Machine Learning Assignment

PROJECT REPORT

TEAM 12

Detecting Payments Fraud Using Machine Learning

Name	SRN
Nagula Anish	PES2UG23CS358
Muskan Goenka	PES2UG23CS355

Problem Statement

Payments fraud is a rapidly growing threat, costing financial institutions and payment operators billions of dollars annually. Detecting fraudulent transactions in real time is difficult because fraudulent activity represents only a tiny fraction of total transactions, creating a severe class imbalance problem. Traditional detection methods often fail to identify rare but costly fraudulent activities while generating large numbers of false positives, impacting customer experience.

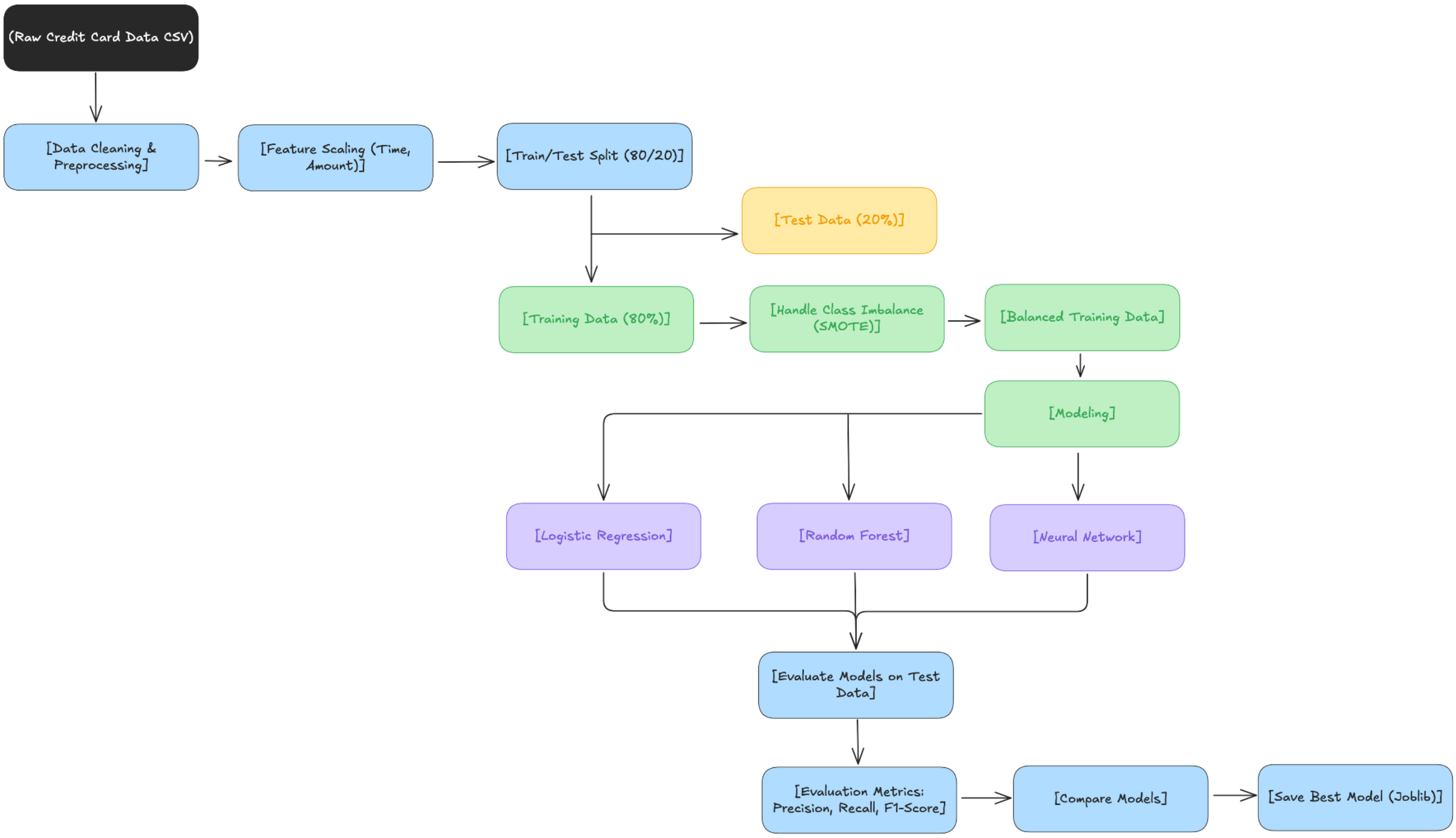
Objective / Aim

- Develop machine learning models that can accurately distinguish between legitimate and fraudulent transactions in real-time.
- Address the challenge of severe class imbalance using advanced sampling techniques like SMOTE.
- Evaluate and compare different classification models using metrics suitable for imbalanced datasets such as Precision, Recall, and F1-Score.
- Deploy the best-performing model into an interactive web application for live demonstrations.

Dataset Details

- Source: Kaggle Credit Card Fraud Detection Dataset
- Size: ~285,000 samples, 31 features
- Key Features:
    - Time ~ Seconds elapsed between each transaction and the first transaction.
    - Amount ~ The transaction amount.
    - V1-V28 ~ Anonymized features resulting from a PCA transformation to protect user privacy.
- Target Variable: Class (0 -> Legitimate, 1 -> Fraudulent).

Architecture Diagram



Methodology

We followed a standard machine learning workflow for a classification problem, with the steps as outlined below:

- **Data Preparation:** Load the credit card fraud dataset using the Pandas library.
- **Feature Scaling:** Apply `StandardScaler` to the `Time` and `Amount` features to normalize their distributions and ensure they are on a comparable scale with the PCA-transformed `V` features.
- **Train/Test Split:** Split the data into an 80% training set and a 20% testing set. Stratification is used to ensure the proportion of fraudulent transactions is identical in both splits, which is crucial for achieving an unbiased evaluation.
- **Class Imbalance Handling:** Apply the *Synthetic Minority Over-sampling Technique* (SMOTE) only to the training set. This balances the class distribution by generating synthetic fraudulent samples, allowing the models to

learn their patterns without developing a bias towards the majority class which are the legitimate transactions in this case.

- **Model Training:** Train three distinct classification models on the balanced training data ~ Logistic Regression (as a baseline), Random Forest (an ensemble method), and a simple Neural Network.
- **Model Evaluation:** Evaluate the trained models on the original, unseen test set. Performance is measured using a classification report and confusion matrix for each model.
- **Persistence:** Save the final, best-performing model and the feature scaler to files using the `joblib` library for later use in a deployment environment.

Results & Evaluation

The model performance is quantified using three key metrics ideal for imbalanced classification:

- **Precision:** Measures the accuracy of positive predictions. In this context, it answers: "Of all transactions flagged as fraud, what percentage were actually fraudulent?" A high precision minimizes false positives.
- **Recall (Sensitivity):** Measures the model's ability to find all positive samples. It answers: "Of all actual fraudulent transactions, what percentage did the model catch?" High recall is critical to minimize false negatives (missed fraud).
- **F1-Score:** The harmonic mean of Precision and Recall. It provides a single score that balances the two, which is useful for comparing overall model performance.

Key Results

The evaluation of the three models on the test set yielded the following results for the **Fraud (Class 1)** category:

Metric	Logistic Regression	Random Forest	Neural Network	Interpretation
Precision	0.06 (6%)	**0.85 (85%)**	0.52 (52%)	The Random Forest model is highly trustworthy when it flags a transaction.
Recall	**0.92 (92%)**	0.84 (84%)	0.82 (82%)	Logistic Regression catches the

				highest percentage of fraud but at a great cost.
F1-Score	0.11	**0.84**	0.63	The Random Forest provides the best balance between precision and recall.

Conclusion

The project successfully implemented and evaluated a machine learning pipeline to detect credit card fraud, confirming that model selection is critical when dealing with severely imbalanced data.

The results clearly show the trade-offs between different models. The **Logistic Regression** model, while achieving the highest recall of **0.92**, is impractical for real-world use due to its extremely low precision of **0.06**. This would lead to an unacceptably high number of legitimate customer transactions being incorrectly flagged as fraud.

The **Random Forest** classifier emerged as the clear winner. It achieved an excellent **F1-score of 0.84**, demonstrating a strong balance between a high precision (**0.85**) and a robust recall (**0.84**). This indicates that the model is both effective at catching fraud and reliable in its predictions.

Recommendation:

The **Random Forest model is the recommended solution** for this fraud detection task. Its superior, well-balanced performance makes it the most suitable choice for deployment, as it effectively minimizes financial losses from fraud without causing significant disruption to legitimate customers.