AI BOOST Team Take-Home Exercise

Overview

You'll build a small, end-to-end prototype—similar to the rapid AI experiments we ship on the AI Boost Team. The challenge blends **product thinking with applied AI + execution**. The scope is intentionally tight but open enough for creativity.

- **Timebox:** Aim for ~2 hours of focused work for the core requirements (stretch goals optional).
- **We're evaluating:** Signal over polish—your product reasoning, prompt/AI design choices, data plumbing, and how you make trade-offs.
- **Stack:** Any language/stack you're comfortable with (Python/TypeScript are common). Open-source libraries and hosted LLM APIs are fine—please cite them.

What we'll be looking for
We're less concerned with polish and more interested in how you think and build. In particular, we'll look at:

- **Product thinking** — Are the insights and recommendations concrete and actionable?
- **LLM design** — How you framed prompts and structured AI outputs.
- **Data handling** — How you worked with and joined the data to support your reasoning.
- **Communication** — How clearly you explain your approach, trade-offs, and next steps.

The Challenge

Customer Feedback Incident Detection & Action Recommender

Goal: Detect recurring customer issues from support feedback and recommend concrete actions for the business.

You'll build a small **AI-powered prototype** that:

1. **Surfaces recurring themes/issues from customer feedback** (e.g., "Chime linking errors", "approval criteria complaints", "too many repayment reminders").

2. **Shows how the frequency of these themes changes over time** (weekly or monthly).

3. **Connects feedback to product usage** (e.g., “High-usage customers reporting Chime linking errors” or “Frequent borrowers raising approval criteria concerns”).

4. **Generates 3–5 prioritized, actionable recommendations** that a PM or Ops lead could take next week.

Data You’ll Use

We’ll provide two static files.

1. feedback.jsonl — Zendesk-style tickets

- `id`: string
- `customer_id`: string
- `created_at`: string (ISO8601, UTC)
- `message`: string (free text)

Note: Sentiment and themes are not provided—you’ll need to infer them using an LLM.

2. product_usage.csv — Product snapshot data

- `customer_id`: string
- `last_login`: datetime (ISO8601, UTC)
- `feature_usage_count`: integer (last 30d)
- `total_spend`: float (last 30d)
- `subscription_tier`: enum("free","plus","pro")
- `advance_attempts_30d`: integer
- `advance_approvals_30d`: integer

Core Requirements (MVP)

- **Data ingestion:** Load both data sources (JSONL + CSV).
- **Theme mining:**
    - Use an LLM to assign a concise theme (≤5 words) to each feedback message.
    - Also extract sentiment.
    - Consolidate into 3–5 recurring themes.

- **Theme trends:** Plot how frequently these themes occur over time (weekly or monthly). Include a short summary.
- **Anomaly detection:** Highlight unusual spikes in negative sentiment for a theme.
- **Link feedback ↔ usage:** Join themes with product usage (e.g., which tiers, spend cohorts, or approval ratios are most affected).
- **Actionable recommendations:** Use an LLM to generate 3–5 prioritized, concrete recommendations with rationale based on theme trends and usage analysis. Where possible, tie each recommendation to a business outcome (e.g., reducing churn, improving approval rates, increasing engagement).

Stretch Goals (Optional)

- **Segmentation:** Contrast issues by subscription_tier or total_spend.
- **Interaction:** Add a lightweight dashboard (Streamlit/Jupyter/Gradio/Next.js) with filters (date range, tier, theme).
- **Explainability:** Show which example comments map to each theme.
- **Hosting:** Deploy a hosted demo (Streamlit Cloud, Render, Netlify, etc.).

Deliverables

- **Code** in a public repo or zipped package.
- **README** including:
    - Problem framing & assumptions
    - Setup instructions
    - How to run (ingest, process, serve UI if any)
    - Decisions & trade-offs
    - What you'd do next with more time
- **Outputs:**
    - One trend chart (PNG or in-app)
    - A themes table with counts + sample comments
    - A joined view (at least one theme × usage metric)
    - 3–5 actionable recommendations (LLM-generated) with rationale
    - A brief explanation of your LLM prompts and approach