**Project Title**

**AI-Driven Financial Risk Analysis System**

**1. Project Overview**

This project automates the analysis of financial transaction data to identify potential fraud risks. It leverages OpenAI's GPT API to extract insights, calculate risk scores, and provide structured outputs for decision-making. The system reduces manual effort, enhances accuracy, and provides actionable insights for analysts.

**2. Key Features**

**Entity Extraction:** Extracts relevant entities from structured (CSV) and unstructured (text) transaction data.

**Risk Scoring Framework:**

Evaluates transactions across five categories:

* Entity Reputation (20 pts)
* Transaction Amount (15 pts)
* Geographic Risk (15 pts)
* Transaction Pattern (20 pts)
* External Data (30 pts)

Calculates a Total Risk Score by summing sub-category scores.

**Data Enrichment:** Incorporates external data (e.g., IP geolocation) to enhance analysis.

**Output Structuring:** Converts AI-generated responses into structured JSON for easy integration.

**Error Handling:** Manages incomplete or invalid inputs gracefully.

**3. Workflow**

**Input Handling:**

Reads transaction data from files (CSV or text) and processes it into a suitable format.

**AI Analysis:**

Sends data to OpenAI for scoring and reasoning based on the predefined framework.

**Output Processing:**

Parses AI responses, extracts category-wise scores, and calculates the total risk score.

**Final Output:**

Returns structured JSON with detailed scores and reasoning for decision-making.

**4. Technologies Used**

Python: Core programming language for building the system.

OpenAI GPT API: For natural language processing and risk analysis.

Pandas: For handling structured data (e.g., CSV files).

Regular Expressions (re): For cleaning and structuring text data.

JSON: For structured output and integration.

**5. Challenges Faced**

OpenAI API Integration: Ensuring consistent and accurate AI responses while managing rate limits.

Data Handling: Processing diverse input formats and managing inconsistencies.

Output Structuring: Parsing and formatting AI responses into usable JSON.

Error Handling: Managing invalid inputs and incomplete AI outputs.

Scalability: Designing the system to handle large datasets efficiently.

**6. Sample Code Snippets**

1. **Open AI Integrations**

client = OpenAI(api\_key="your\_api\_key\_here")

system\_message = {"role": "system", "content": "Act as a financial data analyst..."}

completion = client.chat.completions.create(

model="gpt-4o-mini",

messages=[

system\_message,

{"role": "user", "content": "Here is the transaction detail for you to analyze: 'data'"},

]

)

1. **Risk Scoring Fundamentals**

### Scoring System (Total Risk Score: 0-100):

1. Entity Reputation (20 pts): 0-5 (Clean history) to 16-20 (Sanctions).

2. Transaction Amount (15 pts): 0-5 (Normal) to 11-15 (Extremely high/low).

3. Geographic Risk (15 pts): 0-5 (Low-risk) to 11-15 (High-risk regions).

4. Transaction Pattern (20 pts): 0-5 (Normal) to 16-20 (No logical purpose).

5. External Data (30 pts): 0-10 (No red flags) to 21-30 (Sanctions/legal actions).

1. {

"Entity Reputation": 5,

"Transaction Amount": 5,

"Geographic Risk": 10,

"Transaction Pattern": 10,

"External Data": 10,

"Total Risk Score": 40

}

**7. Outcome**

The system automates manual transaction analysis, reduces errors, and provides actionable insights for financial analysts. It is scalable, customizable, and integrates seamlessly with external data sources for enhanced risk assessment.