Business Transformation Recommendation Report – Movie recommendation system

Business problem: How can we recommend movies to users using LLM?

The basic problem is to increase user satisfaction through effective recommendations. In our project we introduced such solutions as:

* • understanding natural language by using LlaMa
* understanding the description of movie to develop semantic interpretation of the content
* understanding new users who haven't saved their favorites

Structured problem:

1. Define the LLM deadlock, identify where it might struggle to interact with user.
2. Gather and prepare data, ensure data quality by cleaning and finding dataset is relevant to the use case.
3. Learn the LLM architecture, research various models and choose one which aligns with requirements.
4. Implement and integrate LLM with server, set up APIs and enable communication between frontend, backend and LLM.
5. Observe performance and improve the model, use evaluation tools to iterate fine-tuning and prompt engineering.
6. Deploy the system and continuous monitoring, automate monitoring dashboards and alerts to track key performance.

Collect relevant information:

* Large movie datasets: Kaggle, API or scrapping
* Natural Language User Queries
* User and feedback data: save user data and feedback in database

Considering Options

* using LLM to extract semantic features from movie text
* using LLM to understand users' natural language (movie descriptions)
* using LLM to generate movie recommendations based on users' feedback
* using an LLM model that balances performance

Making Recommendation

* Use LLM to enable users to search for their movie preferences in natural language
* Use A/B testing to improve the system based on user feedback
* Prioritize data quality by cleaning and preprocessing
* Focus on user experience, create intuitive design
* Include an explanation of how LLM recommendation works and ensure unbiased recommendations
* Combining LLM and traditional collaborative or content-based filtering models to create a powerful recommendation system
* Evaluate the performance of recommended movies and refine the prompt