

Agentic Financial Workflow Using AWS Bedrock

Bushra Paracha (Prompt Engineer)
Abdul Sheikh (Data Engineer)
Mishar Mozumder (Financial Analyst)



Introduction & Dataset Overview

Project Goal:

- Automate financial insights using an agentic AI workflow powered by AWS Bedrock
- Our workflow processes raw transactions → categorizes them → computes KPIs → produces a summary → and evaluates errors through reflection

Dataset Overview:

- File: transactions.csv
- Personal Finance Data
- Columns: date, merchant, amount, description
- Represents realistic monthly spending and income

	date	merchant	amount	description
1	10/1/2024	Amazon	45.99	Online shopping - household items
2	10/1/2024	Starbucks	8.25	Coffee and snack
3	10/2/2024	Trader Joe's	76.4	Grocery shopping
4	10/2/2024	Uber	22.5	Transportation ride
5	10/3/2024	Blue Bottle	6.5	Coffee
6	10/3/2024	Costco	185.75	Monthly groceries

Agentic Workflow: The 5 Stages

- PLAN: Model creates a 5-step plan to analyze transactions
- ACT (Categorization): Each transaction assigned to one category
Shopping, Dining, Utilities, Income, Other
- OBSERVE (KPIs): Compute financial metrics such as total spend, income, top merchants, and averages
- SUMMARIZE: Model generates a ≤100-word narrative summary of the month.
- REFLECT: Model reviews for errors, misclassifications, KPI mistakes, and improvements.

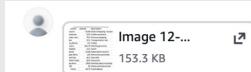


Image 12...

153.3 KB

You are a financial analysis agent.
You are given this month's transaction dataset with the following columns: date, merchant, amount, and description.
Your job is to design a clear 5-step analysis plan that follows this agentic reasoning pattern:
1. PLAN – Determine how you will approach analyzing the financial data.
2. ACT – Explain how you will process or classify the transactions.
3. OBSERVE – Describe how you will review or compute key financial metrics based on the processed data.
4. SUMMARIZE – Explain how you will turn the insights into a concise monthly financial summary.
5. REFLECT – Describe how you will evaluate the results and identify improvements for future analysis.
Each step must be specific to financial transaction analysis and must clearly reflect its position in the sequence Plan → Act → Observe → Summarize → Reflect.
Return your response ONLY in valid JSON using the exact structure below:
{ "plan_steps": ["step 1 description...", "step 2 description...", "step 3 description...", "step 4 description...", "step 5 description..."] }
Do not include any text outside of the JSON. Do not add extra fields, commentary, labels, or explanations.

```
[  
  "plan_steps": [  
    "Organize transactions by category (e.g., groceries, utilities, income) and calculate total income and expenses",  
    "Create a cash flow statement by subtracting total expenses from total income",  
    "Calculate key financial ratios such as savings rate, debt-to-income ratio, and discretionary spending percentage",  
    "Generate a concise report highlighting total income, expenses, savings, and top spending categories",  
    "Identify spending patterns, potential areas for budget optimization, and set financial goals for the next month"  
  ]
```

FinTech Relevance

- Small businesses use similar pipelines to automate bookkeeping and expense tracking.
- Categorization + KPIs provide insights into cash flow, spending habits, and vendor costs.
- Structured JSON outputs resemble the internal logic of tools like QuickBooks, Mint, and Wave.
- Reflection mirrors real auditing loops where systems self-correct classification errors.
- Visual KPI charts replicate the dashboards used by personal finance apps.



Lessons Learned

Challenges Faced

- Several category misclassifications (e.g., Uber and Shell Gas → Transportation)
- Spotify labeled as Other due to missing “Subscriptions” category
- Top merchants list incorrect at first (AT&T mistakenly replaced by Whole Foods)
- Needed stricter JSON-only prompting and explicit sorting instructions

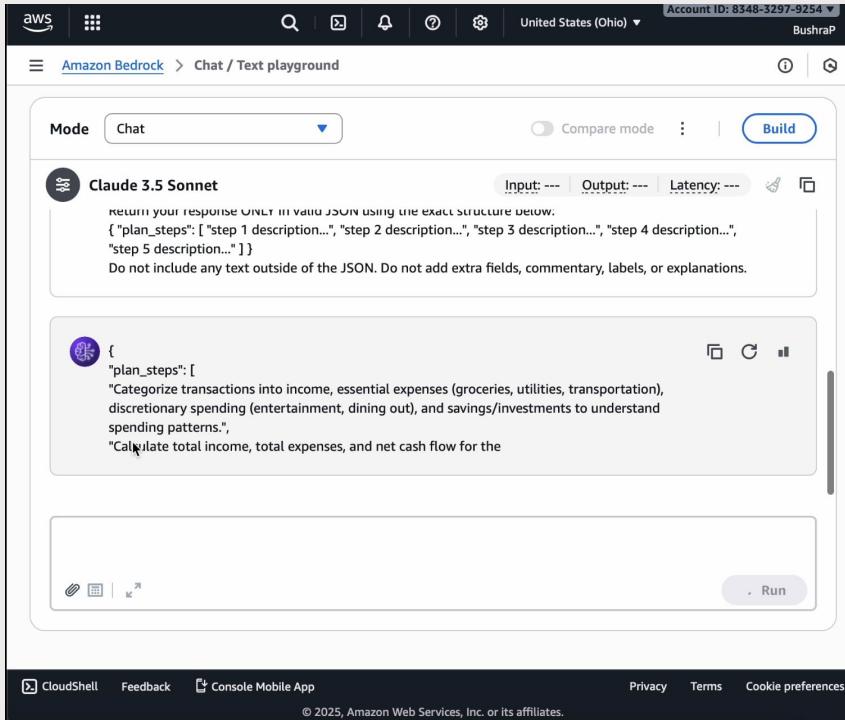
Improvements Implemented

- Strengthened prompts with explicit schemas (“ONLY return JSON”)
- Manually validated KPI totals and merchant rankings
- Added guidelines for category rules in future iterations
- Created optional KPI visualizations for clearer insights

Team Participation

- Bushra: Prompt engineering, model iteration, JSON cleanup.
- Abdul: Data generation, CSV creation, pipeline setup.
- Mishar: KPI validation, financial insight checks, summary + reflection refinement.

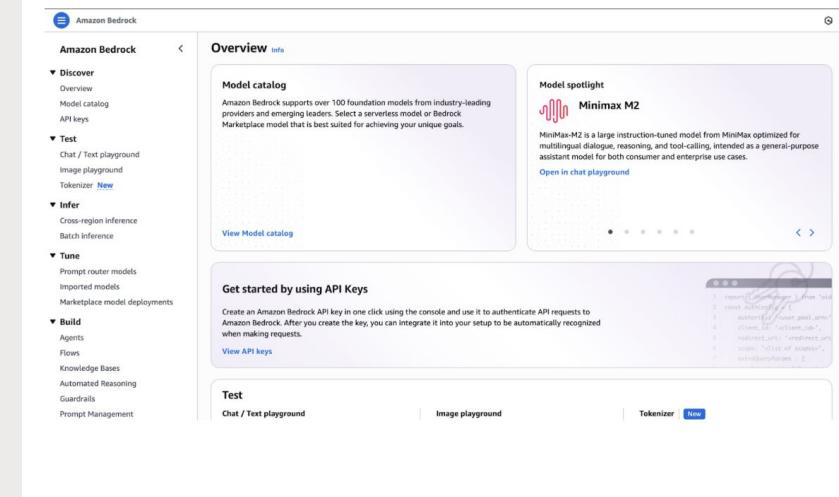
Demonstration Video



Prompt Demonstration

Prompting process

Step 1: After logging into AWS Bedrock, go to Chat/ Text Playground under Text option



ISE 391 Prompts doc