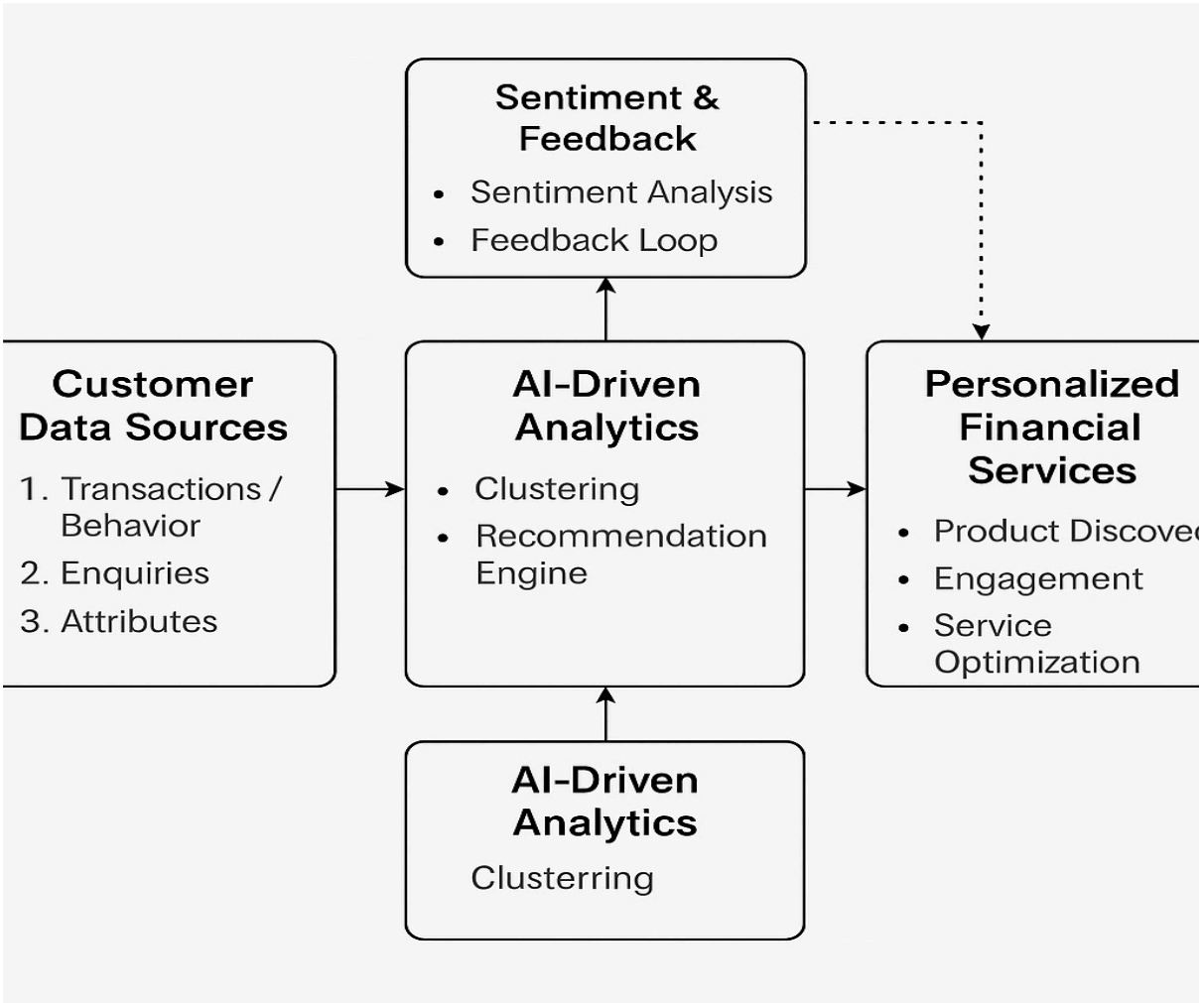
Personalized Financial Recommendation System - Documentation

Overview

This system delivers hyper-personalized financial product recommendations, engagement strategies, product discovery suggestions, and service optimization guidance based on customer clustering and user feedback. It combines rule-based insights with AI (via Hugging Face Inference API) and supports real-time interaction through a Flask API and a React frontend.

Architecture

Below is the system architecture of the proposed solution -



System Components

1. Data Layer

- **customer_data.csv**: Core profile info (age, income, spending habit, etc.)
- **financial_behavior_data.csv**: Credit/loan behavior
- **enquiries_data.csv**: Product enquiry history
- **clustered_customer_data.csv**: Combined and clustered customer data

2. Backend API (Flask)

- **Model training**: RandomForest classifier trained on clustered data
- **Recommendations**:
    - Rule-based mappings for each customer segment (cluster)
    - Rephrased using Hugging Face model (`flan-t5-large`) for natural language output
- **Feedback system**:
    - In-memory feedback store
    - Feedback influences future recommendations

API Endpoints

Method	Route	Description
`GET`	`/recommend?customer_id=<id>`	Fetch personalized recommendation for a customer
`POST`	`/feedback`	Submit feedback to improve future recommendations

Sample `/feedback` payload:

```
{
  "customer_id": 5,
  "feedback": ["fixed deposit", "too conservative"]
}
```

Recommendation Engine

Rule-Based Mapping

Each cluster maps to a set of tags (e.g., "stable income", "digital savvy"), which in turn map to recommendations like:

- "Suggest a recurring deposit"
- "Recommend a mobile-only savings account"

AI-Powered Rephrasing

Recommendations are passed to Hugging Face's `flan-t5-large` via API and rephrased into customer-friendly text.

Feedback Integration

User feedback is stored in memory and:

- Filters out disliked tags
- Optionally adds new tags (e.g., "travel", "tech")
- Regenerates recommendations

Frontend (React)

- User enters `Customer ID` and gets recommendations
- Displays cluster ID, personalized suggestions, and 3 strategic insights:
    - Engagement Strategy
    - Product Discovery
    - Service Optimization
- Allows submitting feedback (e.g., "too risky", "add travel focus")

Sample UI Features

- Axios integration for API calls
- Real-time updates on feedback
- Clear, clean UX with minimal state handling

Future Enhancements

- Feedback persistence using database
- Multi-language or tone control
- Export as PDF or email
- Frontend for feedback analytics
- Deploy Flask backend on Render and frontend on Vercel

Technologies Used

- **Python, Flask, scikit-learn, pandas** — data pipeline & model
- **React, Axios** — frontend
- **Hugging Face Inference API** — LLM rephrasing
