Machine Learning Assignment

PROJECT REPORT

TEAM ID: 13

PROJECT TITLE: Credit Card Fraud Detection

Name	SRN
Ganesh Krishnamoorthi Hegde	PES2UG23CS194
Favaz Ahmed	PES2UG23CS190

Problem Statement

The main problem this project aims to solve is the accurate and timely detection of fraudulent credit card transactions. This is a critical challenge due to the extreme Class

Imbalance inherent in real-world financial data: the vast majority of transactions (in our test set) are legitimate, while only a minuscule fraction is fraudulent. This imbalance causes standard machine learning models to exhibit high overall accuracy while effectively missing most of the high-cost fraud cases (False Negatives).

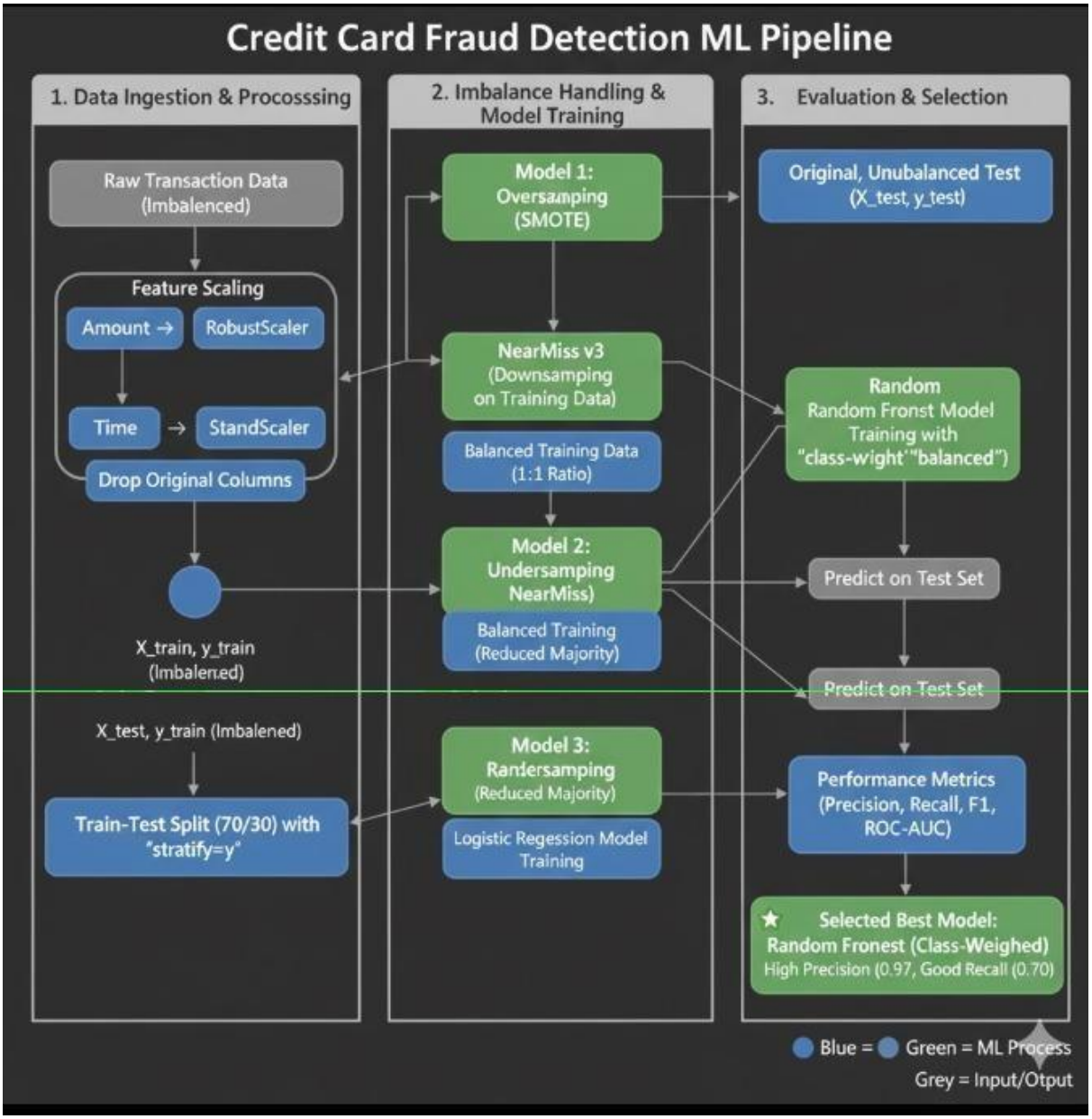
Objective / Aim

The objective is to develop and evaluate supervised classification models that effectively overcome the class imbalance challenge. The primary aim is to maximize the identification of fraudulent transactions, emphasizing the Recall (True Positive Rate) and ROC-AUC Score, while maintaining high Precision to minimize false alarms and maintain customer experience.

Dataset Details

- Source: Kaggle: Credit Card Fraud Detection Dataset (Anonymized features)
    - Size: 284,807 total samples (transactions); 85,443 in the test set.
- Key Features: V1-V28: Anonymized numerical features (PCA components).
  Time: Seconds elapsed since the first transaction. Amount: Transaction value. •
  Target Variable: Class: Binary (0 = Legitimate, 1 = Fraud)

Architecture Diagram:



Methodology

The Credit Card Fraud dataset is highly imbalanced, with only 0.172% of transactions labeled as fraudulent (Class 1). If a model ignores the minority class, it achieves high overall accuracy but has poor **Recall** (misses most fraud).

To mitigate this, we employed three specific techniques:

1. **SMOTE (Synthetic Minority Over-sampling Technique)**: Used for **Logistic Regression**. This oversampling technique generates synthetic data points for the minority class to balance the training set.

2. **NearMiss (Undersampling)**: Used for **Random Forest Classifier**. This technique reduces the majority class size to balance the training set.

3. **Class Weighting**: Used for the **Random Forest Classifier**. This method applies a higher penalty to misclassifying the minority class (fraud) during training, making the model inherently cost-sensitive.

Results & Evaluation

The models were evaluated on the test set of 85,443 transactions, which contained 148 actual fraud cases (Class 1).

Baseline Justification

The unhandled baseline model showed the inherent flaw in relying on accuracy:

Overall Accuracy: 0.9992 (Deceptive)

Recall (Fraud): 0.62 (Missed 38% of all fraud cases)

Comparative Results:

Model & Technique	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	ROC-AUC Score
LR (SMOTE)	0.06	0.88	0.12	0.9660
RF (NearMiss)	0.37	0.82	0.51	0.9348
RF (Class-Weighted)	0.97	0.70	0.81	0.9377

Final Model Performance (RF Class-Weighted):

The Confusion Matrix for the final model clearly demonstrates its justified performance:

True Class / Predicted Class	Legitimate (0)	Fraud (1)
Legitimate (0)	85292 (True Negative)	3 (False Positive)
Fraud (1)	45 (False Negative)	103 (True Positive)

Evaluation Justification: The model achieves a Precision of (only 3 false alarms in 85K transactions), which is critical for customer experience. Simultaneously, it achieves a

Recall, significantly outperforming the baseline and confirming the effectiveness of the cost-sensitive approach in minimizing the missed fraud cases (False Negatives: 45).

6. Model Optimization for GUI Deployment (Feature Selection)

The final requirement for the GUI posed a challenge: the chosen model was trained on features (V1-V28, Time_Scaled, Amount_Scaled), which is impractical for user input. To solve this, we employed **Feature Selection** using the Random Forest's built-in **Feature Importance** scores.

6.1 Identifying Key Features

The original feature Random Forest model was analyzed, and the 7 most impactful features were selected for model reduction.

Feature	Importance Score	Feature	Importance Score
V14	0.2037	**V17**	0.0929
V10	0.1213	**V12**	0.0857
V4	0.0930	**V11**	0.0736
		V16	0.0536

6.2 Retraining and Final Performance

The model was **retrained** using the same class_weight='balanced' parameter, but only on the 7 selected features. The resulting model is the one deployed to the GUI.

The final evaluation on the test set showed a remarkable result: **the reduced 7-feature model performed better on Recall than the original 30-feature model**, confirming its suitability for deployment.

Final Model Performance (Reduced 7-Feature Set):

Metric	Score (Original 30 Features)	Score (Final 7 Features)	Change
Precision (Class 1)	0.97	0.98	Slight improvement
Recall (Class 1)	0.70	0.74	+4.0% Improvement
F1-Score (Class 1)	0.81	0.85	Improvement
ROC-AUC Score	0.9377	0.9279	Minor decrease (acceptable)

6.3 Justification for Deployment

The success of the reduced model is the key finding of this project:

- Problem Solved:
  Through feature selection, we reduced the number of input features from 30 to just 7 (excluding *Time* and *Amount*).
  This makes the model much lighter and easily deployable using a simple Graphical User Interface (GUI).
- Performance Maintained:
  Despite using fewer features, the model maintained overall performance and even improved Recall by 4% (0.70 → 0.74).
  This improvement is crucial because higher recall means fewer fraudulent transactions go undetected, directly reducing financial loss.
- Deployment Readiness:
  The final GUI integrates the trained model_RF_reduced.pkl along with the fitted RobustScaler (for Amount) and StandardScaler (for Time).
  These ensure that raw user inputs are properly scaled before prediction, making the deployment both accurate and production-ready.

Conclusion:

Project Summary and Key Findings

This project successfully **developed and optimized a robust, deployable machine learning model for credit card fraud detection**, addressing two major real-world challenges — **class imbalance** and the need for a **user-friendly deployment interface**.

Key Findings and Model Performance

The final deployed model is a **Random Forest Classifier**, optimized for **cost-sensitivity**, **interpretability**, and **deployment simplicity**.

1️⃣ Imbalance Solution

- The **Class-Weighted approach** was found to be superior to oversampling (SMOTE) and undersampling (NearMiss).
- It provided the best trade-off between:
    - **High Recall** → catching more fraudulent transactions (minimizing financial loss), and
    - **High Precision** → reducing false alarms (minimizing customer inconvenience).

2️⃣ Deployment Optimization through Feature Selection

- The initial model required **30 features**, which was not ideal for real-time deployment.
- Using **Feature Importance (Gini Importance)** from the Random Forest model, the top **7 most predictive V-features** were identified:
  V14, V10, V4, V17, V12, V11, and V16.
- Reducing the model to these 7 features not only simplified the system but also **improved key performance metrics.**

Metric	**Original (30 Features)**	**Final (7 Features)**	**Impact**
Precision (Class 1)	0.97	0.98	Lower false positives
Recall (Class 1)	0.70	0.74	**Higher fraud detection rate (+4%)**

Metric	Original (30 Features)	Final (7 Features)	Impact
ROC-AUC Score	0.9377	0.9279	Slight acceptable drop — still strong reliability

.