# BA882 Project Final Deliverable

## Important Links

**Team 05 GitHub Repository:** GitHub-Team05
**Streamlit Application:** Streamlite-Team05

## Pipeline Improvement

In the final phase of our project, we significantly enhanced our data pipeline and machine learning workflow. Here's a detailed overview of the improvements:

### Extending the Pipeline

We expanded our original ETL data pipeline to include downloading the final Netflix API data to Google Cloud Storage. This step is crucial for preparing the data for model training and ensures a seamless flow from data extraction to model input.

### Machine Learning Integration

Building upon our previous KNN model for recommendations, we integrated the machine learning pipeline directly into our main workflow. This integration includes:

1. Model Training: We created two separate Cloud Functions responsible for training models for movies and TV shows respectively.
2. Metrics Archiving: New schemas were created in MotherDuck to store model metrics. The Cloud Functions now archive these metrics into dedicated tables, allowing for performance tracking over time.
3. Prefect Flow Integration: The model training process is now incorporated into the Prefect flow, ensuring automated and scheduled execution.

## Prefect Orchestration

Bringing together all the flows from building the data pipeline to integrating machine learning we connected the pipelines and orchestrated it altogether on Prefect Cloud.

### Automated Daily Deployment

Our final pipeline consists of three main components that are automatically deployed in sequence every day:

1. EtLT data pipeline
2. Training data extraction
3. Model training (KNN)

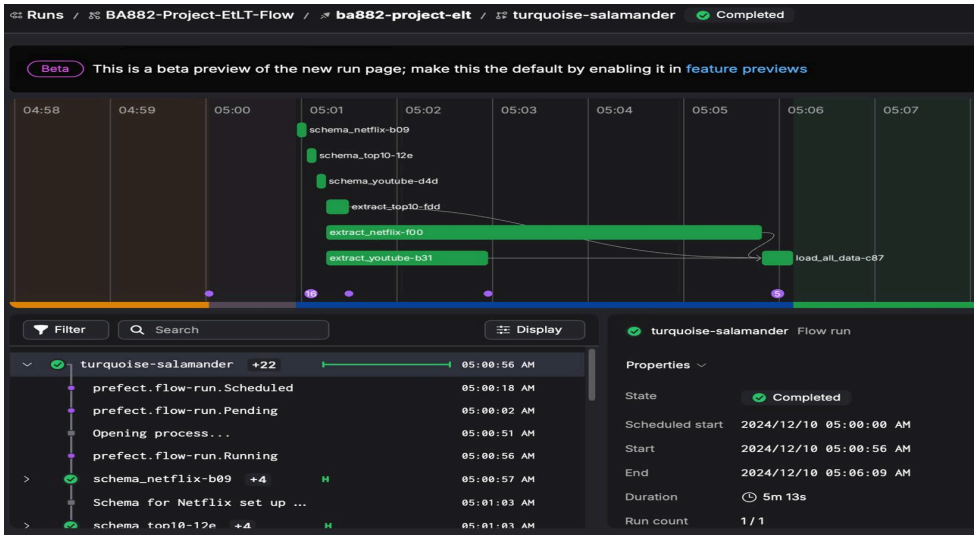
This automated sequence ensures that our data and models are consistently up-to-date, reflecting the latest information from our data sources.

### Pipeline Visualization

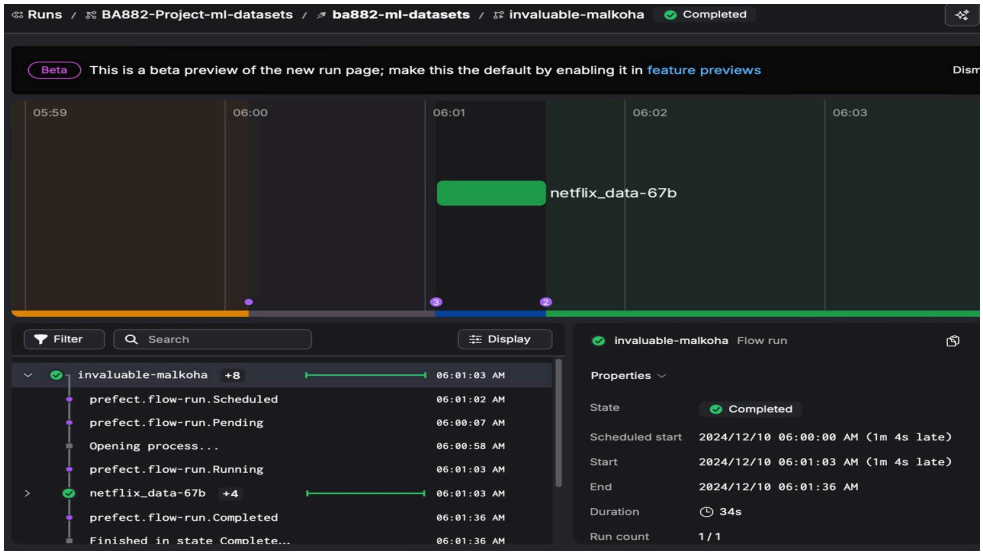
The new deployment on Prefect Cloud provides a clear visualization of our improved pipeline, showcasing the interconnected flows and their scheduled execution

Below are screenshots of the overall pipeline deployment on Prefect Cloud:

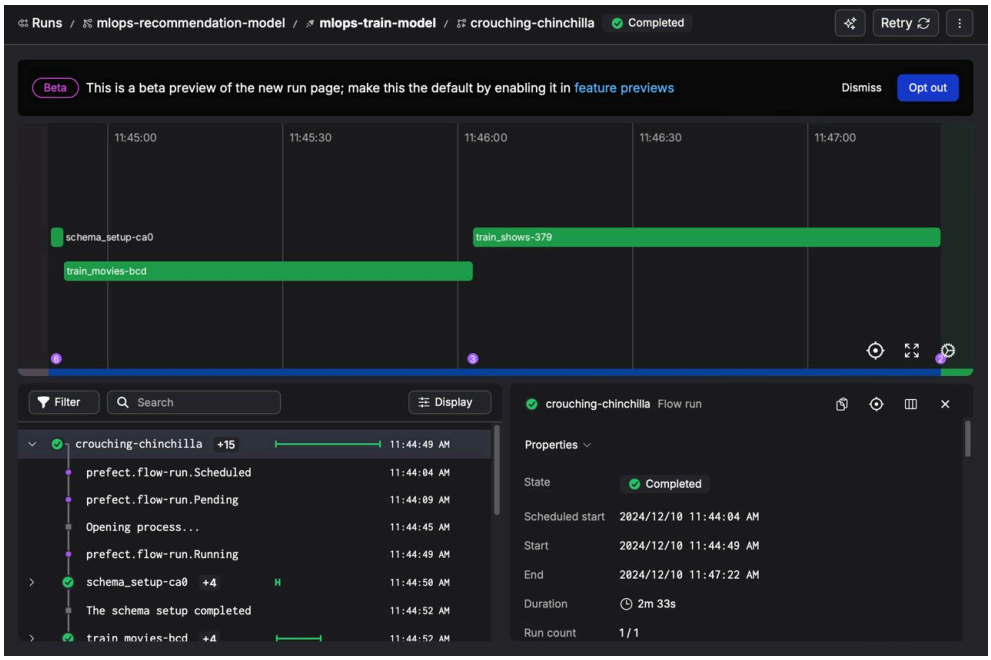
1. **EtLT flow**




2. **Training data extraction**




3. **Model training flow**




## Streamlit Application

Our Streamlit application offers a comprehensive Netflix recommendation system that combines traditional machine learning with advanced language models. Users can search for specific titles or select from a list, triggering our KNN-based recommendation engine deployed on Google Cloud Run. The system initially provides recommendations using models trained on movie and TV show data stored in DuckDB via MotherDuck. For more personalized suggestions, users can provide feedback, which is processed by the Gemini 1.5 Pro LLM. This AI-powered feature considers the user's input, initial recommendations, and additional web-sourced information to generate five tailored movie or show suggestions, complete with titles and brief descriptions. The application ensures secure handling of sensitive data through Google Cloud Secret Manager and maintains a clear separation between the frontend and backend recommendation logic.

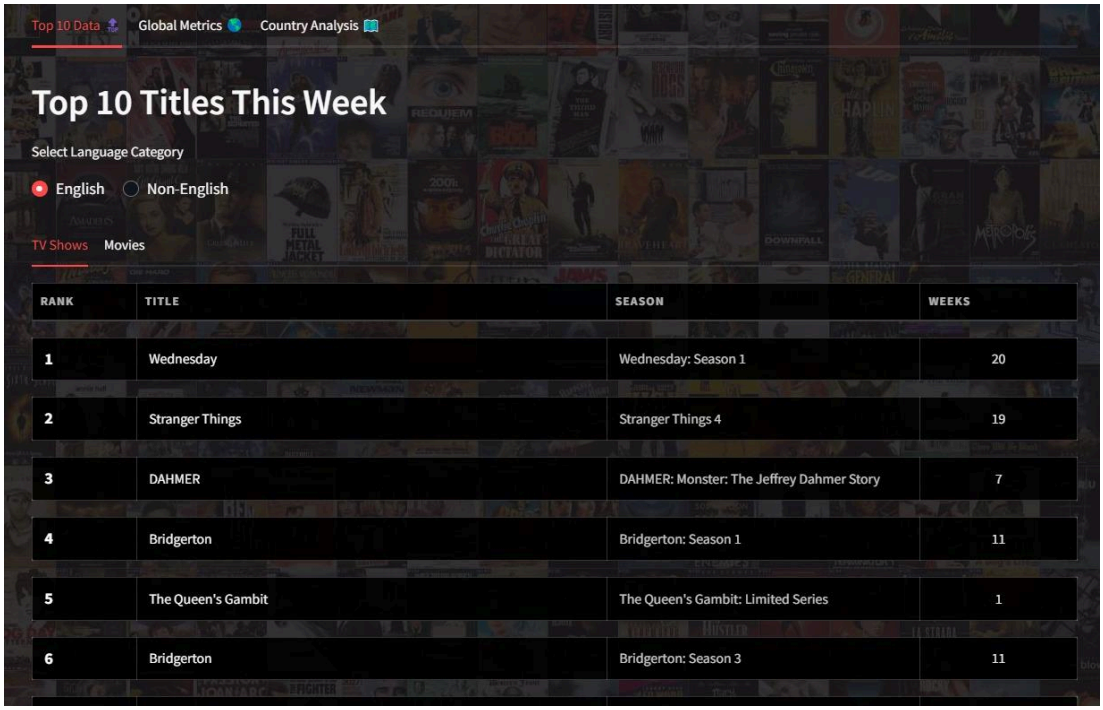
### Netflix Data Analytics Page

We've added a dedicated page for Netflix data analytics, which includes:

- Visualizations of key metrics from our Netflix dataset
- Trends in viewership, genre popularity, and content distribution
- Interactive charts allowing users to explore the data in depth

This page ties back to our Phase 1 work, showcasing the insights gained from our initial data collection and analysis.

Below is a sample screenshot of the Netflix Data Analytics page:



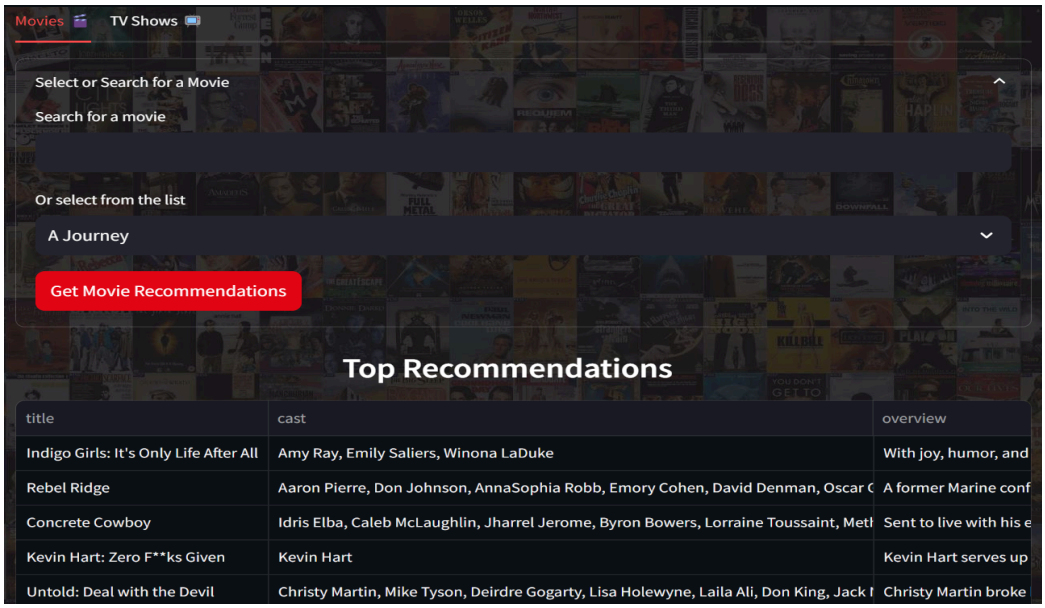

### Recommendation System Page

The recommendation system page has been enhanced with the following features:

1. **Title Selection:** Users can select a movie or TV show from a dropdown list or use a search feature to find specific titles.
2. **KNN Model Recommendations:** Upon selecting a title and clicking "Get Recommendations," the application deploys the corresponding KNN model (movie or TV show) to generate initial recommendations.
3. **LLM Integration:** We've introduced a large language model (LLM) to provide more personalized recommendations:
    - Users can expand the "Want to get customized recommendations?" section
    - They can enter feedback or specific preferences in a text box
    - Clicking "Get New Recommendations" triggers the LLM process.
4. **LLM Processing**:
    - The system combines the initially selected title, preliminary recommendations, full database information, and user feedback into a prompt
    - This prompt is submitted to the "gemini-1.5-pro-001" model
    - The LLM can also search the web for additional context to enrich recommendations.
5. **Enhanced Recommendations Display**:
    - The application displays five new LLM-generated recommendations

- Each recommendation includes the title and a brief description.

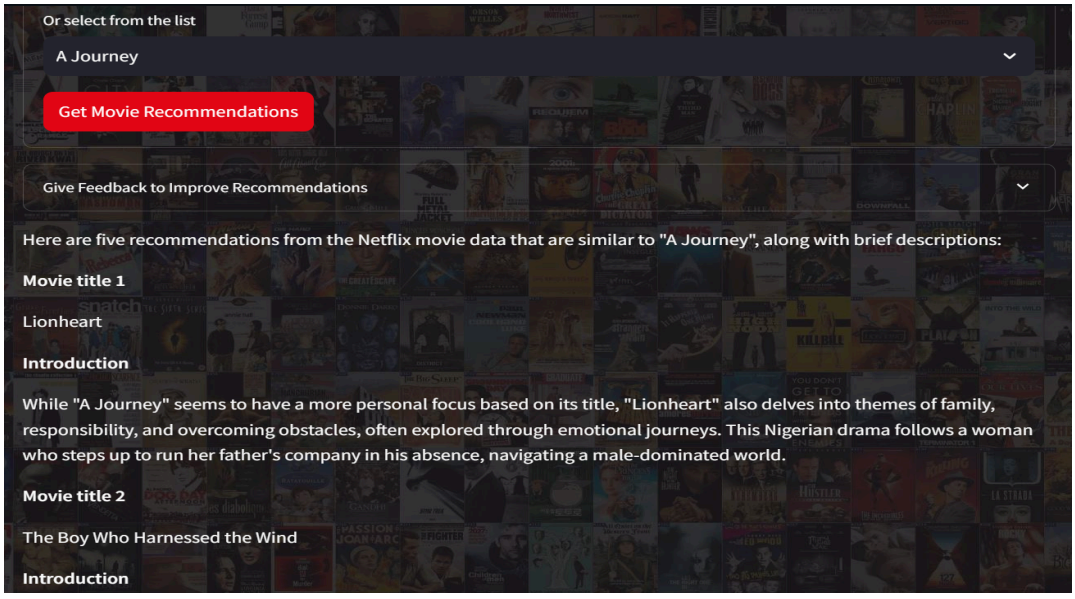
Below are screenshots of the KNN Recommendation implementation:




## User Interaction and Feedback Loop

- Users can provide feedback on recommendations, which is then used to refine future suggestions
- This feedback loop helps improve the recommendation quality over time and provides a more personalized experience

Below is a screenshot of the User Feedback implementation:



Team 05 - Final Deliverable

## Technical Implementation

- We use DuckDB connected via MotherDuck for efficient data querying and retrieval
- Google Cloud Secret Manager is implemented for secure handling of sensitive information like database tokens
- The KNN algorithm is deployed on Google Cloud Run functions, separating the recommendation logic from the frontend
- API calls are used to interact with Cloud Run functions, ensuring a modular and scalable architecture.

## Visualization Integration

- The application incorporates visualizations from our earlier phases, providing a comprehensive view of Netflix content trends alongside personalized recommendations.

Here is the link to our Streamlit interface: Streamlit-05

For the best experience, please go to the top-right corner of the interface, open **Settings**, and set the **app theme, colors, and fonts** to **Dark**.