Movie Recommendation System Using Machine Learning

Shivam Shishodia
Dept. of Computer Science and Engineering
Galgotias University
shivamshishodia2000@gmail.com

Tarun Kumar
Dept. of Computer Science and Engineering
Galgotias University
bagheltarun114@gmail.com

Ms. Shweta Mayor Sabharwal
Assistant Professor
Galgotias University
shweta.sabharwal@galgotias university.edu.in

Abstract: *A recommender system is a system that provides you with similar types of products or solutions and results that you are looking for. For example, if you go to a clothing store, you want a t-shirt with another design or another color, then the store recommends another color. Movie recommendation systems aim to help moviegoers by suggesting which movie to watch without going through the lengthy process of choosing from a large number of thousands of movies, which is time-consuming and confusing. This content-based recommender system recommends tasks for web pages. Recommendation engines use various algorithms to filter data and then recommend the most relevant items to consumers.A movie recommendation system will recommend the most relevant and relevant movies for a given search category, and if a user is visiting a movie website for the first time, the website will have no history for that user. In this case, users can search their movie recommendations by genre, release year, director or actor, and their favorite movie itself to get new movie recommendations.*

Keywords: Movie Recommendation Systems, Content-Based , Movie recommendation, machine learning project.

INTRODUCTION

At the beginning of the 21st century, electronic commerce on the Internet was developing. The online shopping and entertainment industry is at its peak. For years to come, all online content will be the new normal. Suppose you are shopping online on a site like amazon.com, with more than 60 million products sold, the same goes for Flipkart and other e-commerce sites.Entertainment sites like Netflix, Amazon Prime and Hotstar allow you to watch over 10 million movies and series. If you need specific content from these sites, just search. But what about other products? You are lost and never find your way back. Recommender systems are there.Recommender systems play an important role as guides in systems like Amazon and Netflix. Without a recommendation system, many e-commerce and entertainment sites would act like a database, asking you what you're looking for. It's a huge loss for these companies if people don't buy their products or watch their movies. Likewise, I would be a huge loss if my users couldn't find the product they needed. Therefore, industries and users need to integrate recommender systems across multiple websites.

OVERVIEW OF RECOMMENDATION SYSTEM

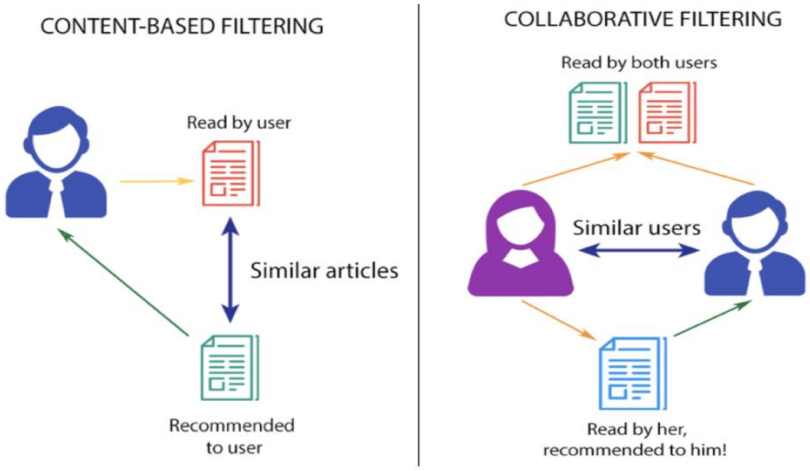
A. Recommender System

Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them. They are among the most powerful machine learning systems that online retailers implement in order to drive sales.. For example .They're used by various large name companies like Google, Instagram, Spotify, Amazon, Reddit, Netflix etc.

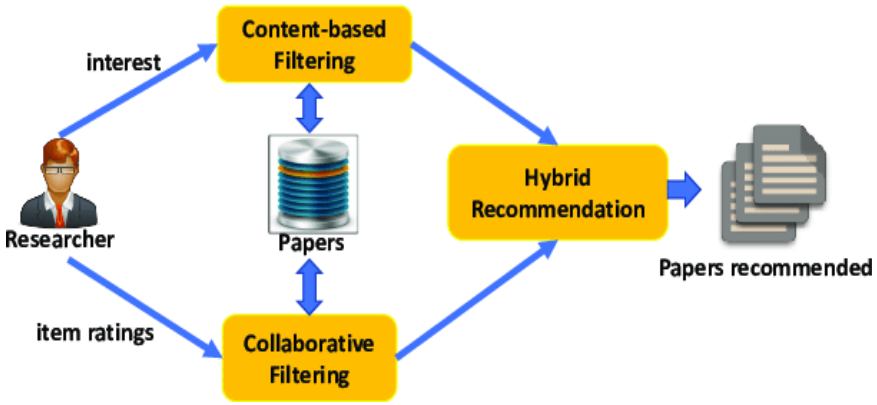
B. Types of Recommender System

1) *Content Based Recommender System*: A Content-Based Recommender works by the data that we take from the user, either explicitly (rating) or implicitly (clicking on a link). By the data we create a user profile, which is then used to suggest to the user, as the user provides more input or take more actions on the recommendation, the engine becomes more accurate . For example, while watching movies, YouTube suggests similar movies.

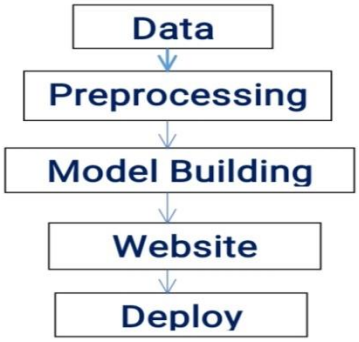
2) *Collaborative Recommender System:* In Collaborative Recommender System, The Data from users is collected to recommend different products. Collaborative filtering is currently one of the most frequently used approaches and usually provides better results than content-based recommendations. Some examples of this are found in the recommendation systems of Youtube, Netflix, and Spotify.



1) *Hybrid Recommendation System:* This approach overcomes the limitations of both content-based and collaborative filtering methods. In this article, we will discuss the hybrid recommendation systems in detail and we will learn how to build a hybrid recommendation system Google Search results and suggestions are a great example of Hybrid Recommendation System. Youtube, Instagram and Python implementation named LightFM.



Flow Chart



A. Data

Data is the most important and foundation for machine learning projects. We are takeing the data form "TMDB 5000 Movie Dataset" datasets is key for a recommendation system. The more the data is useful for better the recommending results.

B. Pre-processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

C. Model Building

We will Building a model in machine learning project is creating a mathematical representation by

generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain result. The different test cases are performed to test the project module is giving the expected outcome .

D. Website Designing

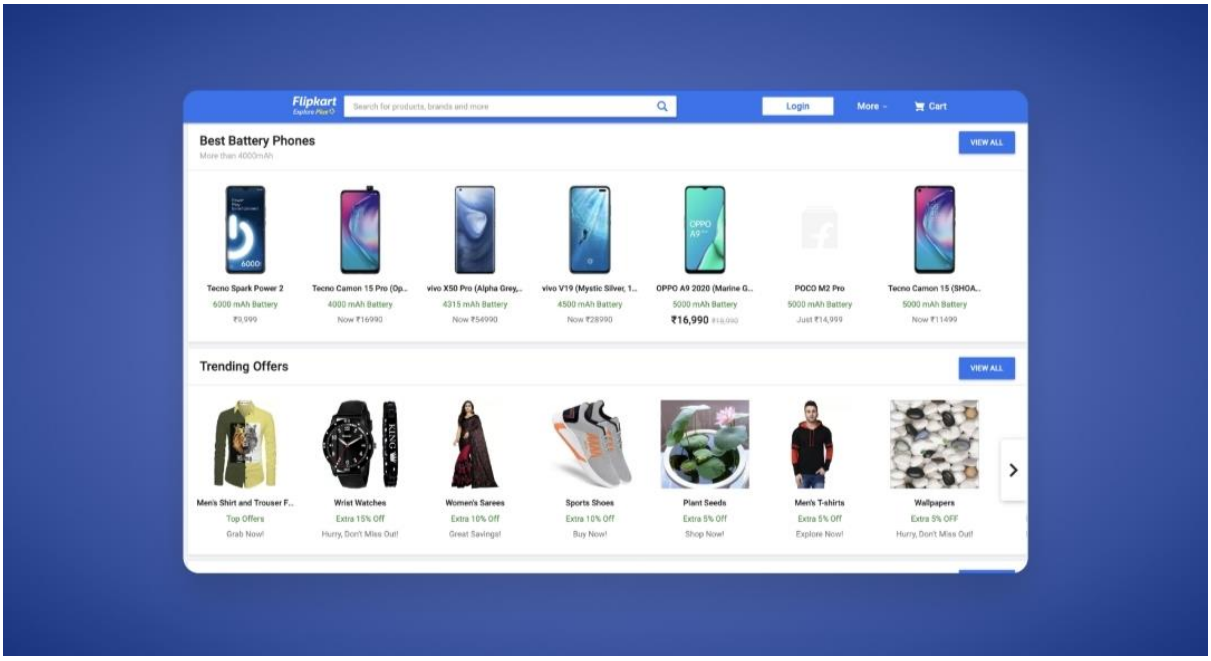
After we have created a model and we will create a website in this website same as a model . for web designing we can use different types of platform.

E. Deployment of System

After website Desiging than we are deploy our project on any Deployment sit, the product is deployed in the virtual environment or released into the local hosting like Heroku ,etc.

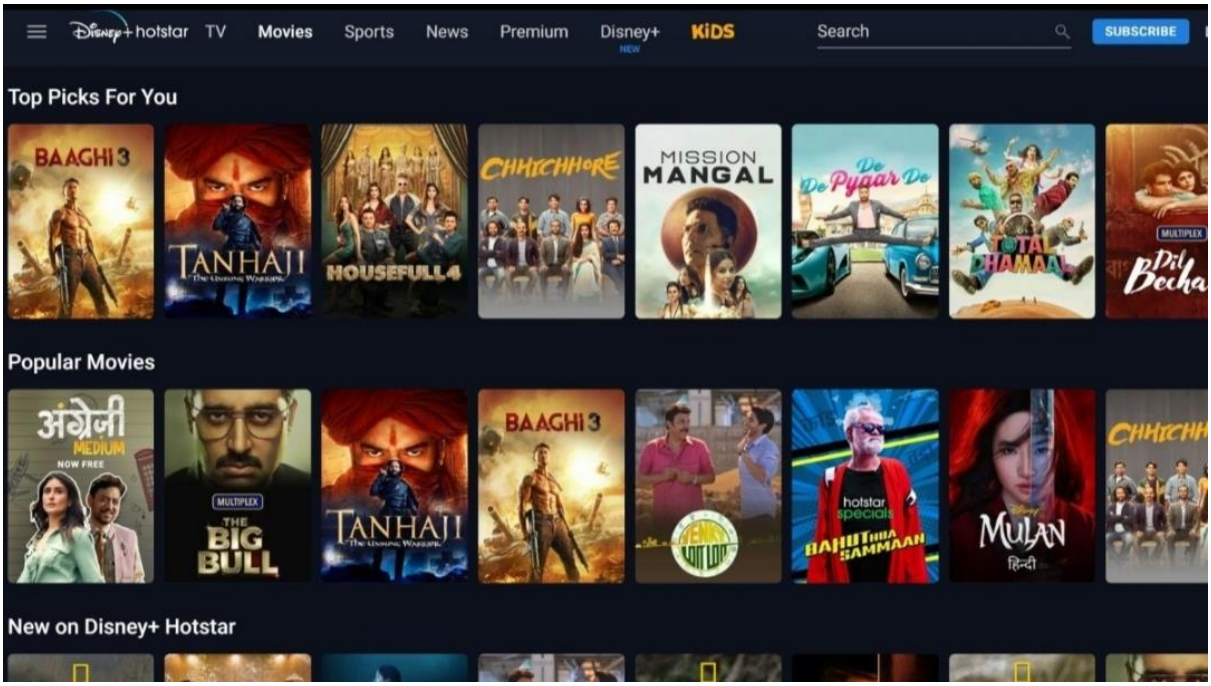
C. E-Commerce

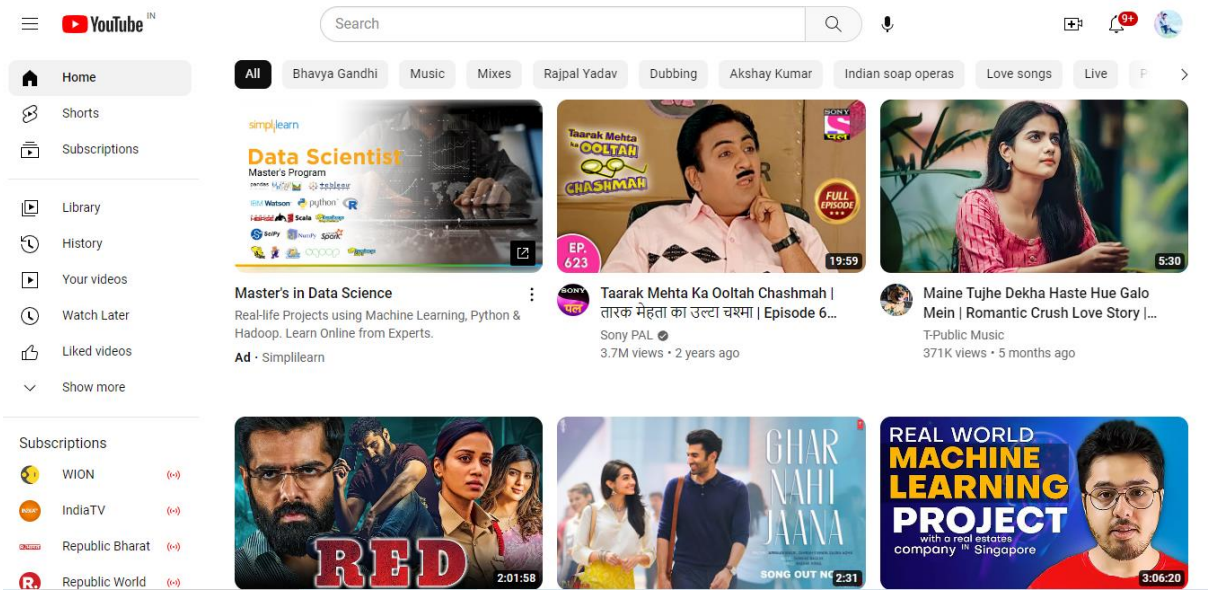
Sites such as Amazon.com and Flipkart use recommendation systems based on a user's past purchase history. These recommender systems also use filters to select the right product to recommend. The Recommender also recommends products that can be used with other products or purchased at the same time by other users/customers . There are some of the recommending products over Flipkart:



D. Movie and Video Recommenders

Netflix and Hotstar are prime examples of movie and video recommendation systems. However, most upgrades and user-friendly recommendations can be found on YouTube's website. Recommend similar content that the user has already seen or liked. The system uses a content-based system which also has settings called tags that define genres, languages, etc. Include what defines the movies.





INTELLIGENT MOVIE RECOMMENDER SYSTEM

A. Datasets and Pre-processing

Under normal circumstances, collecting movies and their information is a difficult task. However, for this project, we used the publicly available dataset of the serious 5000 movie dataset. It is mainly popular with Hollywood and Bollywood movies that are well known to most users. The dataset contains useful information about movies such as movie title, cast (actors, actresses and their roles), crew (director and other members), genre, release date, description , etc.

B. Extraction of Data

The data extraction takes into account the required tables and information about the movie. All unnecessary data should be discarded. For example, movie length and budget do not affect movie similarity. Null values should also be ignored to avoid further errors and underreporting issues.

C. Creation of Tags

A data block is a set of results that contains only the necessary values. In our example, this would be the movie title, genre, release date, cast, and crew. To make smart movie-to-movie recommendations, we need to create a table called "tags", which can be a combination of all available data and keywords. Each label will represent a movie in the dataset.

D. Normalization

Reduce the occurrence of similar worlds and stop words (on, the, is, is, that). All available data must first be normalized with each variable. Count Vectorizer is used to manipulate data and remove these stop words. We may (not) use natural language processing models to perform such operations. A feature called Porter Stemmer will replace similar words with a single word.For example, 'Loving, Loved, Lover' would be replaced with the single word 'love'. This will make calculating the similarity between tags easier and more accurate.

E. Bag of Words

In Bag of Words, We Combine all the words in Tags into one Single long word.
For ex. Tag1 + Tag2 + Tag3 + …. = Tag
As we have 5000 movies dataset, we get 5000 tags representing 5000 movies. Now, we have to calculate the 5000 common words which describe each movie.
By calculating the frequency of words in each tag, we can get these 5000 words.

Let's say, Word 1 = Action
Word 2 = Adventure
Word 3 = War
Word 5000 = 2021

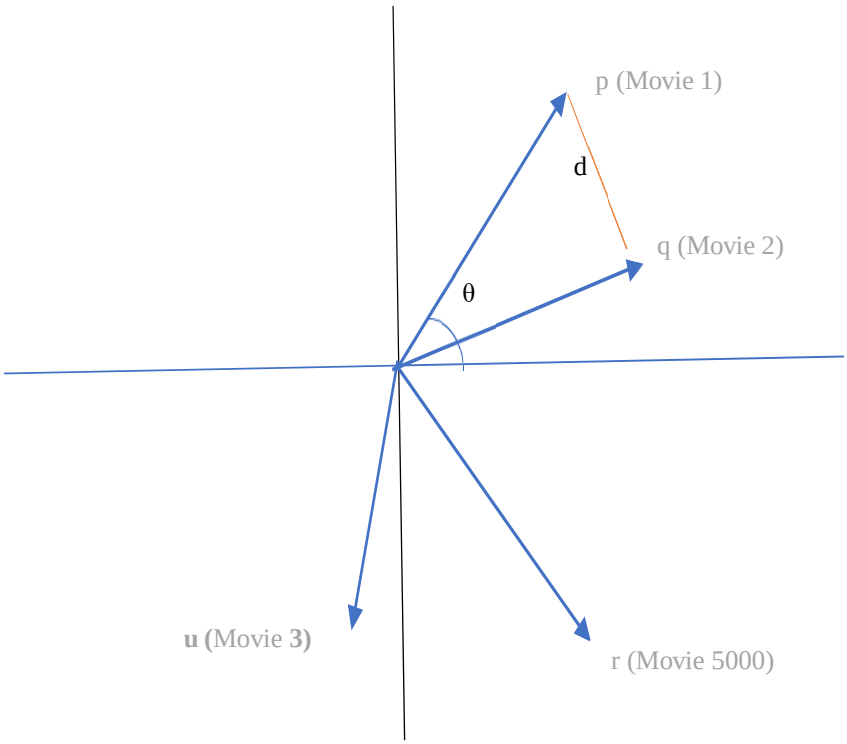
F. Matrix of Vectors

In Vectorization, we got 5000 words representing 5000 movies
Now, we create a matrix of 5000 movies x 5000 words (5000,5000)

	Word 1	Word 2	Word 3	Word 5000
Movie 1	5	7	1	4
Movie 2	5	5	0	4
Movie 3	6	1	4	0
Movie 4	3	3	0	3
Movie 5000	0	2	3	7

Movie Vector Representation
Every movie in the dataset is converted into A vector.
Graphically it can be represented as,

G. Similarity



To Calculate the Similarity between two movies, we need to calculate the distance between their vectors.There are two methods to calculate this distance:

1) Euclidian Distance
2) Cosine Distance

I. Euclidian Distance

Euclidean distance is the distance between a point on one line segment and a point on another line segment. We can call the end-to-end distance between these two line segments. It is a useful distance calculation technique, but at the same time not very efficient.
Formula for Euclidian distance:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

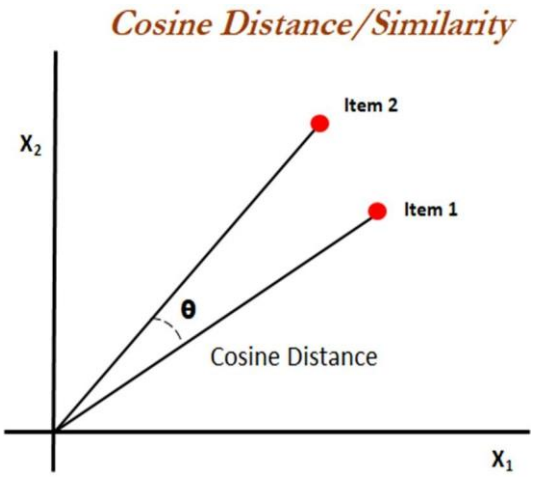
J. Cosine Distance

Cosine distance is measured by taking the cosine of the angle between two vectors. This is a very efficient way to calculate distances, since the values are between Zero to One.
Formula for Cosine distance:
Cos (θ) = Distance between p (movie 1) and q (movie 2)

K. 3.7 Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$



L. **Retaining the calculated Similarity**

All vectors will be sent through the Cosine-Summit function. Each movie will have its own similarity score with all the other movies. We will sort them first using the highest similarity score. The calculated protocol of the will be saved in a separate file so that the system can work as quickly as possible. Each time the recommendation function is called, the similarity file is retrieved to query the desired similar movies.

IMPLEMENTATION

We use a content-based recommendation system because it is simpler than collaborative and hybrid systems. Many small businesses use content-based filtering on their e-commerce sites and online marketplaces.

We will use cosine similarity to calculate a numerical value representing the similarity between two films. We use the cosine similarity score because it is independent of size and relatively easy and quick to calculate. Mathematically, it is defined as follows:

We can now properly define our recommendation function. Here are the next steps we will take:

1) Get the index of the movie based on its title.
2) Get a list of cosine similarity scores for this particular movie to all movies. Convert it to a list of tuples where the first element is its position and the second element is the similarity score.
3) Sort the above list of tuples by similarity score; i.e. the second element.
4) Get the first 10 items from this list. Ignore the first item as it refers to itself (the closest match to a particular movie is the movie itself). Returns the title corresponding to the index of the top element.

Although our system did a decent job of finding movies with similar plot descriptions, the quality of the recommendations wasn't that good. Batman: The Dark Knight Rises brought back every Batman movie, and those who liked it were more likely to like Christopher Nolan's other films.This is something that the current system cannot grasp. We are going to use a method called text vectorization

The credits genres actors and keywords are all combined and converted into tags.

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
```

```
vectors = cv.fit_transform(new_df['tags']).toarray()
```

The “tags” are something that describes the movie

```
In [41]: movies.head()
```

Out[41]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[culturecIash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

```
In : import nltk

In : from nltk.stem.porter import PorterStemmer
     ps = PorterStemmer()

In : def stem(text):
         y = []

         for i in text.split():
             ps.stem(i)
```

Labels are converted to vectors via text.vectorization.
We use the "bag of words" technique to calculate the similarity between tags.
We are using the scikit.learn library.

```
]: from sklearn.metrics.pairwise import cosine_similarity

]: cosine_similarity(vectors)
```

We are going to calculate theta.
This theta is the distance between two segmentas i.e. vectors.
Cosine_similarity has a value from 0 to 1.
It calculates the angles between the vectors and converts them into 0 to 1 value

```
In [34]: similarity

Out[34]: array([[1.        , 0.08964215, 0.06071767, ..., 0.02519763, 0.0277885 ,
                 0.        ],
                [0.08964215, 1.        , 0.06350006, ..., 0.02635231, 0.        ,
                 0.        ],
                [0.06071767, 0.06350006, 1.        , ..., 0.02677398, 0.        ,
                 0.        ],
                ...,
                [0.02519763, 0.02635231, 0.02677398, ..., 1.        , 0.07352146,
                 0.04774099],
                [0.0277885 , 0.        , 0.        , ..., 0.07352146, 1.        ,
                 0.05264981],
                [0.        , 0.        , 0.        , ..., 0.04774099, 0.05264981,
                 1.        ]])
```

We will create a "command" function that can take "MovieName" as input and find its index position in the corresponding files. Based on the labels we provide and the scores calculated by the similarity function, the top 5 scores will be the most similar movies. Their names are also preserved with their index positions.

```
]: def recommend(movie):
       movie_index = new_df[new_df['title'] == movie].index[0]
       distances = similarity[movie_index]
       movies_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

       for i in movies_list:
           print(new_df.iloc[i[0]].title)
```

We will pass all the vectors (movies) through the "command" function. It will calculate the similarity from one vector to another. We will sort this array by the calculated maximum similarity. The first five films will be our release. As following,

```
In [37]: recommend('Batman')

         Batman
         Batman & Robin
         The Dark Knight Rises
         Batman Begins
         Batman Returns
```

A. Creating other Recommending Filters

Similarly, we can do the same for the remaining tables in the dataset. We will calculate the similarity based on genre, category, release year, actor, director and save their similarity in a separate file.

```
In [37]: genrelist

Out[37]: ['action',
          'adventure',
          'animation',
          'comedy',
          'crime',
          'documentary',
          'drama',
          'family',
          'fantasy',
          'foreign',
          'history',
          'horror',
          'music',
          'mystery',
          'romance',
          'sciencefiction',
          'thriller',
          'tvmovie',
          'war',
          'western']
```

```
In [38]: recommend ('crime')

Wall Street: Money Never Sleeps
Black Mass
Catch Me If You Can
Casino
American Hustle
Mean Streets
21
Black Water Transit
Blow
Once Upon a Time in America
```

```
In [55]: recommend2 ('2015.0')

Avengers: Age of Ultron
Jurassic World
Furious 7
The Good Dinosaur
Jupiter Ascending
Inside Out
The Lovers
Tomorrowland
The Hunger Games: Mockingjay - Part 2
Terminator Genisys
```

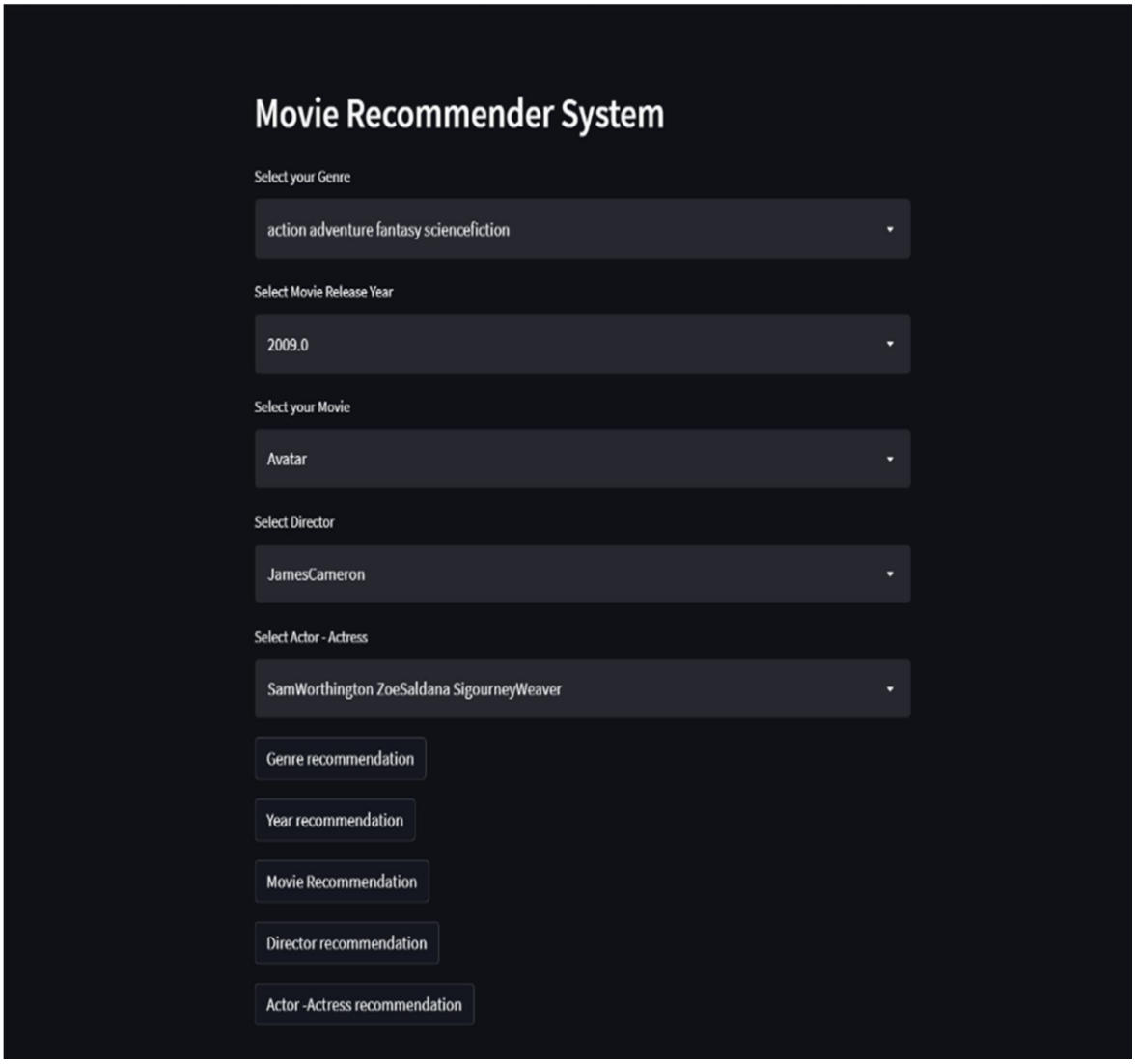
```
In [69]: recommend4 ('JohnnyDepp OrlandoBloom KeiraKnightley')

         Pirates of the Caribbean: Dead Man's Chest
         Pirates of the Caribbean: The Curse of the Black Pearl
         The Lone Ranger
         Pirates of the Caribbean: On Stranger Tides
         Alice in Wonderland
         Alice Through the Looking Glass
         Charlie and the Chocolate Factory
         Dark Shadows
         Troy
         Rango
```

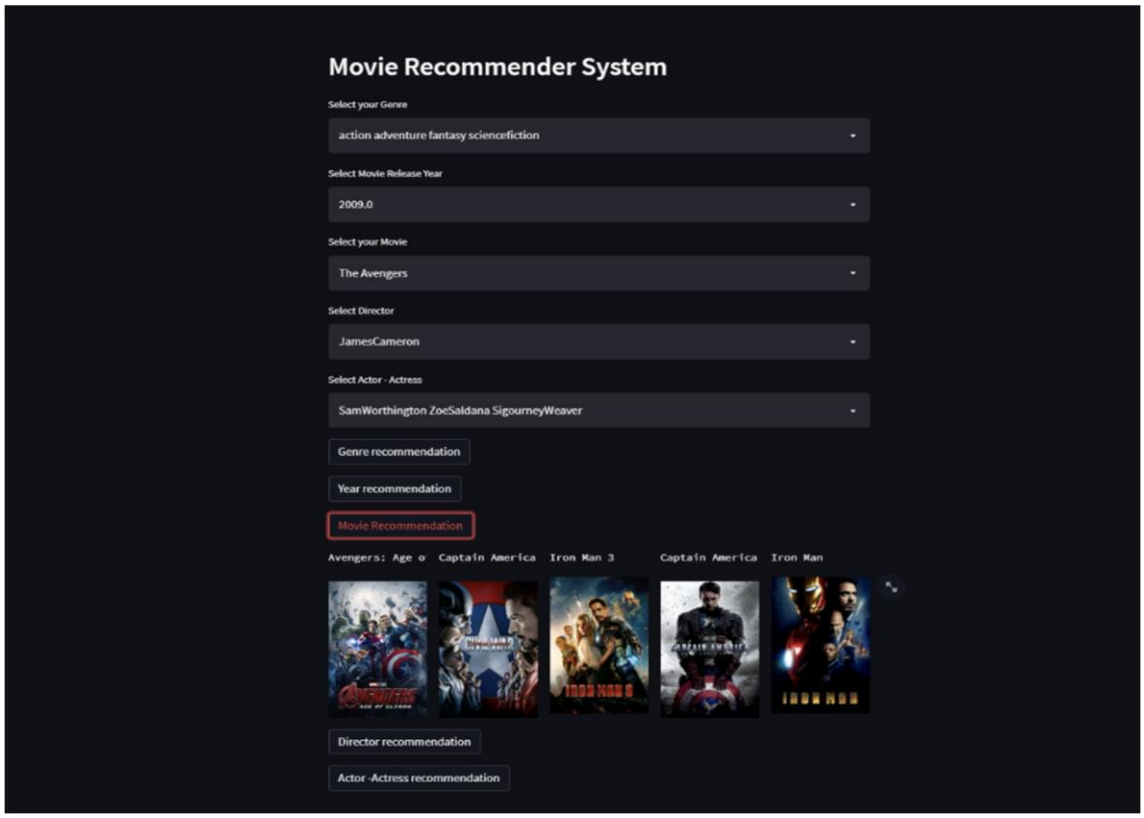
Each Function will have a different result based on the selected names from the respective entity.

B. **System Design and Model:**

Use Juypter Notebook as for Python application development. By creating a virtual environment, we can create a private website interface using the Python Streamlit library. Streamlit is an open-source Python library for building and sharing web applications for data science and machine learning projects. This library makes it possible to build and deploy data science solutions in minutes with just a few lines of code.

RESULTS





CONCLUSION

Recommender systems can be improved based on current and future requirements to improve quality and achieve better recommendation results. Recommender systems can be your virtual guides on e-commerce platforms, and if people don't buy or watch their products, it's a huge loss for companies like Amazon and Netflix While demand of machines is growing - "ML" automation solution has become one of the fastest growing technologies along with AI and data science. In the future, the recommender system will be used to predict product demand, connect buyers and sellers, and ultimately become the backbone of the supply chain. Large companies such as Amazon, Netflix and Facebook now need recommendation systems to cope with the ever-increasing number of products and users.

REFERENCES

[1] Mahesh Giyani and Neha “A Review of Movie Recommendation System: Limitations, Survey and Challenges”
[2] Nirav Raval, Vijayshri Khedkar contant Filtering Based Moive Recommendation System”
[3] Bhusan K. and Sripant “Recommendation System: Literature Survey and Challenges.
[4] Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: You might also like: privacy risks of collaborative filtering.
[5] Research.ijcaonline.org
[6] Dataset: tmdb-5000-movies dataset.