Archisha Nangia

Smart Financial Coach Case Study

This serves as a personal finance assistant designed to highlight intelligent insights and help users manage spending more proactively.

My vision was not just to visualize user financial data, but to **explain it** as how a real financial coach would and provide the full picture. Simply, this project empowers anyone, regardless of financial literacy, to visualize, understand, and improve their spending behavior. It's an abstraction layer - designed to make AI accessible and actionable for everyday decisions.

Design Choices

The application is centered on 3 MVPs which are: AI-Powered Spending Insights, Personalized Goal Forecasting, and Subscriptions & Gray Charges Detector.

1. AI-Powered Spending Insights

The AI Powered Spending Insights feature scans all of your transactions and mimics the feedback of what a financial coach might say.
For this I used:

- First, I used **Theil–Sen Regression** to detect long-term category-level spending trends in a way that's robust to outliers. For example, if dining out increases steadily month-over-month, it flags that trend.
- Second, I applied **Isolation Forest**, an unsupervised anomaly detection method, to surface unusually high expenses in specific categories, even if they aren't part of a broader trend like a sudden spike in shopping or travel spend.
- Finally, I layered in **domain-specific heuristics**. These trigger contextual nudges for categories like coffee or groceries, offering actionable advice. For example, frequent coffee runs might prompt a tip like 'brew at home 3x a week.'

These insights are then sorted by their severity and color coded Info, High, Medium, and Low = positive reinforcer.

2. Personalized Goal Forecasting

This feature forecasts whether you're on track to meet a custom savings goal for example, saving $1,000 in the next 3 months based on your current income and expenses.

Here, I used:

- **per-category Linear Regression** models to forecast future monthly spending for example, projecting how your grocery or shopping spend will evolve.
- With that forecast and your provided income, the app computes whether you'll have enough monthly leftover savings to reach your target in time.
- To visualize this, I included a **goal progress chart** it shows your predicted total monthly spend stacked against the maximum spend allowed to stay on track. If you're projected to overspend, the app suggests **smart spending cuts**, capped at 20% per category, to help close the gap.

This feature balances predictive modeling with real-world constraints

3. Subscriptions & Gray Charges Detector

This goes beyond simply listing recurring payments.

I implemented a custom heuristic engine based on unsupervised learning principles specifically clustering and anomaly detection but without model training. SO to detect recurring subscriptions, I group transactions by merchant and compute the coefficient of variation (CV) of payment amounts, along with the average time between charges. A low CV and consistent cadence suggest a subscription. For example, if we think about Netflix : if a user is charged around $15 every 30 days, that pattern is flagged as a recurring subscription by this heuristic.

Similarly, free trials are detected by identifying an initial $0 or $1 transaction followed by a spike the next month a common tactic used by services to auto-convert to paid plans. Other examples of ambiguous gray charges is international phone fees or even apple storage fees that are small repeated microtransactions and these are are flagged by looking for low-value, high-frequency patterns that users themselves often overlook or forget.

To avoid false positives, I excluded essentials like rent and utilities and used natural language matching for fee-like keywords.

Tech Stack

- **FastAPI** on the backend
- **Jinja2 + Chart.js** for the frontend
- **scikit-learn** for ML models (LinearRegression, Theil–Sen, IsolationForest)
- **NumPy & pandas** for all data transformations
- No external APIs everything runs locally for quick iteration

---

Trade-offs

One key trade-off was between simplicity and realism. For speed, I used in-memory data but realistically to make it scalable and usable I would utilize a database like Postgres to store all user data.

Also the AI logic is currently rule-based and statistical, and for future vision if allotted more time, I would next work on integration with LLMs for deeper personalization. For example, I would finetune an LLM model and includes different features such as bag of words to emphasize or downplay certain categories. For example, when creating insights, I would downplay rent, as it is a fixed cost and should never be an insight to cut rent as that doesn't make any sense.

Key Learnings & Challenges

One key design decision was to **exclude rent from both the AI-generated insights and personalized goal forecasting**. While rent is typically a user's largest monthly expense, it's also one of the least flexible. Initially I included rent in the features and saw skewed results that distracted us from actionable spending categories like dining, shopping, or entertainment since it was always the category to work on. By excluding rent, by hard-coding it out, the app ensures that insights focus on **areas where users have the agency to adjust their behavior**, making the feedback more relevant and actionable.

I learned how to:

- Blend frontend UX with backend intelligence,
- Use real-world financial reasoning to guide product design,
- And design for future extensibility like letting insights come from LLMs or user chat.