Fraud‑Detection FastAPI + Streamlit – Quick‑Start Guide

# 1. Project layout (high‑level)

fraud\_fastapi\_app\_v5/  
├─ main.py # FastAPI entry‑point  
├─ streamlit\_app.py # Streamlit UI  
├─ requirements.txt  
├─ project/  
│ ├─ api/ # FastAPI routers (train / test / predict)  
│ ├─ services/ # ML logic (training, inference, evaluation)  
│ ├─ utils/ # helpers (save/load, etc.)  
│ └─ resources/  
│ ├─ data/ # sample CSVs (carclaims.csv, new\_data.csv)  
│ └─ models/ # persisted artifacts are written here at run‑time  
└─ README.md

# 2. Installing & launching

|  |  |  |
| --- | --- | --- |
| Step | Command | Notes |
| Create a virtual environment | python -m venv .venv && source .venv/bin/activate (Linux/Mac) .\.venv\Scripts\activate (Windows) | Python 3.11 recommended |
| Install dependencies | pip install -r requirements.txt | CuDNN/GPU optional but speeds training |
| Start FastAPI (optional) | uvicorn main:app --reload | Swagger UI available at /docs |
| Start Streamlit UI | streamlit run streamlit\_app.py | Opens http://localhost:8501 |

# 3. Using the Streamlit interface

Sidebar contains:

* Train Model
* Test Model for Optimal Threshold
* Predict Fraud

## 3.1  Train Model

|  |  |  |
| --- | --- | --- |
| What you see | What to do | What happens under the hood |
| “Start Training” button | Click once. Button greys out with “Training in progress…” | • Pre‑processing pipeline • Mutual‑info + DNN feature selection • Train DNN & XGBoost + soft‑voting ensemble • Save artifacts under project/resources/models/ • Return training summary JSON |

## 3.2  Test Model for Optimal Threshold

|  |  |  |
| --- | --- | --- |
| What you see | What to do | Output |
| “Test & Get Threshold” button | Click after at least one training run | \* Confusion‑matrix heat‑map \* Best threshold value \* Classification report |

## 3.3  Predict Fraud

|  |  |  |
| --- | --- | --- |
| What you see | What to do | Output |
| CSV uploader + download link | Upload CSV (same schema, no FraudFound\_P column) | \* Threshold echoed \* Table with fraud\_prediction column \* Download predictions CSV |

# 4. Typical end‑to‑end workflow

1. Train Model after each batch of labeled data.
2. Test Model for Optimal Threshold immediately afterwards.
3. Hand threshold to operations.
4. Use Predict Fraud for incoming unlabeled claims.
5. Periodically append outcomes and retrain.

# 5. Troubleshooting & FAQ

|  |  |  |
| --- | --- | --- |
| Symptom | Likely cause | Fix |
| “Training in progress” never clears | Error during training | Inspect terminal logs; reduce batch size / check GPU RAM |
| ModuleNotFoundError: project… | Ran Streamlit from wrong path | cd fraud\_fastapi\_app\_v5 then run command |
| Predictions all zeros | Skipped Test Model step | Run threshold search or lower cut‑off |
| Browser frozen on wait | Training very long | Open second tab for monitoring or reduce epochs |

# 6. Where to go next

* Swagger UI: http://127.0.0.1:8000/docs
* Edit project/services/model\_service.py to tweak models.
* Enhance streamlit\_app.py with extra visualisations (ROC, SHAP).