and characteristics indicative of fraudulent activities.
2. To build and optimize machine learning models, focusing on Random Forest and Logistic Regression, with the goal of improving fraud detection accuracy and system performance.
3. To compare and evaluate the performance of various machine learning models—including Random Forest, Logistic Regression, and other relevant techniques—to determine the most effective method for detecting fraudulent transactions.
4. To develop and implement a robust credit card fraud detection system that can be integrated into banking and financial institutions, aimed at reducing the occurrence of fraudulent transactions and minimizing associated financial losses.

II. REVIEW OF RELATED LITERATURE

Credit Card is a thin rectangular piece of either metal or plastic issued by banks or financial institutions that allows cardholders to borrow funds with which to pay for goods and services with merchants that accept cards for payment (Ekwealor et al, 2021). On its front are the bank name, card number, cardholder's name the chip and the expiry date then at its reverse are the magnetic strip, signature, hologram, and the Card Verification Code (CVC) as shown Figure 1 below.

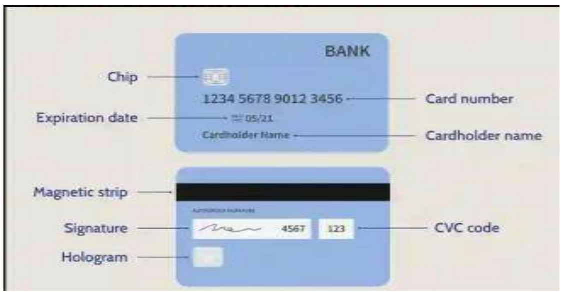


Figure 1: Diagram of Credit Card

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 52-66, March-2025
https://doi.org/10.47001/IRJIET/2025.903008

With credit cards, shopping online, reserving airline tickets and ordering from a catalog becomes a breeze. Mailing a cheque is almost a thing of the past as a credit card is faster, easier, and generally a more secure way of doing business. Sadly, it seems that fraudsters are keeping track and even thriving in this growing environment (Kurshan, et al., 2020). Though frauds can be online or offline; credit card frauds are categorized in various ways but split into application frauds and behavioral frauds. In application frauds, the fraudsters apply for a credit card with a false ID whereas in behavioral frauds, the fraudsters find a way to obtain the cardholder's credential in other to use a pre-existing credit card. (Lebichot, et al., 2021).

Credit card fraud has a long and complex history, evolving alongside technological advancements and the changing financial landscape. The persistence and adaptability of fraudsters in exploiting emerging vulnerabilities reflect an ongoing battle between security measures and criminal ingenuity. Despite significant advancements in fraud prevention technologies, the global financial system continues to face substantial challenges, with billions of dollars lost annually to credit card fraud. This historical review traces the development of credit card fraud from its inception to the present, highlighting key technological shifts and the corresponding evolution of fraudulent tactics.

1. The Emergence of Credit Cards (1950s-1960s): The advent of credit cards in the 1950s, marked by the introduction of the Diners Club card, represented a major leap in the evolution of financial transactions. With the expansion of credit card usage, the nascent system became a target for criminal exploitation. Early fraud schemes primarily involved the use of counterfeit cards, taking advantage of the limited verification systems in place at the time.

2. Manual Transaction Processing (1960s-1970s): During the early years of credit card adoption, transactions were processed manually, with merchants taking imprints of cards and submitting them to issuers for verification. This slow, manual process created significant vulnerabilities, particularly in terms of the delayed verification of transactions. Fraudsters capitalized on this delay, using stolen or counterfeit cards without immediate detection.

3. The Introduction of Magnetic Stripe Technology (1970s-1980s): The introduction of magnetic stripe technology in the 1970s marked a significant improvement in transaction efficiency and data storage. However, this innovation also facilitated the rise of card cloning. The ability to store card information on magnetic stripes allowed criminals to exploit card-skimming devices, which could illicitly capture card data, leading to a surge in fraud related to cloned cards.

4. The Internet Era and Online Transactions (1990s): The growth of e-commerce in the 1990s introduced a new realm for credit card fraud. As consumers increasingly turned to the internet for purchasing goods and services, fraudsters adapted by exploiting vulnerabilities in online transaction systems. Phishing attacks, wherein criminals deceived individuals into revealing their credit card information through fraudulent emails or websites, became increasingly prevalent.

5. The Shift to EMV Chip Technology (2000s-Present): To address the growing threat of card cloning, many countries transitioned to EMV (Europay, Mastercard, and Visa) chip technology in the 2000s. Unlike magnetic stripe cards, EMV chips generate unique transaction codes for each purchase, significantly enhancing security and making card duplication more difficult. Despite the advantages of this technology, criminals found new ways to exploit other vulnerabilities, particularly in online and card-not-present transactions.

6. Data Breaches and the Dark Web (2000s-Present): Large-scale data breaches at major retailers and financial institutions have become a significant concern in recent decades. These breaches expose millions of credit card details, which are often sold on the dark web. Stolen card information frequently ends up in the hands of criminals who engage in identity theft or unauthorized transactions, further complicating efforts to combat fraud.

7. Contactless Payments and Mobile Wallets (2010s-Present): The rise of contactless payments and mobile wallets in the 2010s introduced new conveniences for consumers. Technologies such as tokenization—where sensitive payment data is replaced by a unique, encrypted token—provided an additional layer of security. However, the proliferation of mobile payment systems also introduced new risks, particularly the potential for unauthorized access to mobile devices, leading to concerns over the security of these platforms.

8. The Integration of Machine Learning and Artificial Intelligence (2010s-Present): In recent years, financial institutions have increasingly relied on machine learning and artificial intelligence to combat credit card fraud. These technologies allow for real-time monitoring of transactions, enabling the detection of unusual patterns that may indicate fraudulent activity. By analyzing vast amounts of data, machine learning models can identify subtle anomalies, improving the accuracy and speed of fraud detection systems.

9. The Rise of Cybercrime (2010s-Present): The 2010s witnessed a dramatic increase in cybercrime, with criminals employing sophisticated techniques such as ransomware attacks to compromise financial institutions. These attacks often target the infrastructure of financial organizations,

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 52-66, March-2025
https://doi.org/10.47001/IRJIET/2025.903008

stealing sensitive customer data and facilitating fraud. As cybercrime tactics continue to evolve, the financial industry faces growing challenges in securing payment systems against increasingly advanced threats.

More so, the history of credit card fraud is a testament to the ingenuity of criminals and the continuous evolution of payment technologies. While security measures have advanced, so too have the tactics used by fraudsters, underscoring the ongoing challenge of staying ahead of emerging threats. The rise of new technologies such as mobile wallets, machine learning, and artificial intelligence offers promising tools for combating fraud. However, as the sophistication of fraud continues to increase, it remains essential for both individuals and businesses to remain vigilant and proactive in their efforts to mitigate the risks associated with credit card fraud.

Credit card fraud detection has been a prominent area of research, attracting considerable attention due to its critical importance in the financial sector. Over the years, a variety of techniques have been proposed, with a strong focus on data mining and neural networks. This review examines key contributions to the field, highlighting the evolution of methods and their applications in detecting fraudulent credit card transactions.

Ghosh and Reilly (1994) were among the pioneers in applying neural networks for credit card fraud detection. They developed a detection system trained on a large sample of labeled credit card transactions, which included various fraud types such as lost cards, stolen cards, application fraud, counterfeit fraud, mail-order fraud, and non-received issues (NRI). Their work laid the foundation for subsequent developments in fraud detection systems that leverage machine learning. In a more recent study, Syeda et al. (2002) introduced Parallel Granular Neural Networks (PGNNs) to enhance the speed and efficiency of the data mining and knowledge discovery process in fraud detection. Their approach, which implemented a complete fraud detection system, aimed to optimize the performance of neural networks in real-time fraud detection applications.

Stolfo et al. (2000) expanded on previous work by employing meta-learning techniques for credit card fraud detection. Meta-learning is a strategy that integrates multiple classifiers or models to improve the detection process. Their approach involves training a meta-classifier that analyzes the correlations between base classifiers' predictions, thereby enhancing the accuracy of fraud detection systems. Additionally, this group worked on developing a cost-based model for fraud and intrusion detection, utilizing distributed data mining through Java agents for Meta-learning (JAM). This system has been noted for its ability to handle the complexities of fraud detection with respect to key performance metrics, such as True Positive - False Positive spread and accuracy.

Aleskerov et al. (1997) presented CARDWATCH, a database mining system for credit card fraud detection that integrates a neural learning module. The system provides an interface to a variety of commercial databases, facilitating the detection of fraudulent activities across multiple data sources. This approach underscores the growing need for robust, scalable systems in the fight against credit card fraud. Kim and Kim (2002) highlighted two primary challenges in credit card fraud detection: the skewed distribution of data and the mix of legitimate and fraudulent transactions. To address these challenges, they proposed the use of fraud density as a confidence value, generating a weighted fraud score to minimize misdetections. Their work emphasizes the importance of data preprocessing and the adjustment of detection thresholds to improve detection performance. Brause et al. (1999) developed an innovative approach that integrates advanced data mining techniques and neural network algorithms to achieve high fraud coverage. This combination of methods has proven effective in detecting previously unrecognized fraud patterns, demonstrating the efficacy of hybrid approaches in the domain.

Chiu and Tsai (2004) proposed a collaborative fraud detection scheme for the banking industry, utilizing web services and data mining techniques. This system enables participating banks to share knowledge about fraud patterns in a distributed, heterogeneous environment, thus enhancing the collective ability to detect fraud. The use of web service technologies such as XML, SOAP, and WSDL ensures seamless data exchange among institutions, contributing to more robust fraud detection networks. Phua et al. (2007) conducted a comprehensive survey of existing data mining-based fraud detection systems, providing valuable insights into the state of the field. Their report summarizes various methods and highlights the challenges and successes encountered in the application of data mining techniques to credit card fraud detection.

Stolfo (1999) employed an agent-based approach combined with distributed learning to detect credit card fraud. Their system leverages artificial intelligence, integrating inductive learning algorithms and meta-learning methods to enhance detection accuracy. This multi-layered approach aims to optimize decision-making in real-time fraud detection scenarios. Phua et al. (2004) extended their work on meta-learning by applying meta-classifiers, such as naive Bayes, C4.5, and back propagation neural networks, for fraud detection. Their method focuses on adjusting the classifier selection based on data skewness, though their approach was not directly applied to credit card fraud detection, its general applicability makes it a valuable contribution to the field.

Vatsa et al. (2005) introduced a novel game-theoretic approach to fraud detection, modeling the interaction between an attacker and the fraud detection system as a multi-stage game. This model treats the fraud detection system and the attacker as two players each striving to maximize their respective payoffs. This approach offers a strategic framework for understanding the dynamics of fraud detection and developing countermeasures.

In addition to the established techniques outlined above, recent advancements in credit card fraud detection have gained attention. These newer methods continue to evolve, incorporating innovative approaches such as deep learning, hybrid models, and real-time transaction monitoring. The ongoing development of these techniques underscores the importance of adapting fraud detection systems to keep pace with increasingly sophisticated fraud strategies.

In conclusion, the evolution of credit card fraud detection techniques demonstrates a significant shift towards more intelligent, data-driven approaches. From early neural network models to modern, collaborative, and game-theoretic frameworks, the field has progressed in response to the growing complexity of fraud schemes. Continued research into advanced machine learning, distributed systems, and innovative methodologies will likely play a key role in improving the detection and prevention of credit card fraud in the future. Although machine learning algorithms have demonstrated their ability to identify fraudulent credit card transactions, there is still room for improvement. While these models can detect, classify, and potentially prevent fraud, their accuracy is often constrained by the limitations of the training data. To further enhance fraud detection capabilities, the integration of deep learning techniques is recommended. Deep learning models have the potential to significantly improve the accuracy, reliability, and efficiency of fraud detection systems, providing more robust solutions for financial institutions in combating fraud. The evolving nature of fraudulent activities necessitates the development of increasingly sophisticated detection methods to address the challenges faced in the financial sector.

III. METHODOLOGY ADOPTED

The Development and Implementation of a Machine Learning-Based Framework for Credit Card Fraud Detection: A Comparative Study of Random Forest and Logistic Regression Models employ supervised machine learning techniques to accurately detect and predict fraudulent credit card transactions. The methodology is structured to optimize detection accuracy while minimizing false positives, thereby enhancing financial security.

3.1 Model Selection and Evaluation

In this study, a comparative analysis is conducted between Random Forest and Logistic Regression models, along with Decision Trees, to determine their effectiveness in detecting fraudulent credit card transactions. The Random Forest algorithm is selected as the primary model due to its robust ensemble-learning capabilities, which allow it to function as an efficient binary classifier. Its ability to aggregate multiple decision trees enhances predictive performance and reduces overfitting, making it well-suited for distinguishing between fraudulent and non-fraudulent transactions. The Logistic Regression model, known for its statistical reliability and interpretability, is also evaluated to provide a comparative benchmark.

3.2 Framework Development and Model Training

The framework for fraud detection follows a structured machine learning pipeline consisting of the following key stages:

1. Data Preprocessing – Credit card transaction data is extracted, cleaned, and preprocessed by handling missing values, removing duplicates, and normalizing feature distributions to improve model efficiency.
2. Data Partitioning – The dataset is divided into training and testing subsets, ensuring a balanced representation of fraudulent and legitimate transactions.
3. Model Training – Both Random Forest and Logistic Regression classifiers are trained using supervised learning on labeled transaction data to identify fraud patterns.
4. Prediction Generation – The trained models are applied to new transaction data, generating probability scores for fraud likelihood.
5. Threshold Application – A decision threshold is applied to classify transactions as fraudulent or non-fraudulent based on probability scores from each model.
6. Performance Evaluation – The models are evaluated using performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve) to determine their effectiveness in fraud detection.

3.3 Comparative Analysis and Fraud Detection Efficiency

The study aims to compare the performance of Random Forest and Logistic Regression models to determine the most effective approach for detecting fraudulent credit card transactions. The framework is designed to simulate real-world banking environments, enabling automated detection of unauthorized transactions in electronic payment systems.

By leveraging machine learning-based fraud detection, this study provides valuable insights into the strengths and weaknesses of different classification models. The results contribute to the ongoing development of more accurate and reliable fraud detection frameworks, equipping financial institutions with improved tools to mitigate fraud risks and enhance transaction security.

3.4 Analysis of the Existing System

The current credit card fraud detection systems employed by banks primarily rely on the verification of card information, such as the Card Verification Value (CVV), expiration date, and other details printed on the card. While this provides basic security measures, it is inadequate in preventing fraud once hackers obtain secure card details. Despite the use of online secure passwords for transactions, these measures do not verify the legitimacy of a transaction in real time. As a result, fraudulent transactions are often detected only after the fraud has occurred, typically when the cardholder reports the issue. This delay in detection leaves cardholders vulnerable, and the subsequent investigation process can be lengthy and inconvenient.

In the existing system, fraud detection is reactive rather than proactive. Although online transactions are routed through secure gateways, fraud detection is typically not integrated into the transaction process. Once credit card details are compromised—whether through data breaches, phishing, or other means—tracking and preventing fraudulent transactions becomes significantly more challenging. The reliance on post-transaction reporting means that fraud is not detected until after it has been committed, placing a significant burden on both the cardholder and the financial institution.

Moreover, the current system faces difficulties due to the rapid and widespread use of credit cards, both physically and online. Monitoring every transaction, especially in a global and digital context, is an increasingly complex task. While the capture of online transaction-related data such as IP addresses can assist in verifying suspicious activities, this alone is not sufficient for real-time fraud detection. Consequently, the financial institution often requires external intervention from cybercrime experts, which further complicates and delays the fraud investigation process.

In response to these limitations, some models have sought to integrate machine learning algorithms such as K-Nearest Neighbors (KNN) and Logistic Regression to detect fraudulent transactions more effectively. These systems aim to improve the detection process by analyzing transaction patterns and identifying anomalies that are indicative of fraud. While these algorithms offer a step forward in fraud detection by automating and enhancing the analysis of transaction data, the existing system still struggles with challenges such as false positives, imbalanced datasets, and the difficulty of processing large-scale, real-time transactions.

The core challenge of the existing system is that it remains largely dependent on post-transaction fraud detection methods. Even though machine learning can improve detection rates and reduce false alerts, it is still difficult to prevent fraud before it occurs, especially in the case of advanced fraud techniques such as card-not-present (CNP) fraud. Furthermore, many machine learning algorithms used in these systems rely on historical transaction data, which limits their ability to detect new or emerging fraud patterns in real-time.

In conclusion, the existing credit card fraud detection system has made significant strides in improving the identification of fraudulent transactions, primarily through machine learning techniques. However, it remains limited in its capacity to offer proactive, real-time fraud prevention. The system still relies on detecting fraud post-transaction and requires improvement in its integration of real-time analysis and broader security measures, such as IP address validation, behavioral analysis, and advanced anomaly detection. The existing approach needs further optimization to effectively combat the evolving landscape of credit card fraud.

3.5 Dataflow of the Existing System

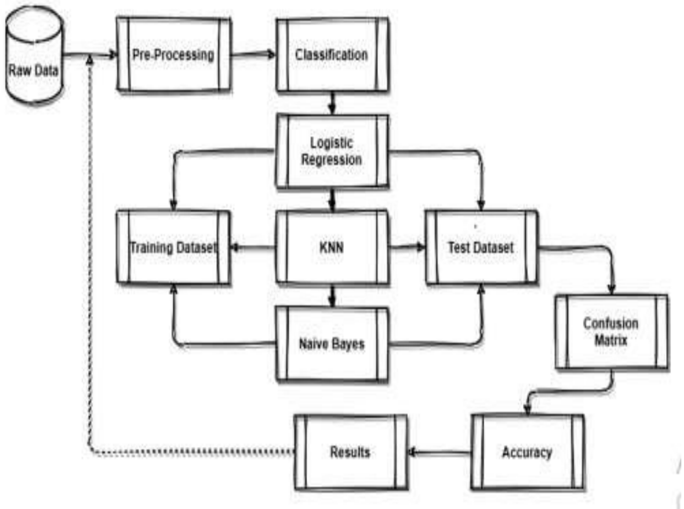


Figure 2: A dataflow of an Existing System

3.6 Flowchart of the Existing System

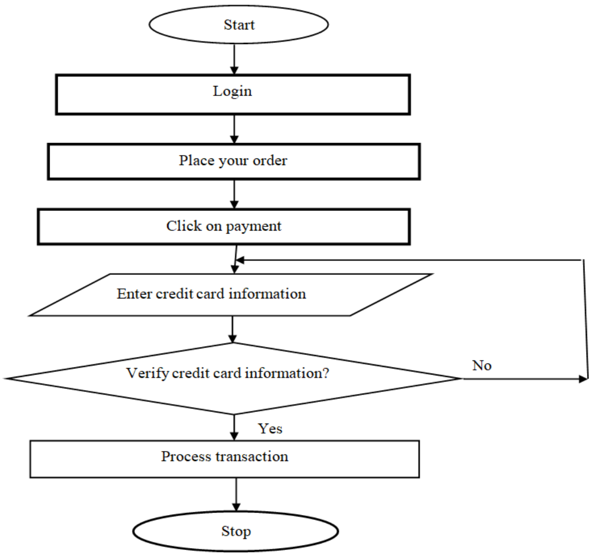


Figure 3: Flowchart of the Existing System

3.3 Analysis of the Proposed System

The proposed system utilizes a systematic methodology to detect credit card fraud using machine learning algorithms, specifically Random Forest Classifier (RFC) and Logistic Regression (LGR). The approach follows a well-structured, multi-stage process consisting of data collection, processing, feature engineering, model training, evaluation, and validation. Below is a detailed analysis of the steps involved in the proposed system.

A. Data Collection (Dataset)

The dataset used for this study is sourced from Kaggle, containing credit card transaction data from Europe. This dataset represents transactions over a two-day period and includes 492 fraudulent transactions. The features in the dataset are primarily derived from Principal Component Analysis (PCA), with attributes such as V1 to V28 capturing key characteristics of each transaction. The Time attribute tracks the number of seconds elapsed between each transaction, while the Amount attribute represents the transaction value. The Class feature is the target variable, indicating whether the transaction is fraudulent (1) or not (0). The sample-based method, using readily available datasets like those from Kaggle, is ideal as it allows for practical, large-scale testing with real-world data.

B. Data Processing and Reading

The first step in the data processing phase involves cleaning the dataset to address issues such as data anomalies, missing values, and redundant entries. By preparing the data for analysis, these challenges are mitigated to ensure the reliability and accuracy of the results. Once cleaned, the dataset is divided into training and testing sets to facilitate the evaluation of machine learning models. This division is crucial for ensuring that the model's performance is assessed on unseen data, preventing over-fitting and ensuring generalizability. Additionally, the research introduces Supervised Learning algorithms, namely Random Forest and Logistic Regression, as the core methods for classifying fraudulent transactions.

C. Histogram Check

A function is created to analyze the distribution of feature values over time. This histogram check helps visualize how the different features (such as the transaction amount and time) are distributed across the dataset. By doing so, it can highlight any patterns, anomalies, or imbalances in the data, particularly with respect to fraudulent transactions. This stage is essential for identifying potential outliers or skewed distributions that may affect the performance of the machine learning models.

D. Feature Engineering

Feature engineering involves selecting and constructing relevant features that contribute significantly to the detection of fraud. In this research, the primary task is to identify the key attributes that help distinguish between legitimate and fraudulent transactions. Features such as V1 to V28 and Amount are considered independent variables, while the Class feature serves as the dependent variable or target. Effective feature selection ensures that the machine learning models are trained with the most informative variables, which directly impacts the accuracy and effectiveness of fraud detection.

E. Model Training

Once the data is pre-processed and the features are engineered, the next step is to train the machine learning models. The dataset is divided into training and testing sets, with the training set used to teach the models how to detect fraudulent patterns in transaction data. Random Forest Classifier (RFC), an ensemble learning method, is particularly chosen for its ability to handle complex data structures and its robustness in classification tasks. Logistic Regression (LGR), a simpler but highly interpretable model, is used for comparison. These models are trained on the training set to learn the underlying patterns of fraud and non-fraud transactions.

F. Model Evaluation and Validation

After training, the models are evaluated on the testing set to assess their performance. Key evaluation metrics include precision, recall, and accuracy. Precision measures how many of the predicted fraudulent transactions were actually fraudulent, while recall assesses how many of the actual fraudulent transactions were correctly identified. Accuracy provides an overall measure of correct predictions across both fraudulent and legitimate transactions. These metrics are crucial in determining the models' effectiveness in detecting fraud, as high precision and recall are essential for minimizing false positives and false negatives. Model validation ensures that the system performs well on new, unseen data, providing a reliable and efficient fraud detection solution.

The proposed system provides a comprehensive framework for credit card fraud detection by integrating Random Forest and Logistic Regression models into a structured pipeline. By systematically collecting and processing data, performing feature engineering, and training machine learning models, the system is designed to detect fraud efficiently and accurately. The use of precision, recall, and accuracy as evaluation metrics ensures that the models are rigorously tested for reliability and effectiveness. With its emphasis on both performance and practicality, this system provides a robust approach to combating credit card fraud in real-world applications.

3.4 Dataflow of the Proposed System

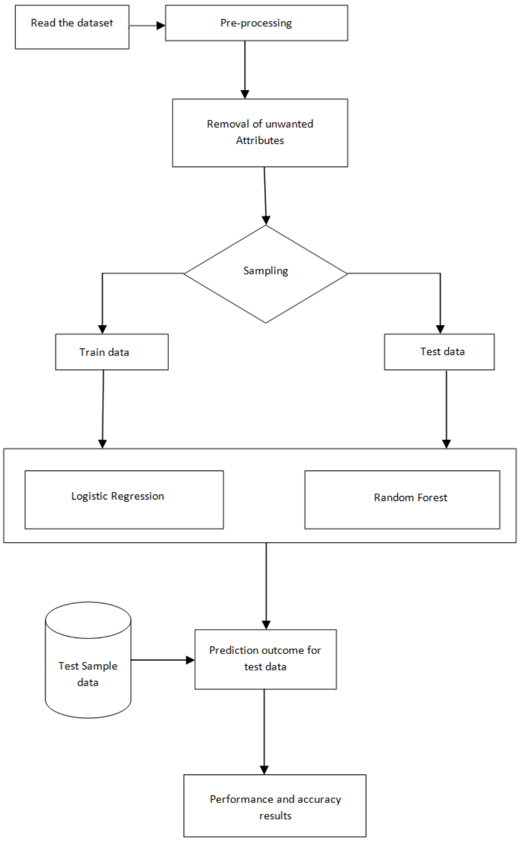


Figure 4: A Data Flow of the Proposed System

3.5 High Level Model of the Proposed System

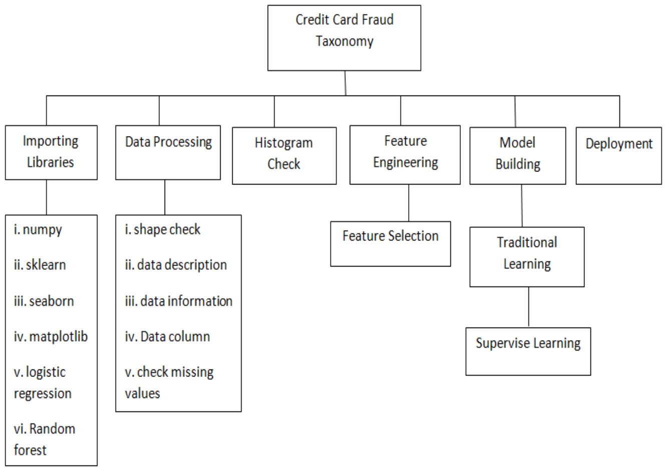


Figure 5: High Level Model of the Proposed System

3.6 Control Centre/Main Menu

The focus is primarily on the algorithms and the data processing. Designing a main menu for a credit card fraud detection system involves presenting key functionalities and options in a user-friendly and intuitive manner. Below is the main menu:

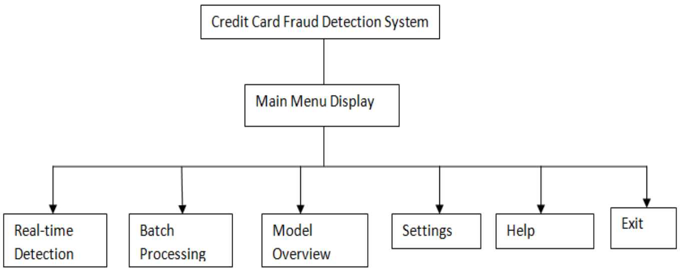


Figure 6: System Main Menus

3.7 The Submenus/Subsystem

3.7.1 Real-time Detection Subsystems and Brief Description of each item

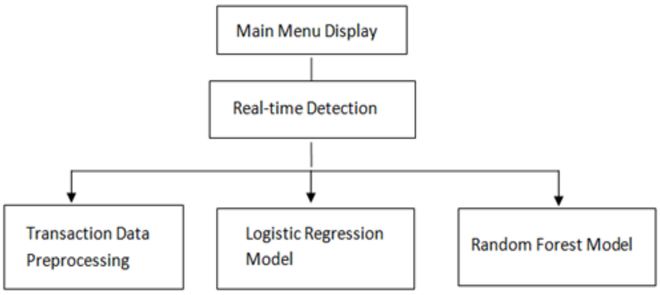


Figure 7: Subsystem for Real-time Detection Menu

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 52-66, March-2025
https://doi.org/10.47001/IRJIET/2025.903008

3.7.2 Input/output format

The input module has an input which is text provided in a form field. The following input is expected to be provided by the user:

i. Time
ii. v1, v2, v3, v4, v5, v6, v7 & v8
iii. Amount
iv. Class

The output module provides details if the transaction is fraudulent or not.

3.8 System Flowchart

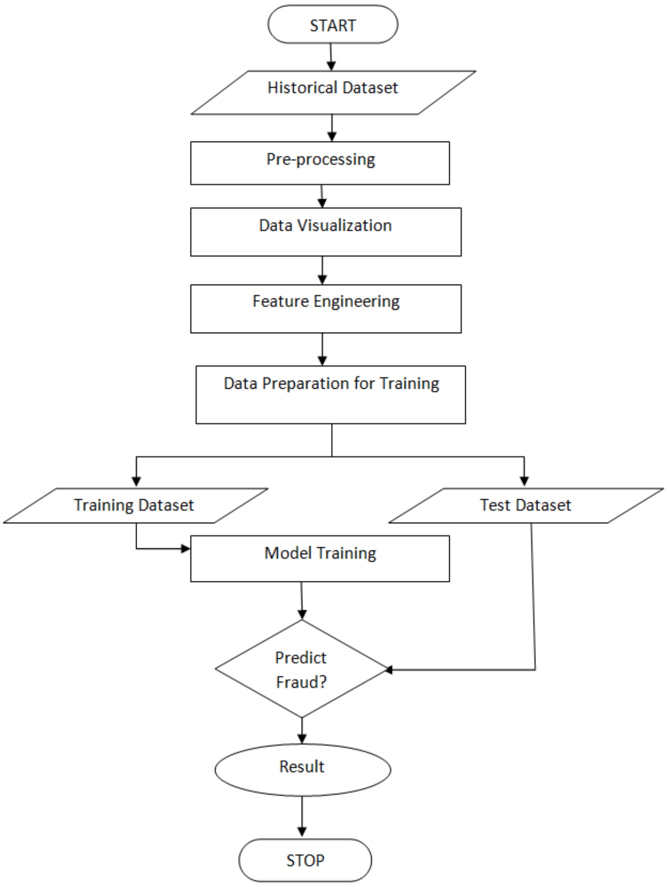


Figure 8: System Flowchart of the System

IV. SUMMARY

This study develops and implements a machine learning-based framework for credit card fraud detection, comparing Random Forest and Logistic Regression models. Addressing the limitations of traditional fraud detection methods, such as CVV verification, the research employs machine learning to proactively identify fraudulent transactions. The framework uses Random Forest for handling imbalanced datasets and capturing complex transaction patterns, and Logistic Regression to predict fraud by detecting data anomalies. Implemented with Jupyter Notebook, Visual Studio Code, and FastAPI, the system supports real-time data analysis and continuous monitoring. The study concludes that integrating machine learning algorithms significantly enhances fraud detection, offering a robust solution to reduce financial losses from credit card fraud.

4.1 Conclusion

Credit card fraud detection systems are essential for preventing fraudulent activities, with machine learning algorithms such as Random Forest and Logistic Regression proving highly effective in this area. Random Forest, using an ensemble of decision trees, excels in handling imbalanced datasets and accurately identifying fraud. Logistic Regression, a statistical method, predicts fraud by detecting patterns and anomalies in transaction data. A real-time fraud detection system is built using tools like Jupyter Notebook, Visual Studio Code, and FastAPI. The process involves data analysis, feature engineering, model training, and continuous deployment for effective fraud detection. Overall, integrating these machine learning techniques and tools significantly enhances fraud detection, minimizing risks and financial losses for organizations.

REFERENCES

[1] Aleskerov, E., Freisleben, B., AndRao. B(1997) Card watch: A Neural Network Based Database Mining System For Credit Card Fraud Detection, Proceedings of IEEE/Iafe: Computational Intelligence For Financial Eng.. Pp. 220-226.

[2] Anusiuba O. I. A, (2025), Design and Implementation of a Credit Card Fraud Detection System Using Random Forest and Logistics Regression Models, International Research Journal of Innovations in Engineering and Technology-IRJIET, 9(2),152-166

[3] Anusiuba O. I. A, Okechukwu O. P., Ekwealor O. U, Anusiuba A. A.,(2022), The Application of Hidden Markov Model in Credit Card Fraud Detection System, iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE), 10(2), 1-20

[4] Brause, R., Langsdorf, T., Flepp, M. (1999), Neural Data Mining for Credit Card Fraud Detection, International Conference on Tools with Artificial Intelligence, IEEE, pp. 103-106.

[5] Chiu C., and Tsai, C., (2004). A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection, Proceedings of IEEE International Conference e-Technology, e-Commerce and e-Service. pp. 177-181.

[6] Dornadul, V.N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithm 165 631-641 http://creativecommons.org/licenses/by-nc-nd/4.0/

[7] Ekwealor O.U, Anusiuba O. I. A, Ezuruka E. O, Uchefuna C.I., (2021). An Intelligent Credit Card Fraud Detection System, iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE) 9(2), 25-54

[8] Ghosh, S., Reilly, D.L. (1994). Credit Card Fraud Detection with a Neural- Network, Proc. 27th Hawaii Int'l Conf. System Sciences: Information Systems: Decision Support And Knowledge-Based Systems, Vol. 3, Pp. 621-630.

[9] Kim, M.T. and Kim. T.S. 2002. A Neural Classifier with Fraud Density Map for Effective Credit Card Fraud Detection, Proc. International Conference on Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, Springer Verlag, no. 2412, pp. 378-383.

[10] Kurshan, E., Shen, H., & Yu, H. (2020). Financial crime & fraud detection using graph computing: Application considerations & outlook, in: 2020 Second International Conference on Transdisciplinary AI (TransAI), IEEE, pp. 125–130.

[11] Lebichot, B., Siblini, G.M.P.W., & Bontempi, L.H.F.O.G. (2021). Incremental learning strategies for credit cards fraud detection: International Journal of Data Science and Analytics, 12(2), 165–174.

[12] Phua, C., Lee. V.. Smith. K. and Gayler R (2007) . A comprehensive Survey of Data Mining based Fraud Detection Research, Pp. 190-200.

[13] Sulaiman, B.R., Schetinin, V. & Sant, P. (2022). Review of Machine Learning Approach on Credit Card Fraud Detection: Human-Centered Intelligent Systems, 2, 55–68. https://doi.org/10.1007/s44230-022-00004-0

[14] Syeda. M. Zhang. Y.Q., Pan, Y. (2002). Parallel Granular Networks for Fast Credit Card Fraud Detection, Proc. IEEE Int'l Conf. Fuzzy Systems, Pp. 572-577.

[15] Stolfo, S. J., Fan. D. W, Lee, W,, Prodrorakis. A. And Chan, P. K.. (2000). Cost-Based Modeling for Fraud and Intrusion Detection Results from the Jam Project, Proceedings of Darpa Information Survivability Conference And Exposition Vol. 2 Pp. 130-144.

[16] Vatsa.Y..Sural, S. and Majumdar, A.K.(2005). A Game-theoretic Approach to Credit Card Fraud Detection, Proc. 1st International Conference on Information Systems Security, Lecture Notes in Computer Science, Springer Verlag, pp. 263-276.

Citation of this Article:

Anusiuba, Overcomer Ifeanyi Alex. (2025). Development and Implementation of a Machine Learning-Based Framework for Credit Card Fraud Detection: A Comparative Study of Random Forest and Logistic Regression Models. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(3), 52-66. Article DOI https://doi.org/10.47001/IRJIET/2025.903008

APPENDIX A

SOURCE CODE LISTING (PYTHON PROGRAMMING LANGUAGE)

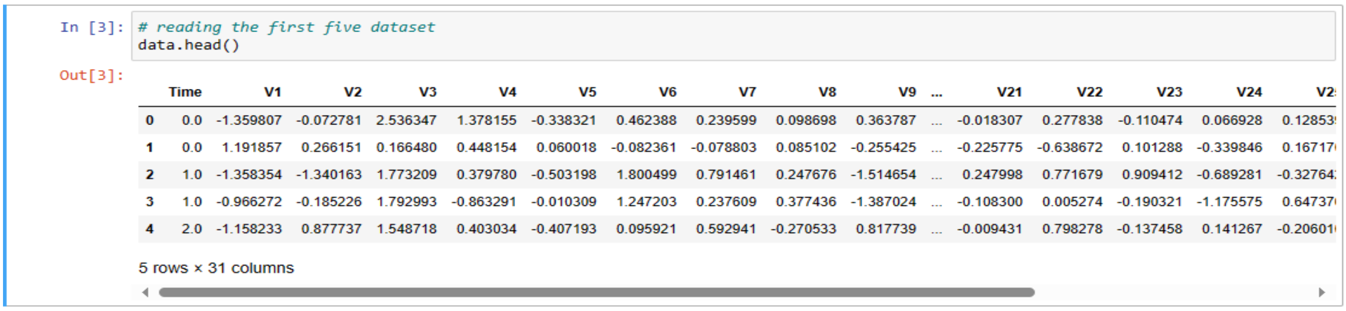
```
# Importing libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
# loading the dataset
data = pd.read_csv('creditcard.csv')
# reading the first five dataset
data.head()
# Data preprocessing
#checking for shape
data.shape
data.columns
```

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 52-66, March-2025
https://doi.org/10.47001/IRJIET/2025.903008

```
data.describe()
#checking for value_counts
data['Class'].value_counts()
# checking for info
data.info()
# checking for missing values
sns.heatmap(data.isna(),yticklabels=False,cbar=False,cmap='viridis')
data.isnull().sum()
# Historical check
def draw_histograms(dataframe, features, rows, cols):
    fig=plt.figure(figsize=(20,20))
    for i, feature in enumerate(features):
        ax=fig.add_subplot(rows, cols, i + 1)
        dataframe[feature].hist(bins = 20, ax = ax, facecolor = 'midnightblue')
        ax.set_title(feature +'Distribution', color='DarkRed')
        ax.set_yscale('log')
    fig.tight_layout()
    plt.show()
draw_histograms(data, data.columns, 8, 4)
# Feature engineering
## independent and dependent features
X = data.drop('Class',axis=1)
y = data.Class
#Model Building
# Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.model_selection import KFold
import numpy as np
from sklearn.model_selection import GridSearchCV
log_Class=LogisticRegression()
grid={'C':10.0**np.arange(-2,3),'penalty':['l1','l2']}
cv=KFold(n_splits=5,random_state=None,shuffle=False)
from sklearn.model_selection import train_test_split
```

APPENDIX B

SAMPLE OUTPUT



```
In [3]: # reading the first five dataset
        data.head()
```

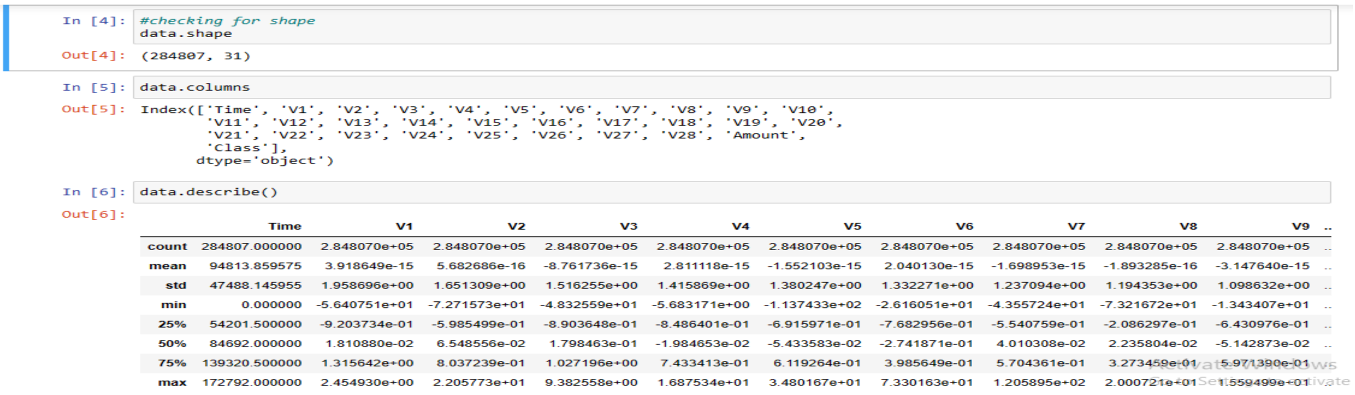
Out[3]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

5 rows × 31 columns

Plate 1: Reading the first five dataset

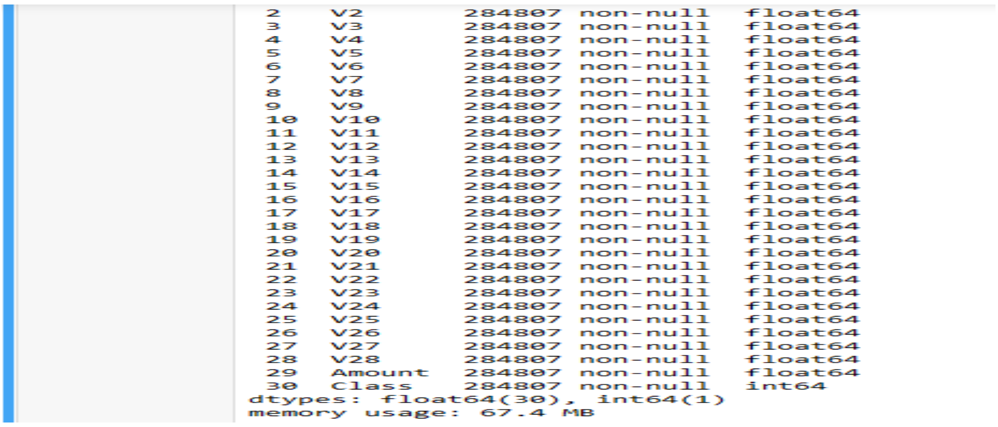
ISSN (online): 2581-3048
Volume 9, Issue 3, pp 52-66, March-2025
https://doi.org/10.47001/IRJIET/2025.903008



```
In [4]: #checking for shape
        data.shape

Out[4]: (284807, 31)

In [5]: data.columns

Out[5]: Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
               'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
               'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
               'Class'],
              dtype='object')

In [6]: data.describe()

Out[6]:
```

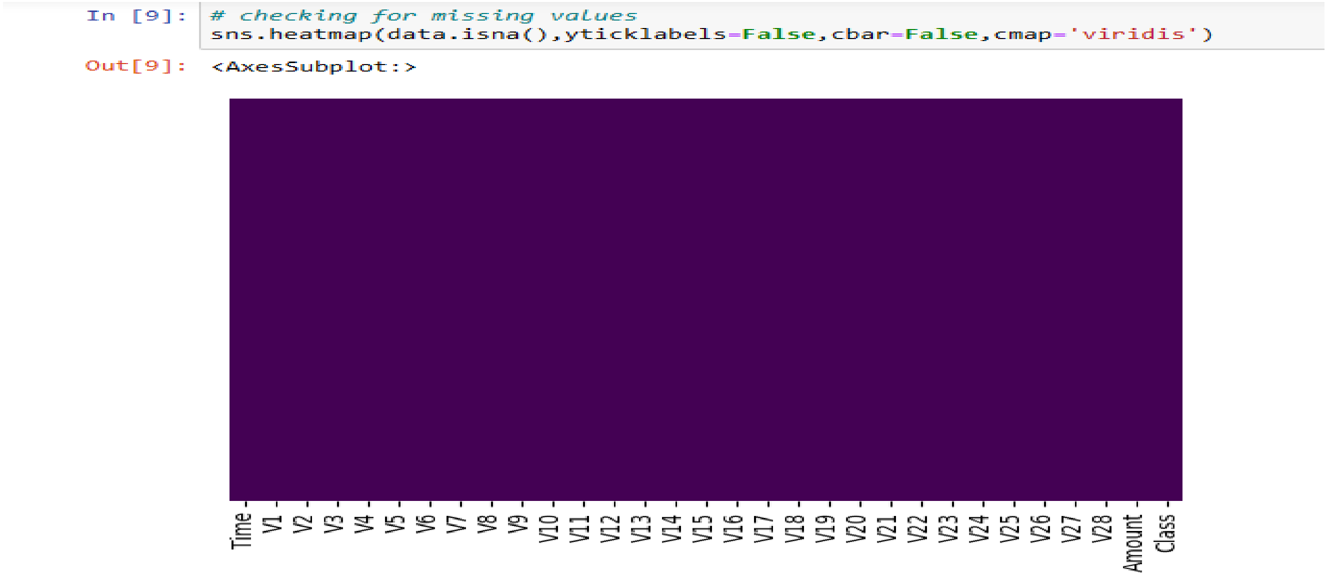
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	..
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	..
mean	94813.859575	3.918649e-15	5.682686e-16	-8.761736e-15	2.811118e-15	-1.552103e-15	2.040130e-15	-1.698953e-15	-1.893285e-16	-3.147640e-15	..
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	..
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	..
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	..
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	..
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	..
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	..

Plate 2: Data Preprocessing



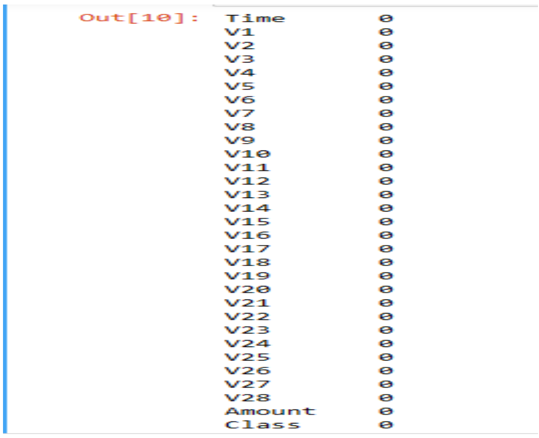
```
 2   V2      284807 non-null   float64
 3   V3      284807 non-null   float64
 4   V4      284807 non-null   float64
 5   V5      284807 non-null   float64
 6   V6      284807 non-null   float64
 7   V7      284807 non-null   float64
 8   V8      284807 non-null   float64
 9   V9      284807 non-null   float64
 10  V10     284807 non-null   float64
 11  V11     284807 non-null   float64
 12  V12     284807 non-null   float64
 13  V13     284807 non-null   float64
 14  V14     284807 non-null   float64
 15  V15     284807 non-null   float64
 16  V16     284807 non-null   float64
 17  V17     284807 non-null   float64
 18  V18     284807 non-null   float64
 19  V19     284807 non-null   float64
 20  V20     284807 non-null   float64
 21  V21     284807 non-null   float64
 22  V22     284807 non-null   float64
 23  V23     284807 non-null   float64
 24  V24     284807 non-null   float64
 25  V25     284807 non-null   float64
 26  V26     284807 non-null   float64
 27  V27     284807 non-null   float64
 28  V28     284807 non-null   float64
 29  Amount  284807 non-null   float64
 30  Class   284807 non-null   int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Plate 3: Checking for data types



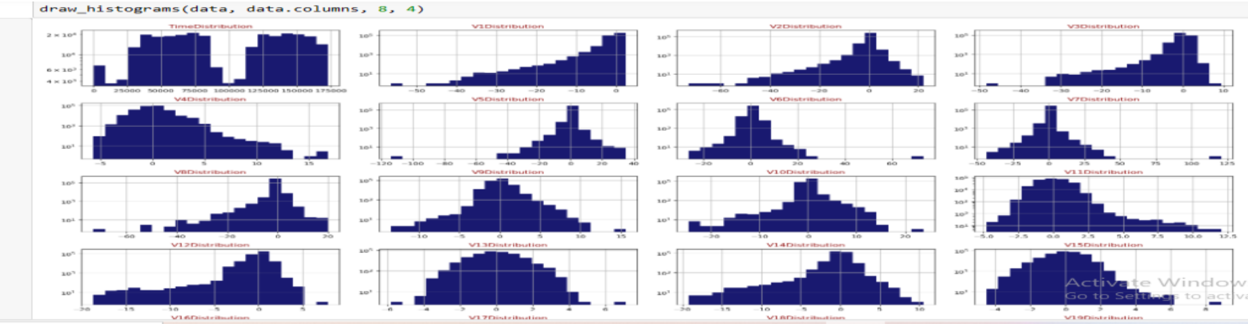
```
In [9]: # checking for missing values
        sns.heatmap(data.isna(),yticklabels=False,cbar=False,cmap='viridis')

Out[9]: <AxesSubplot:>
```

Plate 4: Checking for missing values



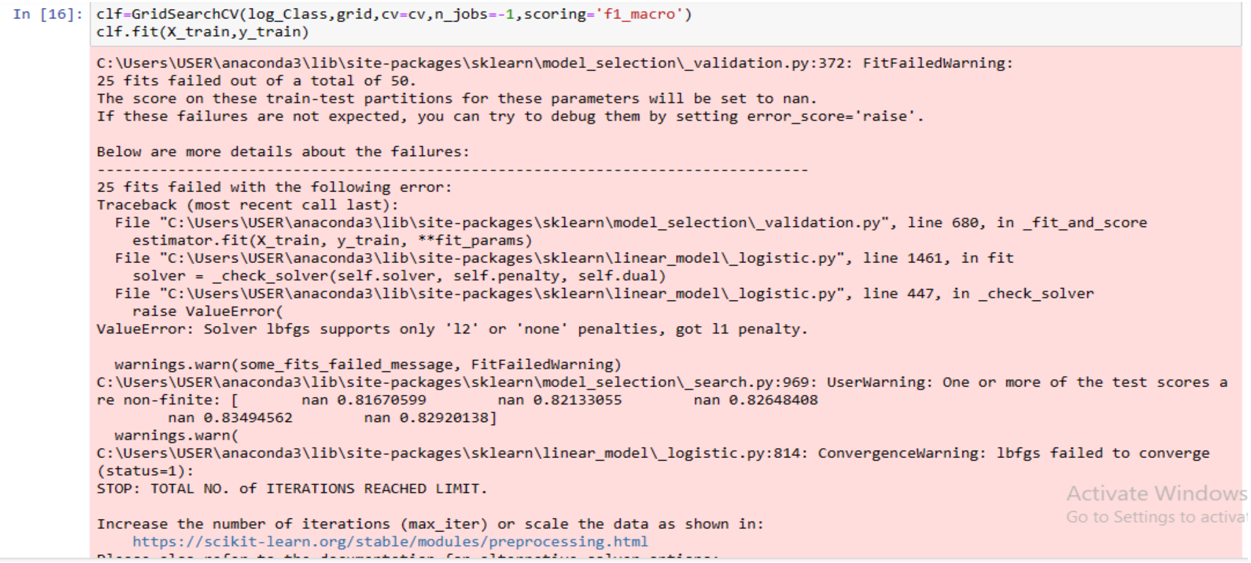
```
Out[10]:  Time      0
          V1        0
          V2        0
          V3        0
          V4        0
          V5        0
          V6        0
          V7        0
          V8        0
          V9        0
          V10       0
          V11       0
          V12       0
          V13       0
          V14       0
          V15       0
          V16       0
          V17       0
          V18       0
          V19       0
          V20       0
          V21       0
          V22       0
          V23       0
          V24       0
          V25       0
          V26       0
          V27       0
          V28       0
          Amount    0
          Class     0
```

Plate 5: Checking missing values



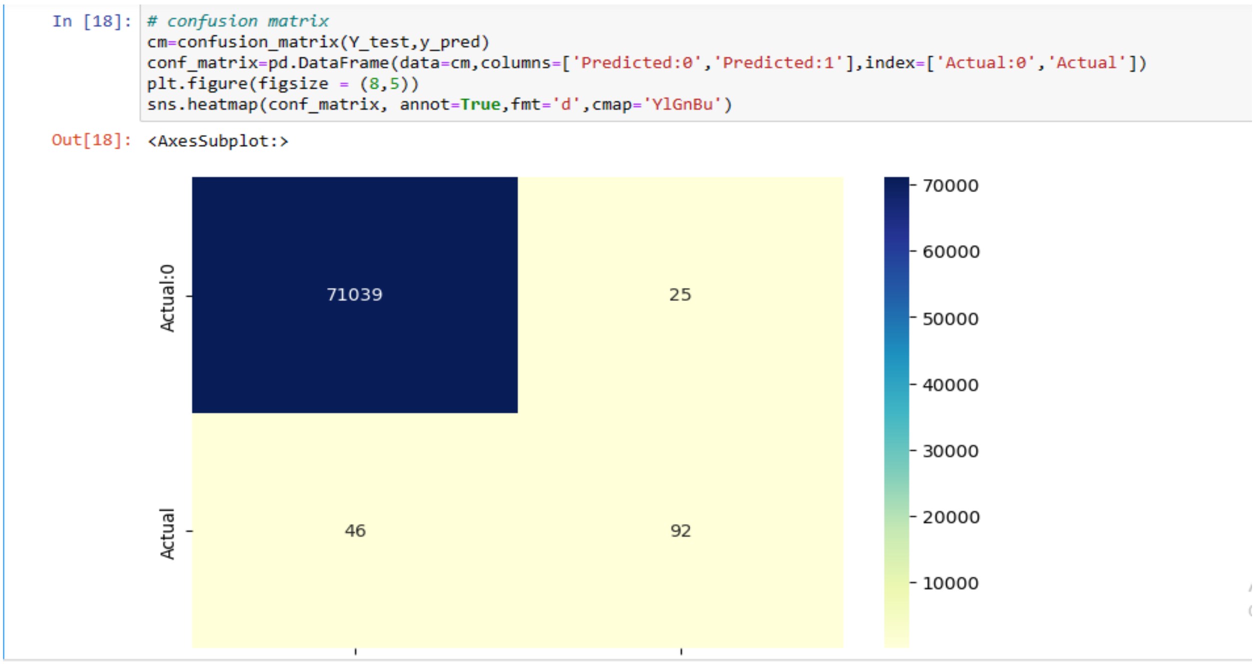
```
draw_histograms(data, data.columns, 8, 4)
```

Plate 6: Histogram check



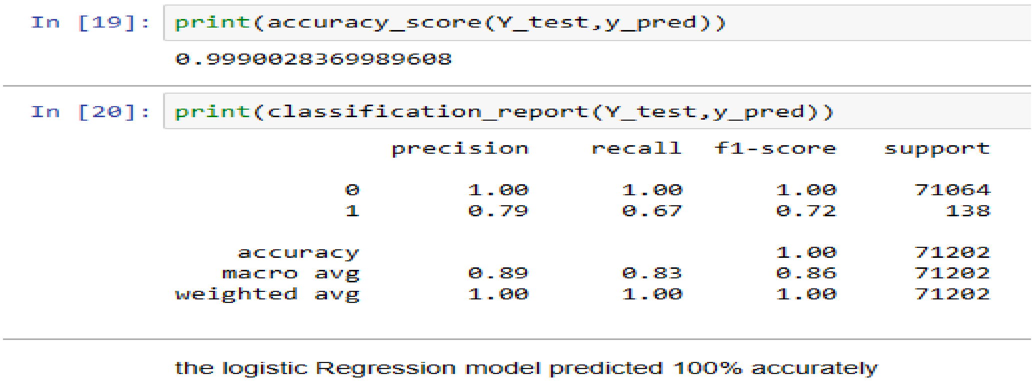
```
In [16]:  clf=GridSearchCV(log_Class,grid,cv=cv,n_jobs=-1,scoring='f1_macro')
          clf.fit(X_train,y_train)

C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:372: FitFailedWarning:
25 fits failed out of a total of 50.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
25 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py", line 680, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\USER\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py", line 1461, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\USER\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py", line 447, in _check_solver
    raise ValueError(
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:969: UserWarning: One or more of the test scores a
re non-finite: [       nan 0.81670599        nan 0.82133055        nan 0.82648408
        nan 0.83494562        nan 0.82920138]
  warnings.warn(
C:\Users\USER\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```

Plate 7: Feature Engineering



```
In [18]: # confusion matrix
         cm=confusion_matrix(Y_test,y_pred)
         conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual'])
         plt.figure(figsize = (8,5))
         sns.heatmap(conf_matrix, annot=True,fmt='d',cmap='YlGnBu')

Out[18]: <AxesSubplot:>
```

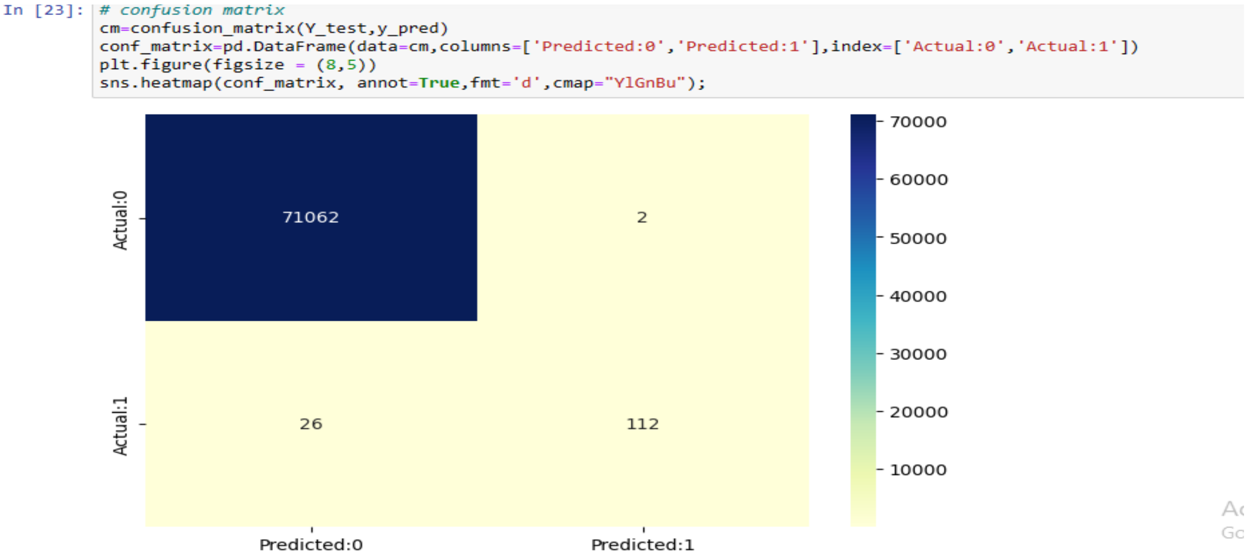
Plate 8: Validation of logistic regression model



```
In [19]: print(accuracy_score(Y_test,y_pred))

         0.9990028369989608

In [20]: print(classification_report(Y_test,y_pred))

                       precision    recall  f1-score   support

                    0       1.00      1.00      1.00     71064
                    1       0.79      0.67      0.72       138

             accuracy                           1.00     71202
            macro avg       0.89      0.83      0.86     71202
         weighted avg       1.00      1.00      1.00     71202
```

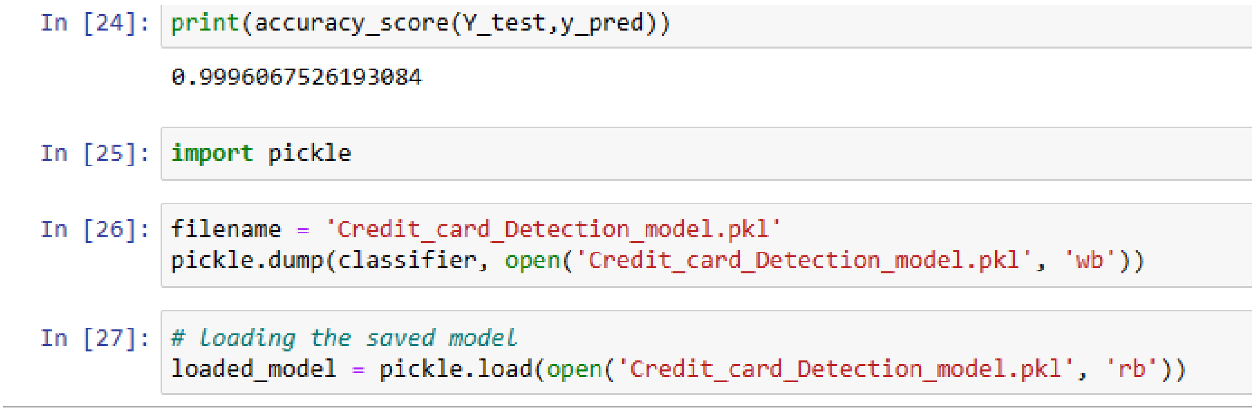
the logistic Regression model predicted 100% accurately

Plate 9: Accuracy of the logistic regression model which gives 100% accuracy



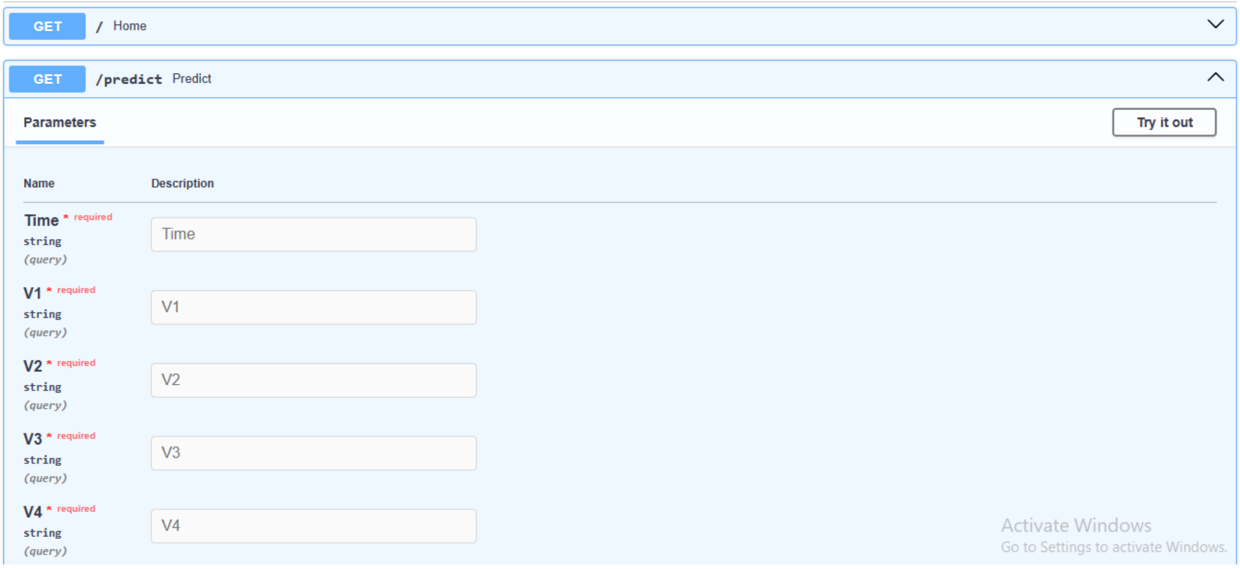
```
In [23]: # confusion matrix
         cm=confusion_matrix(Y_test,y_pred)
         conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])
         plt.figure(figsize = (8,5))
         sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu");
```

Plate 10: Validation of random forest model



```
In [24]: print(accuracy_score(Y_test,y_pred))

         0.9996067526193084

In [25]: import pickle

In [26]: filename = 'Credit_card_Detection_model.pkl'
         pickle.dump(classifier, open('Credit_card_Detection_model.pkl', 'wb'))

In [27]: # loading the saved model
         loaded_model = pickle.load(open('Credit_card_Detection_model.pkl', 'rb'))
```

Plate 11: Accuracy of the random forest model which gives 100% accuracy



Name | Description

Time * required
string
(query)
Time

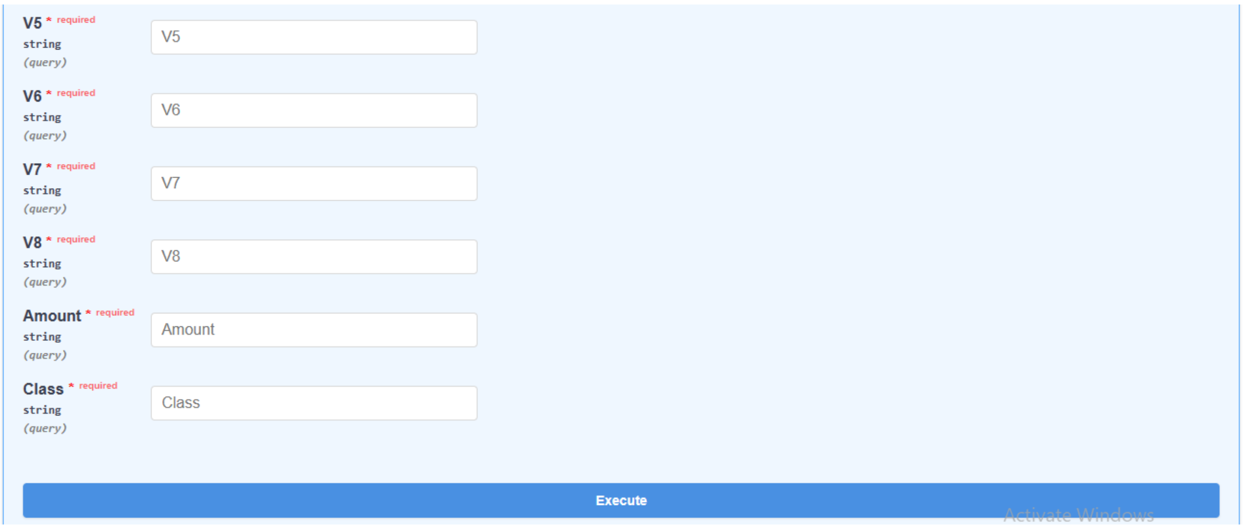
V1 * required
string
(query)
V1

V2 * required
string
(query)
V2

V3 * required
string
(query)
V3

V4 * required
string
(query)
V4

Plate 12: Deployment Stage



V5 * required
string
(query)
V5

V6 * required
string
(query)
V6

V7 * required
string
(query)
V7

V8 * required
string
(query)
V8

Amount * required
string
(query)
Amount

Class * required
string
(query)
Class

Plate 13: Deployment Stage

****