**AI-Powered Personal Finance Alert System**

* **Goal: Predict suspicious or unusual spending patterns from personal finance data.**
* **Data Collection: Use synthetic or anonymized datasets (Kaggle finance datasets).**
* **ML Skills: Anomaly detection (Isolation Forest, Autoencoders), time-series modeling.**
* **Evaluation: Precision/Recall on injected anomalies, visual dashboards.**
* **Appeal: Practical, multi-modal ML (numeric + temporal data), highly portfolio friendly.**

**✅ Combined Project Flow**

1. **Data collection & preprocessing:** TestDataBox finance dataset → handle missing values, feature engineering.
2. **EDA & statistical analysis:** Correlation analysis, outlier z-scores, feature distributions.
3. **Modeling:** Isolation Forest, Autoencoder, XGBoost → anomaly/fraud detection.
4. **MLOps setup:** Track experiments in MLflow or W&B.
5. **API Deployment:** Dockerize model + FastAPI → expose anomaly/fraud prediction endpoint.
6. **Distributed processing:** Use PySpark or Dask for feature processing on large datasets.
7. **Monitoring & Reporting:** Log metrics, model drift, or prediction statistics.
8. **Visualization & Dashboard:** Interactive plots (Plotly/Streamlit) for anomalies and stats.

💡 **Outcome:**  
You now have a **production-ready ML system** that demonstrates:

* ML modeling & evaluation
* Statistical inference
* MLOps workflow
* Containerization (Docker + optional Kubernetes)
* API deployment
* Big data / distributed computing

This is **exactly the type of project that impresses AI/ML internship recruiters**.

If you want, I can **update the 2-month roadmap to integrate all these MLOps, API, and big-data enhancements** so you have a **complete end-to-end production-ready timeline**.

Do you want me to do that?

**You said:**

Yes and add the optional things too. Since I have adding things, I can give more than 1 hour

**ChatGPT said:**

Perfect 🔥 — since you’re willing to invest **more than 1 hour/day**, we can upgrade this from a “class project” into a **production-ready, MLOps-enabled ML system**. This will *seriously* stand out on LinkedIn and in interviews because it combines:

* **ML research side** (EDA, anomaly detection, fraud classification, statistics)
* **MLOps/Engineering side** (Docker, APIs, MLflow, Spark, Kubernetes, monitoring)

Here’s the **expanded 8-week roadmap** with all optional pieces included.

**📅 Expanded 8-Week Roadmap (with MLOps + Deployment)**

**Week 1 – Setup & Data Understanding**

* Download **Synthetic Finance Dataset (TestDataBox)**.
* Set up repo structure (src/, notebooks/, data/, models/, api/, docker/).
* Explore dataset schema (customers, accounts, loans, transactions).
* Identify data quality issues: missing values, duplicates, invalid dates, anomalies.

📌 **Deliverable:** Clean repo structure + initial data exploration notebook.

**Week 2 – Data Preparation & Feature Engineering**

* Handle missing values (impute, drop, flag as anomaly features).
* Fix duplicates & formatting issues (currency, dates, IDs).
* Feature engineering:
  + Transaction frequency, mean spend, std deviation, loan-to-income ratio, debt ratio.
* Save **finance\_cleaned.csv** for modeling.

📌 **Deliverable:** Clean dataset + reproducible preprocessing script (src/preprocess.py).

**Week 3 – EDA + Statistical Analysis**

* Visualize distributions (income, loan size, spending patterns).
* Correlation analysis (which features drive anomalies?).
* Apply statistical anomaly methods: z-scores, IQR outlier detection.
* Compare **statistical vs ML anomaly detection** later.

📌 **Deliverable:** EDA & stats notebook with clear visualizations.

**Week 4 – Baseline ML Models (Anomaly Detection & Classification)**

* Implement **Isolation Forest** + visualization.
* Implement **Local Outlier Factor (LOF)**.
* Train supervised classifiers (Random Forest, XGBoost) on labeled/injected fraud cases.
* Evaluate with metrics: Accuracy, Precision, Recall, F1, ROC-AUC.

📌 **Deliverable:** Baseline ML models with evaluation report.

**Week 5 – Advanced ML + Explainability**

* Build **Autoencoder** for anomaly detection.
* Compare Autoencoder vs Isolation Forest vs LOF vs XGBoost.
* Add explainability:
  + **SHAP values** (feature importance)
  + Partial dependence plots
* Document which features most influence fraud/anomalies.

📌 **Deliverable:** Advanced ML notebook + explainability results.

**Week 6 – MLOps Integration**

* Add **MLflow** (or Weights & Biases) for experiment tracking.
* Log preprocessing, model parameters, metrics.
* Save best-performing model to **models/** with versioning.
* Add **Airflow/Prefect pipeline** for automation (ETL → Train → Evaluate).
* Implement **model monitoring** (track drift, anomaly frequency over time).

📌 **Deliverable:** Automated ML pipeline with experiment tracking + monitoring.

**Week 7 – Deployment (API + Containerization)**

* Build **FastAPI service**:
  + /predict → takes transaction JSON, returns anomaly score/fraud label.
  + /health → health check endpoint.
* Add input validation (pydantic).
* Containerize app with **Docker**.
* Optional advanced: Deploy to **Kubernetes** (local minikube or cloud cluster).
* Add **logging + error handling** in API.

📌 **Deliverable:** REST API containerized with Docker (and optionally orchestrated with Kubernetes).

**Week 8 – Big Data + Final Polish**

* Re-implement preprocessing + anomaly detection using **PySpark/Dask** for scalability.
* Compare performance (pandas vs Spark).
* Build **Streamlit dashboard**:
  + Transaction stats
  + Anomalies detected over time
  + Fraud detection ROC curve
* Write final **project report** (problem, methods, results, deployment).
* Push to **GitHub** with:
  + README (setup, usage, demo)
  + Screenshots & visuals
  + Clear folder structure
* Post polished **LinkedIn project story** (with dashboard GIF/demo video).

📌 **Deliverable:** End-to-end production-ready ML system (GitHub + demo + LinkedIn showcase).

**🚀 Final Deliverables**

* **GitHub Repo**
  + /src → preprocessing, training, evaluation scripts
  + /notebooks → EDA, models, visualizations
  + /models → saved models
  + /api → FastAPI code
  + /docker → Dockerfile, Kubernetes manifests
  + README.md → overview, setup, usage
* **Deployed API** (local Docker or cloud)
* **Streamlit dashboard** for anomaly insights
* **MLflow/W&B experiment logs**
* **Final report + LinkedIn post**

✅ With this roadmap, your project will cover:

* Data collection, cleaning, feature engineering
* ML anomaly detection & fraud classification
* Statistical inference & explainability
* MLOps (experiment tracking, pipelines, monitoring)
* Deployment (API, Docker, Kubernetes)
* Big Data (PySpark/Dask)
* Dashboard + LinkedIn polish

This makes it **resume-grade + recruiter eye-catching**.

**GitHub Repo Structure**

synthetic-finance-mlops/

│

├── data/ # Datasets (raw + processed)

│ ├── raw/ # Original dataset (not committed if too large)

│ ├── processed/ # Cleaned + feature-engineered data

│

├── notebooks/ # Jupyter notebooks for exploration + modeling

│ ├── 01\_data\_exploration.ipynb

│ ├── 02\_feature\_engineering.ipynb

│ ├── 03\_baseline\_models.ipynb

│ ├── 04\_advanced\_models.ipynb

│ ├── 05\_explainability.ipynb

│

├── src/ # Python modules (reusable code)

│ ├── preprocess.py # Data cleaning & feature engineering

│ ├── train.py # Model training pipeline

│ ├── evaluate.py # Evaluation metrics & plots

│ ├── inference.py # Model inference (used by API)

│ ├── utils.py # Helper functions

│

├── models/ # Saved models (versioned with MLflow too)

│ └── best\_model.pkl

│

├── api/ # REST API (FastAPI)

│ ├── main.py # FastAPI app entrypoint

│ ├── requirements.txt # API-specific dependencies (optional)

│

├── docker/ # Containerization & orchestration

│ ├── Dockerfile # For containerizing the project

│ ├── docker-compose.yml # For local multi-service testing

│ ├── k8s-deployment.yaml # Optional: Kubernetes manifest

│

├── airflow/ # (Optional) Workflow orchestration

│ ├── dags/

│ │ └── finance\_pipeline.py

│

├── dashboard/ # Streamlit dashboard

│ └── app.py

│

├── tests/ # Unit tests

│ ├── test\_preprocess.py

│ ├── test\_train.py

│ ├── test\_api.py

│

├── .gitignore # Ignore venv, data, logs, etc.

├── README.md # Project documentation

├── requirements.txt # Full dependencies (for pip install)

└── setup.sh # Quick setup script (create venv + install deps)

Github Token: github\_pat\_11AYA6EQI0bXKf2J8QYHvZ\_8RQ3MZ5UZbqOK8r7I7DG9mE99HONzUt5T16AlWf9LbDXLU7JQCVYWWd9iaT

Pat\_token:  
github\_pat\_11AYA6EQI0vVjhA5sVzGCj\_SoHn74yPFcd5eyDQSFO5gXWWbnhcDSxKZwMhzvWSrxUBFZULRRJqUw2FfEH

**DAY 1**

* Created project directory and added all necessary things
* Setup python environment and installed dependencies
* Creted github repo with dev and main branch and pushed changes to both
* Loaded Dataset and added to project raw data folder