**Surya Partap Singh**

**CS 470**

**Movie Recommendation System**

**Final Paper**

For my project I chose to work on Movie Recommendation system. The system shows the current popular movies along with the movies which user might like. There is a search option through which user can get recommendations for movies like the searched movies. User can read about the recommended movies and can watch a trailer; also the posters of the movies are also shown.

I have used python and its libraries in order to make the recommendation system work. Flask microframework was also used to make the machine learning model web friendly. HTML/CSS and JavaScript was used to design and implement API’s on the website.

**Working**

* I have used dataset from Kaggle which has 5000 movies and it has all the information about the movies and has an id of the movie which links the dataset to TMDB database. This id helps to make API calls.
* When we search any movies in the search bar then 3 movies are recommended, so when we enter any movie then we create a count matrix using count vectorizer with English stop words and fit over the soup column which contains keywords of all the movies. Then using cosine similarity, we check how similar the document is. After building cosine similarity for our dataset, we return about ten similar movies with their title, overview, id, names, date of release, ratings of the movie. I showed only three movies as sometimes the output of the program is not more than three movies, it is because the dataset only has 5000 movies. So, to remove this complexity I have shown only three movies.
* Let’s look at the **Cosine Similarity** how it works**: -** Itis a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.
* The id from the returned movies is used to make call to the TMDB API to request posters of the movie and its trailer. Also, I can get any information about the movie using this API call. Suppose for future I can continue working on this and can make a tab for cast of an particular movie.
* Now on the front page there are some recommended movies for the user and some popular movies are shown. The popular movies are shown by making an API call to the TMDB API and shown with their posters and their ratings, it changes with the change in the TMDB API.
* Now, the recommended movies which are shown on the front page are based on the user searches. As user search any movie then those movies are stored in movieR.csv file. And when user revisits the main page, then we randomly select multiple movies from csv file and cosine similarity is applied to get some movie recommendations to show the user movies they may like.

The libraries used for the following recommendations system are:- Flask, difflib, pandas, sklearn random and csv.

I didn’t wrote this code completely from scratch but I have used help from multiple articles about recommendation systems and that is how I learnt about the cosine similarity scores and multi arm bandit algorithm. The part in which another csv file is created using the user searches is based on the principle of multi arm bandit algorithm which is used by Netflix, Netflix keeps all the data related to the user to recommend them movies but my algorithm only saves the searched movies and shows recommendation based on that. I wanted to make an multiuser experience but I couldn’t develop the method on time, instead it works well with one user as it can store the search activity of the user and recommends movies to it.

This project has helped me a lot in learning a lot of things. Before starting this project, I didn’t know about JavaScript but in the process of developing this system I learnt about JavaScript and how API calls in JavaScript works. I was able to make successful call to the TMDB movie data base which was difficult in the first place as not much documentation was available on its website. Also, double arm bandit system was difficult to understand but I implemented accordingly but the algorithm is not entirely based on double arm but it’s a little different as it only implements for one user. For the design part of the website I took help from codepen, I took design ideas and implemented it accordingly as per the requirements of the website. There were some complications as there were so many things going on, on a single page but I was able to make an clean design.