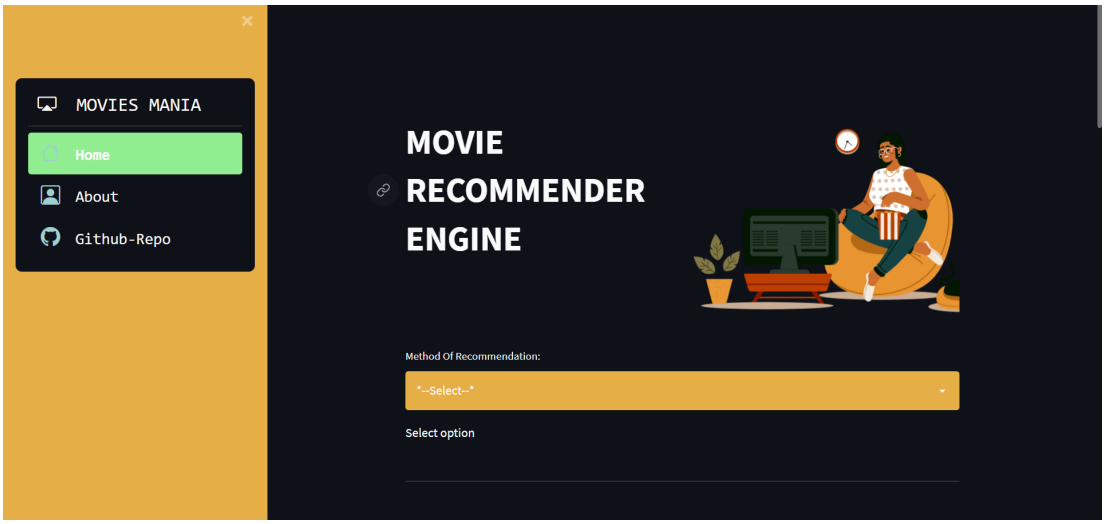
MOVIE RECOMMENDATION ENGINE

Recommender System is a system that seeks to predict or filter preferences according to the user's choices. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

Movie recommendation systems usually predict what movies a user will like based on the attributes present in previously liked movies. Such recommendation systems are beneficial for organizations that collect data from large amounts of customers, and wish to effectively provide the best suggestions possible.



Acknowledgements

- Streamlit Documentation
- Content Based Filtering
- KNN Algorithm
- Awesome README

Deployment

Check Out The Deployment Of this Project here :

`MovieRecommenderEngine`

(https://share.streamlit.io/pragyabisherwal/movie-recommendation-engine/app.py)

Video Demo And Working

https://youtu.be/DiNjSPCjkMU

Presentation

This is the slide show for the project I created on `MOVIE RECOMMENDATION ENGINE`

MOVIE RECOMMENDER ENGINE

Features

- Content-based Filtering
- Imdb Direct Link
- Movie And Genre Based Filering
- Option To Select Multiple Genres
- A movie could also be chosen based on its rating.
- The range of movies available can be increased.

Installation

Install my-project with npm

```
npm install my-project
cd my-project
```

Run Locally

Clone the project

```
git clone https://github.com/pragyabisherwal/movie-recommendation-engine.git
```

Go to the project directory

```
cd movie-recommendation-engine
```

Install dependencies

```
pip install -r requirements.txt
```

Start the server

```
streamlit run app.py
```

Tech Stack

WEB TECHNOLOGIES

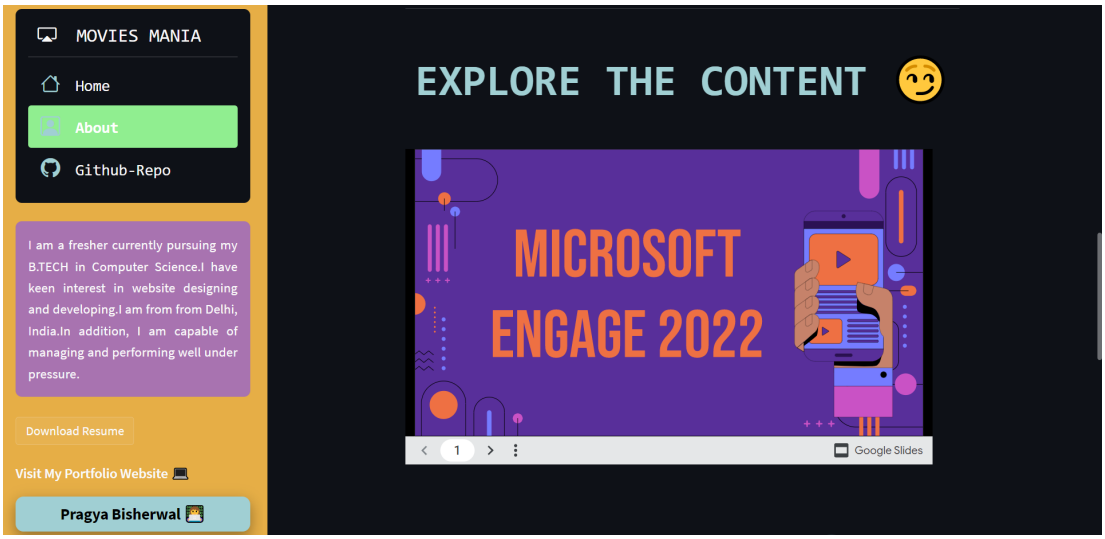
```
Backend -- Python , Numpy , Pandas

FrontEnd -- HTML,CSS,Streamlit
```

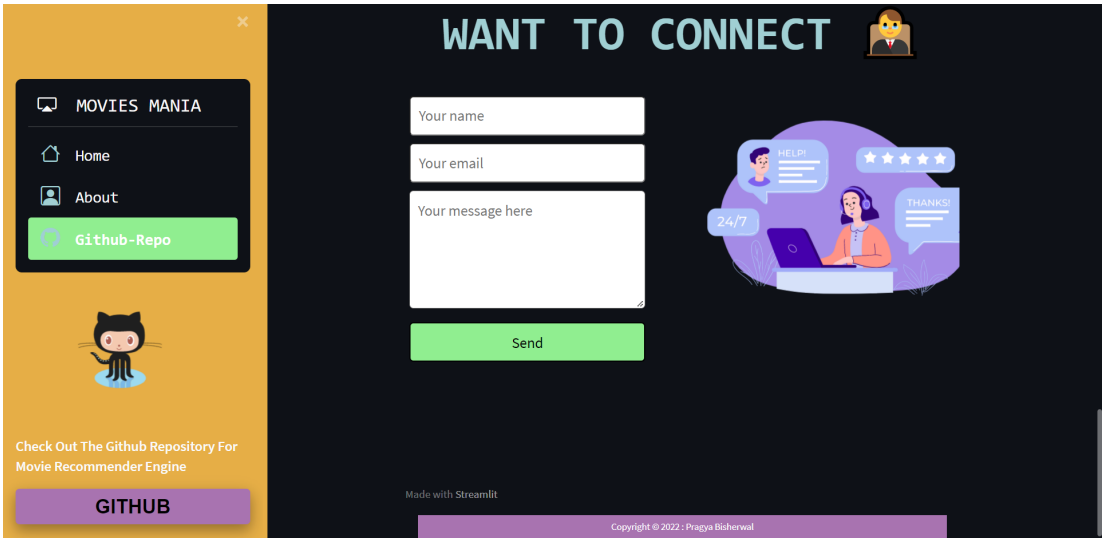
DATA SET

```
https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset
```

Screenshots

```
    GLIMPSE OF THE WEBSITE
```





Lessons Learned

KNN ALGORITHM

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

CONTENT BASED FILTERING

Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. Content-based Filtering is a Machine Learning technique that uses similarities in features to make decisions. This technique is often used in recommender systems, which are algorithms designed to advertise or recommend things to users based on knowledge accumulated about the user.

License

MIT

GNU General Public License v3.0

Contributing

Contributions are always welcome!

See `contributing.md` for ways to get started.

For Discussion Just Head Over to

`Pragya Bisherwal` (https://www.linkedin.com/in/pragya-bisherwal/)

Please adhere to this project's `code of conduct`.

Support

For support, email pragyabisherwal@gmail.com or

Connect with me here Pragya Bisherwal

🚀 About Me

- 👋 Hi, I'm @pragyabisherwal
- 👀 I'm interested in WEB DEVELOPMENT
- 🌱 I'm currently learning Django
- 💞️ I'm building my problem solving abilities.
- 📫 Want To Contact : pragyabisherwal@gmail.com

🔗 Links

MY PORTFOLIO LINKEDIN

Feedback

If you have any feedback, please reach out to me at pragyabisherwal@gmail.com

Author

- @PragyaBisherwal