

Charlottecool / project-5

Search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master

project-5 /

Charlottecool

Create README.md

7500d6a · 5 minutes ago

| Name | Name | Last commit date |
|--|--|------------------|
| <div></div> .gitignore | Add director information to the dataset and fix o... | last week |
| <div></div> Neural Network and NLP-Based Movie ... | last polishing is done | 26 minutes ago |
| <div></div> Neural Network and NLP-Based Movie ... | add presentation to the folder | 7 minutes ago |
| <div></div> README.md | Create README.md | 5 minutes ago |

README.md

Neural Network and NLP-Based Movie Recommendation System

Ever wondered how streaming platforms always seem to know the perfect movie for your mood? What if you could uncover the secrets behind those recommendations? Let's get started!

1. Overview and Business Understanding

In today's entertainment landscape, movie recommendation systems are a vital tool for streaming platforms to enhance user engagement and satisfaction. The goal of this project is to develop a personalized movie recommendation system that can effectively predict user preferences using collaborative filtering, content-based models, and advanced neural network techniques such as Transformers. By analyzing user ratings, movie genres, and other attributes, the system provides highly tailored recommendations.

From a business perspective, a robust recommendation system can significantly boost user retention and increase viewing time, ultimately leading to higher subscription rates and revenue growth. Additionally, accurate recommendations improve the overall user experience by reducing the time spent searching for content, making users more likely to explore new movies based on personalized suggestions. This system not only serves individual users but also helps platforms optimize their content offerings.

2. The Datasets

- [MovieLens 20M Dataset](#)
- [US Film Industry Top Movies & Directors](#)

3. Project Workflow: From SVD Collaborative Filtering to Transformer-based Recommendations

The project began with the development of a traditional collaborative filtering system using Singular Value Decomposition (SVD). In this step, I created a user-movie rating matrix and used SVD to predict missing ratings based on user preferences and movie features. The model performed well, but it had limitations, especially in handling cold-start problems (i.e., new users or movies with little data) and lacked the ability to account for more complex content features.

To address this, the next step focused on a content-based recommendation approach using cosine similarity. I shifted from just user-movie interactions to analyzing movie genres and tags, which allowed me to recommend movies based on how similar their content was. The cosine similarity function compared movie vectors based on these features, producing recommendations that relied more on the intrinsic properties of the movies themselves.

Following this, I introduced a neural network model to improve the recommendation system's performance. By using a deeper learning method, I aimed to capture non-linear relationships between user preferences and movie features, further refining recommendations. While this model offered better performance than the basic SVD, it still showed some limitations in diversity and accuracy.

In the final stage, I implemented a transformer-based model to enhance the recommendation system's capability. Using pre-trained embeddings from SentenceTransformer, I generated vector representations for movie genres and tags, providing richer semantic content. The transformer model enabled more context-aware recommendations, resulting in a system that better understood user preferences by leveraging more complex content and relational data.

Overall, this multi-phase development process led to a robust recommendation system, combining collaborative filtering, content-based analysis, and advanced deep learning techniques. The final product was able to provide accurate, diverse, and contextually relevant recommendations.

4. Deployment

I've deployed the recommendation system as a simple and user-friendly [web app](#).

For more information

Check out the full [Jupyter notebook](#) and [the presentation](#).