Smarter Reconciliation and Anomaly Detection System - Runbook

🔧 Project Overview

This project provides a FastAPI-powered real-time and batch anomaly detection system for financial reconciliation data. It integrates ML models, LLMs for categorization/resolution, WebSockets for live updates, and email notifications.

🗂️ Project Structure

econ\_anomaly\_detection/

├── api/

│ ├── \_\_init\_\_.py

│ ├── main.py

│ ├── models.py

│ └── websocket\_manager.py

├── config/

│ └── config.yaml

├── data/

│ └── generated\_reconciliation\_data.csv

├── models/

│ ├── isolation\_forest\_model.pkl

│ ├── kmeans\_model.pkl

│ └── label\_encoder.pkl

├── modules/

│ ├── data\_ingestion.py

│ ├── data\_preparation.py

│ ├── model\_layer.py

│ ├── llm\_integration.py

│ ├── data\_persistence.py

│ ├── data\_validation.py

│ ├── agentic\_ai.py

│ └── utils.py

├── scripts/

│ └── model\_training.py

└── ui/

└── ui\_integration.py

🚀 How to Run

1. Setup Environment

python -m venv venv

source venv/bin/activate # Linux/macOS

venv\Scripts\activate # Windows

pip install -r requirements.txt

2. Configure the App

Edit config/config.yaml with proper email credentials, paths, and thresholds.

Example:

paths:

data\_file: data/generated\_reconciliation\_data.csv

anomaly\_output: output/detected\_anomalies\_{timestamp}.csv

log\_file\_name: logs/recon\_anomaly\_detection\_{timestamp}.log

data\_validation:

quantity\_threshold: 10

email:

sender: your\_email@gmail.com

password: your\_password

recipient: recipient@example.com

api\_keys:

openai: YOUR\_OPENAI\_API\_KEY

3. Run the API

uvicorn api.main:app --reload

# Or if renamed

uvicorn api.anomalydetection:app --reload

4. Test API Endpoints

- POST /realtime\_anomaly/ – Detect anomaly in a single record

- POST /batch\_anomaly/ – Detect and categorize batch data

- GET /health – Check app status

- WS /ws – Real-time WebSocket anomaly updates

💼 Usage Examples

Realtime Example:

POST /realtime\_anomaly/

{

"TRADEID": "T123",

"COMMENT": "Mismatch detected",

"QUANTITY": 50,

"AMOUNT": 2000.5,

...

}

Batch Example:

POST /batch\_anomaly/

[

{

"TRADEID": "T001",

"COMMENT": "Late settlement",

...

},

{

"TRADEID": "T002",

"COMMENT": "Negative amount",

...

}

]

🧠 LLM & Agentic AI

- llm\_integration.py uses OpenAI/Gemini to categorize anomalies & generate summaries.

- agentic\_ai.py can apply feedback or auto-resolution logic.

📬 Notifications

Sends an email with the CSV report when anomalies are detected.

🧹 Cleanup / Cache Issues

If FastAPI app behaves unexpectedly:

find . -type d -name "\_\_pycache\_\_" -exec rm -r {} +

find . -name "\*.pyc" -delete

PowerShell:

Get-ChildItem -Recurse -Include "\_\_pycache\_\_" | Remove-Item -Recurse -Force

Get-ChildItem -Recurse -Include "\*.pyc" | Remove-Item -Force

🛠️ Model Training

python scripts/model\_training.py

📈 Future Enhancements

- Database integration (PostgreSQL/MongoDB)

- UI dashboard for monitoring

- Docker containerization

- CI/CD pipeline

- Feedback learning loop API