PROJECT TITLE: Fraud Detection in Credit Card Transactions Using Machine Learning
PREPARED BY: Prajwal M
DATE: 30/10/2025

---

1. INTRODUCTION

Credit card fraud has become a serious challenge for financial institutions due to the rapid growth of online transactions and increasingly complex fraud strategies. Conventional rule-based systems often fail to detect new and evolving fraud patterns, resulting in both undetected fraudulent transactions and a high number of false positives. To address these limitations, this project develops a machine learning–based fraud detection system that analyzes historical transaction data to identify suspicious activities and supports real-time or near–real-time decision making.

---

2. PROBLEM STATEMENT

A financial institution is facing a steady rise in fraudulent credit card transactions, causing direct financial losses and negatively affecting customer confidence. Existing manual and rule-based detection approaches are not sufficiently accurate or fast to handle large transaction volumes and rapidly changing fraud patterns. There is a need for a robust, automated fraud detection model that can accurately classify transactions as legitimate or fraudulent and operate in real time or near real time.

---

3. OBJECTIVES

The main objective of this project is to design and implement a credit card fraud detection system using machine learning techniques.

Specific objectives are:

- Analyze historical credit card transaction data to understand patterns of normal and fraudulent behavior.
- Handle highly imbalanced data effectively to train a reliable fraud detection model.
- Develop and evaluate machine learning models capable of detecting fraudulent transactions with high precision and recall.
- Deploy the selected model into a real-time or near–real-time detection pipeline with low latency.
- Enable monitoring and continuous improvement of the fraud detection system as fraud patterns evolve.

---

4. SCOPE

The scope of this project includes:

- Data collection (public credit card transaction dataset), analysis, and preprocessing.
- Design, training, and evaluation of fraud detection models using machine learning.
- Design of a real-time or near–real-time system architecture for fraud detection based on the trained model.
- Implementation of basic monitoring strategies and feedback loops for model improvement.

The project does not include changes to core banking systems, full-scale production deployment, or legal and compliance policy design.

---

5. LITERATURE REVIEW

Previous studies have used various machine learning algorithms such as Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machines, and Neural Networks for credit card fraud detection. Research consistently highlights the challenge of severe class imbalance, where fraudulent transactions represent a very small fraction of total transactions, and recommends techniques such as resampling and cost-sensitive learning. Recent work also emphasizes real-time architectures that combine ensemble models with streaming platforms and microservices to achieve low-latency fraud detection at scale.

---

6. DATASET DESCRIPTION

This project uses a publicly available credit card fraud detection dataset containing anonymized transaction records. The dataset includes numerical features such as principal components (V1–V28) derived using PCA, along with “Time,” “Amount,” and a binary target label indicating non-fraud (0) or fraud (1). The data is highly imbalanced, with fraudulent transactions accounting for a very small percentage of total records, which reflects realistic financial transaction behavior.

---

7. EXPLORATORY DATA ANALYSIS (EDA)

EDA focuses on understanding the distribution of classes, transaction amounts, time-based patterns, and relationships between features and the target label. Class imbalance is visualized to confirm the rarity of fraud cases and motivate the use of specialized imbalance-handling methods. Correlation and feature-distribution analysis help identify features that contribute most to distinguishing fraudulent from legitimate transactions.

---

8. DATA PREPROCESSING

Data preprocessing steps include cleaning, feature scaling, and data splitting. The dataset is checked for missing or inconsistent values, which are handled appropriately to maintain data quality. Numerical features such as “Amount” are typically standardized, and the data is split into training, validation, and test sets using stratified sampling to preserve class proportions.

Handling imbalance is a critical part of preprocessing. Techniques such as random oversampling, random undersampling, and synthetic sampling (e.g., SMOTE) are considered to increase the representation of fraudulent cases in the training set. Cost-sensitive learning may also be applied by assigning higher misclassification costs to fraud samples in the model training process.

---

9. MODEL DEVELOPMENT

Several supervised learning models are explored for fraud detection, including:

- Logistic Regression
- Random Forest
- Gradient Boosting (e.g., XGBoost or similar)
- Support Vector Machine
- Neural Network

Studies show that ensemble techniques like Random Forest and Gradient Boosting often yield strong performance on fraud detection tasks due to their ability to capture complex relationships and handle noisy data. For each algorithm, hyperparameters are tuned using grid search or random search with cross-validation, focusing on metrics that reflect minority-class performance, such as precision, recall, F1-score, and Area Under the Precision–Recall Curve.

Decision thresholds on predicted probabilities are adjusted to align with business requirements, trading off between higher fraud detection (recall) and fewer false alerts (precision). The best-performing model is selected based on validation metrics and then evaluated on the test set to estimate its generalization performance.

---

10. EVALUATION METRICS

Given the strong class imbalance, overall accuracy is not a suitable performance measure for fraud detection. Instead, the following metrics are emphasized:

- **Precision**: Proportion of predicted frauds that are actually fraudulent.
- **Recall (Sensitivity)**: Proportion of actual frauds that are correctly detected.
- **F1-Score**: Harmonic mean of precision and recall.
- **AUC-PR**: Area under the Precision–Recall curve, which is effective for imbalanced problems.

The project aims to achieve precision and recall scores above 90% on the fraud class while maintaining acceptable latency for real-time decision making. Confusion matrix analysis is used to understand the distribution of true positives, false positives, true negatives, and false negatives.

---

11. REAL-TIME / NEAR–REAL-TIME SYSTEM DESIGN

To support real-time or near–real-time fraud detection, the system is designed as a set of loosely coupled components connected through a streaming or messaging platform. A typical architecture

includes a message broker (such as Apache Kafka or similar technology) that receives transaction events and forwards them to a model-scoring microservice.

The trained model is deployed as a RESTful or gRPC-based microservice, which receives transaction data, applies the necessary feature transformations, and returns a fraud risk score or classification. Depending on the predicted outcome and confidence, downstream services can trigger actions such as approving the transaction, flagging it for review, or blocking it automatically.

Low latency is achieved using lightweight model-serving frameworks, horizontal scaling of the scoring service, and efficient data serialization formats. The architecture is designed to be scalable so that it can handle high transaction throughput without performance degradation.

---

12. IMPLEMENTATION PLAN

The implementation proceeds in the following phases:

1. **Data Preparation and EDA**
    - Load the dataset, perform cleaning, handle missing values, and conduct exploratory analysis.
    - Visualize class imbalance and key feature distributions.
2. **Preprocessing and Imbalance Handling**
    - Apply feature scaling and train–test–validation split with stratification.
    - Experiment with oversampling, undersampling, and synthetic sampling techniques.
3. **Model Training and Selection**
    - Train multiple algorithms (Logistic Regression, Random Forest, Gradient Boosting, etc.).
    - Perform hyperparameter tuning and threshold adjustment using validation data.
4. **Model Evaluation**
    - Evaluate selected models on the test set using precision, recall, F1-score, and AUC-PR.
    - Choose the best model based on performance and robustness.
5. **Deployment Design and Prototype**
    - Package the chosen model and preprocessing steps into a microservice (e.g., using Flask/FastAPI).
    - Integrate with a simple message or REST-based interface to simulate real-time predictions.
6. **Monitoring and Feedback**
    - Log predictions, decisions, and ground-truth outcomes where available.
    - Define a plan for periodic retraining as new data arrives.

---

13. MONITORING AND CONTINUOUS IMPROVEMENT

After deployment, continuous monitoring is required to maintain model effectiveness and reliability. Key metrics to track include fraud detection rate, false positive rate, system latency, and transaction volumes. Significant changes in these metrics may indicate concept drift or operational issues that require retraining or model updates.

A feedback loop is established where confirmed frauds and false alarms are added to the training dataset for periodic retraining. New features, such as device identifiers, IP geolocation, or behavioral patterns, can be introduced over time to enhance the model's predictive power.

---

14. EXPECTED OUTCOMES

By the end of this project, the following outcomes are expected:

- A well-documented fraud detection model trained and evaluated on real credit card transaction data.
- Precision and recall above 90% for the fraud class on the test dataset, subject to the chosen decision threshold.
- A prototype real-time or near–real-time detection system design demonstrating how the model can be integrated into a streaming or microservice architecture.
- A monitoring and improvement strategy that allows the system to adapt to new fraud patterns and maintain performance over time