

PROJECT REPORT FOR
RECOMMENDATION SYSTEM

As a project work for Course

PYTHON PROGRAMMING (INT 213)

NAME: VINEET SINGH

REG. NO: 12010670

ROLL: RK20SDA03

PROGRAM: B. Tech CSE

COLLEGE: SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING

UNIVERSITY: LOVELY PROFESSIONAL UNIVERSITY



L OVELY
P ROFESSIONAL
U NIVERSITY

Project: A Movie Recommendation System

Abstract:

Recommendation system is a tool that helps users to find content and overcome information overload. It predicts interests of users and makes recommendation according to interest model of users.

The original content-based recommender system is the continuation and development of collaborative filtering, which doesn't need the user's evaluation for items. Instead, the similarity is calculated based on the information of items that are chose by users, and then make the recommendation accordingly. With the improvement of machine learning, current content-based recommender system can build profile for users and products respectively. Building or updating the profile according to the analysis of items that are bought or visited by users. The system can compare the user and the profile of items and then recommend the most similar products. So this recommender method that compare user and product directly cannot be brought into collaborative filtering model. The foundation of content-based algorithm is acquisition and quantitative analysis of the content. As the research of acquisition and filtering of text information are mature, many current content-based recommender systems make recommendation according to the analysis of text information.

Functionalities of this project:

- It can recommend movies to the user based on the content.
- Consists of database of more than thousands of movies to recommend.
- It can recommend movies based on the reviews, cast, crew and directors as well.
- It can recommend as many movies as the user likes.
- Reliable data used in this system to ensure proper recommendations.

Introduction

A recommender system is a simple algorithm whose aim is to provide the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the user the items that they would rate highly. An example of recommendation in action is when you visit Amazon and you notice that some items are being recommended to you or when Netflix recommends certain movies to you. They are also used by Music streaming applications such as Spotify and Deezer to recommend music that you might like.

Different types of Recommendation Systems:

The most common types of recommendation systems are **content-based** and **collaborative filtering** recommender systems. In collaborative filtering, the behaviour of a group of users is used to make recommendations to other users. The recommendation is based on the preference

of other users. A simple example would be recommending a movie to a user based on the fact that their friend liked the movie. There are two types of collaborative models **Memory-based** methods and **Model-based** methods. The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are divided into two categories:

- **User-based collaborative filtering:** In this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user. For example, if Derrick and Dennis like the same movies and a new movie come out that Derrick like, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the same movies.
- **Item-based collaborative filtering:** These systems identify similar items based on users' previous ratings. For example, if users A, B, and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B, and C.

Content-based systems use metadata such as genre, producer, and actor, musician to recommend items say movies or music. Such a recommendation would be for instance recommending Infinity War that featured Vin Diesel because someone watched and liked The Fate of the Furious. Similarly, you can get music recommendations from certain artists because you liked their music. Content-based systems are based on the idea that if you liked a certain item you are most likely to like something that is similar to it.

Datasets used in this project:

In this project, we used the Movie Lens dataset. This dataset was put together by the Group lens research group at the University of Minnesota. It contains 1, 10, and 20 million ratings. Movie lens also has a website where you can sign up, contribute reviews and get movie recommendations.

Libraries used in this project:

In this movie recommendation system we used the following libraries:

- **Pandas**
Pandas is a software library written for the Python programming language for data manipulation and analysis.
- **Numpy**
This is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Scikit-Learn**

It features various classification, regression and clustering algorithms that are designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Below is the screenshot of imported libraries:

```

In [ ]: #Description: Building a movie recommendation system using python

In [4]: #Creating Libraries
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer

In [5]: #Loading the data
df = pd.read_csv('C:/Users/vinee/Desktop/recommendation_system/movie-data.csv', low_memory = False)

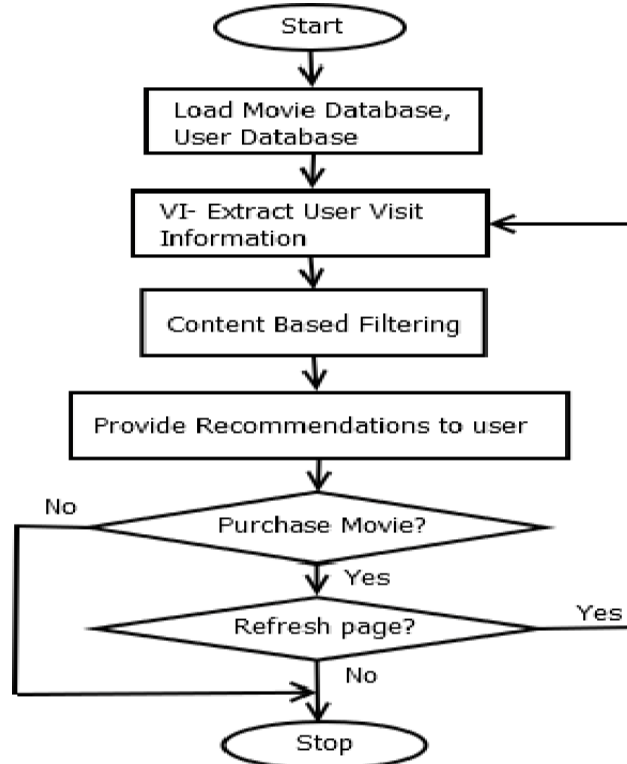
In [6]: #Viewing the data
df.head(2)

```

Out[6]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0

Flowchart of the project:



Screenshots of code and Output:

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: #Description: Building a movie recommendation system using python
In [1]: #Creating libraries
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer
In [2]: #Loading the data
df = pd.read_csv('C:/Users/vinee/Desktop/recommendation_system/movie-data.csv', low_memory = False)
In [3]: #Viewing the data
df.head(2)
Out[3]:
```

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [4]: #Providing movie id to the dataset
df['Movie_id'] = range(0,1000)
In [5]: df.head(3)
Out[5]:
```

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore	Movie_id
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0	0
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0	1
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	62.0	2

```
In [6]: #Checking the number of rows/movies and number of columns
df.shape
```

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Out[6]: (1000, 13)
In [7]: #Creating list of important columns in the dataset
imp_columns = ['Actors', 'Director', 'Genre', 'Title']
In [8]: df[imp_columns].head(2)
Out[8]:
```

	Actors	Director	Genre	Title
0	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	James Gunn	Action,Adventure,Sci-Fi	Guardians of the Galaxy
1	Noomi Rapace, Logan Marshall-Green, Michael Fa...	Ridley Scott	Adventure,Mystery,Sci-Fi	Prometheus

```
In [9]: #Checking if any column has null values
df[imp_columns].isnull().values.any()
Out[9]: False
In [10]: #Creating a function to combine important columns in a single string
def get_important_features(data):
    important_features = []
    for i in range(0, data.shape[0]):
        important_features.append(data['Actors'][i]+' '+data['Director'][i]+' '+data['Genre'][i]+' '+data['Title'][i])
    return important_features
```

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [11]: #Creating a column to hold the combined strings
df['important_features'] = get_important_features(df)
#Viewing the above data
df.head()

Out[11]:
Rank Title Genre Description Director Actors Year Runtime (Minutes) Rating Votes Revenue (Millions) Metascore Movie_id in
0 1 Guardians of the Galaxy Action,Adventure,Sci-Fi A group of intergalactic criminals are forced ... James Gunn Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S... 2014 121 8.1 757074 333.13 76.0 0
1 2 Prometheus Adventure,Mystery,Sci-Fi Following clues to the origin of mankind, a te... Ridley Scott Noomi Rapace, Logan Marshall-Green, Michael Fa... 2012 124 7.0 485820 126.46 65.0 1
2 3 Split Horror,Thriller Three girls are kidnapped by a man with a diag... M. Night Shyamalan James McAvoy, Anya Taylor-Joy, Haley Lu Richar... 2016 117 7.3 157606 138.12 62.0 2
In a city of humanoid Christopher McConaughey, Reese ... 7.5 157606 138.12 62.0 2

In [12]: #Converting the text to a matrix using CountVectorizer
cm = CountVectorizer().fit_transform(df['important_features'])

In [13]: #Getting the cosine similarity matrix and then printing it
cs = cosine_similarity(cm)

In [14]: #Checking the number of rows in cosine similarity matrix
cs.shape

Out[14]: (1000, 1000)

In [15]: #Getting the title of movie that the user Likes
title = 'Ant-Man'
#Finding the movie_id
movie_id = df[df.Title == title]['Movie_id'].values[0]
#Creating a list of enum for similarities of the above movie
scores = list(enumerate(cs[movie_id]))

In [16]: #Sorting the above formed List
sorted_scores = sorted(scores, key = lambda x:x[1], reverse = True)
sorted_scores = sorted_scores[1:] #So that the first movie's name does not get repeated

In [ ]: #Printing the sorted scores
print(sorted_scores)
```

Conclusion

Do we need recommendation systems?

Yes, we definitely need recommendation systems in today's world. On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services. This paper explores the different characteristics and potentials of different prediction techniques in recommendation systems in order to serve as a compass for research and practice in the field of recommendation systems.