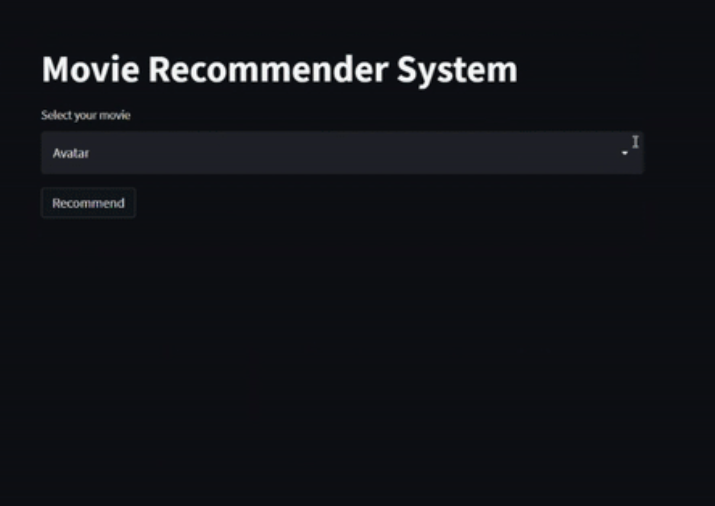
**MOVIE RECOMMENDER SYSTEM**

**Abstract:** Digital information, content and the number of users is growing exponentially. This rampant growth has created an issue of information overload, making it difficult to instantly access the required data available in cyberspace. To provide the solution, our Recommender system plays a crucial role for websites dedicated to movies, e-commerce, music and more, as it assists users in discovering items they require based on their experience and offers them relevant information.

As part of the project, while developing a movie recommender system, numerous factors were considered, including genre, cast, overview, keywords, the movie's director, and many other variables.

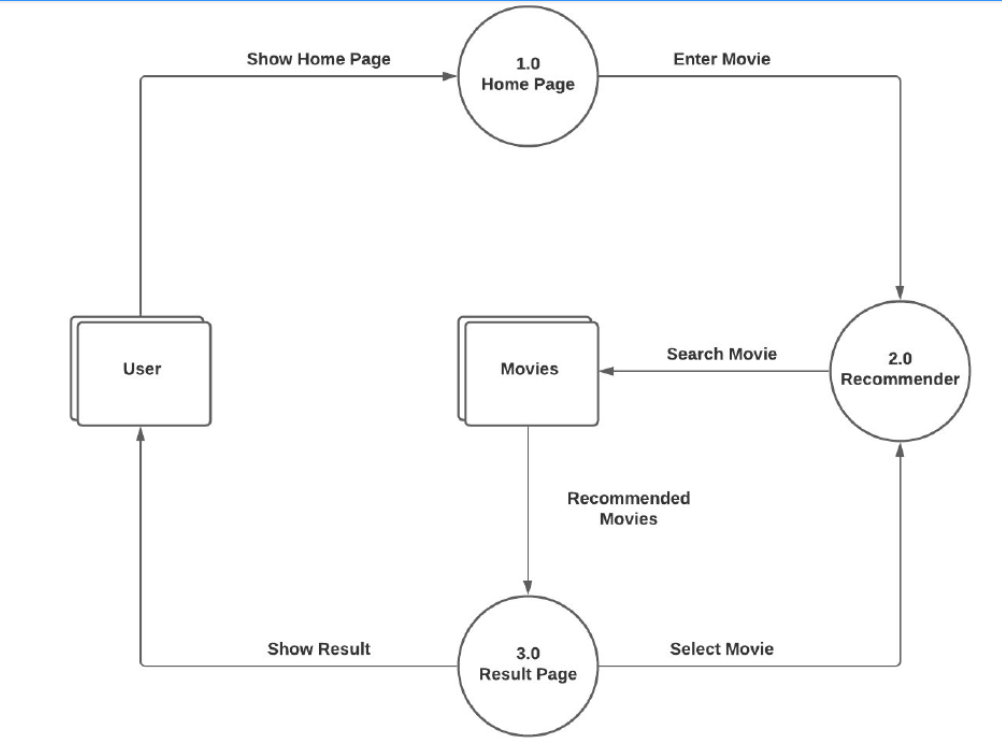
**Overview:** This recommender system provides the user with an option to select a movie which he/she likes and then recommends five movies the user may want to watch based on selection. The movie which is recommended to the user is selected by checking the movie name, genre, actors, and directors of each film.

 A screen shot of a movie

Description automatically generated

**Working**: Gathered a dataset which consisted of 5,000+ movies and used a TMDB API to access data which included information related to cast, movie genres, overview, language, release date, name of director and more. **Python** served as the programming language for both server-side and front-end development. The server-side programming leveraged libraries such as **Pandas** and **Numpy**, while the creation of the local host for the recommender system was accomplished using the **Streamlit** library.

Subsequently a list was created which concatenated lists of movie overview, cast, genre, keywords, and director, so it becomes easier to find similarities between different movies. Next **Vectorization** was performed where each concatenated list of a movie was treated as a vector and then cosine similarity was performed to find the five most similar vectors. Another library **Pickle** was used for transferring the title of the movies as a dictionary from the dataset to the front-end notebook file. Finally, **Heroku** was used to deploy the recommender system as a website.



**Conclusion**: Recommendation systems have evolved into a vital aspect of our daily lives. Given the vast number of movies already released and releasing globally each year, individuals frequently miss out on exceptional works of art because they lack accurate recommendations. Therefore, deploying a Machine Learning based recommender system is crucial for obtaining the most accurate recommendation.

Likewise, these systems can be further enhanced by implementing neural network embedding techniques to elevate the quality of recommendations and render them for more personalization.

Author: Rohit Lahori