**Student Name:** Akshaya S

**Register Number:** 613023104006

**Institution:** Vivekanandha College of Technology For Women

**Department:** B.E Computer Science And Engineering

**Date of Submission:** 29.04.2025

### **1. Problem Statement**

### In today's digital economy, credit card fraud poses a serious threat to financial institutions and consumers alike. Based on further understanding of the dataset, this project focuses on detecting fraudulent transactions using machine learning to flag suspicious activity in real time.

### Problem Type: Classification (binary classification – fraud or not fraud)

### Why It Matters: Timely and accurate fraud detection helps reduce financial losses, protect consumer trust, and comply with regulations.

### **2. Project Objectives**

### **Technical Objectives:**

### Develop and compare multiple classification models.

### Minimize false negatives (missed frauds) and optimize the F1-score.

### Ensure the model can generalize to unseen transaction patterns.

### **Goal Evolution:** After EDA, the focus shifted to improving recall and managing class imbalance due to the low number of fraud cases in the dataset

### **3. Flowchart of the Project Workflow**

### Data Collection and Preparation:

### Gather transaction data, clean and preprocess it, and handle missing values.

### Exploratory Data Analysis (EDA):

### Explore the dataset, identify patterns, and analyze relationships between features and the target variable.

### Feature Engineering:

### Create new features based on EDA insights and domain knowledge to enhance model performance.

### Model Building and Selection:

### Develop and compare different machine learning models (e.g., Logistic Regression, Decision Tree, Random Forest).

### Model Evaluation:

### Evaluate the performance of each model using appropriate metrics (e.g., accuracy, precision, recall, F1-score).

### Visualization and Interpretation:

### Visualize model results, identify important features, and interpret model behavior.

### Model Deployment and Monitoring:

### Deploy the best-performing model and monitor its performance over time.

### **4. Data Description**

* **Dataset Name**: Credit Card Fraud Detection Dataset (e.g., from a financial institution or open source repository like Kaggle).
* **Data Type:** Structured tabular data, including transaction details like amount, time, location, and cardholder information.
* **Number of Records:** A large number of records, potentially millions, depending on the source.
* **Number of Features:** Several features, including transaction amount, time, location, and other relevant details.
* **Target Variable:** Binary classification (fraudulent or legitimate).

### **5. Data Preprocessing**

### **Missing Values:** No missing values found.

### **Duplicate Records:** Removed 108 duplicate records.

### **Outliers:** Boxplots used to detect and retain extreme values (important for fraud).

### **Data Types:** Confirmed numeric types.

### **Encoding:** Not needed (PCA-transformed features).

### **Scaling:** Applied Standard Scaler to Amount and Time

### **6. Exploratory Data Analysis (EDA)**

### **Univariate Analysis:**

### Fraud cases make up only 0.17% of all transactions.

### Skewed distribution in Amount, hence scaling was critical.

### **Bivariate/Multivariate Analysis:**

### Correlation matrix shows V14, V17, and V10 are strongly linked to fraud.

### Fraud transactions cluster in certain feature ranges.

### **Insights Summary:**

### Severe class imbalance requires techniques like SMOTE or under sampling.

### PCA features help anonymize sensitive info but require interpretability analysis.

### **7. Feature Engineering**

### **Created Features:** Hour Of Day derived from Time feature.

### **Transformation:** Normalized Amount and engineered Amount\_per\_Hour.

### **Justification:** Time-based trends may expose fraud patterns (e.g., overnight transactions).

### **8. Model Building**

### **Models Used:**

### Logistic Regression

### Random Forest Classifier

### **Why Chosen:**

### Logistic Regression for baseline interpretability

### Random Forest for handling nonlinear relationships and feature importance

### **Data Split:** 80% training, 20% test (stratified by churn)

### **Evaluation Metrics:**

### Accuracy, Precision, Recall, F1-Score

### Logistic Regression F1: 0.78

### Random Forest F1: 0.86

### **9. Visualization of Results & Model Insights**

### **Confusion Matrix:** Shows high true positive rate for Random Forest.

### **ROC Curve:** Random Forest outperforms baseline with AUC = 0.99

### **Feature Importance:** V17, V14, V10 are top predictors of fraud.

### **Interpretation:** Time-independent PCA features hold significant predictive power.

### **10. Tools and Technologies Used**

### **Programming Language**: Python

### **IDE/Notebook**: Google Colab & Jupyter Notebook

### **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, imbalanced-learn, XGBoost

### **Visualization Tools**: seaborn, matplotlib, Plotly

### **11. Team Members and Contributions:**

* Team Head: Akshaya S

Responsibilities: Handled data cleaning, deal with missing values, and ensured data consistency.

* Hemavarni S

Responsibilities: Conducted Exploratory Data Analysis and prepared visual summaries.

* Iniya S

Responsibilities: Performed feature engineering and implemented model training pipelines.

* Bhavani Nachiyar S

Responsibilities: Focused on model evaluation, performance metrics, and created visualizations and reporting.

* Kanishkaa V

Responsibilities: Worked on hyperparameter tuning, testing and different algorithms, and integrating final output