

# Credit Card Fraud Detection using Random Forest

**Student Name:** Vinith K M

**Internship:** AI & ML Internship

**Task Number:** Task 9

**Project Title:** Credit Card Fraud Detection using Random Forest

**Tools Used:** Python, Pandas, Scikit-learn, Matplotlib

**Dataset:** Credit Card Fraud (Synthetic Dataset)

**Date:** 29-01-2026

## Introduction

Credit card fraud is a serious problem in the financial industry, causing significant monetary losses each year. Detecting fraudulent transactions is challenging because fraud cases are rare compared to legitimate transactions, leading to highly imbalanced datasets.

This project focuses on building a **Credit Card Fraud Detection system using the Random Forest algorithm**. The model aims to accurately identify fraudulent transactions while minimizing false positives, making it suitable for real-world fraud detection systems.

## Dataset Description

The dataset used in this project is a **synthetic credit card fraud dataset** designed to simulate real transaction behavior.

- **Total Records:** Varies based on dataset
- **Target Variable:** Class
  - 0 → Non-Fraud Transaction
  - 1 → Fraud Transaction

The dataset contains multiple numerical features representing transaction characteristics.

## Data Preprocessing

The following preprocessing steps were performed:

1. Loaded the dataset using Pandas
2. Checked class distribution to understand data imbalance
3. Separated feature columns and target variable
4. Removed non-useful identifiers if present

5. Performed stratified train-test split to maintain fraud ratio

## Handling Imbalanced Data

Fraud detection datasets are highly imbalanced, which makes accuracy an unreliable metric. Therefore, this project focuses on **precision, recall, and F1-score** to evaluate model performance.

A stratified sampling technique was used to ensure the proportion of fraud cases remains consistent across training and testing sets.

## Baseline Model

A **Logistic Regression model** was trained as a baseline to compare performance against the Random Forest model. This helps demonstrate the effectiveness of ensemble learning in handling complex and imbalanced datasets.

## Model Used

### Random Forest Classifier

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting. It aggregates the predictions of several trees to produce a more robust and accurate result.

The model was trained using:

- n\_estimators = 100
- Default Gini impurity criterion

## Model Evaluation

The Random Forest model was evaluated using the following metrics:

- **Precision:** Measures how many predicted fraud cases are actually fraud
- **Recall:** Measures how many actual fraud cases were correctly identified
- **F1-Score:** Harmonic mean of precision and recall

The Random Forest model outperformed the baseline Logistic Regression model, especially in recall and F1-score, making it more suitable for fraud detection.

## Feature Importance Analysis

Feature importance was extracted from the Random Forest model to identify the most influential features in fraud detection. The feature importance plot helps understand which transaction attributes contribute most to fraud predictions.

This improves model interpretability and aids business decision-making.

## Model Saving

The trained Random Forest model was saved using the **joblib** library in .pkl format. This allows the model to be reused for future predictions without retraining.

## Results and Discussion

The Random Forest model demonstrated strong performance on imbalanced data by effectively identifying fraudulent transactions. The use of ensemble learning significantly improved prediction stability and reduced overfitting compared to the baseline model.

The results highlight the importance of choosing appropriate evaluation metrics and models for fraud detection problems.

## Conclusion

This project successfully implements a Credit Card Fraud Detection system using Random Forest. The model effectively handles class imbalance and provides reliable predictions using precision, recall, and F1-score metrics.

## Tools and Technologies Used

- Python
- Pandas
- Scikit-learn
- Matplotlib
- Google Colab
- GitHub