MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Movie Recommendation System

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by:

Ashutosh Soni (0901IT213D03)

Vishal Arya (0901IT213D06)

Minor Project-I – 160516

Faculty Mentor:

Dr. Dhananjay Bisen, Assistant Professor, Department of Information Technology, MITS

Submitted to:

DEPARTMENT OF INFORMATION TECHNOLOGY
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005

July-December 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Ashutosh Soni (0901IT213D03)** and **Vishal Arya (0901IT213D06)** has submitted the project report titled **Movie Recommendation System** under the mentorship of **Dr. Dhananjay Bisen**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Information Technology** from Madhav Institute of Technology and Science, Gwalior.

Dr. Dhananjay Bisen
Assistant Professor
Information Technology

Dr. Akhilesh Tiwari
Assistant Professor and Head,
Department of IT

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Information Technology at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Dhananjay Bisen**, **Assistant Professor**, **Department of Information Technology, MITS**.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Date: 22/11/2022
Place: Gwalior

Ashutosh Soni (0901IT213D03)
Vishal Arya (0901IT213D06)
III Year
Information Technology

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Information Technology,** for allowing me to explore this project. I humbly thank **Dr. Akhilesh Tiwari**, Assistant Professor and Head, Department of Information Technology, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Dr. Dhananjay Bisen**, **Assistant Professor**, **Department of Information Technology**, **MITS** for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Ashutosh Soni (0901IT213D03)
Vishal Arya (0901IT213D06)
III Year
Information Technology

TABLE OF CONTENTS

ABSTRACT

A recommendation system is a system that, depending on some data, makes suggestions to users for specific resources like books, movies, songs, etc. The characteristics of previously liked movies are typically used by movie recommendation algorithms to anticipate what movies a user will enjoy. Such recommendation systems are advantageous for businesses that gather data from a lot of clients and want to successfully offer the finest recommendations.

When creating a movie recommendation system, many variables can be taken into account, including the movie's genre, cast, and even director. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movies dataset. The language used in this project is Python.

SUMMARY

Recommender system is a kind of information filtering system try to predict user preferences and to do recommendations based on those preferences. There are many recommended apps systems. They have become increasingly popular over the past few years. old and now used in most of the online platforms we use. The content of these platforms is diverse from movies, music, books and videos, for friends and stories on social media platforms, to products on e-commerce sites, for people on professional sites and dating sites, to the search results returned on Google.

Often these systems can collect information about user’s choices and can use this information to improve their recommendations Future. For example, Facebook can track your interactions with various stories on your feed to find out what kind stories that captivate you. Sometimes recommender systems can make improvements based on the operations of a large number Everyone. For example, if Amazon finds that a large number of customers who buy the latest Apple MacBook also buy a USB-C-to USB adapter, they can recommend the adapter to new users just added a MacBook to their cart.

Chapter 1: PROJECT OVERVIEW

The recommendation system analyses the past preferences of the user concerned, and then it uses this information to try to find similar movies. This information is available in the database (e.g., lead actors, director, genre, etc.). After that, the system provides movie recommendations for the user.

1.1 Introduction:

A recommendation system or recommendation engine is a model used to information filtering where it tries to predict user preferences and provide recommendations based on those preferences. These systems are becoming more and more popular today and widely used today in fields such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about users' preferences and behaviour and then use that information to improve their recommendations in the future.

Many companies use recommender systems to increase user interaction and enrich the user's shopping experience. recommendation system has a number of benefits, the most important being customer satisfaction and revenue. Movie recommendation system is a very powerful and important system.

1.2 Problem Statement:

The project's objective is to suggest a movie to the user. Providing customers of online service providers with related content chosen from relevant and irrelevant collections of objects.

1.3Objectives and Scope:

The goal of this project is to provide accurate movie recommendations for user. The purpose of the project is to improve the quality of the film proposed system, such as system accuracy, quality and scalability than pure methods. This is done using the combined method by combining Content-based filtering and collaborative filtering, to eliminate overloads data, the recommendation system is used as a tool to filter information in social networking sites. Therefore, there is a huge scope to explore in this area. scope improves film scalability, accuracy and quality very powerful movie recommender system and important systems. But, because of matters related to purity cooperative approach, the movie recommendation system is also poor quality and scalability issues of the recommendations.

CHAPTER 2: SYSTEM REQUIREMENTS SPECIFICATION

2.1 Software Requirement:

- Text Editor (VS-Code/Jupyter)
- Python
- Data Set
- Python libraries
    - Pandas
    - Numpy
    - Difflib
    - TfidfVectorizer
    - cosine_similarity

2.2 Hardware Requirement:

- A PC with Windows/Linux OS
- Processor with 1.7-2.4gHz speed
- Minimum of 4gb RAM

CHAPTER 3: TECHNOLOGIES AND FLOW CHART

Our system should meet the following minimum specifications: OS – Windows, Mac OS etc. We have used Python and some of its libraries like Pandas, NumPy, Difflib, TfidVectorizer (to convert textual data into meaningful numerical values) and Cosin Similarity (for finding highest similarity score). It is implemented on Jupyter Notebook, VS Code & Terminal.

3.1 Technologies Used:

- **Python**

  Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

- **Libraries**
  - **Pandas**

    Pandas is an open-source library designed primarily for working quickly and logically with relational or labelled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library. Pandas is quick and offers its users exceptional performance & productivity.

  - **NumPy**

    NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

- **Difflib**

  This module provides classes and functions for comparing sequences. It can be used for example, for comparing files, and can produce information about file differences in various formats, including HTML and context and unified diffs. For comparing directories and files

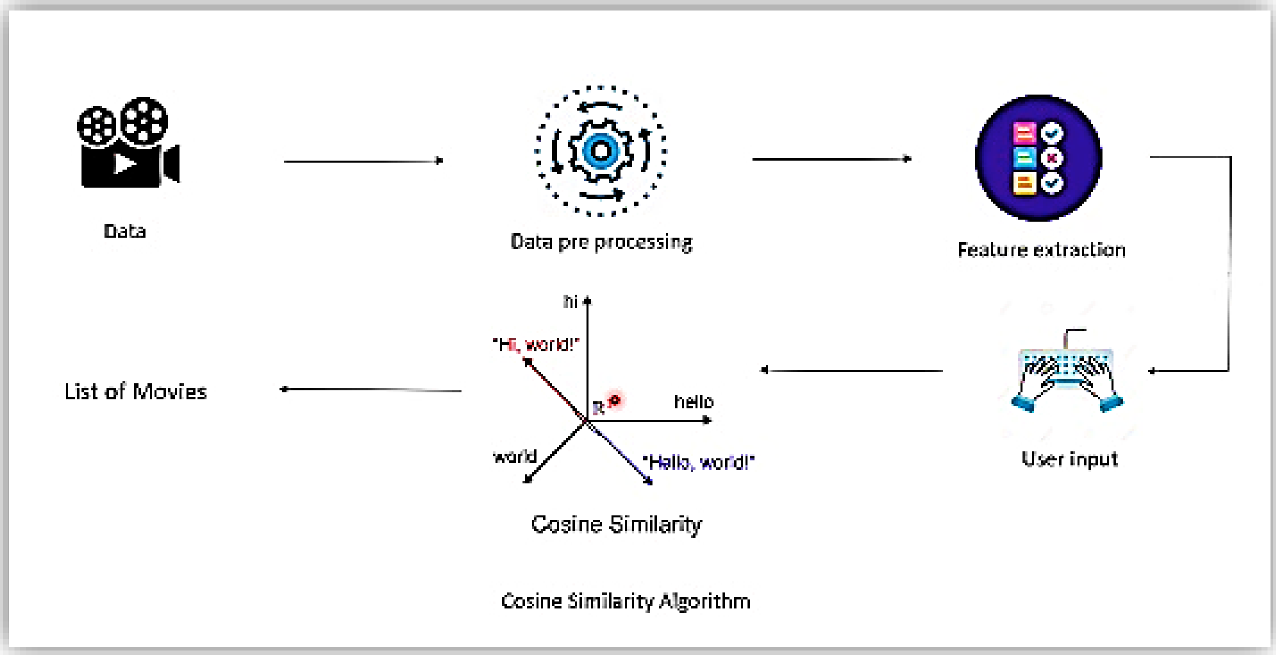
- **TfidVectorizer**

  Tf-idf, or term frequency-inverse document frequency, is a technical word. It is a mathematical statistic designed to show how important a word is to a given record in a corpus or collection. The more often a term appears in a document, the higher it is regarded by the tf-idf. In order to account for the fact that a few terms appear more frequently overall, it is balanced by the number of documents in the corpus that contain the word.

- **Cosin Similarity**

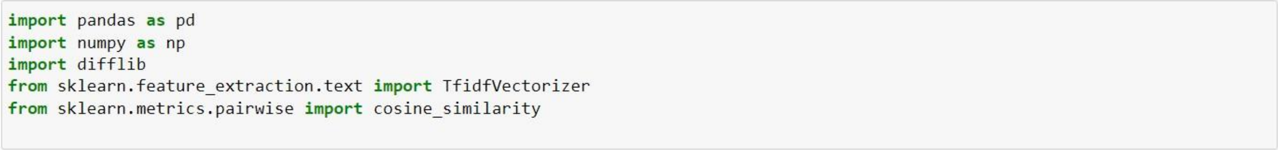
  Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far 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.

3.2 Flow Chart:

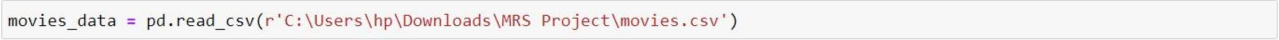


CHAPTER 4: SOURCE CODE AND OUTPUTS

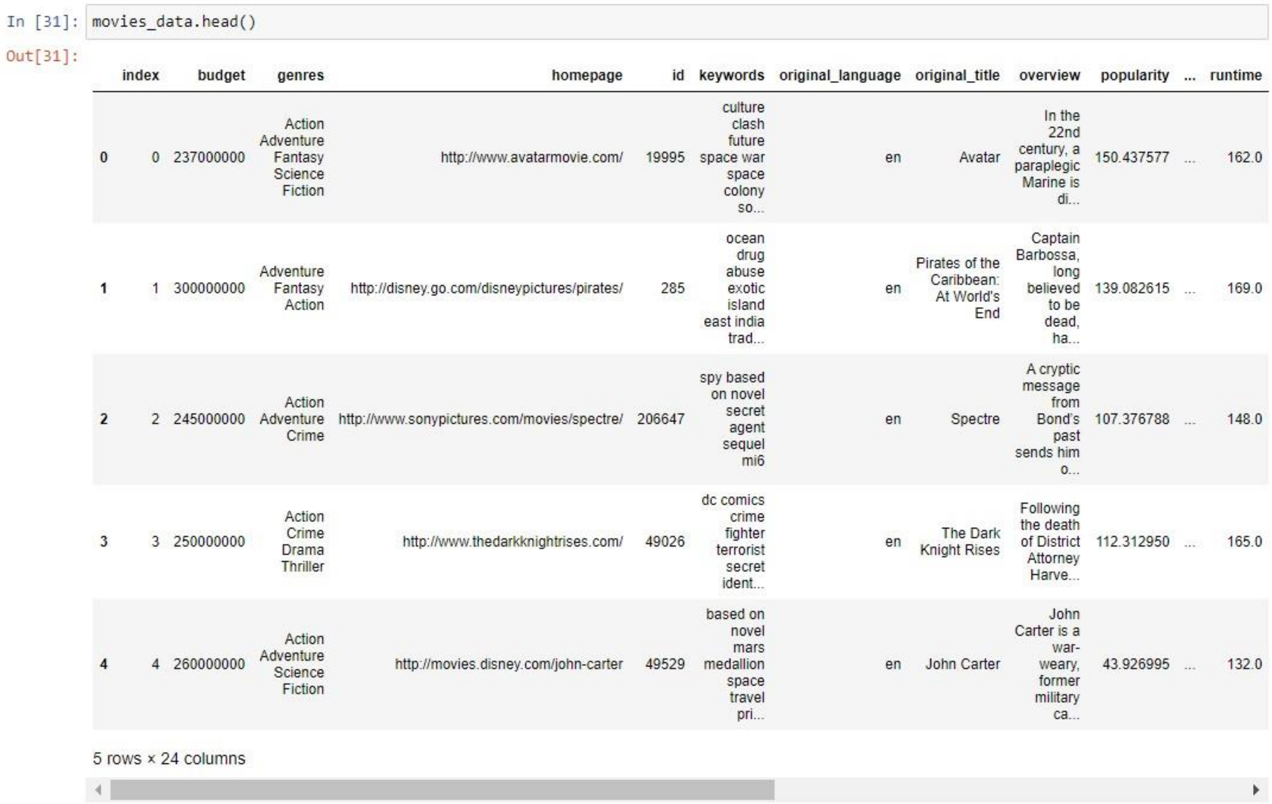
4.1 Source Code



```
import pandas as pd
import numpy as np
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```



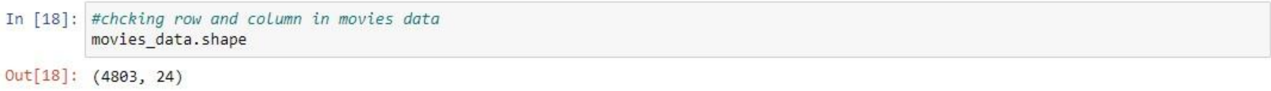
```
movies_data = pd.read_csv(r'C:\Users\hp\Downloads\MRS Project\movies.csv')
```



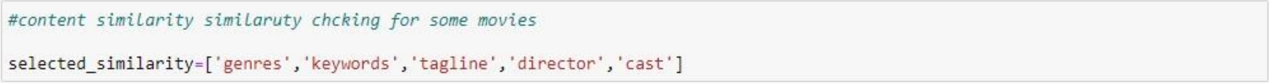
```
In [31]: movies_data.head()
Out[31]:
```

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...	runtime
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	culture clash future space war space colony so...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	...	162.0
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	ocean drug abuse exotic island east india trad...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	...	169.0
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy based on novel secret agent sequel mi6	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	...	148.0
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026	dc comics crime fighter terrorist secret ident...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	...	165.0
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49529	based on novel mars medallion space travel pri...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	...	132.0

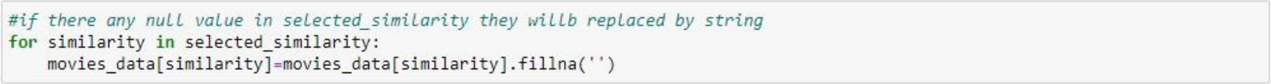
5 rows × 24 columns



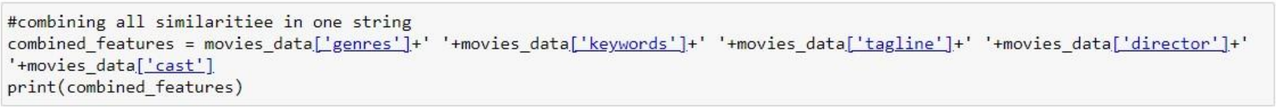
```
In [18]: #chcking row and column in movies data
         movies_data.shape

Out[18]: (4803, 24)
```



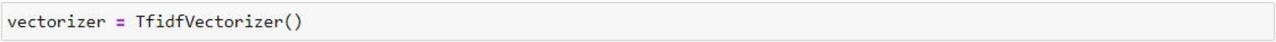
```
#content similarity similaruty chcking for some movies

selected_similarity=['genres','keywords','tagline','director','cast']
```



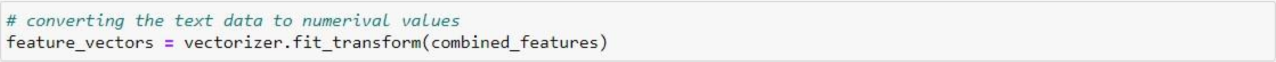
```
#if there any null value in selected_similarity they willb replaced by string
for similarity in selected_similarity:
    movies_data[similarity]=movies_data[similarity].fillna('')
```



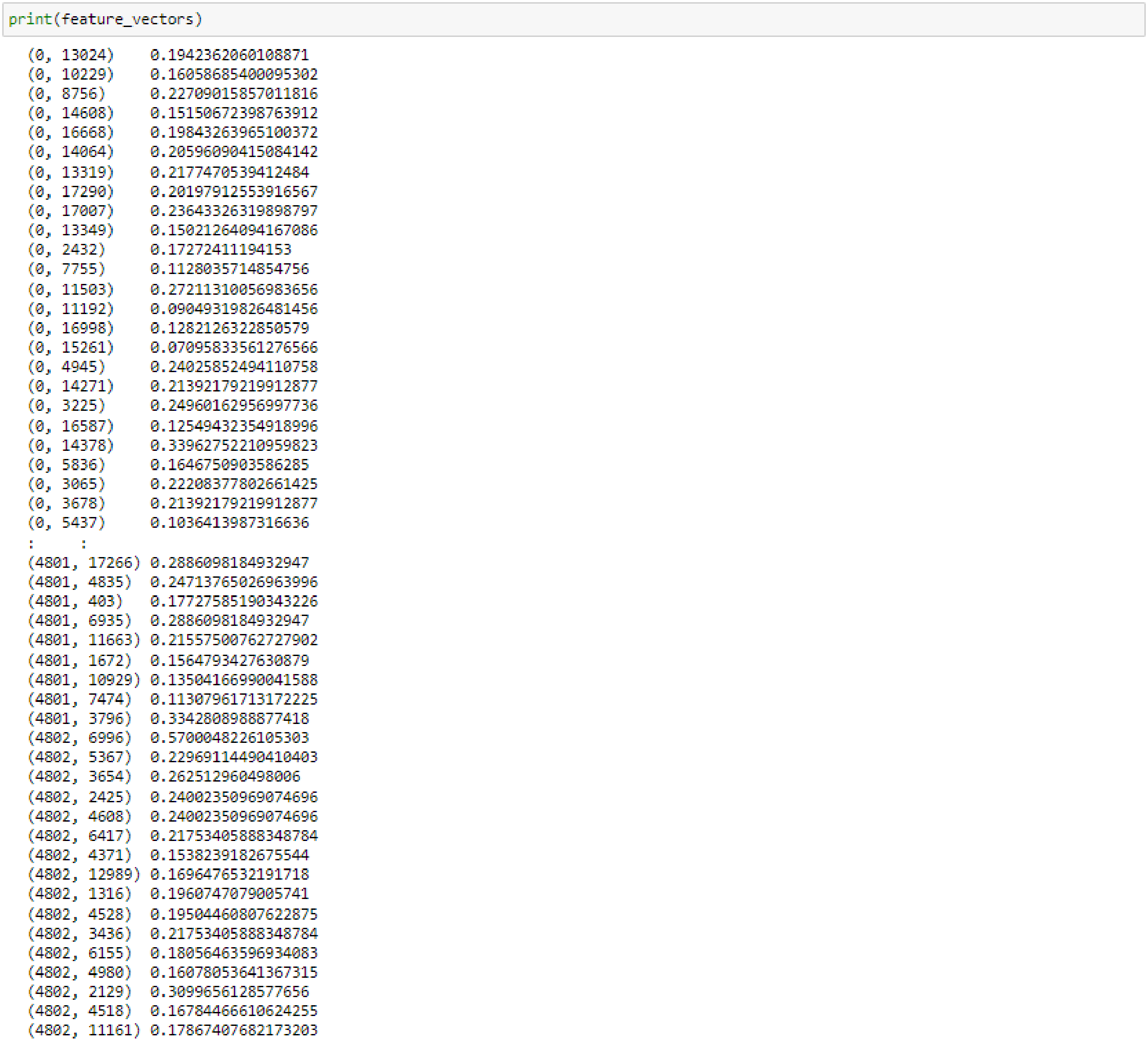
```
#combining all similaritiee in one string
combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['director']+'
'+movies_data['cast']
print(combined_features)
```



```
vectorizer = TfidfVectorizer()
```

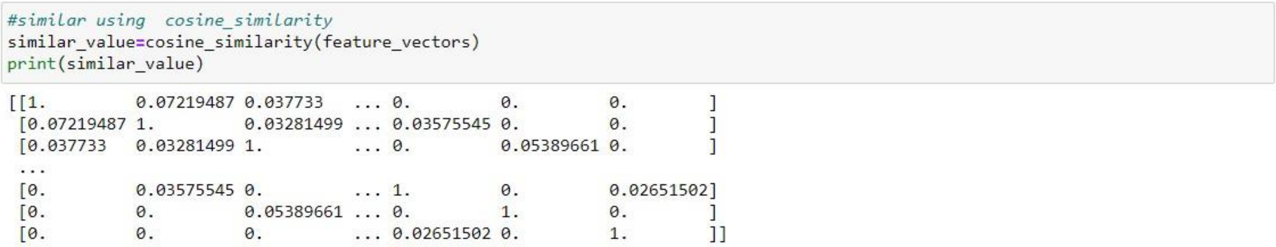


```
# converting the text data to numerival values
feature_vectors = vectorizer.fit_transform(combined_features)
```



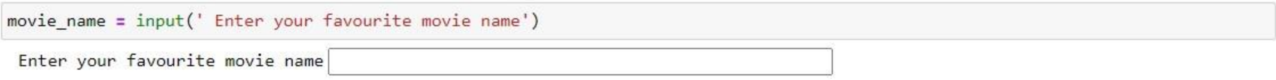
```
print(feature_vectors)
```

```
  (0, 13024)	0.1942362060108871
  (0, 10229)	0.16058685400095302
  (0, 8756)	0.22709015857011816
  (0, 14608)	0.15150672398763912
  (0, 16668)	0.19843263965100372
  (0, 14064)	0.20596090415084142
  (0, 13319)	0.2177470539412484
  (0, 17290)	0.20197912553916567
  (0, 17007)	0.23643326319898797
  (0, 13349)	0.15021264094167086
  (0, 2432)	0.17272411194153
  (0, 7755)	0.1128035714854756
  (0, 11503)	0.27211310056983656
  (0, 11192)	0.09049319826481456
  (0, 16998)	0.1282126322850579
  (0, 15261)	0.07095833561276566
  (0, 4945)	0.24025852494110758
  (0, 14271)	0.21392179219912877
  (0, 3225)	0.24960162956997736
  (0, 16587)	0.12549432354918996
  (0, 14378)	0.33962752210959823
  (0, 5836)	0.1646750903586285
  (0, 3065)	0.22208377802661425
  (0, 3678)	0.21392179219912877
  (0, 5437)	0.1036413987316636
  :	:
  (4801, 17266)	0.2886098184932947
  (4801, 4835)	0.24713765026963996
  (4801, 403)	0.17727585190343226
  (4801, 6935)	0.2886098184932947
  (4801, 11663)	0.21557500762727902
  (4801, 1672)	0.1564793427630879
  (4801, 10929)	0.13504166990041588
  (4801, 7474)	0.11307961713172225
  (4801, 3796)	0.3342808988877418
  (4802, 6996)	0.5700048226105303
  (4802, 5367)	0.22969114490410403
  (4802, 3654)	0.262512960498006
  (4802, 2425)	0.24002350969074696
  (4802, 4608)	0.24002350969074696
  (4802, 6417)	0.21753405888348784
  (4802, 4371)	0.1538239182675544
  (4802, 12989)	0.1696476532191718
  (4802, 1316)	0.1960747079005741
  (4802, 4528)	0.19504460807622875
  (4802, 3436)	0.21753405888348784
  (4802, 6155)	0.18056463596934083
  (4802, 4980)	0.16078053641367315
  (4802, 2129)	0.3099656128577656
  (4802, 4518)	0.16784466610624255
  (4802, 11161)	0.17867407682173203
```



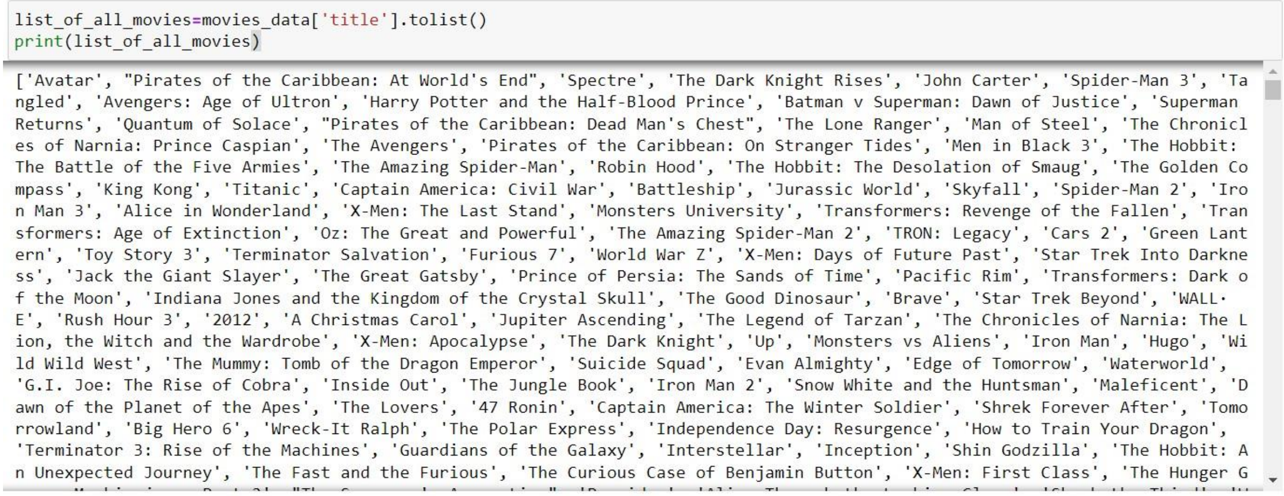
```
#similar using  cosine_similarity
similar_value=cosine_similarity(feature_vectors)
print(similar_value)
```

```
[[1.         0.07219487 0.037733   ... 0.         0.         0.        ]
 [0.07219487 1.         0.03281499 ... 0.03575545 0.         0.        ]
 [0.037733   0.03281499 1.         ... 0.         0.05389661 0.        ]
 ...
 [0.         0.03575545 0.         ... 1.         0.         0.02651502]
 [0.         0.         0.05389661 ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.02651502 0.         1.        ]]
```



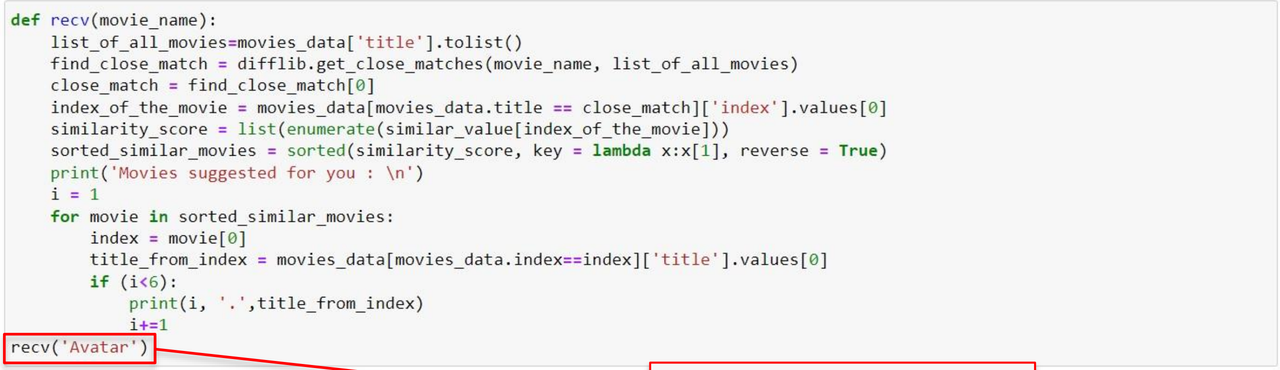
```
movie_name = input(' Enter your favourite movie name')
```

```
 Enter your favourite movie name
```



```
list_of_all_movies=movies_data['title'].tolist()
print(list_of_all_movies)
```

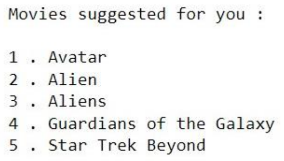
```
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'Ta
ngled', 'Avengers: Age of Ultron', 'Harry Potter and the Half-Blood Prince', 'Batman v Superman: Dawn of Justice', 'Superman
Returns', 'Quantum of Solace', "Pirates of the Caribbean: Dead Man's Chest", 'The Lone Ranger', 'Man of Steel', 'The Chronicl
es of Narnia: Prince Caspian', 'The Avengers', 'Pirates of the Caribbean: On Stranger Tides', 'Men in Black 3', 'The Hobbit:
The Battle of the Five Armies', 'The Amazing Spider-Man', 'Robin Hood', 'The Hobbit: The Desolation of Smaug', 'The Golden Co
mpass', 'King Kong', 'Titanic', 'Captain America: Civil War', 'Battleship', 'Jurassic World', 'Skyfall', 'Spider-Man 2', 'Iro
n Man 3', 'Alice in Wonderland', 'X-Men: The Last Stand', 'Monsters University', 'Transformers: Revenge of the Fallen', 'Tran
sformers: Age of Extinction', 'Oz: The Great and Powerful', 'The Amazing Spider-Man 2', 'TRON: Legacy', 'Cars 2', 'Green Lant
ern', 'Toy Story 3', 'Terminator Salvation', 'Furious 7', 'World War Z', 'X-Men: Days of Future Past', 'Star Trek Into Darkne
ss', 'Jack the Giant Slayer', 'The Great Gatsby', 'Prince of Persia: The Sands of Time', 'Pacific Rim', 'Transformers: Dark o
f the Moon', 'Indiana Jones and the Kingdom of the Crystal Skull', 'The Good Dinosaur', 'Brave', 'Star Trek Beyond', 'WALL·
E', 'Rush Hour 3', '2012', 'A Christmas Carol', 'Jupiter Ascending', 'The Legend of Tarzan', 'The Chronicles of Narnia: The L
ion, the Witch and the Wardrobe', 'X-Men: Apocalypse', 'The Dark Knight', 'Up', 'Monsters vs Aliens', 'Iron Man', 'Hugo', 'Wi
ld Wild West', 'The Mummy: Tomb of the Dragon Emperor', 'Suicide Squad', 'Evan Almighty', 'Edge of Tomorrow', 'Waterworld',
'G.I. Joe: The Rise of Cobra', 'Inside Out', 'The Jungle Book', 'Iron Man 2', 'Snow White and the Huntsman', 'Maleficent', 'D
awn of the Planet of the Apes', 'The Lovers', '47 Ronin', 'Captain America: The Winter Soldier', 'Shrek Forever After', 'Tomo
rrowland', 'Big Hero 6', 'Wreck-It Ralph', 'The Polar Express', 'Independence Day: Resurgence', 'How to Train Your Dragon',
'Terminator 3: Rise of the Machines', 'Guardians of the Galaxy', 'Interstellar', 'Inception', 'Shin Godzilla', 'The Hobbit: A
n Unexpected Journey', 'The Fast and the Furious', 'The Curious Case of Benjamin Button', 'X-Men: First Class', 'The Hunger G
```



```
def recv(movie_name):
    list_of_all_movies=movies_data['title'].tolist()
    find_close_match = difflib.get_close_matches(movie_name, list_of_all_movies)
    close_match = find_close_match[0]
    index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
    similarity_score = list(enumerate(similar_value[index_of_the_movie]))
    sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
    print('Movies suggested for you : \n')
    i = 1
    for movie in sorted_similar_movies:
        index = movie[0]
        title_from_index = movies_data[movies_data.index==index]['title'].values[0]
        if (i<6):
            print(i, '.',title_from_index)
            i+=1
recv('Avatar')
```

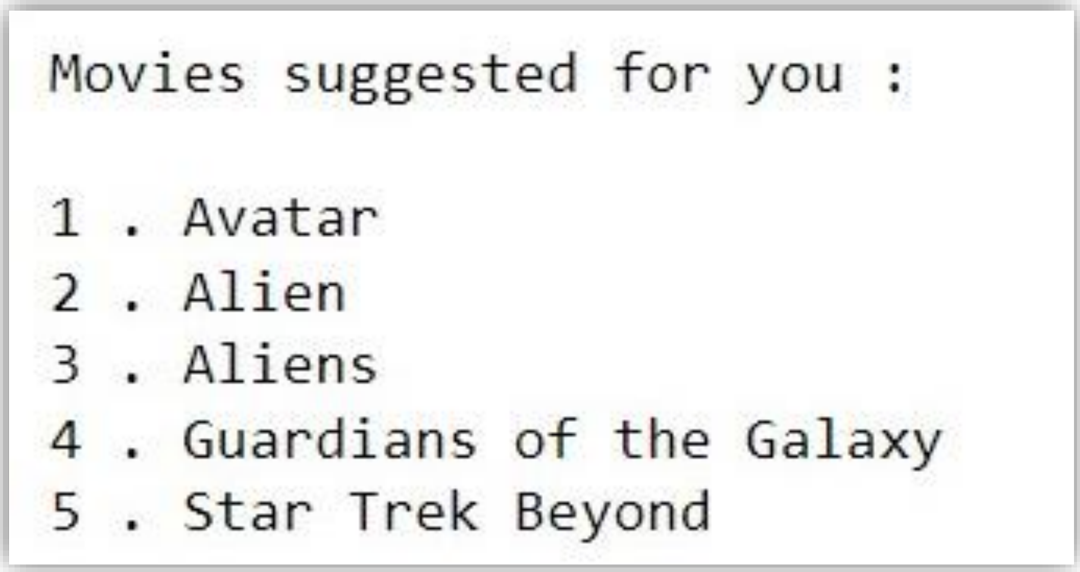
```
recv('Avatar')
```

recv(‘ < To search for any movie > ’)



```
Movies suggested for you :

1 . Avatar
2 . Alien
3 . Aliens
4 . Guardians of the Galaxy
5 . Star Trek Beyond
```

4.2 Output:



```
Movies suggested for you :

1 . Avatar
2 . Alien
3 . Aliens
4 . Guardians of the Galaxy
5 . Star Trek Beyond
```

CONCLUSION

In this project, to improve the accuracy, quality and scalability of movie recommendation system, we use content based filtering, using some python libraries (Pandas, NumPy, Difflib, TfidVectorizer) and Cosine Similarity is presented in the proposed methodology. Existing pure approaches is implemented on a Movie dataset and the results are compared among them. Comparative results depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches. Also, computing time of the proposed approach is lesser than the other approaches.

REFERENCES

- **YouTube** (https://www.youtube.com/watch?v=7rEagFH9tQg&t=2180s)\
- **Dataset - movies.csv** (https://drive.google.com/file/d/1cCkwiVv4mgfl20ntgY3n4yApcWqqZQe6/view)

*****