

Final Project Overview

Online Payment Fraud Detection using Machine Learning

Date	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	10 Marks

Project Summary:

This project delivers a complete, end-to-end machine learning solution for detecting fraudulent online payment transactions. Starting from raw transactional data sourced from Kaggle, the project progresses through systematic phases of data exploration, preprocessing, multi-model training, evaluation, and Flask-based web deployment. The final system classifies transactions as FRAUD or NOT FRAUD in real time through a clean web interface.

Key Outcomes:

- Trained and compared five classification algorithms: Random Forest, Decision Tree, ExtraTrees, SVM, and XGBoost on the Kaggle fraud detection dataset.
- Achieved up to 79% classification accuracy with the SVM (Support Vector Classifier) selected as the best-performing model.
- Conducted comprehensive EDA including univariate, bivariate, and descriptive analysis with 20+ visualizations.
- Built and deployed a Flask web application with three HTML pages (home, predict, result) for real-time fraud prediction.
- Saved the trained model using Pickle for efficient loading and inference in the production Flask application.
- Delivered complete 7-phase project documentation covering Ideation, Requirements, Design, Planning, Development, Documentation, and Demonstration.

Technology Stack:

Component	Technology
-----------	------------

Programming Language	Python 3.x
ML Framework	Scikit-learn, XGBoost
Data Processing	Pandas, NumPy
Visualization	Matplotlib, Seaborn
Web Framework	Flask
Model Storage	Pickle (.pkl)
Frontend	HTML5, CSS3, Bootstrap
Dataset	Kaggle — PS_20174392719 (6.3M records)
Development Env	Anaconda Navigator / VS Code

Project Flow:

Step 1: Data Collection → Download Kaggle fraud dataset (PS_20174392719_1491204439457_logs.csv)

Step 2: Data Preprocessing → Remove unnecessary columns, handle null values, encode categorical variables, treat outliers

Step 3: Exploratory Data Analysis → Univariate, bivariate, and descriptive statistical analysis with visualizations

Step 4: Model Building → Train and evaluate 5 classification algorithms

Step 5: Model Evaluation → Compare accuracy, confusion matrix, classification report; save best model with Pickle

Step 6: Application Building → Develop Flask app with 3 HTML pages; integrate model for real-time prediction

Step 7: Documentation & Demonstration → Complete 7-phase documentation + end-to-end demo video