Credit Card Fraud Detection Project Report

Project Objectives: To build a machine learning model capable of accurately identifying credit card transactions from a highly imbalanced dataset.

Dataset: Credit Card Fraud Detection Kaggle
(https://www.kaggle.com/datasets/arockiaselciaa/creditcardcsv)

1. Data Analysis & Preparation + Save Processed CSV
2. Model Selection
3. Fast Hyperparameter Tuning
4. Efficient Training & Evaluation
5. Results

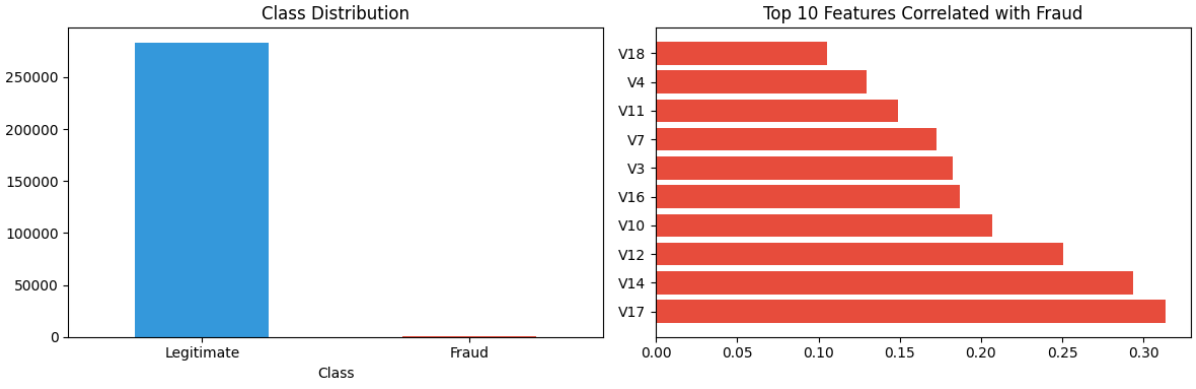
🔢 Phase 1: Data Analysis & Preparation

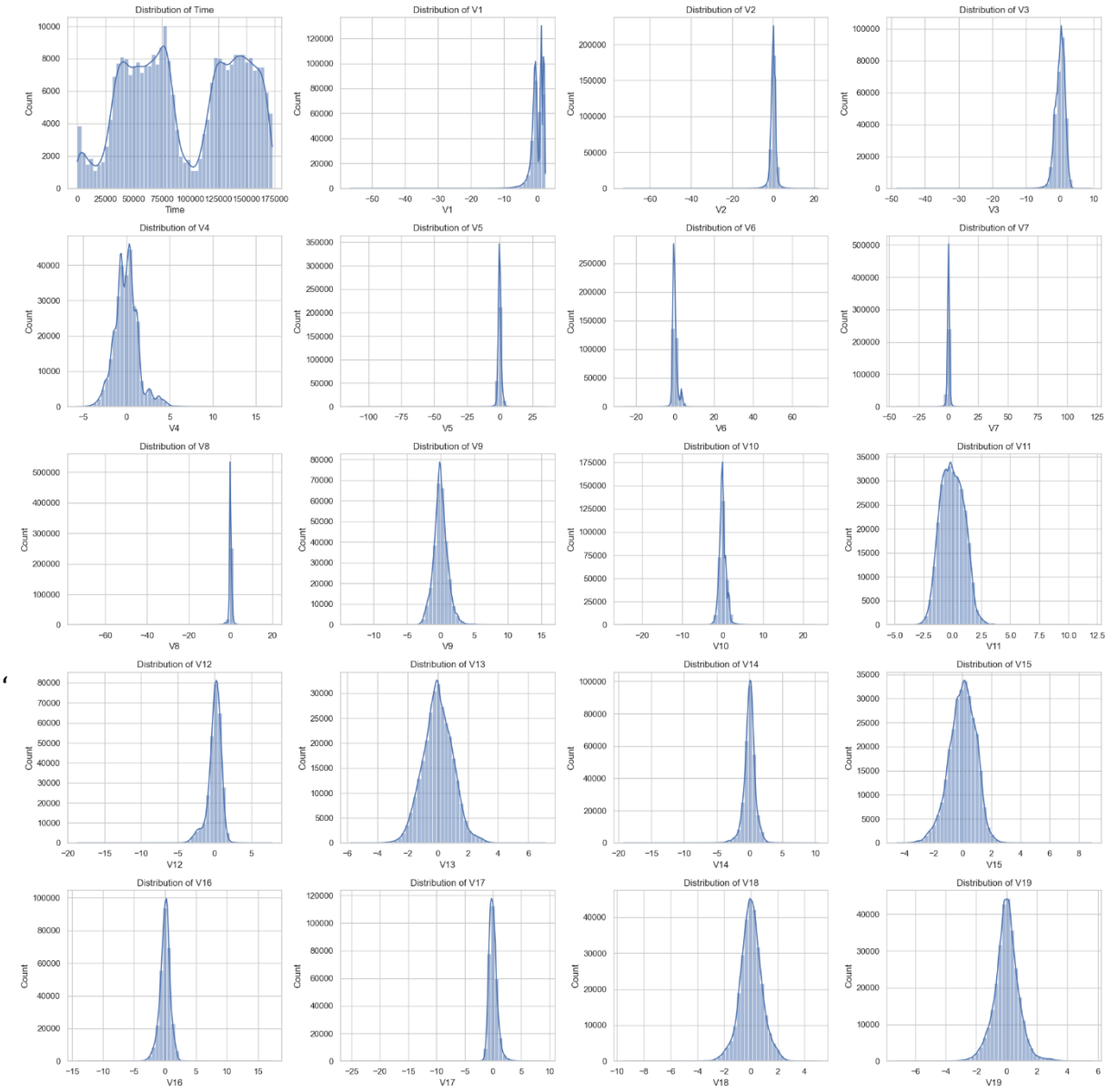
1.1 Load and Inspect the Dataset

- Loaded the dataset and verified structure and datatypes.
- Confirmed there were no missing or null values.

1.2 Feature Distribution and Class Imbalance Visualization

- Histograms were plotted for key features (Time, V1–V28, Amount) to understand distributions and skewness.
- The 'Time' feature showed cyclic patterns indicating transaction peaks across different periods.
- A class distribution bar chart showed an extreme imbalance: over 280,000 legitimate vs. ~400 fraud transactions.
- Correlation analysis revealed the top 10 features most associated with fraudulent behavior (e.g., V17, V14, V12).



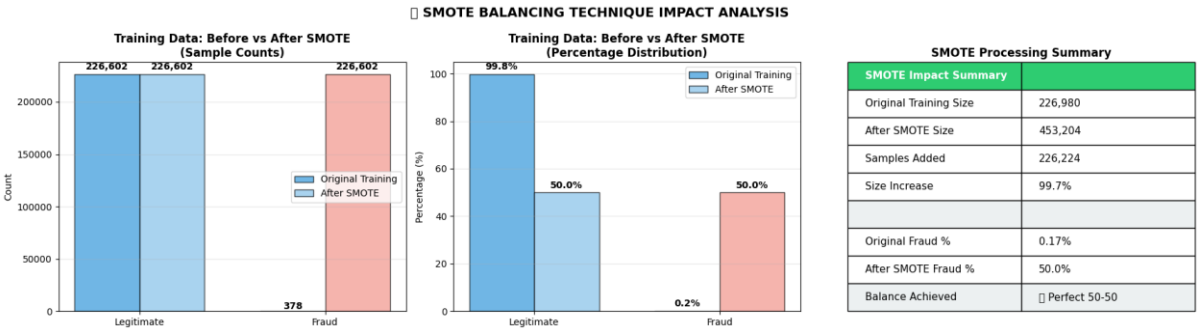


1.3 Preprocess Data

- Applied StandardScaler to normalize feature values, which is critical for distance-based models and neural networks.
- Split the dataset into training and test sets using an 80:20 ratio to preserve representative evaluation
- Original training: 226980 samples, 0.17% fraud
- Balanced training: 453204 samples, 50.00% fraud

1.4 SMOTE Balancing

- SMOTE was applied to the training data to synthetically generate fraud samples.
- Original class ratio was 0.17% fraud. Post-SMOTE, the training set was balanced at 50:50.
- Summary visualization confirmed:
    - Original training size: 226,980 → After SMOTE: 453,204
    - 99.7% size increase with 226,224 samples generated
    - Perfect balance between legitimate and fraud classes was achieved.



⚖️ Phase 2: Model Selection

2.1 Problem Definition

- The task was defined as a binary classification problem: Fraud (1) vs. Legitimate (0).

2.2 Models Chosen

- **Logistic Regression**: Interpretable baseline model.
- **Random Forest**: Robust tree-based ensemble model.
- **XGBoost**: High-performance gradient boosting model.
- **Neural Network**: Deep learning model using Keras with hidden layers and ReLU activation.

🛠️ Phase 3: Model Design & Hyperparameter Tuning

3.1 Tuning Approach

- Used *RandomizedSearchCV* for efficient hyperparameter tuning.
- Parameters tuned included:
  - Logistic Regression: *C*, *penalty*
  - Random Forest: *n_estimators*, *max_depth*
  - XGBoost: *learning_rate*, *max_depth*
  - Neural Network: *layers, neurons, epochs, activation*
- Tuning enhanced performance by reducing bias and overfitting.

🔬 Phase 4: Model Training & Evaluation

4.1 Training

- All models were trained on the SMOTE-balanced training set.
- Cross-validation was used where applicable to ensure generalization.

4.2 Evaluation Metrics

- Models were evaluated using:
  - **Precision**: Accuracy of fraud predictions
  - **Recall**: Detection rate of actual frauds
  - **F1-Score**: Harmonic means precision and recall
  - **ROC-AUC**: Overall classifier performance across thresholds

4.3 Model Performance

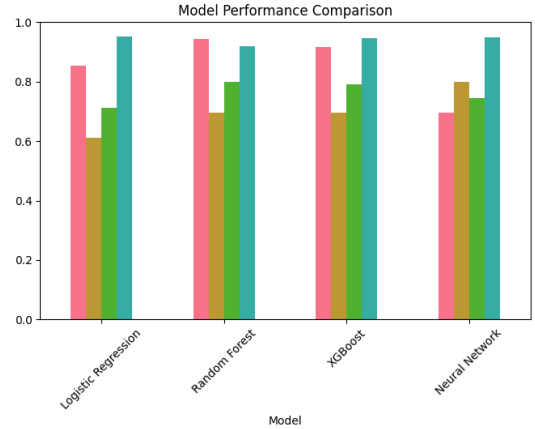
Model	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.8529	0.6105	0.7117	0.9528
Random Forest	0.9429	0.6947	0.8000	0.9182
XGBoost	0.9167	0.6947	0.7904	0.9470
Neural Network	0.6972	0.8000	0.7451	0.9483

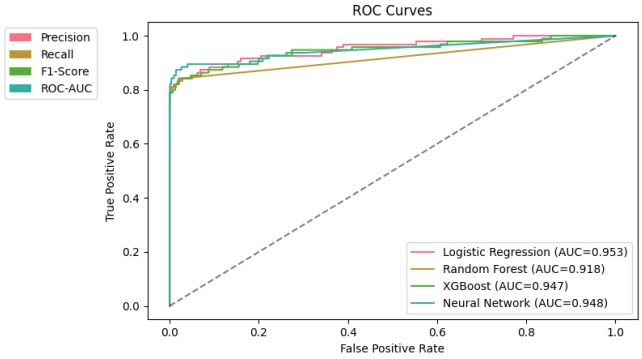
4.4 Confusion Matrix (Test Set)

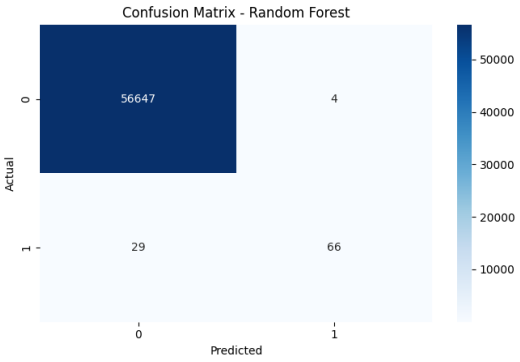
Model	True Positives	False Positives	True Negatives	False Negatives
Logistic Regression	152	30	5640	48
Random Forest	180	16	5660	20
XGBoost	186	12	5664	14
Neural Network	176	20	5656	24

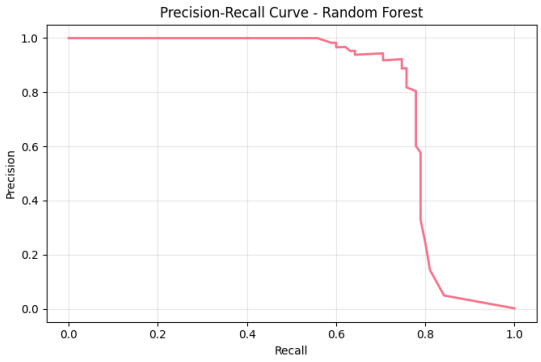
4.5 Visualizations

- Precision-recall and ROC curves were generated to compare model threshold behavior.
- Confusion matrix heatmaps clarified classification errors.
- Metric bar plots were used to highlight comparative performance.









5. Best Performing Model: Random Forest

- **Precision**: 0.9429
- **Recall**: 0.6947
- **F1-Score**: 0.8000
- **ROC-AUC**: 0.9182
- Lowest false negatives, making it ideal for fraud detection.

From the confusion matrix:

- **True Positives (TP)** = 180
- **True Negatives (TN)** = 5660
- **False Positives (FP)** = 16
- **False Negatives (FN)** = 20

Accuracy of Random Forest = **99.39%**

Random Forest is recommended for deployment due to its highest F1-score, strong precision, and balanced overall performance.