Movie Recommendation System

A COURSE PROJECT REPORT

By

RIYANSH DAGAR (RA2011027010070)

DIVYANSH GAUR (RA2011027010090)

RISHABH SHAH (RA2011027010133)

Under the guidance of

Ms.D.Hemavathi

Assistant professor

Department of Data Science and Business Systems

In partial fulfillment for the Course

of

18CSE396T – DATA SCIENCE

in

Department of Data Science and Business Systems

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

It is Certified that this mini project report "**Movie Recommendation System**" is the bonafide work of **Riyansh Dagar (RA2011027010070), Divyansh Gaur (RA2011027010090)** and **Rishabh Shah (RA2011027010133)** who carried out the project work under my supervision.

Ms.D.Hemavathi
Assistant professor
Department of Data Science and Business Systems SRM
institute of science and technology

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our Registrar **Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our Dean of College of Engineering and Technology, **Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to the Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to Course Audit Professor **Dr. Annapurani Panaiyappan**, Professor and Head, Department of Networking and Communications and Course Coordinators for their constant encouragement and support.

We are highly thankful to our course project faculty **Ms.D.Hemavathi**, Assistant Professor , Department of DSBS for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our HoD, Professor **Dr. M. Lakshmi ,** Department of DSBS and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project

INDEX

1. ABSTRACT

This report is concerned with a movie recommendation system. It is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies.

The classification analysis that deals with the training and test dataset and modelling using logistic regression.

A recommendation system is used for the purpose of suggesting items to purchase or to see. They direct users towards those items which can meet their needs through cutting down large database of Information.

A recommender system, or a recommendation system (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

2. INTRODUCTION

The evolution of technology brings us many advanced platforms such as Machine Learning, Deep Learning, Data Mining, the Internet of Things (IoT), etc. To satisfy the need of society, almost in each work, we use this technology.

Recommendation systems have become well known nowadays, be it in the field of entertainment, education, etc. Earlier, the users needed to settle on choices on what books to purchase, what music to tune in to, what motion pictures to watch and so on.

With the improvement of recommendation systems, the customers will have no agony in settling on choices and organizations can keep up their client gathering and draw in new clients by improving users' satisfaction

Additionally, nowadays the modern technologies like machine learning and deep learning also plays a vital role in the process flexible technologies for day to day operations.

3. PROBLEM STATEMENT

This recommendation system recommends different movies to users. Since this system is based on a collaborative approach, it will give progressively explicit outcomes contrasted with different systems that are based on the content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users' ratings, hence limiting your choice to explore more. While our system which is based on a collaborative approach computes the connection between different clients and relying upon their ratings, prescribes movies to others who have similar tastes, subsequently allowing users to explore more.

4. LITERATURE REVIEW

Kumar et al. [29] proposed MOVREC, a movie recommendation system based on collaborative filtering approaches. Collaborative filtering takes the data from all the users and based on that generates recommendations. A hybrid system has been presented by Virk et al. [30]. This system combines both collaborative and content-based method. De Campos et al. [34] also made an analysis of both the traditional recommendation techniques. As both of these techniques have certain setbacks, he proposed another system which is a combination of Bayesian network and collaborative technique. Kużelewska [35] proposed clustering as an approach to handle the recommendations. Two methods for clustering were analyzed: Centroid-based solution and memory-based methods. The result was that accurate recommendations were generated. Chiru et al. [27] proposed Movie Recommender, a system that uses the user's history in order to generate recommendations. Sharma and Maan [36] in their paper analyzed various techniques used for recommendations, collaborative, hybrid and content-based recommendations. Also, it describes the pros and cons of these approaches. Li and Yamada [37] proposed an inductive learning algorithm. Here a tree had been built which shows the user recommendation. Some of the major contribution in recommendation system is discussed in Table 1.

Table 1. Literature review of recommendation systems.

Authors	Year	Descriptions
Scharf & Alley [38]	1993	The authors proposed a flexible multicomponent rate recommendation system to predict the optimum rate of fertilizer for winter wheat.
Basu et al. [39]	1998	The authors proposed an approach to the recommendation that can exploit both ratings and content information.
Sarwar et al. [40]	2001	The authors proposed various techniques for computing item-item similarities.
Bomhardt [41]	2004	The author proposed an approach for a personal recommendation of news.
Manikrao & Prabhakar [42]	2005	The authors presented the design of a dynamic web selection framework.
Von Reischach et al. [43]	2009	The authors proposed a rating concept that allows users to generate rating criteria.
Choi et al. [44]	2012	The authors proposed approaches for integrating various techniques for improving the recommendation quality.

Table 2. Literature review of filtering techniques.

Authors	Year	Descriptions
Goldberg et al. [45]	1992	The authors introduced the collaborative filtering technique.
Herlocker et al. [46]	1997	Authors applied filtering techniques to Usenet news.
Miyahara & Pazzani [47]	2000	The authors introduced an approach to calculate the similarity between a user from negative ratings to positive ratings separately.
Hofmann [48]	2004	The author introduced a new-family of model-based algorithms.
Dabov et al. [49]	2008	The authors proposed an image restoration technique using collaborative filtering.
Pennock et al. [50]	2013	The authors proposed various approaches for filtering by personality diagnosis.
Liu et al. [51]	2014	The authors introduced a new method to provide an accurate recommendation.

5.OBJECTIVE

KNN algorithm is called the K nearest neighbor algorithm. The center thought of this algorithm is if most of the k most comparable neighbors of the test in the component space have a place with a specific class, at that point the example is considered to have a place with this category. Most of w's nearest neighbors have a place with the x class, w has a place with the X classification.

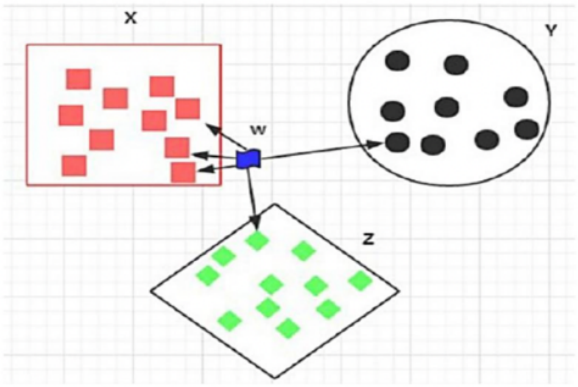
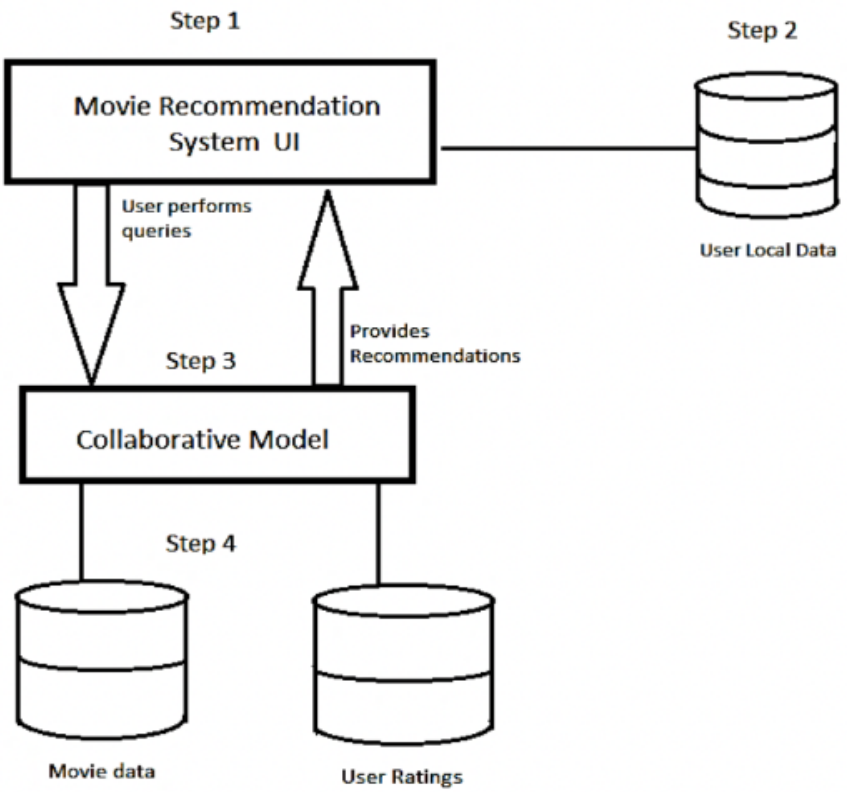


Figure 1. K nearest algorithm [33].

Calculating the similarity
This similarity value is going to play an eminent role in the collaborative filtering technique to select trustworthy users from the given set of users. Hence, they give us a method to increase or decrease the significance of a particular user or item. At present, we are using adjusted cosine similarity for calculation of similar items.

Choosing a Neighbourhood
In this technique, the neighbors that we are going to use as a part of prediction also makes an influence on the recommendations that are going to be generated. Consequently, this determination of neighbors must be accomplished more cautiously to not influence the nature of suggestions created. Hence, we will be choosing the most related neighbors who have the highest match compared to others. So, this value must be chosen more delicately.

Predicting Undetermined Ratings
In this for the user, to whom we want to predict those movies for which he hasn't rated, should be predicted using similar weights that are calculated in the previous steps.


Step 1
Step 2
Movie Recommendation System UI
User Local Data
User performs queries
Provides Recommendations
Step 3
Collaborative Model
Step 4
Movie data
User Ratings

6. SOFTWARE PACKAGES

Following software packages has been imported for machine learning and training the model:-

1. **Language Platform:** Python 3.5.2

2. **Python – Pandas:** Pandas is a Python package focused on reading, writing and manipulating data. There are a variety of required and recommended pandas dependencies which has to installed prior
to installation of python-pandas

3. **Python-NumPy:** Python dependencies can be installed using pip repository. Numpy can also be installed using python-pip. NumPy is a python package for scientific computing. It provides
functions to handle N-dimentional array objects and tools to integrate C++ code with random number capabilities.

4. **from sklearn.model_selection import train_test_split:**

Split arrays or matrices into random train and test subsets. Quick utility that wraps input validation and next (ShuffleSplit().split(X, Y)) and application to input data into a single call for splitting (and optionally subsampling) data in a one liner.

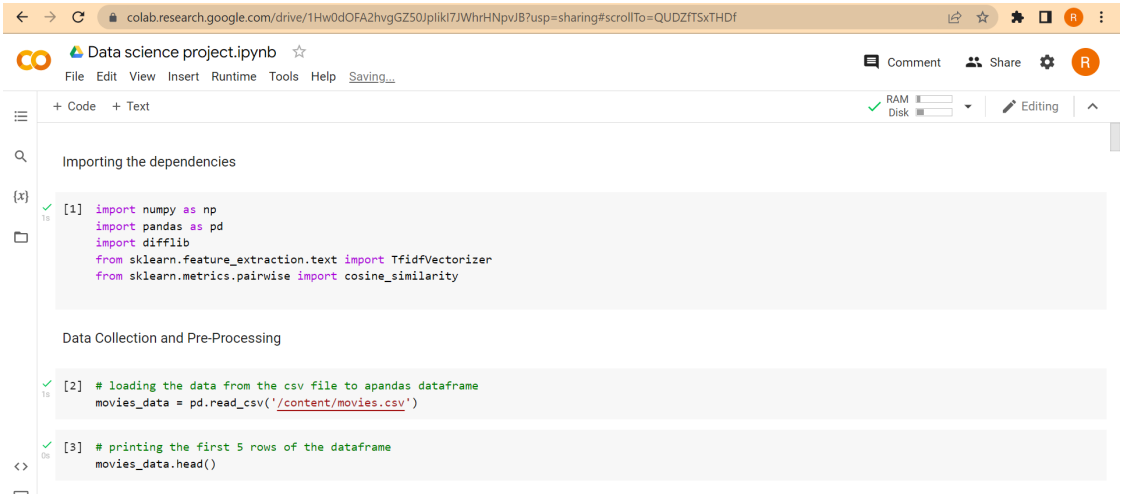
5. from sklearn.linear_model import LogisticRegression:

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'.

6. from sklearn.metrics import accuracy_score:

In multilabel classification, this function computes subset accuracy, the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.

7. CODE



Data science project.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

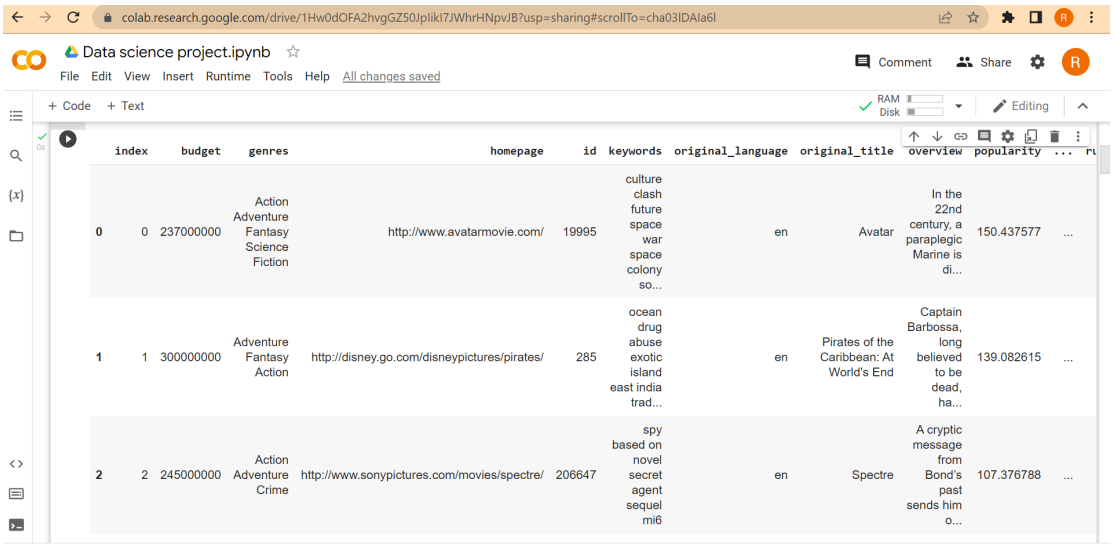
Importing the dependencies

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

Data Collection and Pre-Processing

```
[2] # loading the data from the csv file to apandas dataframe
    movies_data = pd.read_csv('/content/movies.csv')
```

```
[3] # printing the first 5 rows of the dataframe
    movies_data.head()
```



Data science project.ipynb

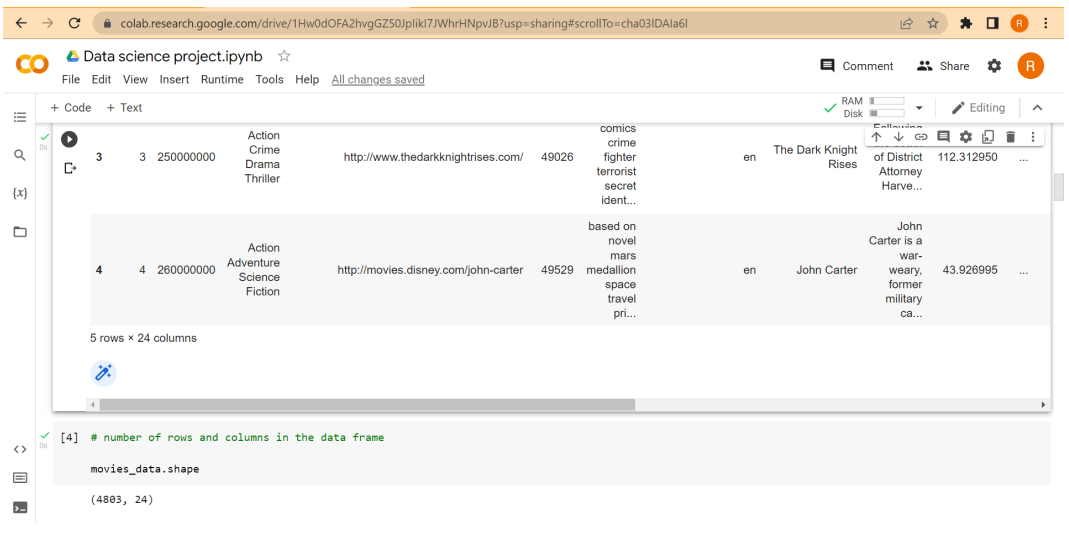
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...	r
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	...	
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	...	
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	...	



Data science project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026	comics crime fighter terrorist secret ident...	en	The Dark Knight Rises	... of District Attorney Harve...	112.312950	...
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	...

5 rows × 24 columns

```
[4] # number of rows and columns in the data frame
    movies_data.shape
```

```
(4803, 24)
```

Data science project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[5]  # selecting the relevant features for recommendation

     selected_features = ['genres','keywords','tagline','cast','director']
     print(selected_features)
```

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

```
[6]  # replacing the null valuess with null string

     for feature in selected_features:
       movies_data[feature] = movies_data[feature].fillna('')
```

```
[7]  # combining all the 5 selected features

     combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['cast']+' '+movies_data['director']
```

```
[8]  print(combined_features)
```

```
0       Action Adventure Fantasy Science Fiction cultu...
1       Adventure Fantasy Action ocean drug abuse exot...
2       Action Adventure Crime spy based on novel secr...
3       Action Crime Drama Thriller dc comics crime fi...
4       Action Adventure Science Fiction based on nove...
                              ...
4798    Action Crime Thriller united states\u2013mexic...
4799    Comedy Romance  A newlywed couple's honeymoon ...
4800    Comedy Drama Romance TV Movie date love at fir...
4801      A New Yorker in Shanghai Daniel Henney Eliza...
4802    Documentary obsession camcorder crush dream gi...
Length: 4803, dtype: object
```

```
[9]  # converting the text data to feature vectors

     vectorizer = TfidfVectorizer()
```

Data science project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[10] feature_vectors = vectorizer.fit_transform(combined_features)
```

```
[11] print(feature_vectors)
```

```
  (0, 2432)     0.17272411194153
  (0, 7755)     0.1128035714854756
  (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, 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
```

colab.research.google.com/drive/1Hw0dOFA2hvgGZ50JpIikI7JWhrHNpvJB?usp=sharing#scrollTo=qNbeeMSnMb2F

Data science project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

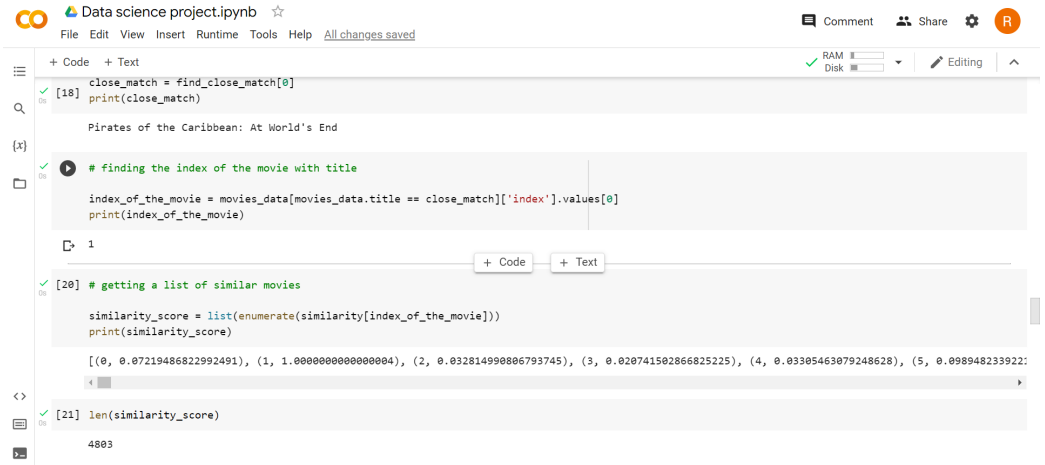
+ Code + Text

RAM Disk Editing

```
  :     :
(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
```


colab.research.google.com/drive/1Hw0dOFA2hvgGZ50JplikI7JWhrHNpvJB?usp=sharing#scrollTo=qNbeeMSnMb2F
Data science project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM Disk
Editing
Cosine Similarity
[12] # getting the similarity scores using cosine similarity
similarity = cosine_similarity(feature_vectors)
print(similarity)
[[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.]]
[14] print(similarity.shape)
(4803, 4803)

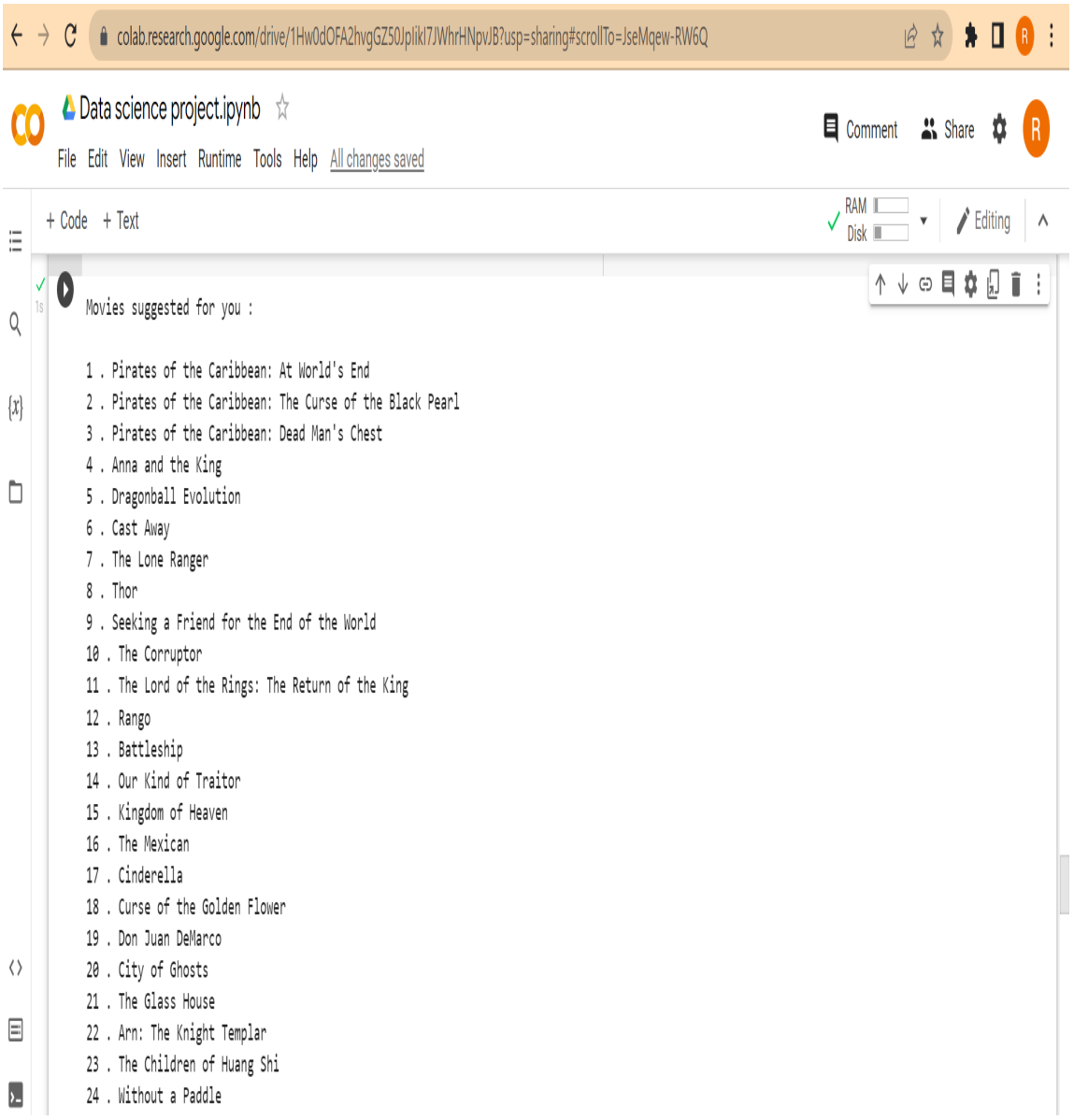

Data science project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM Disk
Editing
Getting the movie name from the user
# getting the movie name from the user
movie_name = input(' Enter your favourite movie name : ')
Enter your favourite movie name : Pirates of the carribean
[16] # creating a list with all the movie names given in the dataset
list_of_all_titles = movies_data['title'].tolist()
print(list_of_all_titles)
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'Tangled', 'Avengers: Age of
[17] # finding the close match for the movie name given by the user
find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
print(find_close_match)
["Pirates of the Caribbean: At World's End", 'Rise of the Guardians', "Pirates of the Caribbean: Dead Man's Chest"]


Data science project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM Disk
Editing
close_match = find_close_match[0]
[18] print(close_match)
Pirates of the Caribbean: At World's End
# finding the index of the movie with title
index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
print(index_of_the_movie)
1
+ Code + Text
[20] # getting a list of similar movies
similarity_score = list(enumerate(similarity[index_of_the_movie]))
print(similarity_score)
[(0, 0.07219486822992491), (1, 1.0000000000000004), (2, 0.032814990806793745), (3, 0.020741502866825225), (4, 0.03305463079248628), (5, 0.098948233922
[21] len(similarity_score)
4803


colab.research.google.com/drive/1Hw0dOFA2hvgGZ50JplikI7JWhrHNpvJB?usp=sharing#scrollTo=bhSWJgqgNDo7
Data science project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM Disk
Editing
[23] # print the name of similar movies based on the index
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<30):
print(i, '.',title_from_index)
i+=1

8. CONCLUSION AND RESULTS

OUTPUT



24 . Without a Paddle

25 . Bulletproof Monk

26 . Love Actually

27 . Noah

28 . Crouching Tiger, Hidden Dragon

29 . Star Trek II: The Wrath of Khan

Enter your favourite movie name : John carter
Movies suggested for you :

1 . John Carter
2 . Heaven is for Real
3 . Alien
4 . The Specials
5 . The Helix... Loaded
6 . Finding Nemo
7 . Transformers
8 . Mission to Mars
9 . The Astronaut's Wife
10 . American Psycho
11 . Max
12 . The English Patient
13 . The Last Temptation of Christ
14 . Enter Nowhere
15 . The Martian
16 . Notes on a Scandal
17 . Sideways
18 . Spider-Man 3
19 . Daddy's Home
20 . We Bought a Zoo

21 . George of the Jungle
22 . Treasure Planet
23 . Don McKay
24 . Auto Focus
25 . Savages
26 . The Covenant
27 . X-Men Origins: Wolverine
28 . Daybreakers
29 . Gravity

CONCLUSION

This recommendation system recommends different movies to users. Since this system is based on a collaborative approach, it will give progressively explicit outcomes contrasted with different systems that are based on the content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users' ratings, hence limiting your choice to explore more. While our system which is based on a collaborative approach computes the connection between different clients and relying upon their ratings, prescribes movies to others who have similar tastes, subsequently allowing users to explore more. It is a web application that allows users to rate movies as well as recommends them appropriate movies based on other's ratings.

9. REFERENCES

Mohapatra, H., Panda, S., Rath, A., Edalatpanah, S., & Kumar, R. (2020). A tutorial on powershell pipeline and its loopholes. International journal of emerging trends in engineering research, 8(4), 975-982.

Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). A Pythagorean fuzzy approach to the transportation problem. Complex & intelligent systems, 5(2), 255-263.

Smarandache, F., & Broumi, S. (Eds.). (2019). Neutrosophic graph theory and algorithms. Engineering Science Reference. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019).

https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109

https://colab.research.google.com/