AI-Based Customer Journey Analyzer

Course No: AIMLCZG628T

Course Title: Dissertation

Dissertation Done by:

Student Name: Sumithra Jayaraman

BITS ID: 2023AC05275

Degree Program: M.Tech. - Artificial Intelligence & Machine Learning

Research Area: Customer Analytics, Artificial Intelligence

Dissertation / Project Work carried out at:

Verizon, Chennai

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
VIDYA VIHAR, PILANI, RAJASTHAN - 333031.

December, 2025

Contents

1. Abstract Summary

1.1 Project Overview

The **AI-Based Customer Journey Analyzer** is a modular, full-stack application designed to analyze complex customer interaction logs across multiple channels. It uses unsupervised machine learning (K-Means Clustering) and Natural Language Processing (NLP) to segment customers based on their behavior patterns and provide actionable insights into their journey.

1.2 Problem Statement

Organizations often struggle to synthesize vast amounts of heterogeneous interaction data (Web, App, Call Center, etc.) into a coherent understanding of customer loyalty and friction. Manual analysis is slow, and simple rule-based systems fail to capture nuanced behavioral shifts, leading to missed opportunities for proactive retention and service improvement.

1.3 Key Objective

To provide an automated, AI-driven platform that:

- Segment customers into meaningful loyalty tiers using objective behavioral features.
- Quantifies customer friction and interaction quality.
- Generates human-readable summaries of complex customer journeys using LLMs.
- Offers an interactive dashboard for both individual customer lookups and high-level system monitoring.

---

2. Technical Architecture

2.1 Project Workspace Architecture

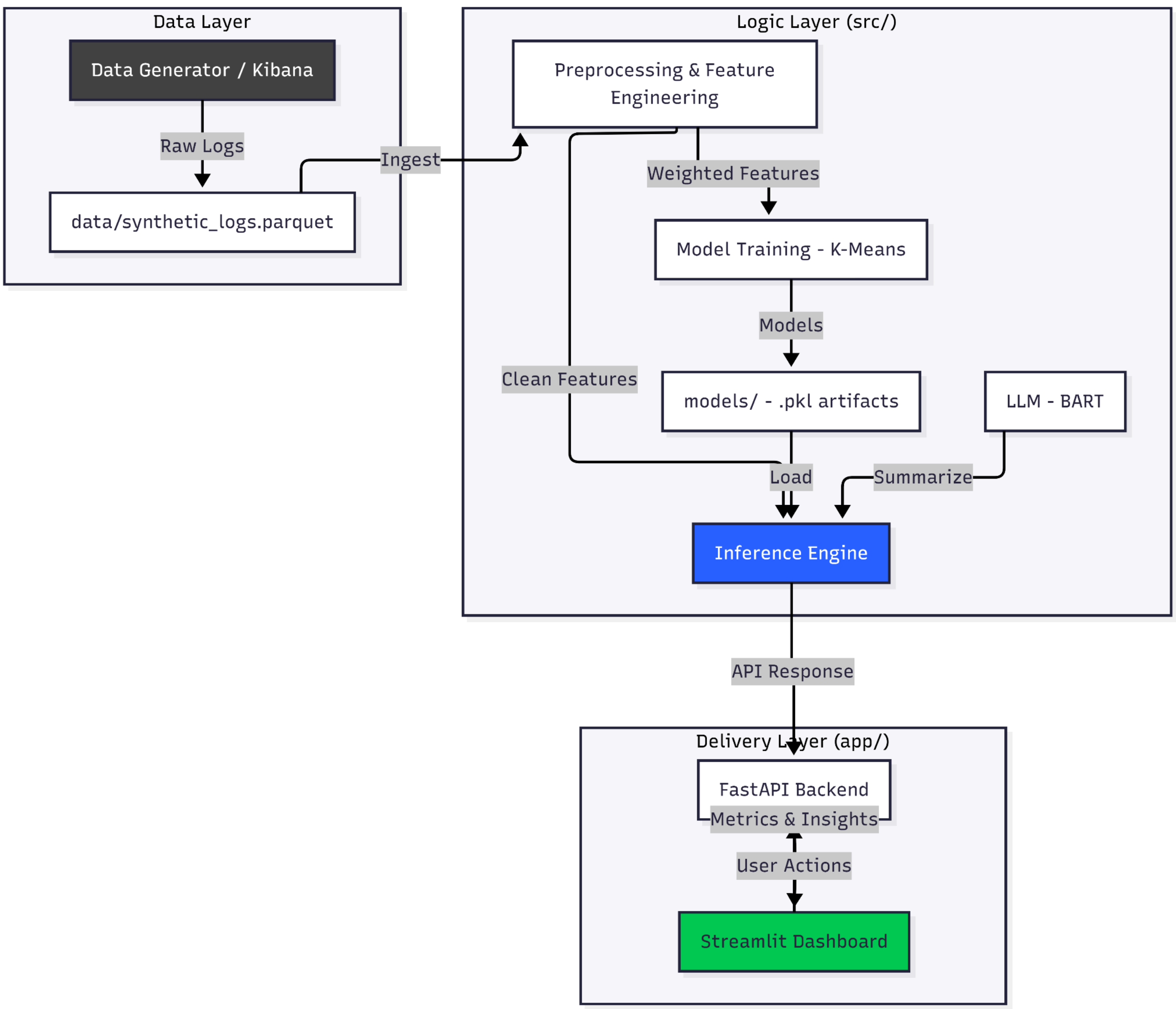
The system is organized into a modular, decoupled structure to ensure scalability and ease of maintenance. Core logic is centralized within the `src/` directory to be shared across both the backend API and standalone training scripts.

- `app/` **— Delivery Layer**
  - `main.py`: FastAPI backend managing asynchronous endpoints for data processing and model lookups.
  - `dashboard.py`: Streamlit-based frontend providing interactive journey mapping and cohort visualizations.
- `src/` **— Logic & Analytics Layer**
  - `data_generator.py`: Logic for simulating complex customer interaction logs across multiple channels.
  - `preprocessing.py`: Data cleaning and feature engineering, including Time-Decay

Weighted Aggregation.

  - `train.py`: Implementation of the K-Means clustering algorithm and model evaluation frameworks.
  - `inference.py`: Core engine for journey prediction, AI-driven summarization (BART), and explainability drivers.
- **Support & Artifacts**
  - `data/`: Persistent storage for raw and processed synthetic logs in `.parquet` format for high-performance querying.
  - `models/`: Repository for serialized model artifacts (`.pkl` files).
  - `metrics.txt`: Log of the latest evaluation scores, such as Silhouette and Davies-Bouldin indices.
  - `requirements.txt`: Comprehensive list of Python dependencies including `Transformers`, `FastAPI`, and `Scikit-learn`.

2.2 Functional Blocks



2.3 Design Considerations

- **Modularity & Scalability**
  - **Decoupled Architecture:** Each layer (Data, Preprocessing, Training, Inference, UI) is decoupled. This allows for replacing the synthetic data generator with a real Kibana feed or upgrading the K-Means model to a Deep Learning model without rewriting the entire system.
  - **Dependency Management:** All core logic resides in `src/`, shared by both the FastAPI backend and standalone training scripts.
- **Data Integrity & Privacy**
  - **Parquet Storage:** Parquet for data storage due to its column-oriented nature, which is significantly faster for analytical queries and feature aggregation compared to CSV or JSON.
  - **Non-PII (Personally Identifiable Information):** The system design focuses on behavioral signals (‘channel_id’, ‘flow_name’) rather than sensitive user details, facilitating easier GDPR/security compliance.
- **Performance Optimization**
  - **Asynchronous Processing:** Training and data generation are handled via FastAPI `BackgroundTasks` to prevent the UI from freezing during long-running operations.
  - **Lazy Loading:** Large LLM models (BART) are loaded only when first requested to ensure fast initial startup of the API.
- **Explainability & User Centricity**
  - **Heuristic-Driven Mapping:** Instead of showing raw cluster IDs (0, 1, 2), the system uses a heuristic to map clusters to human-readable loyalty levels (High, Medium, Low).
  - **Effective Visualization:** Used Altair for the journey map visualization.

2.4 Process & Technology Used

2.4.1 Data Generation & Ingestion

- **Technology:** ‘Pandas’, ‘Faker’
- **Process:** Generates synthetic customer logs simulating interaction across App, Web, Call Center, etc.
- **Signals:** Captures ‘start_time’, ‘end_time’, ‘channel_id’, ‘flow_name’, and ‘log_messages’.

2.4.2 Preprocessing & Feature Engineering

- **Technology:** ‘Pandas’, ‘NumPy’, Regex.
- **Algorithm:** Time-Decay Weighted Aggregation.
  - **Interaction weight** = 1 / (days_since_interaction + 1).
  - Recent interactions carry more weight than older ones.
- **Key Features:** ‘friction_rate’, ‘avg_duration’, ‘last_friction_days’, ‘weighted_digital_count’, ‘weighted_assisted_count’.

2.4.3 Model Training & Evaluation

- **Algorithm**: **K-Means Clustering** (Unsupervised).
- **Process**: Clusters customers into 3 behavioral segments.
- **Loyalty Heuristic**: Dynamically maps clusters to "High", "Medium", and "Low" loyalty levels based on digital adoption vs. assisted channel reliance vs. friction rate.
- **Evaluation**: **Silhouette Score** (cohesion) and **Davies-Bouldin Index** (separation).

2.4.4 Inference & NLP

- **Technology**: 'Transformers' (BART-large-cnn), 'Sklearn'.
- **AI Summary:** Uses a pre-trained BART model to summarize the journey text into concise insights.
- **Explanation**: Calculates distance to cluster centroids to identify the "Top Drivers" (e.g., high friction) for a customer's loyalty assignment.

2.4.5 Delivery (API & Dashboard)

- **Backend**: 'FastAPI' provides asynchronous endpoints for data generation, training, and customer lookup.
- **Frontend**: 'Streamlit' with 'Altair' for interactive journey mapping and cohort visualization.

---

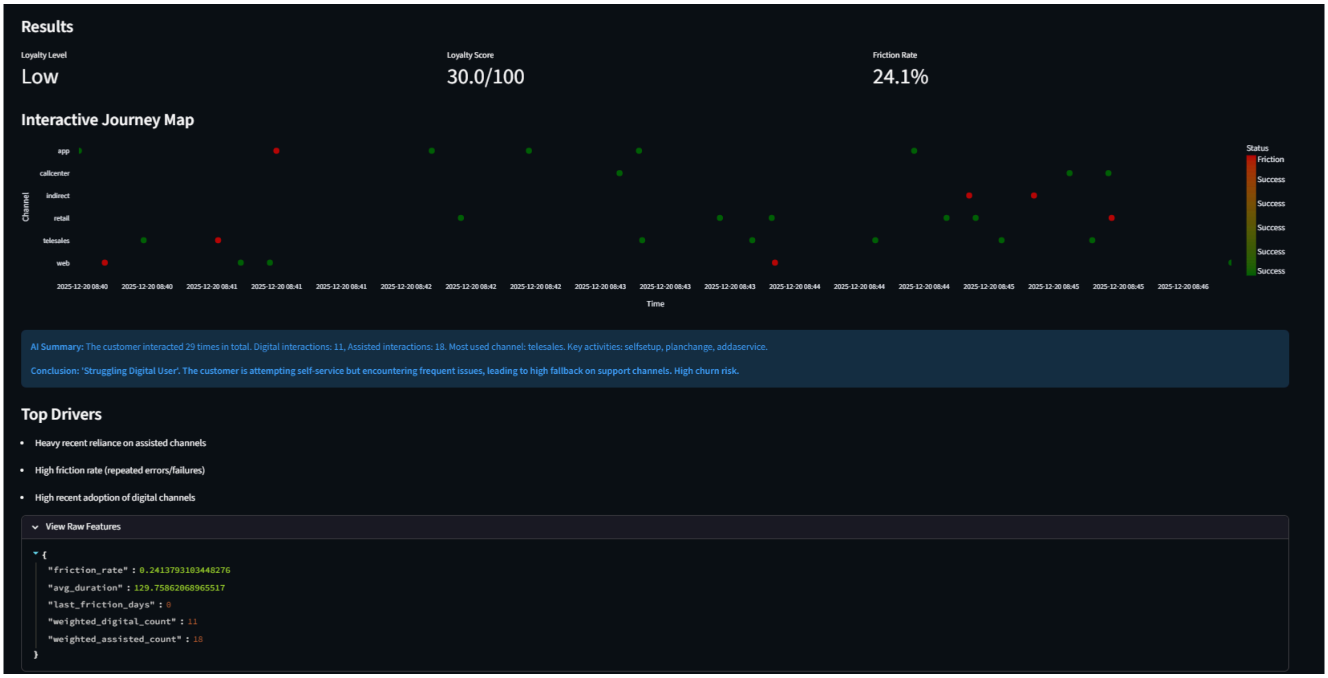
3. Work Accomplished (Completed)

✓ **Modularization**: Created a production-ready, multi-layered Python package structure.
✓ **Data Integration:** Created synthetic data.
✓ **Feature engineering**: Identify key features that would help identify the loyalty score & segmentation. Added sophisticated feature engineering that prioritizes recent customer behavior (**Time-Decay Logic)**
✓ **Model training:** Using K-Means unsupervised algorithm.
✓ **Evaluation Framework**: Established internal validation for unsupervised clustering (Silhouette & DB Index).
✓ **Integrated NLP**: Implemented AI-powered journey summarization using state-of-the-art LLMs (BART).
✓ **Dashboard:** Built an interactive UI that links directly to the AI backend for real-time analysis.

3.1 Current Status

✓ **Active & Functional:** The system is fully operational. Data can be generated, models trained, and journeys analyzed end-to-end.
✓ **Performance:** Clustering achieves a Silhouette Score of ~0.3, indicating stable and meaningful customer segmentation.

Sample user interface screen:



---

4. Future Work

- **Kibana/Elasticsearch Integration:** Replace synthetic data generation with a live data pipeline using the elasticsearch Python client to pull real-time interaction logs directly from Kibana/Elasticsearch indices (This cannot be tested as we will not be able to access the logs outside of the organization - but the implementation will be made ready).
- **Scheduled model training**
- **Explainable AI Implementation:** Evaluate interaction of libraries like **SHAP** or **LIME** to provide mathematical depth to the "Top Drivers" analysis.
- **Downloadable Reporting:** Feature to export individual customer analysis or cluster profiles as PDF/Excel that can be emailed to the operations team for further analysis & actions.