# A PROJECT REPORT

# ON

# “CREDIT CARD FRAUD DETECTION”

# SUBMITTED TO

# KIIT DEEMED TO BE UNIVERSITY

# 

**In Partial Fulfillment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE**

# BY

***Divyesh Kulshreshtha 21052757***

***Jhanvi Jain 21053233***

***Ankita Singh 21053235***

***Somya Sinha 21053365***

***Ayush Pathak 21052747***

**UNDER THE GUIDANCE OF**

**MR. MOHIT RANJAN PANDA**



**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

**May 2024**

# CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

------------------------------------------------------------------------------------------------------------------------------------------

***Abstract- Credit card fraud is a major threat, causing financial losses. This project explores using machine learning in Python to tackle this issue. We'll leverage libraries like Pandas for data cleaning and analysis.[1] By building models like Logistic Regression and Random Forests, we aim to predict fraudulent transactions in real-time. Challenges include the massive amount of data processed daily and the imbalanced nature (mostly legitimate transactions). We'll address these by using efficient models and techniques to handle imbalanced data.[4] The goal is to achieve high accuracy with minimal false positives while understanding the hallmarks of fraudulent activity. This project contributes to developing robust fraud detection systems, offering valuable tools for financial institutions to combat evolving scams and ensure secure transactions.***

***Keywords: Logistic Regression algorithm, Criminal transactions, Credit card, Real-time Model Evaluation***

**1. INTRODUCTION**

The rise of digital transactions has brought a double-edged sword: convenience accompanied by a significant increase in credit card fraud. This poses a threat to both consumers, who can be unfairly charged for items they never purchased, and financial institutions, who face financial losses. Traditional methods struggle to keep pace with the ever-evolving tactics of fraudsters. Machine learning (ML) offers a powerful solution to this problem. By analyzing historical transaction data, ML models can learn to identify patterns and anomalies that signal fraudulent activity. This allows for real-time detection of fraud, improved accuracy in flagging suspicious transactions as new schemes emerge, and ultimately, a reduction in risk for both consumers and financial institutions. The project aims to develop a credit card fraud detection system using Python and machine learning algorithms. This system will analyze features like transaction amount, location, and spending habits to distinguish legitimate purchases from suspicious activity. By continuously learning from new data, the system can stay ahead of evolving fraudster techniques, ultimately creating a more secure digital payments landscape.

**1.1. Advantages:**

Advantages of the Credit Card Fraud Detection Implementation Using Logistic Regression:

1. Efficient Data Handling: The code efficiently handles data using the Pandas library, enabling easy loading, manipulation, and analysis of datasets.
2. Model Evaluation Metrics: It computes various evaluation metrics such as accuracy, precision, recall, and F1-score through the classification\_report() function, providing a comprehensive understanding of model performance.
3. Data Visualization: The code utilizes Matplotlib and Seaborn libraries to create visualizations like bar graphs and confusion matrices, aiding in the interpretation and communication of results.
4. Modular Code Structure: The code is organized into logical sections, making it easy to understand and maintain. Each section focuses on a specific task such as data preprocessing, model training, prediction, and evaluation.
5. Machine Learning Pipeline: It implements a standard machine learning pipeline including data preprocessing, model training, prediction, and evaluation. This standardized approach simplifies the development and deployment of machine learning models.
6. Scalability: The code can be easily scaled to handle larger datasets or incorporate more complex machine-learning algorithms by making minor modifications or additions.
7. Reproducibility: By setting a random seed (random\_state) in the train-test split, the code ensures reproducibility of results, allowing others to replicate the experiments.
8. Interpretability: The code generates a confusion matrix, which provides insights into the model's performance by showing true positives, true negatives, false positives, and false negatives. This aids in understanding where the model excels and where it struggles.
9. Model Selection: Although the code employs a logistic regression model, it can be easily modified to experiment with other algorithms available in scikit-learn, facilitating model selection and comparison.
10. Diagnostic Information: The code prints diagnostic information such as the number of actual and predicted fraud cases, which can be useful for further analysis or business decision-making.

**1.2. Scope of proposed work:**

This project proposes an automated credit card fraud detection system using machine learning. We'll design a model to analyze key transaction features like amount, location, and spending habits. These features become the basis for distinguishing legitimate purchases from fraud. Traditional methods struggle to adapt to new fraud tactics, but machine learning excels here. By continuously training on fresh data, our system can identify emerging patterns that might slip past static rules. This translates to automation, freeing up resources, and superior accuracy due to machine learning's adaptability. The project involves feature engineering, model selection, data training, evaluation, and integration into a real-time system for automatic fraud flagging. Ultimately, this approach aims to build a robust and automated shield against evolving fraud, protecting both institutions and cardholders.

**2. PROBLEM STATEMENT**

**2.1. Project planning:**

* Goal: Build a credit card fraud detection model using machine learning.
* Data: "creditcard.csv" containing transaction information.
* Clean and pre-process data.
* Split features (X) and target (y).
* Train the Logistic Regression model.
* Evaluate model performance (accuracy, report, confusion matrix). Visualize results.
* Outcomes: Trained model, performance metrics, visualizations.
* Contingency: Address data quality, and explore alternative models if needed.

**2.2. Project Analysis:**

* Strengths: Clear goals, transparent code, baseline performance, modular design.
* Weaknesses: Limited data handling, simple algorithm, basic features, narrow evaluation, deployment not considered.
* Opportunities: Enhance data quality, explore better algorithms, advanced feature engineering, comprehensive evaluation metrics, and deployment possibilities.
* Threats: Data availability, achieving high accuracy, computational demands, privacy concerns.
* Recommendations: Focus on data quality, explore alternatives, leverage feature engineering, use diverse evaluation metrics, consider deployment, and ensure data privacy.

**2.3. System design:**

* Data Acquisition: Load credit card transaction data (CSV).
* Preprocessing (Optional): Remove duplicates (if needed).
* Data Exploration: Analyze initial data (e.g., first rows, data shape).
* Feature Engineering (Assumed): Data is assumed preprocessed with relevant features for fraud detection.
* Separate Features & Target: Split data into features (X) and target variable (y) indicating fraud (1) or legitimate (0) transactions.
* Model Training & Evaluation:
* - Split data into training and testing sets (80%/20%).
* - Train a Logistic Regression model to predict fraud on the training data.
* - Evaluate model performance on testing data using accuracy, classification report, and confusion matrix.

**3. LIBRARY PACKAGES**

**Pandas**: For data manipulation and preprocessing.

**Scikit-learn**: For implementing machine learning algorithms such as regression and anomaly detection.

**Matplotlib and Seaborn**: For data visualization and exploratory data analysis.

**3.1. Package Modules:**

* DATA COLLECTION.
* DATA PRE-PROCESSING.
* MODEL TRAINING / FEATURE EXTRACTION.
* EVALUATION MODEL.

**4. DATASET**

The dataset provided contains anonymized credit card transactions labeled as fraudulent or non-fraudulent (class 0 and class 1, respectively). Each transaction is described by 28 numerical features derived from PCA transformation for confidentiality reasons. The time variable indicates the seconds elapsed between transactions. The 'Amount' variable represents the transaction amount. This dataset aims to aid in credit card fraud detection research by providing a diverse set of features and corresponding labels, enabling the development and evaluation of machine learning models for effectively distinguishing between genuine and fraudulent transactions, thereby enhancing fraud detection systems' accuracy and reliability.

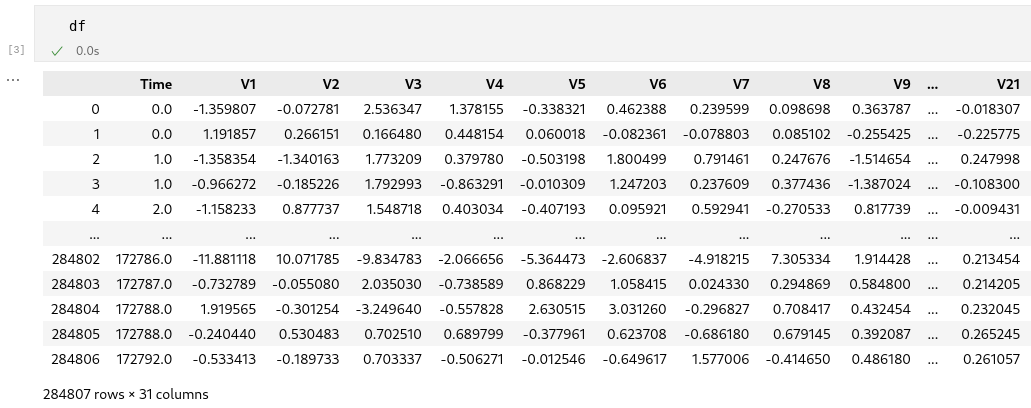
**5. STEP-WISE IMPLEMENTATION**

**5.1. Importing Essential Assets:** The libraries used in the model are pandas, sci-kit learn, matplotlib, and seaborn, from sklearn we fetch accuracy\_score, classification\_report, confusion\_matrix,

logistic regression, and train\_test\_split function.

**5.2. Loading the Data:** Loading the Dataset as a data frame makes the overall data easy to manipulate and access at any point.

**5.3. Data Exploration:** Viewing the Data to sort necessary resources out from it.



**5.3.1: Removing duplicate rows:**

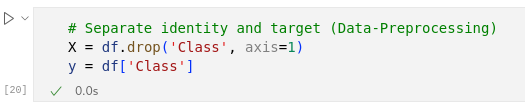
Duplicate entries skew the integrity of the dataset by inflating the frequency of certain observations. In fraud detection, where accurate representation of transactions is paramount, maintaining data integrity is critical. Duplicate entries can introduce bias into the model training process, as they artificially increase the importance of certain data points. This can lead to overfitting and reduce the model's ability to generalize to unseen data.

**5.4. Data Pre-Processing:**

**5.4.1. Separating the target variable (‘class’) from all the features:**

In the machine learning model, you typically need to separate the input features (X) and the target variable (y) before training a model. The X dataframe contains the independent variables or features that the model will use to make predictions, while the y Series contains the corresponding labels or target values that the model should learn to predict. In the credit card fraud detection dataset, X might contain columns like transaction amount, time, location, etc., while y would contain the 'Class' column indicating whether a transaction is fraudulent (1) or not (0).

After separating X and y, you can split the data into training and testing sets, train a machine learning model on the training data (X\_train and y\_train), and evaluate its performance on the testing data (X\_test and y\_test).



**5.5. Model Building and Training:**

The given code is a Python script that performs logistic regression on a credit card fraud detection dataset using scikit-learn. It starts by importing the required libraries, loading the dataset into a Pandas DataFrame, and handling any duplicate rows. The script then separates the features (X) and the target variable (y) from the dataset. Next, it splits the data into training and testing sets using the `train\_test\_split` function. A logistic regression model is created and trained on the training data. The trained model is then used to make predictions on the testing data. The code counts the number of actual frauds and predicted frauds in the testing set, assuming 0 represents non-fraud. Finally, it displays the number of actual frauds, predicted frauds, and the total number of predictions made by the model. Additionally, the script calculates and prints the model's accuracy score and classification report, and visualizes the results using a bar chart and a confusion matrix heatmap.

**5.6. Model Evaluation:**

The code calculates the accuracy score of the logistic regression model by comparing the true labels (y\_test) with the predicted labels (y\_pred) using the accuracy\_score function from sklearn.metrics. The resulting accuracy score, a value between 0 and 1, represents the overall fraction of correctly predicted instances. Additionally, the code prints the calculated accuracy score to provide an overall measure of the model's performance on the testing set. Furthermore, it generates and prints a detailed classification report using the classification\_report function, which provides metrics such as precision, recall, f1-score, and support for each class. This classification report offers more comprehensive insights into the model's performance, particularly in cases of imbalanced datasets or when the cost of different types of errors varies. In the context of credit card fraud detection, the classification report helps evaluate how well the model identifies fraudulent and non-fraudulent transactions, as well as the trade-off between precision and recall for each class.

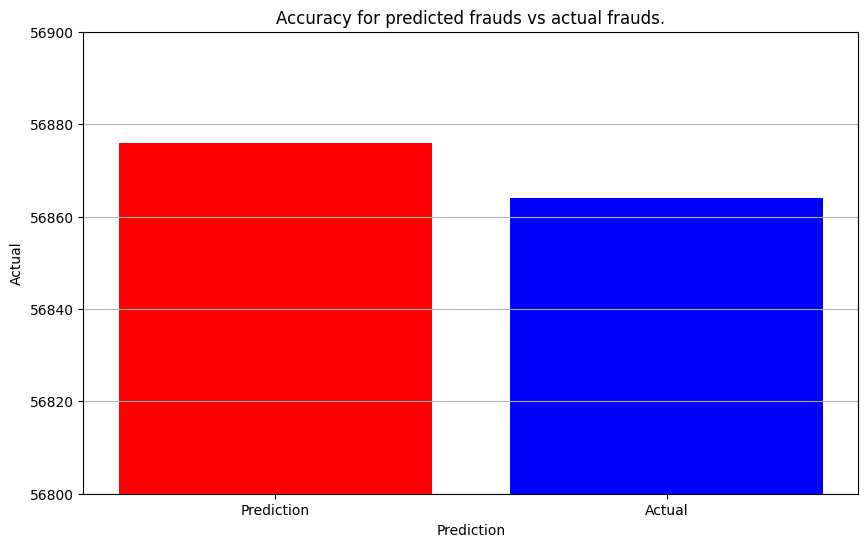
**6. ALGORITHM**

**6.1. Logistic Regression:**

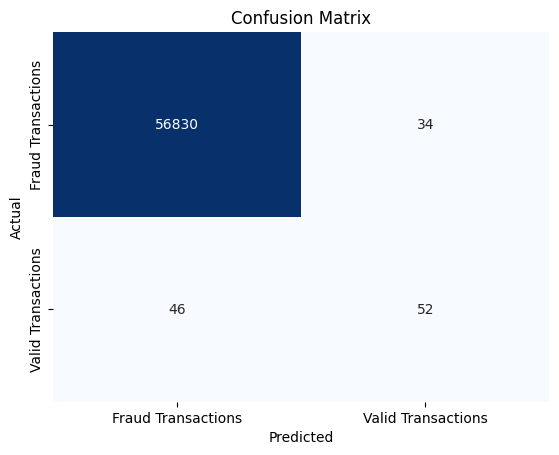
Logistic regression stands out as a valuable tool in credit card fraud detection due to its interpretability, efficiency, and scalability. With the ability to provide easily interpretable results, logistic regression allows financial institutions to understand the factors influencing fraudulent activities, as coefficients associated with each feature indicate their impact on the likelihood of fraud. Moreover, logistic regression's computational efficiency makes it well-suited for processing the massive amounts of transaction data encountered in fraud detection, facilitating real-time analysis. Its scalability ensures that the model remains effective even as transaction volumes increase over time. Additionally, logistic regression's robustness to noise and irrelevant features in the dataset ensures that it can effectively filter out irrelevant information, focusing on features most indicative of fraudulent behavior. Furthermore, logistic regression's probabilistic output enables financial institutions to assess the likelihood of fraud for each transaction, facilitating prioritization for further review or investigation. Lastly, logistic regression's ability to handle imbalanced data through techniques such as adjusting decision thresholds or using class weights ensures that the model adequately captures fraudulent activities while minimizing false positives. Overall, logistic regression offers a powerful and interpretable approach to credit card fraud detection, empowering financial institutions to identify and prevent fraudulent transactions effectively.

**7. MODEL’S ACCURACY**

Accuracy provides a straightforward measure of how well a model is performing, but it should be interpreted in conjunction with other evaluation metrics to gain a more nuanced understanding of the model's strengths and weaknesses.



***A confusion matrix*** serves as a critical evaluation tool. By analyzing the model's predictions against the actual labels in the test dataset, the confusion matrix illustrates the classification performance. Specifically, it captures the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model. This granular breakdown enables a thorough assessment of the model's accuracy and ability to correctly classify fraudulent and non-fraudulent transactions. Visualization of the confusion matrix, typically in the form of a heatmap, offers a clear depiction of the model's classification performance, aiding in the identification of areas for improvement and optimization. Through the analysis of the confusion matrix, practitioners can refine the model's predictive capabilities, ultimately enhancing its effectiveness in detecting credit card fraud.



**8. CONCLUSION**

In conclusion, the rising prevalence of credit card fraud in tandem with the increasing digitization of financial transactions underscores the urgency for robust fraud detection systems. This project endeavors to address this pressing issue by harnessing the power of machine learning algorithms in Python.

Through the utilization of techniques such as Logistic Regression and Random Forests, alongside libraries like Pandas for data pre-processing and analysis, our project aims to enhance the security measures employed by financial institutions. By leveraging historical transaction data, our models seek to discern patterns indicative of fraudulent activity, thereby enabling real-time detection and prevention of unauthorized transactions.

Despite the challenges posed by the massive volume of daily transactions and the imbalanced nature of legitimate versus fraudulent activities, our approach emphasizes the deployment of efficient models and techniques tailored to handle such complexities. By striving for high accuracy while minimizing false positives, we aim to provide valuable tools for financial organizations to combat evolving fraud schemes and safeguard their customers' financial interests.

In a rapidly evolving landscape where cyber threats continue to evolve, the insights gleaned from this project contribute towards the development of more resilient fraud detection systems. By staying proactive and adaptable, we can help mitigate financial losses and bolster consumer confidence in digital payment systems, ultimately fostering a safer and more secure financial environment for all stakeholders.

**9. REFERENCES**

**[1] Basic Idea of Problem**

<https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/>

**[2]How to import libraries?**

<https://towardsdatascience.com/credit-card-fraud-detection-using-machine-learning-python-5b098d4a8edc>

1. **Dataset**

<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

**[4] Machine learning based credit card fraud detection using the GA algorithm for feature selection**

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8>

**[5] Case study of Infosys that is using this same model**

<https://www.infosysbpm.com/blogs/bpm-analytics/machine-learning-for-credit-card-fraud-detection.html>

**[5]Challenges and Techniques**

<https://www.infosysbpm.com/blogs/bpm-analytics/machine-learning-for-credit-card-fraud-detection.html>

**[6]Reviews**

<https://link.springer.com/article/10.1007/s44230-022-00004-0>