

Fraud Detection Machine Learning Project

The Problem: E-commerce and financial businesses lose millions annually to fraudulent transactions. This project addresses this by building an end-to-end machine learning pipeline to detect fraud in credit card data. Inspired by an X post on standing out in job applications, I chose a simple problem (fraud detection) and went all in: training a model, deploying it as an API with a web form, adding monitoring, hosting on cloud, and documenting everything in this PDF.

The Dataset: I used the public Credit Card Fraud Detection dataset from Kaggle, containing 284,807 transactions with features like Time, Amount, and anonymized V1-V28. Fraud cases are rare (~0.17%), so I balanced the dataset using RandomUnderSampler to ~492 fraud and 492 non-fraud samples for effective training.

Training Pipeline:

1. Preprocessed data: Handled imbalance with RandomUnderSampler, split into train/test (80/20), and scaled features using StandardScaler.
2. Trained a RandomForestClassifier (n_estimators=100, random_state=42).
3. Evaluated on test set:
 - Precision (non-fraud): 0.94, Recall: 0.92, F1: 0.93
 - Precision (fraud): 0.92, Recall: 0.94, F1: 0.93
 - Accuracy: 0.93
 - Confusion Matrix: $\begin{bmatrix} 91 & 8 \\ 6 & 92 \end{bmatrix}$ (high true positives for fraud). This demonstrates strong performance in catching fraud while minimizing false positives.

Architecture, Deployment, and Impact

Architecture: The pipeline is modular:

- train_model.py: Loads data, preprocesses, trains, evaluates, and saves model/scaler as .pkl files.
- app.py: Loads saved model, serves predictions via Flask API (/predict endpoint) and web form (HTML template with inputs for Time, V1-V28, Amount).
- Monitoring: Prometheus Counter tracks requests (/metrics endpoint). [Insert Diagram Here: Flowchart showing data flow from training to user prediction]

Deployment: Hosted on Heroku (free tier) for public access:

- Used Gunicorn for production server (Procfile: web: gunicorn app:app).
- Dependencies in requirements.txt (pinned versions like scikit-learn==1.5.2 for compatibility).
- Live URL: <https://rdamon-fraud-detector-f3c10ca7f356.herokuapp.com> (try the form or /metrics).
- Source Code: <https://github.com/Rdamon223/fraud-detection-project>.

Business Impact and Skills Demonstrated: This app could reduce fraud losses by detecting 94% of cases in real-time via API integration. It showcases full-stack ML skills: data processing, modeling, web deployment, monitoring, and documentation. Unlike a resume, this tangible project proves I can create, debug, and ship production-like solutions.