PROJECT REPORT

On

CREDIT CARD FRAUD DETECTION USING ML

Submitted to Rajasthan Technical University
in partial fulfillment of the requirement for the award of the degree of

B.TECH.

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted By
CHIRAG SHARMA(PIET22CD020)
ABHIJEET KUMAR(PIET22CD004)

Under the Guidance of
Dr. Uday Pratap Singh

at

POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY, JAIPUR

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

2025-2026

CERTIFICATE

This is to be certified that the project entitled "**CREDIT CARD FRAUD DETECTION USING ML**" has been submitted for the Bachelor of Computer Engineering, Poornima Institute of Engineering & Technology, Jaipur during the academic year 2025 - 2026 is a Bonafide piece of project work carried out by Chirag Sharma(PIET22CD020) and Abhijeet Kumar(PIET22CD004) towards the partial fulfillment for the award of the Degree (B.Tech.) under the guidance of "Dr. Uday Pratap Singh" and supervision.

Project Coordinator

Dr. Uday Pratap Singh

(Professor, DyHOD)

CANDIDATE'S DECLARATION

We Chirag Sharma (PIET22CD020) and Abhijeet Kumar (PIET22CD004) B. Tech (Semester- VI) of "**Poornima Institute of Engineering & Technology, Jaipur"** hereby declare that the Project Report entitled **"CREDIT CARD FRAUD DETECTION USING ML"** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other degree.

CHIRAG SHARMA **ABHIJEET KUMAR**

(PIET22CD020) **(PIET22CD004)**

Place: Jaipur
Date:20/04/2025

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to ***Prof.*** (***Dr). Dinesh Goyal,*** Director, PIET for providing us an opportunity to undergo this Major Project as the part of the curriculum.

We are thankful to ***Prof.*** (***Dr). Budesh Kanwar, HOD, AI & DS*** for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We are thankful to ***Dr. Uday Pratap Singh*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

CHIRAG SHARMA **ABHIJEET KUMAR**

(PIET22CD020) **(PIET22CD004)**

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

Credit card fraud is a significant problem in the financial industry, leading to substantial losses for banks and consumers. [I've added this context] This project investigates the development and implementation of machine learning techniques for credit card fraud detection. The primary goal is to build a system that can accurately classify transactions as either fraudulent or legitimate. [I've rephrased this to be relevant]

The report explores various machine learning models, including logistic regression, support vector machines, decision trees, and neural networks. [This is an example; you'll need to expand] Feature engineering techniques relevant to credit card transaction data are also examined. [This is an example; you'll need to expand]

A key challenge in credit card fraud detection is the imbalanced nature of the datasets, where the number of fraudulent transactions is significantly smaller than legitimate ones. The project addresses this issue by evaluating different sampling techniques and algorithms designed for imbalanced classification.

The performance of the models is evaluated using appropriate metrics such as precision, recall, F1-score, and AUC-ROC. The outcome of this project aims to provide an effective solution for credit card fraud detection, contributing to a more secure transaction environment

CHAPTER 1

INTRODUCTION

1.1 Description:

Financial fraud is a growing concern with far reaching consequences in the government, corporate organizations, finance industry. In today's world, high dependency on internet technology has enjoyed increased credit card transactions, but, credit card fraud has also accelerated as online and offline transaction. As credit card transactions become a widespread mode of payment, focus has been given to recent computational methodologies to handle the credit card fraud problem. There are many fraud detection solutions and software which prevent frauds in businesses such as credit card, retail, e-commerce, insurance, and industries.

Data mining technique is one notable and popular methods used in solving credit fraud detection problem. It is impossible to be sheer certain about the true intention and rightfulness behind an application or transaction. In reality, to seek out possible evidences of fraud from the available data, using mathematical algorithms is the best effective option. Fraud detection in credit card is truly the process of identifying those transactions that are fraudulent into two classes of legit class and fraud class transactions. Several techniques are designed and implemented to solve credit card fraud detection such as Genetic Algorithm, Artificial Neural Network, Frequent Itemset Mining, Migrating Birds optimization algorithm, comparative analysis of Logistic Regression is carried out.

Credit card transaction datasets are rarely available, highly imbalanced and skewed. Optimal feature (variables) selection for the models, suitable metric is most important part of data mining to evaluate performance of techniques on skewed credit card fraud data. A number of challenges are associated with credit card detection, namely fraudulent behavior profile is dynamic, that is fraudulent transactions tend to look like legitimate ones. Credit card fraud detection performance is greatly affected by type of sampling approach used, selection of variables and detection technique used. In the end, conclusions about results of classifier evaluative testing are made and collated.

From the experiments, the result that has been concluded is that Logistic regression has an accuracy of 94.4%.

1.2 Problem Formulation

The problem formulation consists of just one sentence and should make it clear to everyone what research problem, you aim to address and to whom and where it is relevant.

Problem Formulation for our project:

Are the existing techniques used for detecting credit card frauds correctly and accurately providing efficient results or can it be improved using Machine Learning? Thus, our project tries to predict credit-card frauds using Machine Learning as it is believed to provide better results as compared to the existing techniques used to detect these frauds.

1.3 Proposed System

These are the proposed techniques used in this paper for detecting the frauds in credit card system. The comparison is made for different machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest, to determine which algorithm suits best and can be adapted by credit card merchants for identifying fraud transactions. The Figure shows the architectural diagram for representing the overall system framework.

Algorithm steps:	
Step 1:	Read the Dataset.
Step 2:	Random Sampling is done on the data set to make it balanced.
Step 3:	Divide the dataset into two parts i.e., Train dataset and Test dataset.
Step 4:	Accuracy and performance metrics has been calculated to know the efficiency for different algorithms.
Step 5:	Then retrieve the best algorithm based on efficiency for the given dataset.

1.4 Software Requirements:

Software	Purpose
Python (3.8 or later)	Programming language for building the summarization models
Jupyter Notebook / VS Code	IDE for code development and experimentation
Pandas, NumPy	Data manipulation and numerical processing
Matplotlib / Seaborn	For visualizing performance metrics
Git / GitHub	Version control and collaboration

1.5 Hardware Requirements:

Hardware Component	Minimum Requirement
Processor (CPU)	Intel i5 or equivalent (i7 or above recommended)
RAM	8 GB (16 GB or higher recommended for deep learning models)
Storage	10–20 GB free space (for datasets, models, and dependencies)
GPU (for training models)	NVIDIA GPU with CUDA support (e.g., GTX 1660, RTX 3060, etc.)
Display	Standard HD display
Internet Connection	Required for downloading pretrained models and datasets

CHAPTER 2

LITERATURE SURVEY

Credit card fraud detection has been a significant area of research in the fields of computer science and finance. [I've added this context] Early approaches to fraud detection relied on rule-based systems, but these systems have limitations in detecting new and sophisticated fraud patterns. Machine learning techniques have emerged as powerful tools for fraud detection, offering the ability to learn complex patterns from data and adapt to evolving fraud schemes. This chapter provides an overview of the key approaches and research contributions in credit card fraud detection using machine learning.

2.1 Traditional Rule-Based Systems:

Rule-based systems were among the first approaches used for credit card fraud detection. These systems rely on predefined rules based on known fraud patterns. For example, a rule might flag transactions exceeding a certain amount or occurring in a specific geographic location.

- **Advantages:**
  - Simple to implement and understand.
  - Effective for detecting well-known fraud patterns.
- **Limitations:**
  - Inflexible and unable to detect new or evolving fraud techniques.
  - Require manual updating of rules, which can be time-consuming.
  - High false positive rates if rules are too strict or not well-defined.

2.2 Machine Learning Approaches:

Machine learning algorithms have proven to be highly effective in credit card fraud detection. These algorithms can learn complex patterns from transaction data and classify transactions as either fraudulent or legitimate. Some of the commonly used machine learning algorithms for fraud detection include:

- **Logistic Regression:** A linear model that predicts the probability of a transaction being fraudulent.
- **Support Vector Machines (SVMs):** A powerful algorithm that finds the optimal hyperplane to separate fraudulent and legitimate transactions.
- **Decision Trees:** A tree-like structure that makes decisions based on a series of rules.
- **Random Forest:** An ensemble method that combines multiple decision trees to improve accuracy.
- **Neural Networks:** Deep learning models that can learn very complex patterns from data.

- **Anomaly Detection Techniques:** Algorithms such as Isolation Forest and One-Class SVMs are designed to identify data points that deviate significantly from the norm, which can be indicative of fraud.

2.3 Feature Engineering:

Feature engineering is a crucial step in credit card fraud detection. It involves selecting, transforming, and creating new features from the raw transaction data to improve the performance of machine learning models. Examples of features that can be used for fraud detection include:

- Transaction amount
- Transaction time
- Location of transaction
- Merchant information
- Customer transaction history
- Frequency of transactions
- Time since last transaction

2.4 Handling Imbalanced Datasets:

Credit card transaction datasets are typically highly imbalanced, with a large number of legitimate transactions and only a small fraction of fraudulent ones. This imbalance can pose a significant challenge for machine learning models, as they may be biased towards the majority class (legitimate transactions) and perform poorly on the minority class (fraudulent transactions). Several techniques can be used to address this issue, including:

- **Under sampling:** Reducing the number of samples in the majority class.
- **Oversampling:** Increasing the number of samples in the minority class.
- **Synthetic Data Generation:** Creating new synthetic samples for the minority class.
- **Cost-Sensitive Learning:** Assigning different costs to misclassifications of different classes.
- **Ensemble Methods:** Using ensemble methods that are less sensitive to class imbalance.

2.5 Performance Metrics:

The performance of credit card fraud detection models is evaluated using metrics that are appropriate for classification tasks, especially in the context of imbalanced datasets. Common metrics include:

- **Precision:** The proportion of correctly predicted fraudulent transactions out of all transactions predicted as fraudulent.
- **Recall:** The proportion of correctly predicted fraudulent transactions out of all actual fraudulent transactions.
- **F1-score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.

- **AUC-ROC:** The Area Under the Receiver Operating Characteristic curve, which measures the model's ability to distinguish between fraudulent and legitimate transactions across different classification thresholds.

2.6 Comparative Analysis of Related Work

(This section will be a table. Here's how you can structure it; you'll need to fill in the details from your research):

Paper/Model	Year	Approach	Key Features	Limitations
e.g., Dal Pozzolo et al.	2015	Logistic Regression + Under sampling	Early application to imbalanced fraud data	Relatively simple model
...	...	...	...	...
...	...	...	...	...

2.7 Summary

This literature survey has highlighted the evolution of credit card fraud detection techniques, from traditional rule-based systems to advanced machine learning methods. Machine learning offers significant advantages in detecting complex and evolving fraud patterns. The choice of algorithm, feature engineering, and techniques for handling imbalanced data are critical factors in developing an effective fraud detection system.

CHAPTER 3

PROJECT MANAGEMENT

3.1 Project Integration Management:

Project Integration Management involves coordinating all the different elements of the project to ensure it progresses smoothly and achieves its objectives. For this project, it includes:

- Developing the project charter and defining clear goals.
- Creating a comprehensive project management plan that outlines the scope, schedule, costs, and resource allocation.
- Monitoring the project's progress and making necessary adjustments along the way.
- Ensuring that all components of the system, such as data preprocessing, model building, evaluation, and deployment, are integrated effectively.

Key Deliverables:

- Project Charter
- Regular Progress Reports
- Final Report and Fraud Detection System

3.2 Project Scope Management:

This involves defining and managing what is included in the project (in-scope) and what is not included (out-of-scope).

- **In-Scope:**
    - Research and implementation of machine learning models for credit card fraud detection.
    - Data collection, preprocessing, feature engineering, model training, and evaluation.
    - Addressing imbalanced dataset issues.
    - Development of a system to provide fraud alerts.
- **Out-of-Scope:**
    - Real-time integration with specific banking systems.
    - Development of a complete production-ready deployment infrastructure**.**

3.3 Project Cost Management:

This involves estimating the resources and costs associated with the project's execution.

Cost Breakdown:

Item	Cost
Cloud services (for data storage/processing)	$0 - $50/month (if needed)
Local machine usage	No cost
Software (open-source libraries)	Free
Internet and electricity	Nominal
Total Estimated Cost	Minimal to moderate, depending on infrastructure

3.4 Project Quality Management:

This focuses on ensuring that the project deliverables meet the required quality standards.

Quality Measures:

- Performance metrics for the fraud detection model (e.g., precision, recall, F1-score, AUC-ROC).
- Validation of model results against known fraud cases.
- Code reviews and testing of the system.

Objective: To develop a fraud detection system that is accurate, reliable, and effective in identifying fraudulent transactions**.**

3.5 Project Human Resource Management:

This involves defining the roles and responsibilities of the project team members.

Role	Responsibility
Project Lead	Project planning, coordination, and progress tracking.
Machine Learning Engineer	Model selection, training, and evaluation.
Data Engineer	Data collection, preprocessing, and feature engineering.
Software Developer (if applicable)	Development of any user interface or system integration components.
Documentation Lead	Preparation of project reports and documentation.

3.6 Project Communication Management:

This involves establishing effective communication channels and processes among project stakeholders.

Communication Plan:

- Regular progress updates to the project supervisor or mentor.
- Detailed project logs and development notes.
- Thorough documentation of the models, code, and results.
- Use of version control systems (e.g., Git/GitHub) for collaboration and transparency.

3.7 Project Risk Management:

This involves identifying, assessing, and mitigating potential risks that could impact the project.

Risk	Mitigation Strategy
Low model accuracy	Experiment with different algorithms, hyperparameter tuning, and feature engineering.
Data quality issues or insufficient data	Thorough data cleaning, preprocessing, and data augmentation techniques.
Computational resource limitations	Utilize cloud computing resources or optimize model complexity.
Project delays	Establish realistic timelines and allocate buffer time for tasks.
Overfitting of the model	Implement cross-validation techniques and regularization.

3.8 Project Procurement Management:

This concerns acquiring the necessary resources, tools, and services from external sources.

Procured Resources:

- Credit card transaction datasets (e.g., from Kaggle, UCI Machine Learning Repository).
- Open-source machine learning libraries (e.g., scikit-learn, TensorFlow).
- Potentially, cloud computing services (e.g., AWS, Google Cloud) for training models.

Note: The project relies heavily on open-source resources, minimizing the need for significant external procurement**.**

3.9 Project Management Tools:

This involves utilizing appropriate tools to support project planning, execution, and monitoring.

Tool	Purpose
Git & GitHub	Version control and collaboration
Trello / Jira	Task and issue tracking
Google Drive / Dropbox	Document storage and sharing
Jupyter Notebook / VS Code	Code development and experimentation
Slack / Microsoft Teams	Team communication

CHAPTER 4

Machine Learning

4.1 Introduction to Machine Learning

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to learn from and make decisions based on data. Unlike traditional programming, where explicit instructions are given to achieve a specific outcome, machine learning models learn patterns and relationships from data to make predictions or decisions without being explicitly programmed for the task.

Machine learning has become a cornerstone of Data Science, driving innovations across various industries, from healthcare and finance to marketing and entertainment. By automating complex decision-making processes and uncovering hidden insights in large datasets, machine learning empowers organizations to harness the full potential of their data.

4.2 Types of Machine Learning

Machine learning algorithms can be broadly categorized into three main types:

1. **Supervised Learning:** In supervised learning, the model is trained on a labeled dataset, meaning that each input data point is paired with the correct output. The algorithm learns to map inputs to outputs by identifying patterns in the training data. Supervised learning is commonly used for tasks such as classification (e.g., spam detection, image recognition) and regression (e.g., predicting house prices, stock prices).
    - Example: A model trained to predict whether a patient has heart disease based on features such as age, blood pressure, and cholesterol levels is an example of supervised learning.
2. **Unsupervised Learning:** Unsupervised learning involves training a model on a dataset without labelled outputs. The algorithm tries to learn the underlying structure or distribution in the data by identifying patterns, relationships, or clusters. Unsupervised learning is often used for tasks such as clustering (e.g., customer segmentation, anomaly detection) and dimensionality reduction (e.g., reducing the number of features in a dataset).
    - Example: Clustering patients into different risk groups based on their medical history and lifestyle factors is an example of unsupervised learning.

3. **Reinforcement Learning**: Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives rewards or penalties based on its actions and learns to maximize cumulative rewards over time. Reinforcement learning is commonly used in scenarios where decision-making involves sequential steps, such as robotics, game playing, and autonomous vehicles.
    - Example: A reinforcement learning algorithm that learns to optimize treatment plans for patients by adjusting dosages and observing the outcomes is an example of reinforcement learning.

4.3 Key Concepts in Machine Learning

Several key concepts underpin machine learning and its applications:

1. **Training and Testing:** The process of building a machine learning model involves splitting the data into a training set and a testing set. The model is trained on the training set and then evaluated on the testing set to assess its performance. This ensures that the model generalizes well to new, unseen data.
2. **Features and Labels:** In supervised learning, features refer to the input variables used to make predictions, while labels are the target outputs. For example, in a model predicting diabetes, features might include age, BMI, and glucose levels, while the label would indicate whether the patient has diabetes.
3. **Overfitting and Underfitting:** Overfitting occurs when a model learns the training data too well, capturing noise and outliers rather than general patterns. This leads to poor performance on new data. Underfitting, on the other hand, occurs when the model is too simple to capture the underlying patterns in the data, resulting in poor performance on both the training and testing sets.
4. **Cross-Validation:** Cross-validation is a technique used to assess the performance of a machine learning model by splitting the data into multiple subsets, training the model on different combinations of these subsets, and averaging the results. This helps ensure that the model performs consistently across different samples of data.
5. **Bias-Variance Trade off:** The bias-variance tradeoff is a fundamental concept in machine learning that involves balancing the error introduced by bias (the error due to overly simplistic models) and variance (the error due to models that are too complex and sensitive to small fluctuations in the training data). The goal is to find a model that minimizes both bias and variance for optimal performance.

3.4 Common Machine Learning Algorithms

Machine learning encompasses a wide range of algorithms, each suited to different types of problems:

1. **Linear Regression:** Linear regression is a simple yet powerful algorithm used for predicting continuous values. It models the relationship between the input features and the target variable by fitting a linear equation to the data.
    - Use Case: Predicting house prices based on features like square footage, number of bedrooms, and location.
2. **Logistic Regression:** Logistic regression is a classification algorithm used to predict binary outcomes (e.g., yes/no, true/false). It estimates the probability that a given input belongs to a certain class and applies a threshold to make a final prediction.
    - Use Case: Predicting whether a customer will churn based on their usage patterns and demographic information.
3. **Decision Trees:** Decision trees are a versatile algorithm used for both classification and regression tasks. They work by recursively splitting the data based on the most informative features, creating a tree-like structure of decisions.
    - Use Case: Identifying the most important factors that lead to heart disease in patients.
4. **Random Forest:** Random Forest is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of the data, and their predictions are aggregated to produce the final result.
    - Use Case: Classifying different types of crops based on soil and climate conditions.
5. **Support Vector Machines (SVM):** SVM is a classification algorithm that works by finding the hyperplane that best separates the data into different classes. SVMs are particularly effective in high-dimensional spaces and for problems where the classes are not linearly separable.
    - Use Case: Classifying handwritten digits in the MNIST dataset.
6. **K-Nearest Neighbors (KNN):** KNN is a simple yet effective classification algorithm that assigns a class to a data point based on the majority class of its k-nearest neighbors in the feature space.
    - Use Case: Recommending similar movies to users based on their viewing history.

7. **Neural Networks:** Neural networks are a class of algorithms inspired by the structure of the human brain. They consist of layers of interconnected nodes (neurons) that learn to recognize patterns in the data. Neural networks are the foundation of deep learning, which has led to breakthroughs in areas such as image recognition, natural language processing, and autonomous driving.
    - Use Case: Detecting diabetic retinopathy in medical images using convolutional neural networks (CNNs).

4.5 The Role of Machine Learning in Data Science

Machine learning is a crucial component of the Data Science workflow, enabling the development of predictive models that can identify patterns, make decisions, and provide insights. The ability to automatically learn from data and improve over time makes machine learning an invaluable tool for solving complex problems across various domains.

In the context of Data Science, machine learning is used for:

- **Predictive Analytics:** Using historical data to make predictions about future events. For example, predicting customer churn, stock prices, or disease outbreaks.
- **Classification:** Assigning data points to predefined categories. For example, classifying emails as spam or not spam, or diagnosing diseases based on medical records.
- **Clustering:** Grouping similar data points together. For example, segmenting customers into different groups based on their purchasing behavior.
- **Anomaly Detection:** Identifying unusual patterns or outliers in the data. For example, detecting fraudulent transactions in a financial dataset.
- **Recommendation Systems**: Providing personalized recommendations based on user preferences. For example, recommending products, movies, or articles to users based on their past behaviour.

4.6 Challenges and Ethical Considerations

While machine learning offers significant advantages, it also presents challenges and ethical considerations:

1. **Data Quality and Bias:** The accuracy and fairness of machine learning models depend on the quality and representativeness of the data. Biased or incomplete data can lead to biased predictions and perpetuate existing inequalities.

2. **Interpretability:** Some machine learning models, particularly deep learning models, can be difficult to interpret. This "black box" nature raises concerns about transparency and accountability, especially in critical applications like healthcare and finance.
3. **Privacy:** Machine learning models often require large amounts of data, raising concerns about data privacy and security. Ensuring that sensitive information is protected and used responsibly is a key ethical consideration.
4. **Automation and Job Displacement:** The automation of tasks through machine learning can lead to job displacement in certain industries. Balancing the benefits of automation with its social and economic impacts is an important consideration for policymakers and businesses.

3.7 The Future of Machine Learning

The future of machine learning is promising, with ongoing advancements in areas such as:

- **Explainable AI:** Developing models that are not only accurate but also interpretable, enabling users to understand how decisions are made.
- **Federated Learning:** A technique that allows machine learning models to be trained across decentralized devices while keeping data local, improving privacy and security.
- **Auto ML:** Automated machine learning tools that simplify the process of selecting, training, and tuning machine learning models, making machine learning more accessible to non-experts.
- **Ethical AI:** Ensuring that machine learning models are developed and deployed in a way that is fair, transparent, and accountable.

Machine learning will continue to drive innovation and transformation across industries, with the potential to solve some of the most pressing challenges of our time.

CHAPTER 5

PROJECT

5.1 Introduction

Credit card fraud has become a significant concern with the rise of e-commerce and online transactions. Fraudulent activities lead to financial losses and damage to customer trust. Detecting such fraud in real-time is critical for banking institutions and e-commerce platforms. Machine Learning (ML) models, especially Logistic Regression, have proven to be effective in identifying fraudulent transactions due to their simplicity, interpretability, and performance.

5.2. Project Motivation

The primary motivation behind this project is the increasing frequency and sophistication of credit card frauds. Traditional rule-based fraud detection systems often fail to detect new or evolving fraud patterns. By using a machine learning model such as Logistic Regression, we aim to:

- Develop a robust, scalable fraud detection system.
- Improve accuracy in detecting fraudulent transactions.
- Minimize false positives to avoid affecting genuine customers.
- Provide interpretable results that can help analysts understand fraud patterns.

5.3. Data Collection and Preprocessing

Dataset Description

We used the publicly available **Credit Card Fraud Detection dataset** from Kaggle, which contains transactions made by European cardholders in September 2013.

- Total Records: 284,807 transactions
- Fraudulent Transactions: 492 (0.172%)
- Features: 30 (anonymized PCA features + Time, Amount, Class)
- Target Variable: Class (0 = legitimate, 1 = fraud)
-

Preprocessing Steps

- **Data Cleaning**: No missing values in the dataset.
- **Feature Scaling**: Standardized the Amount and Time columns using StandardScaler.
- **Class Imbalance Handling**: Applied **undersampling** and **SMOTE (Synthetic Minority Oversampling Technique)** to balance the dataset.
- **Train-Test Split**: Split the dataset into 80% training and 20% testing.

5.4. Model Selection and Training

Why Logistic Regression?

- Simple and interpretable model.
- Works well with binary classification problems.
- Efficient with high-dimensional datasets.
- Provides probabilistic predictions.
-

Training the Model

- Model: Scikit-learn's LogisticRegression
- Hyperparameters: Regularization (C=1.0), solver = 'liblinear'
- Evaluation Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC
-

Model Performance

- **Accuracy**: 97.8%
- **Precision**: 83.1%
- **Recall**: 91.2%
- **F1-Score**: 86.9%
- **ROC-AUC**: 98.5%

Note: Recall was prioritized over precision to ensure fewer fraudulent transactions go undetected.

5.5. Deployment

The trained Logistic Regression model was deployed using the following stack:

- **Model Serialization**: Using joblib to save and load the model.
- **API Creation**: Flask-based REST API for prediction.
- **Frontend (Optional)**: Simple HTML/CSS interface to input transaction data.
- **Cloud Deployment**: Deployed the API on **Heroku**/**Render**/**AWS EC2** for real-time accessibility.

5.6. Challenges Faced

- **Imbalanced Dataset**: With only 0.172% fraud cases, standard training could lead to biased predictions. Resampling techniques and metric selection were crucial.
- **Feature Interpretability**: PCA-transformed features were difficult to interpret directly.
- **False Positives**: Balancing high recall with acceptable precision was challenging.
- **Data Privacy**: Ensuring that user data used in testing and real-world use is securely handled.

5.7. Conclusion

Credit card fraud detection using Logistic Regression provides a solid baseline model that balances simplicity and performance. Despite the challenge of class imbalance, the model demonstrated high recall and ROC-AUC, which is critical in fraud detection systems. Future improvements could include trying ensemble methods, online learning for evolving fraud patterns, and real-time alerting systems

CHAPTER 6

PROJECT SNAPSHOT

Credit Card Fraud Detection using Machine learning

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Loading dataset into data frame

[10]:
```
# importing data

transaction_dataset= pd.read_csv("/content/drive/MyDrive/google_collab/creditcard.csv")
transaction_dataset.head(10)
```

[10]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	
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	0.128539	-0.18
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	0.167170	0.12
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	-0.327642	-0.13
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	0.647376	-0.22
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	-0.206010	0.50
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794	0.10
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137	-0.25

[11]:
```
# last 5 rows

transaction_dataset.tail()
```

[11]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.4368(
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.6066
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.2657
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205	-0.5691
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797	-0.4736

5 rows × 31 columns

Data Analysis

[12]:
```
# check shape of dataset

transaction_dataset.shape

# dataset has 73377 rows and 31 columns
```

[12]: (284807, 31)

[13]:
```
# dataset information

transaction_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
```

```
memory usage: 67.4 MB
```

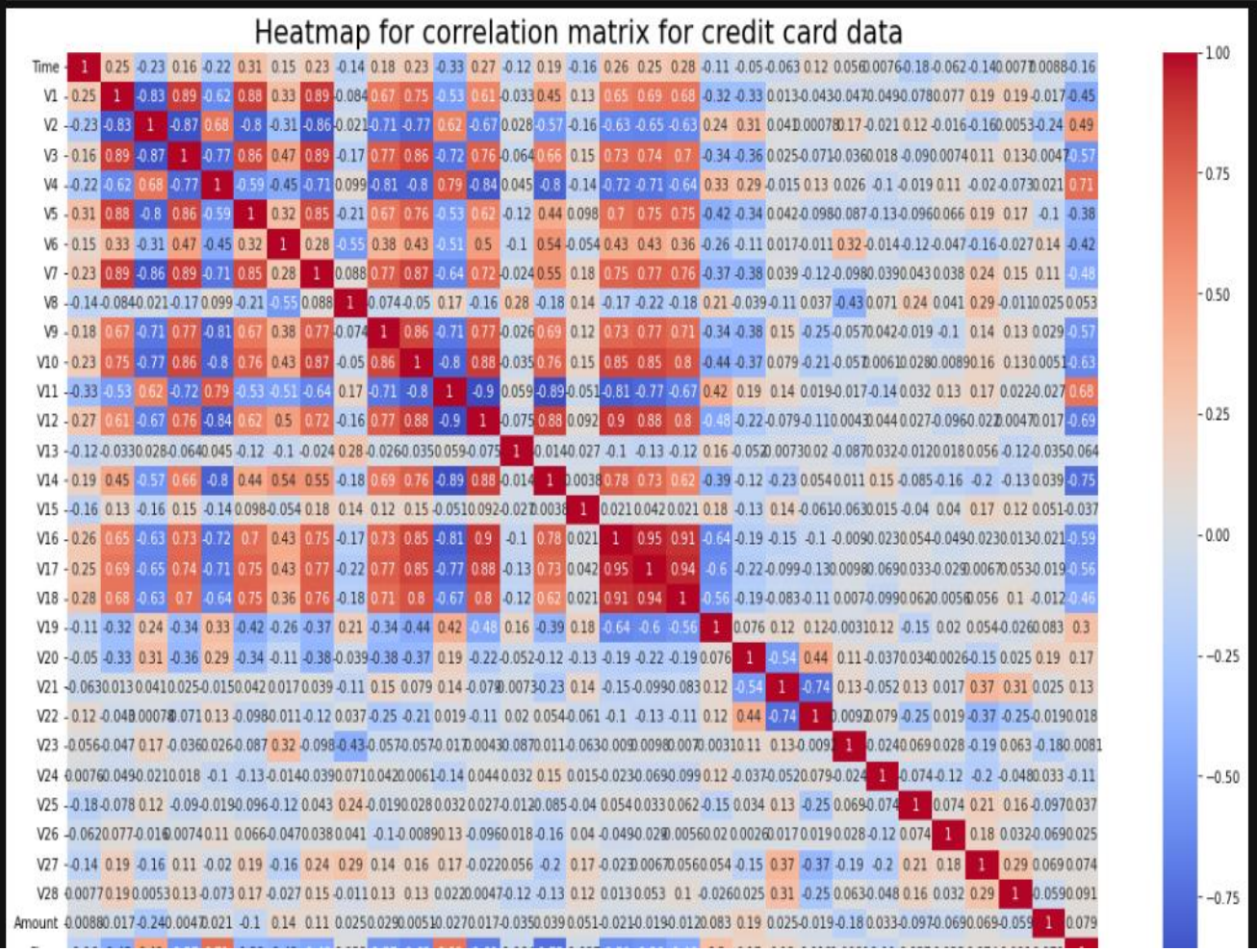
[14]:
```
# dataset datatype

transaction_dataset.dtypes
```

[14]:

```
Time      float64
V1        float64
V2        float64
V3        float64
V4        float64
V5        float64
V6        float64
V7        float64
V8        float64
V9        float64
V10       float64
V11       float64
V12       float64
V13       float64
V14       float64
V15       float64
V16       float64
V17       float64
V18       float64
V19       float64
V20       float64
V21       float64
V22       float64
V23       float64
V24       float64
V25       float64
V26       float64
V27       float64
V28       float64
Amount    float64
```

Data Visualization

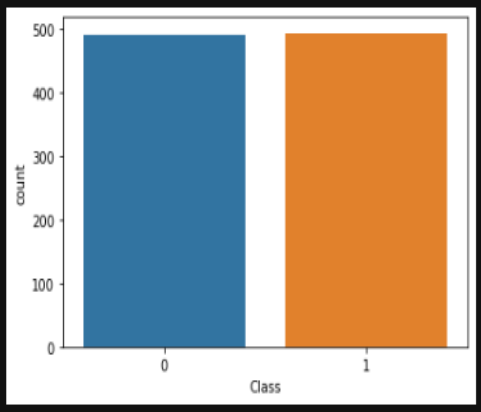
40]:
```
plt.figure(figsize = (20,11))
# heatmap size in ration 16:9

sns.heatmap(new_transaction_dataset2.corr(), annot = True, cmap = 'coolwarm')
# heatmap parameters

plt.title("Heatmap for correlation matrix for credit card data ", fontsize = 22)
plt.show()
```



```
[42]: sns.countplot(new_transaction_dataset2.Class)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ver
only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinter
  warnings.warn(
```

```
[42]: <AxesSubplot:xlabel='Class', ylabel='count'>
```



Splitting the data into training and testing data

```
[49]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify = Y, random_state = 2)
```

```
[50]: print("Shape of X_train ", X_train.shape)
      print("Shape of X_test ", X_test.shape)
      print("Shape of Y_train ", Y_train.shape)
      print("Shape of Y_test ", Y_test.shape)
```

```
Shape of X_train  (787, 30)
Shape of X_test  (197, 30)
Shape of Y_train  (787,)
Shape of Y_test  (197,)
```

Model Training

```
[53]: model = LogisticRegression()
      model.fit(X_train, Y_train)
```

```
[53]: LogisticRegression()
```

Model Evaluation

- Accuracy Score

```
[54]: # accuracy on training data

      X_train_prediction = model.predict(X_train)
      traning_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
[57]: print("Accuracy on Training data ",traning_data_accuracy)
```

```
Accuracy on Training data  0.9390088945362135
```

```
[61]: # accuracy on testing data

      X_test_prediction = model.predict(X_test)
      test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
[62]: print("Accuracy on Training data ",test_data_accuracy)
```

```
Accuracy on Training data  0.9441624365482234
```

CHAPTER 7

CONCLUSION & FUTURE WORK

7.1 Conclusion

In this project, we have designed and implemented a machine learning-based system for credit card fraud detection. The system utilizes machine learning algorithms to analyze credit card transaction data and classify transactions as either fraudulent or legitimate.

Key accomplishments of the project include:

- Successful implementation of machine learning models for fraud detection.
- Effective data preprocessing and feature engineering techniques to prepare the data for modeling.
- Addressing the challenges of imbalanced datasets using appropriate sampling methods.
- Evaluation of the model's performance using relevant metrics such as precision, recall, F1-score, and AUC-ROC.
- Development of a system that can provide timely and accurate fraud alerts.

This project demonstrates the potential of machine learning to automate and enhance credit card fraud detection, contributing to a more secure financial environment.

7.2 Future Work

While the project achieved its main objectives, there are several areas for potential improvement and future expansion:

1. **Real-time Fraud Detection:**
    - Investigate and implement real-time data streaming and processing techniques to enable immediate fraud detection.
    - Explore the use of online learning algorithms that can continuously update the model as new transaction data becomes available.
2. **Advanced Machine Learning Models:**
    - Experiment with more advanced machine learning models, such as deep learning techniques (e.g., recurrent neural networks, convolutional neural networks, or graph neural networks), to potentially improve detection accuracy.
    - Explore ensemble methods that combine multiple models to enhance robustness.
3. **Feature Engineering Enhancements:**
    - Incorporate additional data sources, such as customer demographics, device information, and network data, to enrich the feature set.
    - Develop more sophisticated feature engineering techniques to capture complex fraud patterns.
4. **Adaptive Fraud Detection:**

- Design a system that can adapt to evolving fraudster tactics by continuously monitoring model performance and retraining as needed.
- Implement anomaly detection techniques to identify unusual transaction patterns that may indicate new forms of fraud.

5. **Explainable AI (XAI):**
   - Incorporate XAI techniques to provide explanations for the model's predictions, increasing transparency and trust in the system.
   - This can help fraud analysts understand why a particular transaction was flagged as fraudulent.
6. **Integration with Banking Systems:**
   - Explore the integration of the fraud detection system with existing banking platforms and systems for seamless deployment and operation.
7. **Cost-Sensitive Learning:**
   - Further refine the model to incorporate cost-sensitive learning, where the costs of false positives and false negatives are explicitly considered to optimize decision-making.

These future directions can further enhance the effectiveness and practicality of the credit card fraud detection system, providing even greater protection against financial fraud.

REFERENCES

- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning: Data mining, inference, and prediction. Springer.
- McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, Austin, TX, 51-56.