

# Movies Recommender Engine.

27-5-2022

—  
Srushti Borate  
IIT Roorkee  
Electrical Engineering  
2nd year

## Why this project?

The purpose of a recommender system is to suggest relevant options to a user based on his/her previous choice thus making his/her experience more personalized. In today's rapidly growing world where there is huge information overload people would prefer to have a system which tells them what to buy/watch/read etc and this has a very huge scope in the market that's the reason why I chose this project.

## Overview

The recommender system which I have made is a content-based recommender engine which makes use of parameters like director's name, top 3 actors of the movie and genres. The algorithm used for giving the recommendations is **COSINE SIMILARITY**. For developing the front-end i have used **html, css, javascript, ajax, jquery** and for the back-end i have used **Flask**. Some of the main libraries used are **pandas, scikit\_learn, numpy, nltk...** The dataset which i have used contains 6011 hollywood movies. I have also made use of the **tmdb api** for extracting detailed information of the movies. Along with this I have done sentiment analysis for providing reviews for the selected movie about which i have written in detail in the report further.

## Directory Structure

```

└── MICROSOFT_ENGAGE_2022
    ├── _pycache_
    └── .ipynb_checkpoints
        └── Sentiments-checkpoint.ipynb
    ├── static
    │   ├── action.js
    │   ├── bg1.jpg
    │   ├── file.json
    │   ├── Frame 2.jpg
    │   ├── image.png
    │   ├── script.js
    │   ├── Style1.css
    │   ├── tenor.gif
    │   ├── upScroll.js
    │   ├── Vector.png
    │   ├── Vector2.png
    └── templates
        ├── home.html
        └── recommend.html
    ├── data.csv
    ├── keywords.pkl
    ├── nlp_model.pkl
    └── Procfile
        ├── requirements.txt
        ├── reviews.txt
        └── run.py
    └── Sentiments.ipynb

```

### STATIC FOLDER:

#### Script.js:

This javascript file contains code for **autocomplete of the search** entered in the input.

#### action.js:

This javascript file makes use of ajax for **getting and posting the information from either the backend or from the api.**

### upScroll.js:

This javascript file contains the code for enabling up scroll action on clicking on arrow button

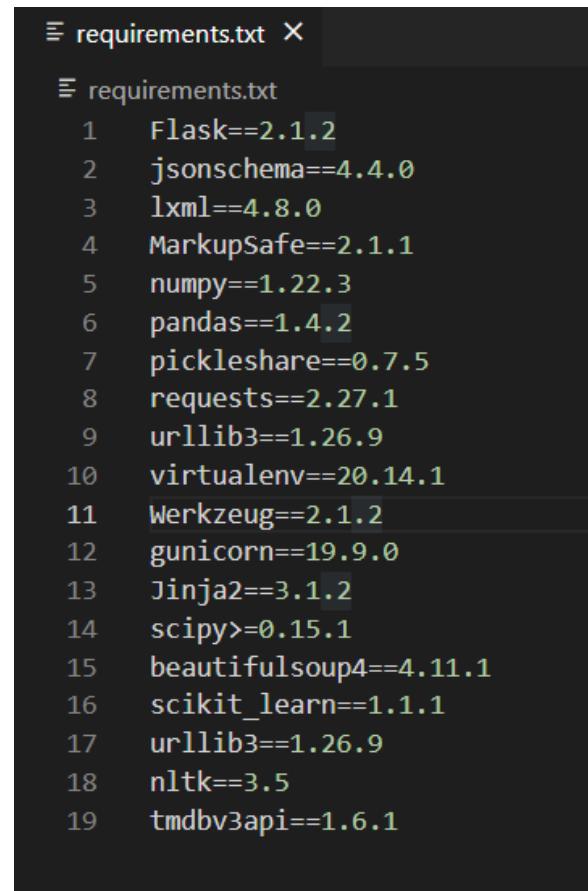
Style1.css: Css design file.

### **TEMPLATES FOLDER:**

Contains home.html and recommend.html

**run.py:** file which contains the recommendation algorithm and flask app.

### **requirements.txt:**



```
requirements.txt
Flask==2.1.2
jsonschema==4.4.0
lxml==4.8.0
MarkupSafe==2.1.1
numpy==1.22.3
pandas==1.4.2
pickleshare==0.7.5
requests==2.27.1
urllib3==1.26.9
virtualenv==20.14.1
Werkzeug==2.1.2
gunicorn==19.9.0
Jinja2==3.1.2
scipy>=0.15.1
beautifulsoup4==4.11.1
scikit_learn==1.1.1
urllib3==1.26.9
nltk==3.5
tmdbv3api==1.6.1
```

The requirements.txt file for a python can be generated by typing the command in the terminal:

**Pip freeze > requirements.txt**

For running the project on local machine after cloning the github repository

Run the command **pip install -r requirements.txt** on the terminal

## data.csv:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	index	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb							
2	1	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar	CCH Pounder Joel David Moore Wes Studi James Cameron Action Adventure Fantasy Sci-Fi							
3	2	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean:	Johnny Depp Orlando Bloom Jack Davenport Gore Verbinski Action Adventure Fantasy							
4	3	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action Adventure Thriller	spectre	Christoph Waltz Rory Kinnear Stephanie Sigman Sam Mendes Action Adventure Thriller							
5	4	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action Thriller	the dark knight rises	Tom Hardy Christian Bale Joseph Gordon-Levitt Christopher Nolan Action Thriller							
6	5	Doug Walker	Doug Walker	Rob Walker	unknown	Documentary	star wars: episode vii - th	Doug Walker Rob Walker unknown Doug Walker Documentary							
7	6	Andrew Stanton	Daryl Sabara	Samantha Morton	Polly Walker	Action Adventure Sci-Fi	john carter	Daryl Sabara Samantha Morton Polly Walker Andrew Stanton Action Adventure Sci-Fi							
8	7	Nathan Greno	Brad Garrett	Donna Murphy	M.C. Gainey	Adventure Animation Comedy Family	tangled	Brad Garrett Donna Murphy M.C. Gainey Nathan Greno Adventure Animation Comedy Family Fantas							
9	8	Joss Whedon	Chris Hemsworth	Robert Downey Jr.	Scarlett Johansson	Action Adventure Sci-Fi	avengers: age of ultron	Chris Hemsworth Robert Downey Jr. Scarlett Johansson Joss Whedon Action Adventure Sci-Fi							
10	9	David Yates	Alan Rickman	Daniel Radcliffe	Rupert Grint	Action Family Fantasy Mystery	harry potter and the half-	Alan Rickman Daniel Radcliffe Rupert Grint David Yates Adventure Family Fantasy Mystery							
11	10	Zack Snyder	Henry Cavill	Lauren Cohan	Alan D. Purwin	Action Adventure Sci-Fi	batman v superman: daw	Henry Cavill Lauren Cohan Alan D. Purwin Zack Snyder Action Adventure Sci-Fi							
12	11	Bryan Singer	Kevin Spacey	Marlon Brando	Frank Langella	Action Adventure Sci-Fi	superman returns	Kevin Spacey Marlon Brando Frank Langella Bryan Singer Action Adventure Sci-Fi							
13	12	Marc Forster	Giancarlo Giannini	Mathieu Amalric	Rory Kinnear	Action Adventure	quantum of solace	Giancarlo Giannini Mathieu Amalric Rory Kinnear Marc Forster Action Adventure							
14	13	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean:	Johnny Depp Orlando Bloom Jack Davenport Gore Verbinski Action Adventure Fantasy							
15	14	Gore Verbinski	Johnny Depp	Ruth Wilson	Tom Wilkinson	Action Adventure Western	the lone ranger	Johnny Depp Ruth Wilson Tom Wilkinson Gore Verbinski Action Adventure Western							

data.csv contains information about the directors name, top 3 actors names and genre for a particular movie.

**reviews.txt, keywords.pkl, npl\_model.pkl and sentiments.ipynb** are for sentiment analysis and generating reviews for the movie.

**procfile:** specifies the commands that are executed by the app on startup.

## About the algorithm: Cosine Similarity

First understands what is **similarity score**:

As you might see in the data.csv we have a column named “comb” it contains combined features of the movie and by using the text similarities of “comb” column for two items we generate a similarity score matrix which has all the values between 0 to 1.

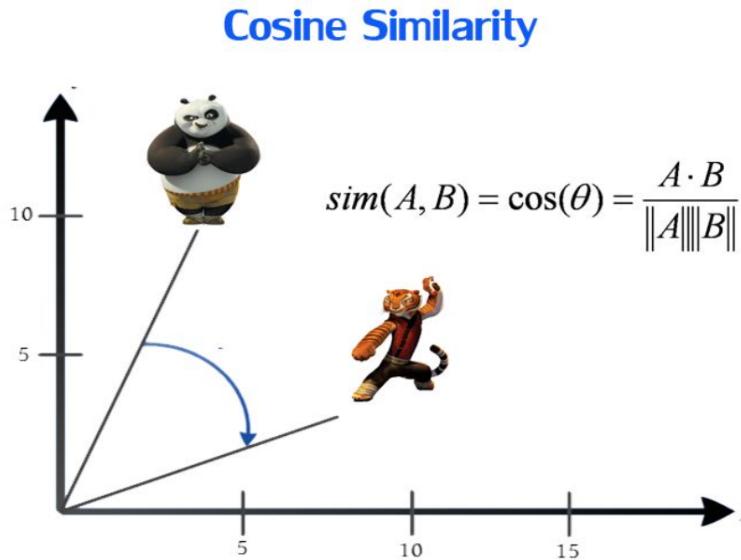
Let's understand this score matrix with simple illustration below:

movies	0	1	2	3
0	1	0.24	0.67	0.93
1		1	0.11	0.43
2			1	0.62
3				1

We can see a particular movie will have similarity count of 1 if its compared to itself

Now the similarity count for movie '3' with movie '0' is 0.93 which means that the movies are pretty much similar to each other.

Now lets see how we get this similarity score matrix:



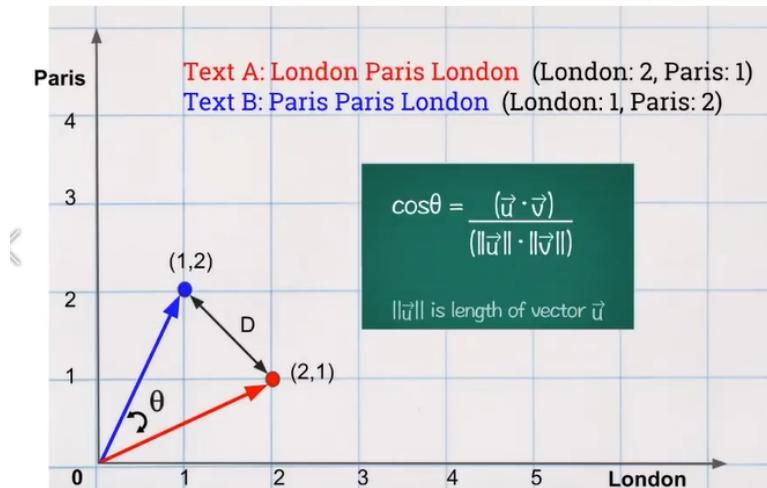
Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. Lets understand this with an example:

Suppose we have

Text A: London Paris London (London: 2, Paris: 1)

Text B: Paris Paris London (London: 1, paris: 2)

### Distance Between Two Vectors



The value of  $\cos(\theta)$  between these two vectors will give us the degree of similarity between the two texts Text A and Text B. The same concept is applied to the combined features column of the data set.

The code for the algorithm is:

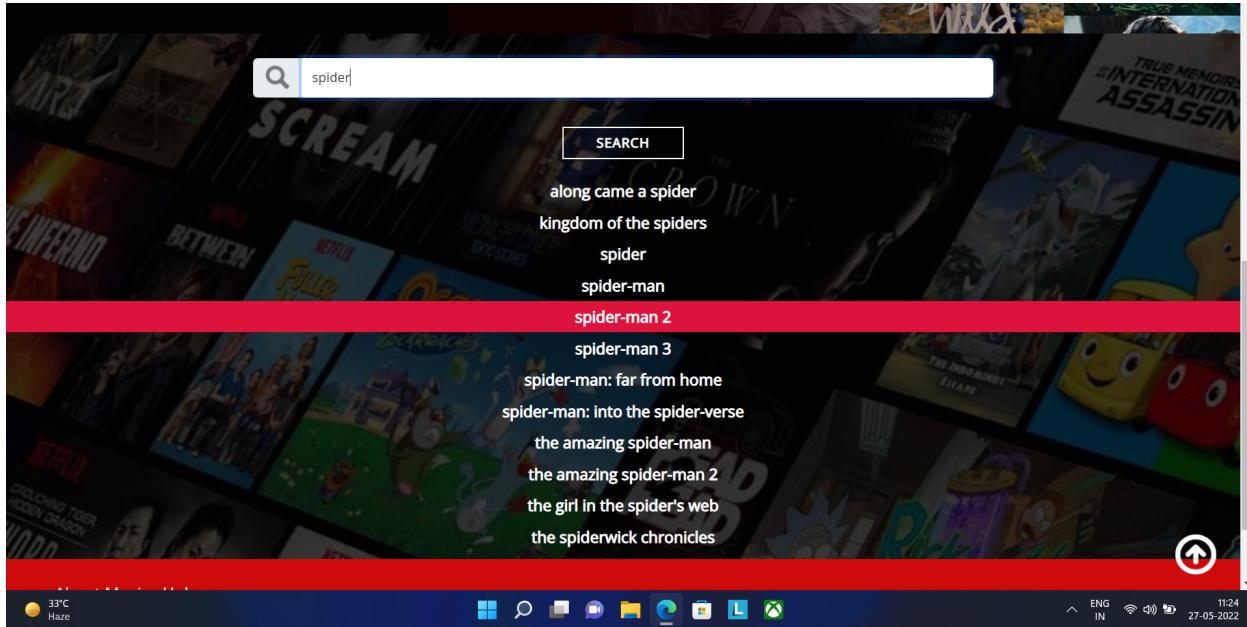
```
#algorithm for finding similarity count-matrix
def create_similarity():
    df=pd.read_csv("data.csv")
    cv=CountVectorizer()
    count_matrix=cv.fit_transform(df["comb"])
    cosine_sim=cosine_similarity(count_matrix)
    return df,cosine_sim
```

## Features:

### Autocomplete Search:

**Search** autocomplete suggests keywords based on what you're typing in the search bar. The feature allows you to predict what shoppers are looking for, helping them complete their search and get their search results faster.

It could happen that sometimes the user is not able to recollect the entire name of the movie or it could be that he/she misspelled it in such cases having an autocomplete search facility will maintain users interest.



## Footer:



I believe having a footer to your website is the best option as it provides a space to summarize the usability of your website and also provides links to some important sites like github and linkedin in my case.

## Extra Details for the movie entered:

1. Cast Info: This is acquired by passing a request for data access from the tmdb api.
2. Details like ratings, overview, genre, duration, year of release is also provided.

## Sentiments Analysis:

The algorithm used for prediction of sentiments is a simple logistic regression giving 99.12% accuracy rate as you will find it in the sentiments.ipynb file.

The data set used for this purpose is reviews.txt which i have then split into test and train i have made use of natural language processing for finding out the frequency of keywords that occur in the review of a specific movie and then predict the sentiment it's depicting for example if the review says whether the movie is bad or good.

Reviews for a specific movie have been acquired from imdb-id for the movie using web-scraping; the library used for the same is beautifulsoup.

USER REVIEWS	
Reviews	Ratings
Really hope, these people stick to their dirty polictics and leave movies to people who actually do it.	Good : 😊
Trashy junk for trash lovers, why do people watch junk like this is beyond me. This is not for intelligent ppl, it is a film for the narrow minded. Live love laugh life is to short for hatred.	Good : 😊
A very scary movie that everyone should see before voting in November. I use the word "scary", not because this film revealed anything to me that I didn't already know. Actually, I'd estimate that I already knew of approximately 95% of the content covered in this movie. What's scary is that the content covered in this film was all readily available before the 2008 election (hence why I voted for the old guy and the dumb broad) and yet so many people ignored it then and continue to ignore it now. How dense are people? Many republicans are against Obama for his politics or think he's dumb or think he's naive, etc. No, this is not the case. I said in 2008 and I'll repeat it now... Barrack Hussein Obama is an evil man who hates the United States of America. That statement seems radical to the sheeple who suck from the teet of Obama's America but it's the truth and if they weren't so immediately selfish it would be painfully obvious. This 2012 election isn't a vote between Romney and Obama or republican and democrat or white and black... it's a vote between America and anti-America. I'd say the choice is pretty obvious.	Bad : 😥