Amity University Noida

2022

Project Title - Content Based Movie Recommender System

Name - Arunya Goojar

Programme - B.Tech. CSE Evening

Semester - 5

Acknowledgement

I have taken efforts during this project. However, it'd not are possible without the type support and help of my faculty guide Mis/Mrs Nidhi Chandra.

I am highly indebted to my faculty guide for his or her guidance and constant supervision further as for providing necessary information regarding the project & also for his or her support in completing the project.

I would wish to express my gratitude towards my parents & friends for his or her kind co-operation and encouragement which help me in completion of this project.

Arunya Goojar

Abstract

With the rising quantity of information that people read on a daily basis, determining how to swiftly access information that meets people's demands has become an essential issue these days. People who tend to obtain information by inputting a query or keywords have benefited greatly from the effort in information retrieval.

If certain information-intensive websites can proactively recommend products or information items that users may be fascinated in, the efficiency and pleasure of users in receiving information items would considerably improve. This is where the work in the field of recommendation systems began. Over the last few years, great progress has been achieved in this field, from non-personalised to personalised, and more recently, deep learning recommendation systems.

Despite the widespread use of recommendation systems, there are still several concerns and obstacles in creating high-quality recommendation systems. A systematic and thorough review procedure is necessary to assess the quality of a recommendation system. This research examines some well-established recommendation systems as well as current measures for assessing them. Furthermore, this study describes the project's implementation - a web application for the unbiased assessment of three important collaborative filtering recommendation algorithms, namely item based, user based, and matrix factorisation.

The programme contains a broad range of easily adjustable assessment measures that allow users to compare the performance of various recommendation systems. The project's goal is to give designers with a complete platform for evaluating recommender systems and guiding them in designing better recommender systems.

Table of Contents

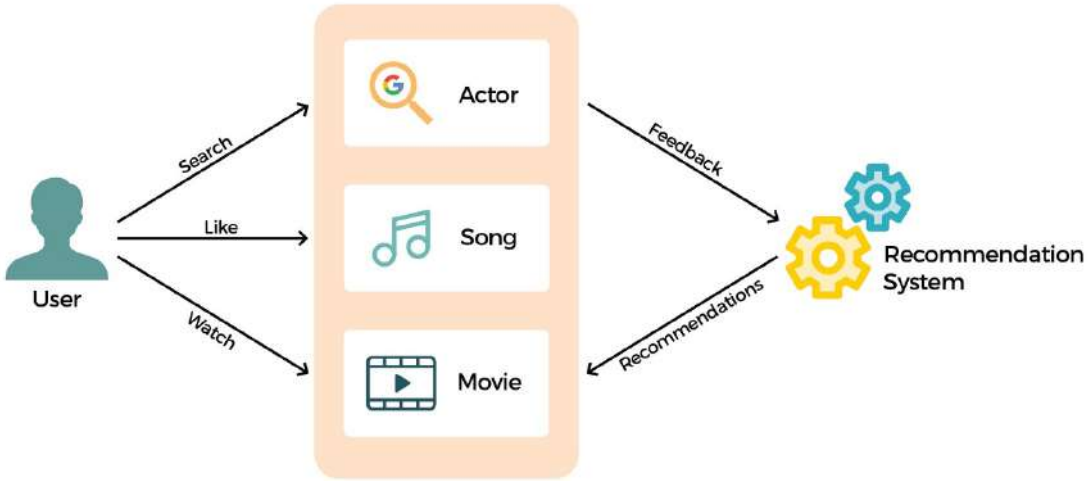
Introduction

Recommendation system is a filtering system which tries to predict user preference for an item. They are mostly used in advertising and business setting. But currently recommendation systems are being used everywhere whether it is a Music platform video platform even the search results are filtered on the user experience.

As the world is moving on towards internet recommendation systems matter a lot both from user perspective and business perspective. If the user finds something which he is interested in let's take music for example, if a person likes hip-hop music the recommendation system will recommend more music in the hip-hop genre.

As for the businesses it is beneficial as on an online marketplace they can show more goods than they actually have, so there is no limit on the online platform but in a physical shop they are limited to the physical space available for storage.

Amazon, Netflix, YouTube, Spotify, Apple Music, and others are some examples. These companies use hybrid method to recommend content to the users



Background

In this day and age, it is quite tough for consumers to find content that they are truly interested in. It is also difficult for content providers to make their material stand out from the throng. As a result, several researchers and businesses create Recommender Systems to address the paradox.

The objective of the Recommendation System is to link users with information, which helps users locate information that is relevant to them while also pushing content to particular users. Both users and content suppliers benefit from this arrangement.

This term paper will provide a more effective suggestion approach that may be applied on an underutilised movie website.

Motivation

A recommender system may now be found on practically every information-heavy website. When a consumer browses the target product on Amazon, for example, a list of probable preferred goods is given to them. Furthermore, when a user watches a video clip on YouTube, the system's recommender system proposes some related videos to the user based on previously created user behaviours. In a sense, recommender systems have fundamentally altered the way we get information.

As mentioned in, recommendation systems not only make it simpler and more convenient for individuals to acquire information, but they also have a high potential for economic growth. As more individuals recognise the relevance and potential of recommendation systems, the search for high-quality recommender systems has been an important issue in the community throughout the last decade. Fortunately, several recommendation systems systems have been created and deployed in a range of fields as a result of ongoing research in the field.

Based on this, a crucial challenge arises: how to know the performance of recommendation systems so that the best ones may be identified for use in certain situations or domains. The answer is to perform thorough and scientific evaluation tests on recommender systems. With the growing use of recommender systems in various applications, evaluating them has become increasingly important. It is common for an application designer to have to pick from a group of possible recommendation algorithms.

This objective can be met by comparing the efficiency of the algorithms in evaluation experiments. Furthermore, analysing recommender systems can assist researchers in selecting, tuning, and designing recommendation systems as a whole. This is because, while creating a recommender system, certain critical elements

determining the system's quality are frequently overlooked throughout the review process.

The project intends to investigate the most scientific technique for assessing recommender systems, motivated by the necessity of analysing recommendation systems and the emphasis on comprehensive metric considerations in assessment trials. The project's implementation is given in the form of a web - based application.

Problem

The retail industry is quickly evolving. Many physical businesses are being displaced by internet retailers, direct-to-consumer businesses, and subscription/ membership services. However, while the breadth of a website's variety attracts users, many online businesses fail to sell a large amount of their items.

This is frequently caused by a bad user browsing experience. Customers often spend hours looking through hundreds, if not thousands, of goods before finding anything they like.

In order to improve shopping experience that enhances sales and increases time spent on a website, shoppers must be given suggestions tailored to individual and wants.

Problem Statement

We confront a number of challenges while developing a recommendation system from start. There are now several recommendation systems based on user information, and what should we do if website does not have enough users?

Then we'll work on the representation of a movie, which is how a system understands a movie. That is the prerequisite for comparing the similarity of two films. Movie attributes such as genre, actor, and director can be used to classify films. However, distinct weights should be assigned to each feature of the film, and each should play a different part in recommendation.

So we got these inquiries:

1. How to propose movies if no user information is available.
2. What types of film features can be employed in the recommendation system?
3. How to determine the similarity between the two films.

4. Can you set a priority for each feature?

Goals

The objectives of this term paper project is to do research on recommendation systems and to identify an appropriate technique to apply it for this project. There are several types of recommendation systems, and not many of them are appropriate for every problem and condition.

My goal is to explore a novel method for enhancing movie categorisation, which is a necessity for developing content-based recommendation system.

Methodology

The first step in achieving the project's aim is to undertake sufficient background research, thus a literature review will be conducted. Because the entire project focuses on a large quantity of movie data, we used a quantitative research strategy.

Because the endeavour is experimental and testing in nature, positivism was chosen as the philosophical premise. As the progress of our study will be assessed by determining and testing a hypothesis, the research technique is deductive.

Overview

In recent years, recommender systems have become a prominent study issue. Many scholars proposed various recommendation methodologies.

The most well-known of these techniques are :

- Content-based filtering
- Collaborative filtering
- Knowledge based filtering
- Hybrid Method

Depending on whether a model is learned from the underlying data, recommender systems are further classified as follows:

- Memory-based
- Model-base Content-base Filtering

Content based filtering

Every user has a different method of choosing a product whether it is music, movie, or a project on commercial website. As the content-based filtering method is based on users utility, and every user has a different method of approaching a product. Let's take movies for example, every user has a different approach to choosing the next movie, leading actors, year of release, rating, genre or some of the data that many users rely on to choose the next movie. The content based recommended system is essentially a user specific filtering. It uses some features to evaluate user preferences (rating, dislikes, likes).

Limitation

A content-based filtering system takes users interaction and then recommends on basis of their interactions. But for a user which has not returned for a long time the recommendations might be very inaccurate. For an active user content based filtering approach is not applicable.

Collaborative Filtering

The collaborative filtering technique is a recommendation method which uses the user interest to recommend a product. In this method many different users data is being used to recommend similar items if the user interaction is similar for two different users. For example if a user **A** likes the romantic songs and there is a user **B** who also likes romantic songs but has different music library, the user **A** can be recommended the music from user **B**.

Limitation

Collaborative different approach depends on user data just like content-based approach. A very less popular product can be affected by these limitations. A new product is a good example as there is no data on that product.

Collaborative Filtering Vs Content-Based Filtering

Here is a list of differences between Content Based Filtering and Collaborative Filtering :

1. Rather than relying on user interactions and feedback, the content-based method necessitates a substantial quantity of knowledge on the features of products. They can be movie properties such as genre, date, filmmaker, actor, and so on, or article text that can be retrieved using Natural Language Processing. Collaborative Filtering, but at the other hand, requires nothing more than the user's previous preference on a set of things to suggest from, and since it is based on previous data, the key premise is that users who agreed in the past would generally agree in the future.

2. Domain information is not required as in case of Collaborative Filtering since the hidden layers are automatically learnt; but, in the case of a Content-based method, because the embedding of the items is custom to some degree, this approach requires a significant amount of domain knowledge to be provided with.

3. The collaborative filtering system will assist users in discovering new interests, and even if the ML system is unaware of the user's interest in a specific item, the model may still propose it since comparable people are interested in that thing. A Content-based model, on the other hand, can only generate suggestions based on the user's existing interests, and the model thus has limited power to build on the user's existing interests.

4. Because the suggestions are personal to a single user, a Content-Based filtering methodology does not require any data about other users. This makes scaling the same to a big number of consumers easy. Collaborative Filtering Methods cannot be spoken or done in the same way.

5. The collaborative algorithm recommends items based only on user behaviour, whereas Content-based filtering requires knowledge of both the user and the item's content.

Knowledge Based Filtering

And knowledge-based filtering method relies on the user input, the users share their location , language , and select many different choices when starting their profile . Which then helps to recommend products to a new user. Let's take a Music application for example, it asks user to select different genres, artist in the beginning of their account set up, this helps the platform to recommend products to that user from the beginning.

Hybrid Methods

Both content based and collaborative approaches address certain problems and it is reasonable to merge these two techniques. Utilising the user profiles to generate recommendations between two different users which is a collaborative filtering method when applied with content-based filtering removes the limitations of both

My ideology

I have chosen content based recommendation system because for the collaborative filtering method a big database of people searching for movies is required . Whereas in content based recommendation system , in this project only a name of a movie is required .

As for the content based recombination system only some important factors need to be considered , such as Title of the movie, genre, keywords, rating, actors, producers and directors . As the cast of the movies is also a big data to deal with.

I have only used the lead actors , Directors in this project . As for the producers they don't matter a lot in recommending movies as they handle the business and advertising side of the movie.

Materials and Method

The task, as the name infers, is worried about suggesting comparable films in view of client input. The undertaking accomplishes comparable film suggestions, which are very exact for such a little dataset.

The project makes use of the TMDB data set, which is a 5000 movie dataset that has been analysed.

Just the main metadata of each movie were examined, such as the title, director, actors, genre and language.

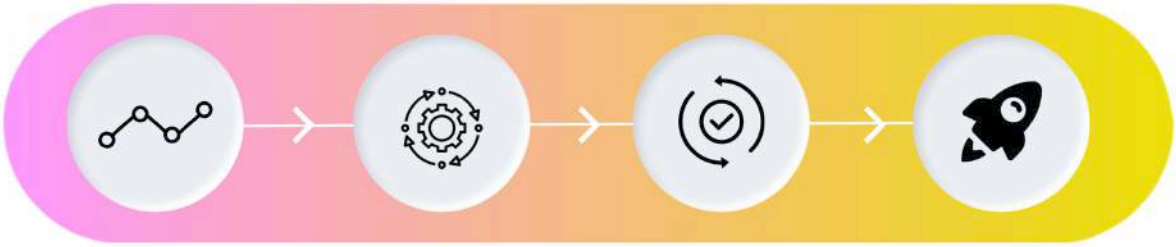
I didn't utilise a bigger dataset in light of the fact that the bigger the data collection, the more RAM is required to handle it, and since nearly everybody has a 8 GB RAM, this is a decent dataset to utilise.

Dataset

All of the film data i used came from TMDB database. Finally, we have 5000 movies and related information (movie cast). A movie may be defined from the standpoint of a recommendation system by a compilation of functionalities, which can include genres, stars, filmmakers, and so on.

- Director: The filmmaker is from IMDb; most films have just one director, but others have two or more.
- Actor: A movie usually features a number of stars, but most of them are worthless for the recommender system and have negative consequences. But we only have three key actors for a single film. They are also from TMDB.
- Release Year: The year the film was released, according to TMDB.
- Language: Language comes from TMDB and is the language used in the film.

Project Flow



Data

TMDB Dataset is being used for the 5000 movie database along with a dataset of casts from TMDB.

Pre Processing

Only important Data from these data sets will be processed.

Model

This project uses content-based filtering and the pre-processed data will be further used to complete the API.

Deploy

Finally the project will be deployed using streamlit and will be deployed as a web app locally.

Modeling

Load the scraped data and import all dependencies

```
In [1]: import numpy as np
        import pandas as pd

In [2]: movies = pd.read_csv('tmdb_5000_movies.csv')
        credits = pd.read_csv('tmdb_5000_credits.csv')

In [3]: movies = movies.merge(credits,on='title')
```

Combine the title, director, genre, and description into a single feature.

```
In [4]: movies.head()

In [5]: movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]

In [6]: movies.head()
Out[6]:
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

		{"id": 80, "nam...			comics"}, {"id": 853,...		Knight Rises	Attorney Harve...		
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"name": " Pictur

This is because our recommendation algorithm works not only by proposing films of a similar genre, but also by combining the cast, director, and narrative of the film.

AST stands for **Abstract Syntax Tree,** which is a potent tool of the Python programming language. It allows us to interact with the Python code itself and can modify it.

```
import ast
```

```
def convert(text):
    L = []
    for i in ast.literal_eval(text):
        L.append(i['name'])
    return L
```

Pandas dropna() method allows the user to analyse and drop Rows/Columns with Null values in different ways. The syntax for it is as follows :

DataFrameName.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)

```
movies.dropna(inplace=True)
```

```
movies['genres'] = movies['genres'].apply(convert)
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
movies['keywords'] = movies['keywords'].apply(convert)
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
import ast
ast.literal_eval('[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id
```

```
[{'id': 28, 'name': 'Action'},
 {'id': 12, 'name': 'Adventure'},
 {'id': 14, 'name': 'Fantasy'},
 {'id': 878, 'name': 'Science Fiction'}]
```

Storing the data after stemming in the same data set allows us to reduce the number of interferences and improves reusability.

```
def convert3(text):
    L = []
    counter = 0
    for i in ast.literal_eval(text):
        if counter < 3:
            L.append(i['name'])
        counter+=1
    return L
```

```
movies['cast'] = movies['cast'].apply(convert)
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weave...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley, ...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman, A...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton,...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
movies['cast'] = movies['cast'].apply(lambda x:x[0:3])
```

```
def fetch_director(text):
    L = []
    for i in ast.literal_eval(text):
        if i['job'] == 'Director':
            L.append(i['name'])
    return L
```

```
movies['crew'] = movies['crew'].apply(fetch_director)
```

```
#movies['overview'] = movies['overview'].apply(lambda x:x.split())
movies.sample(10)
```

	movie_id	title	overview	genres	keywords	cast	crew
2700	34152	Crooklyn	From Spike Lee comes this vibrant semi-autobio...	[Comedy, Drama]	[black people, 1970s, jazz musician, straßenki...	[Alfre Woodard, Delroy Lindo, David Patrick Ke...	[Spike Lee]
3984	55831	Boynton Beach Club	A handful of men and women of a certain age pi...	[Comedy, Drama, Romance]	[independent film]	[Brenda Vaccaro, Dyan Cannon, Joseph Bologna]	[]
1442	35169	Furry Vengeance	When real estate developer Dan Sanders finaliz...	[Comedy]	[bear, animal, aftercreditsstinger, duringcred...	[Brendan Fraser, Ken Jeong, Brooke Shields]	[Roger Kumble]
1293	256040	Baahubali: The Beginning	The young Shivudu is left as a foundling in a ...	[Action, Adventure, War, History]	[kingdom, war, bollywood, medieval india, anci...	[Prabhas, Rana Daggubati, Tamanna Bhatia]	[S.S. Rajamouli]
1748	254473	Brick Mansions	In a dystopian Detroit, grand houses that once...	[Action, Crime, Drama]	[martial arts, atomic bomb, ghetto, parkour, c...	[Paul Walker, David Belle, RZA]	[Camille Delamarre]
4473	838	American Graffiti	A couple of high school graduates spend one fi...	[Comedy, Drama]	[farewell, rock and roll, robbery, love at fir...	[Richard Dreyfuss, Ron Howard, Paul Le Mat]	[George Lucas]
2526	873	The Color Purple	An epic tale spanning forty years in the life ...	[Drama]	[prison, africa, southern usa, rape, black peo...	[Whoopi Goldberg, Margaret Avery, Danny Glover]	[Steven Spielberg]
2531	33217	Diary of a Wimpy Kid	Greg Heffley is headed for big things, but fir...	[Comedy, Family]	[based on novel, coming of age, young boy, bre...	[Zachary Gordon, Robert Capron, Rachael Harris]	[Thor Freudenthal]
4389	39183	Once in a Lifetime: The Extraordinary Story of...	In the 1970s the North American Soccer League ...	[Documentary]	[new york, beckenbauer, pele]	[Pelé, Franz Beckenbauer, Carlos Alberto]	[Paul Crowder, John Dower]

```
def collapse(L):
    L1 = []
    for i in L:
        L1.append(i.replace(" ",""))
    return L1
```

```
movies['cast'] = movies['cast'].apply(collapse)
movies['crew'] = movies['crew'].apply(collapse)
movies['genres'] = movies['genres'].apply(collapse)
movies['keywords'] = movies['keywords'].apply(collapse)
```

```
movies.head()
```

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

```
movies['overview'] = movies['overview'].apply(lambda x:x.split())
```

```
movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords'] + movies['cast'] + movies['crew']
```

```
new = movies.drop(columns=['overview','genres','keywords','cast','crew'])
#new.head()
```

```
new['tags'] = new['tags'].apply(lambda x: " ".join(x))
new.head()
```

	movie_id	title	tags
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...
2	206647	Spectre	A cryptic message from Bond's past sends him o...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

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

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Input In [26], in <cell line: 1>()
----> 1 from sklearn.feature_extraction.text import CountVectorizer
      2 cv = CountVectorizer(max_features=5000,stop_words='english')

ModuleNotFoundError: No module named 'sklearn'
```

```
vector = cv.fit_transform(new['tags']).toarray()
```

```
vector.shape
```

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

```
similarity = cosine_similarity(vector)
```

```
similarity
```

```
def recommend(movie):
    index = new[new['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])
    for i in distances[1:11]:
        print(new.iloc[i[0]].title)
```

```
recommend('Batman')
```

```
import pickle
```

```
pickle.dump(new,open('movie_list.pkl','wb'))
pickle.dump(similarity,open('similarity.pkl','wb'))
```

```
pickle.dump(new,open('movie_list.pkl','wb'))
new['title'].values
pickle.dump(new.to_dict,open('movie_dict.pkl','wb'))
```

Designing the Front End

We can now begin developing a web application now that we have finalised the model.

Streamlit is used in the frontend of our model. Streamlit is a Python open-source package that makes it simple to create stunning bespoke web applications for ML and data research.

```
pip install streamlit
```

There are several decent streamlit tutorials available, such as this one. So I won't go into depth about all of my methods. In summary, I wanted to create a frontend that takes user input and suggests a list of comparable movies.

First we import dependancies, some of the dependancies need to be downloaded manually. In this part I have added a custom font from google fonts website , and implemented it in this website.

We then fetch posters from a already available api by TMDB .

```
def fetch_poster(movie_id):
    url = "https://api.themoviedb.org/3/movie/{}?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US".format(
        movie_id)
    data = requests.get(url)
    data = data.json()
    poster_path = data['poster_path']
    full_path = "https://image.tmdb.org/t/p/w500/" + poster_path
    return full_path


streamlit_style = """
            <style>
            @import url('https://fonts.google.com/specimen/Oswald');
https://fonts.google.com/specimen/Oswald
            html, body, [class*="css"]  {
            font-family: 'Roboto', sans-serif;

            }
            </style>
            """
st.markdown(streamlit_style, unsafe_allow_html=True)
```

Here we declare a design of rows and columns of the movie title and posters on the web app.

```
def recommend(movie):
    index = movies[movies['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    recommended_movie_names = []
    recommended_movie_posters = []
    for i in distances[1:13]:
        # fetch the movie poster
        movie_id = movies.iloc[i[0]].movie_id
        recommended_movie_posters.append(fetch_poster(movie_id))
        recommended_movie_names.append(movies.iloc[i[0]].title)

    return recommended_movie_names, recommended_movie_posters
```

Here we created a title of the page and imported the movie list and the similarity objects in the project.

```
st.markdown("<h1 style='text-align: center; '>Movie Recommender System</h1>", unsafe_allow_html=True)
movies = pickle.load(open('movie_list.pkl', 'rb'))
similarity = pickle.load(open('similarity.pkl', 'rb'))

movie_list = movies['title'].values
selected_movie = st.selectbox(
    "Type or select a movie from the dropdown",
    movie_list
)
```

Here we create the rows and columns with the particular index of movies list .

```
if st.button('Show Recommendation'):
    recommended_movie_names, recommended_movie_posters = recommend(selected_movie)
    col1, col2, col3, col4 = st.columns(4)
    with col1:
        st.subheader(recommended_movie_names[0])
        st.image(recommended_movie_posters[0])
    with col2:
        st.subheader(recommended_movie_names[1])
        st.image(recommended_movie_posters[1])

    with col3:
        st.subheader(recommended_movie_names[2])
        st.image(recommended_movie_posters[2])
    with col4:
        st.subheader(recommended_movie_names[3])
        st.image(recommended_movie_posters[3])
```

Here I have created 4 columns and 3 rows for this project for better visual appeal.

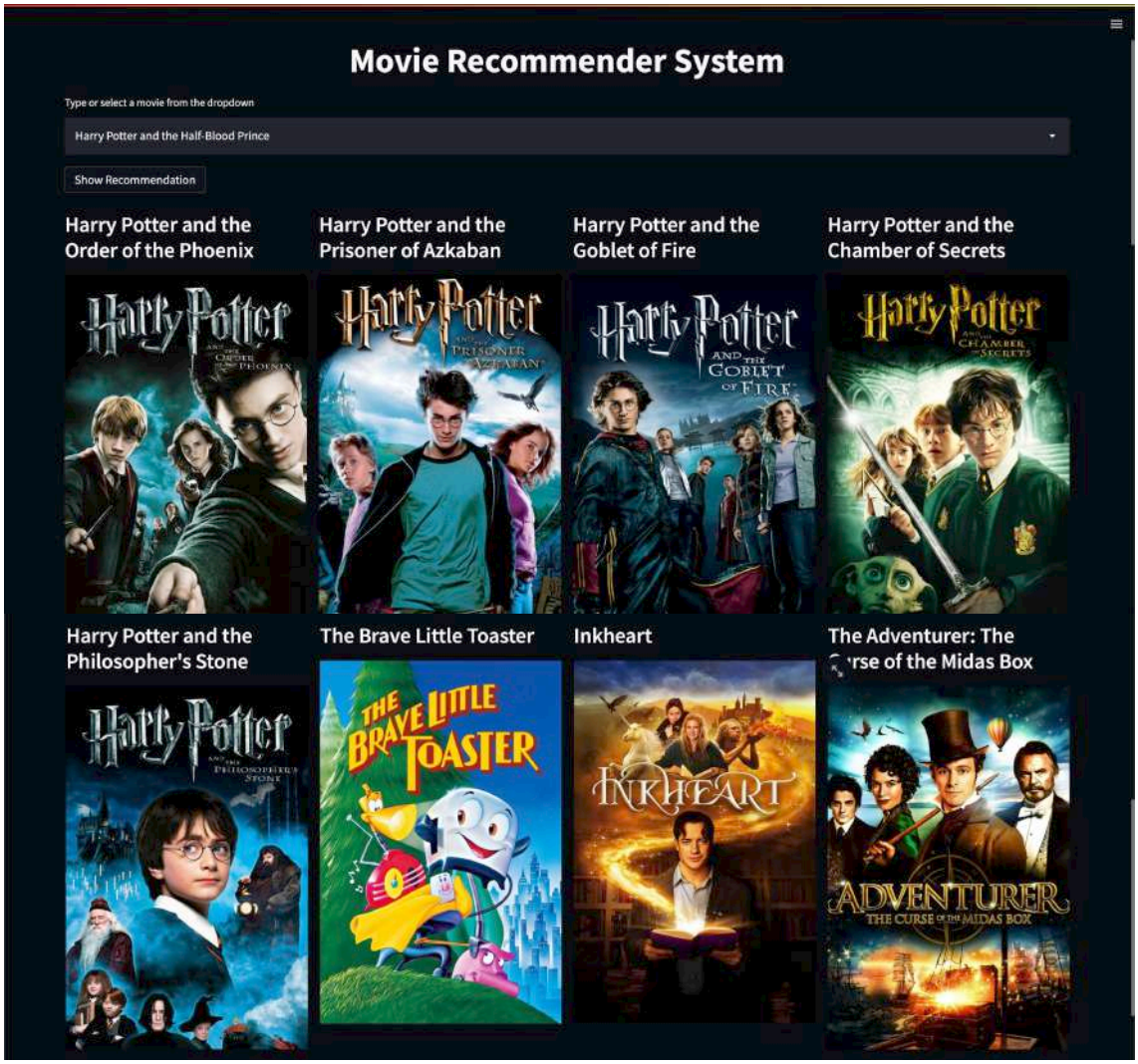
```
    col5, col6, col7, col8 = st.columns(4)
    with col5:
        st.subheader(recommended_movie_names[4])
        st.image(recommended_movie_posters[4])

    with col6:
        st.subheader(recommended_movie_names[5])
        st.image(recommended_movie_posters[5])
    with col7:
        st.subheader(recommended_movie_names[6])
        st.image(recommended_movie_posters[6])

    with col8:
        st.subheader(recommended_movie_names[7])
        st.image(recommended_movie_posters[7])
```

```
col9, col10, col11, col12 = st.columns(4)
with col9:
    st.subheader(recommended_movie_names[8])
    st.image(recommended_movie_posters[8])
with col10:
    st.subheader(recommended_movie_names[9])
    st.image(recommended_movie_posters[9])
with col11:
    st.subheader(recommended_movie_names[10])
    st.image(recommended_movie_posters[10])
with col12:
    st.subheader(recommended_movie_names[11])
    st.image(recommended_movie_posters[11])
```

Result - *The final product*

I have chosen to deploy this project locally. All the posters and titles are retrieved from the internet. As for the movie select list, it is stored locally.

Here in this example I have chosen *Harry Potter and the Half-Blood Prince* , and the results show almost every Harry Potter Movie in the series (the api here finds the genre, actors , directors and the keywords very similar so it is recommending these movies)



Conclusions & Recommendations

Because of information overload, recommender systems have become increasingly vital. Specially in the past two years when we all were facing the pandemic and almost every task was to be held online, this has given us the technology which we were not ready to use yet, millions of people had first time in their life utilised a smart device, recommenders systems have been very vital from the beginning of internet businesses and it will always be important in e-commercial applications.

We try to create a novel technique to increase the accuracy of the representation of the movie for the content-based recommender system in particular.

To address the issues raised before, we first employ a content-based recommender algorithm, which eliminates the need for a cold start.

Cold start is a possible issue in machine data systems that include some automated data modelling. It is specifically concerned with the issue of the system being unable to draw any conclusions for users or products for which it still has not acquired adequate information.

This was a really small project which explores recommended systems, recommendation system is a really vast study. New ways to utilise recommended systems are researched and implemented on different platforms.

Big companies like YouTube, Amazon, Flipkart, Google, Netflix depend on recommenders systems, the more the efficiency of recommenders systems the more beneficial it is for these companies to provide best content for the end user.

Take for example YouTube which was using content based recommended system before the pandemic hit, and now it is using hybrid recommended system method so it can provide the best recommendations to the user.

Implications for future research

For many years, the recommender system has evolved, and it has never reached a low point. The advancement of ML, large scale networks, and high speed computers in recent years has furled new developments in this sector. In future work, we will explore the following points.

- Make use of the collaborative filtering recommendation.

After gathering sufficient user information, collaborative filtering recommendations will be implemented. Collaborative filtering is dependent on user social information, which will be investigated more in the future.

- Introduce more exact and appropriate movie features.

The rating is typically used instead of object characteristics in collaborative filtering recommendations. In the future, we should extract movie elements like colour and subtitles to offer a more appropriate description of the movie.

- Create a list of movies that the user dislikes.

In recommender systems, user data is always helpful. In the future, we will collect additional user data and provide a list of movies that users detest. We will also enter the list of disliked movies into the recommendation system and create ratings that'll be added to the prior result. We can enhance the recommender system's results in this manner.

- Implement machine learning.

In the future, variable parameters will be included into the recommender system, and we will utilise machine learning to automatically alter the weight of each characteristic and identify the most appropriate weights.

- Create an internal service for the recommender system.

The recommender system will no longer be an external website that is only used for testing in the future. We will implement it as an inside API for developers to use. Some of the website's movie listings will be organised by suggestion.

Appendices

This project was mainly based on content-based filtering, but as I have explained in the overview section that there are a total of four types of recommendation systems, namely content-based recommenders systems, collaborative filtering recommender system, knowledge-based recommended system, and hybrid recommended system.

There are multiple platforms which provide deep understanding and learning of recommenders systems here are some of the best recommenders systems which I have gone through to understand this project :

- Machine Learning Mastery
- Free Recommender Systems Specialisation course on Coursera
- Machine Learning advance course on google developers

Referring to these three completely free courses helped me understand this project , it will surely be beneficial for a beginner to start with these courses .

References

1. *Wikipedia - https://en.wikipedia.org/wiki/Recommender_system*
2. Free Recommender Systems Specialisation course on Coursera
3. Medium article on end to end movie recommender system - https://medium.com/geekculture/end-to-end-movie-recommendation-system-49b29a8b57ac
4. Kaggle - https://www.kaggle.com/tmdb/tmdb-movie-metadata?select=tmdb_5000_movies.csv
5. CampusX YouTube channel