A PROJECT REPORT

ON

Recommending Movie with Data Analytics, Machine Learning and AI using Python

Submitted in partial fulfillment for the requirement of the award of

TRAINING

IN

Data Analytics, Machine Learning and AI using Python

Submitted By

ANUSHKA SHARMA(Vellore Institute of Technology Chennai, Chennai**)**

Under the guidance of

Mr. Bipul Shahi

ACKNOWLEDGEMENT

My sincere gratitude and thanks towards my project paper guide **Bipul Shahi**, Corporate Trainer, Developer, Artificial Intelligence, Data Analytics , Machine Learning.

It was only with his backing and support that I could complete the report. He provided me all sorts of help and corrected me if ever seemed to make mistakes. I have no such words to express my gratitude.

And at last but not the least, I acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. I also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

ABSTRACT

Recommendation systems are becoming increasingly important in today's extremely busy world. People are always short on time with the myriad tasks they need to accomplish in the limited 24 hours. Therefore, the recommendation systems are important as they help them make the right choices, without having to expend their cognitive resources.

The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalised lists of useful and interesting content specific to each user/individual. Recommendation systems are Artificial Intelligence based algorithms that skim through all possible options and create a customized list of items that are interesting and relevant to an individual. These results are based on their profile, search/browsing history, what other people with similar traits/demographics are watching, and how likely are you to watch those movies. This is achieved through predictive

modeling and heuristics with the data available. The aim of this model is to familiarise through the process of creating data analytics model using python in order to successfully recommend the movies.

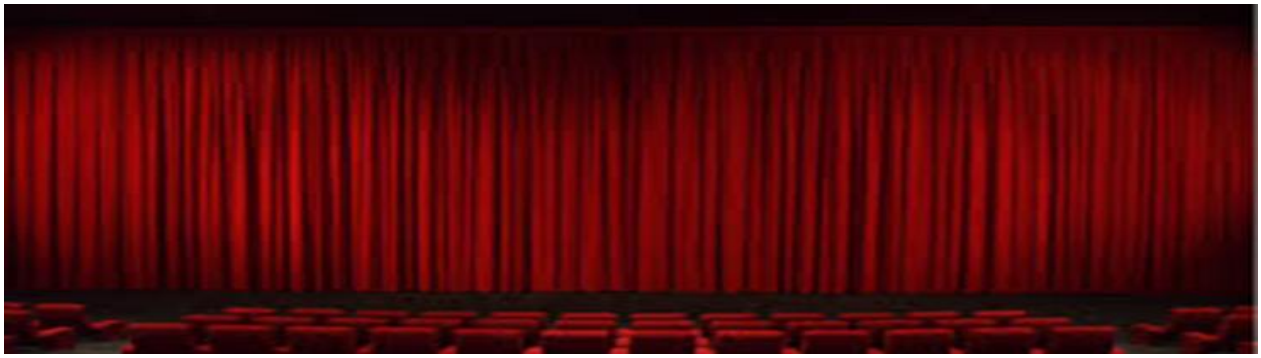
INTRODUCTION

Recommender systems are one of the most popular algorithms in data science today. They possess immense capability in various sectors ranging from entertainment to e-commerce. Recommender Systems have proven to be instrumental in pushing up company revenues and customer satisfaction with their implementation. Therefore, it is essential for machine learning enthusiasts to get a grasp on it and get familiar with related concepts.

As the amount of available information increases, new problems arise as people are finding it hard to select the items they actually want to see or use. This is where the recommender system comes in. They help us make decisions by learning our preferences or by learning the preferences of similar users.

They are used by almost every major company in some form or the other. Netflix uses it to suggest movies to customers, YouTube uses it to decide which video to play next on autoplay, and Facebook uses it to recommend pages to like and people to follow.

This way recommender systems have helped organizations retain customers by providing tailored suggestions specific to the customer's needs. According to a study by McKinsey, 35 percent of what consumers purchase on Amazon and 75 percent of what they watch on Netflix come from product recommendations based on such algorithms.



TECHNOLOGY AND CONCEPTS

DATA ANALYTICS

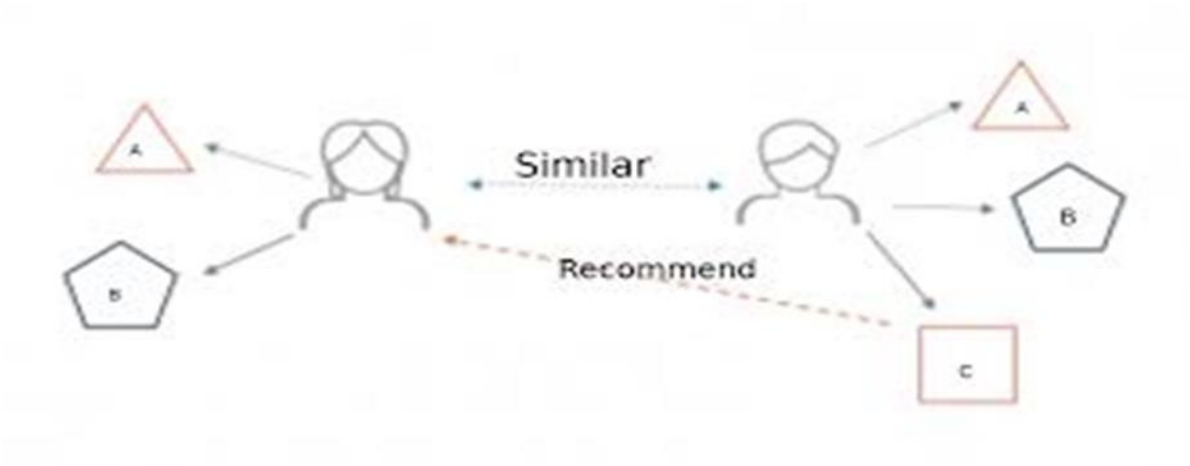
Data analytics is a broad term that encompasses many diverse types of data analysis. Any type of information can be subjected to data analytics techniques to get insight that can be used to improve things.

For example, manufacturing companies often record the runtime, downtime, and work queue for various machines and then analyze the data to better plan the workloads so the machines operate closer to peak capacity.

Data analytics can do much more than point out bottlenecks in production. Gaming companies use data analytics to set reward schedules for players that keep the majority of players active in the game. Content companies use many of the same data analytics to keep you clicking, watching, or re-organizing content to get another view or another click.

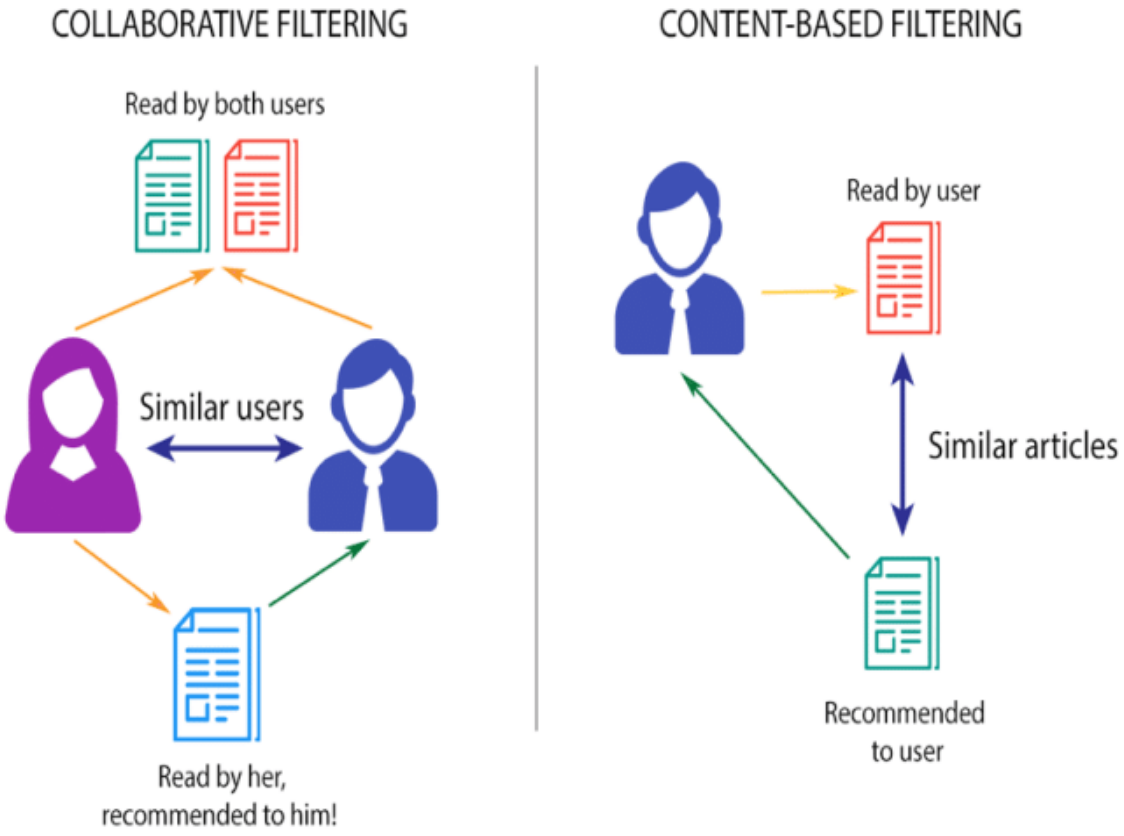
The process involved in data analysis involves several different steps:

1. The first step is to determine the data requirements or how the data is grouped. Data may be separated by age, demographic, income, or gender. Data values may be numerical or be divided by category.
2. The second step in data analytics is the process of collecting it. This can be done through a variety of sources such as computers, online sources, cameras, environmental sources, or through personnel.
3. Once the data is collected, it must be organized so it can be analyzed. Organization may take place on a spreadsheet or other form of software that can take statistical data.
4. The data is then cleaned up before analysis. This means it is scrubbed and checked to ensure there is no duplication or error, and that it is not incomplete. This step helps correct any errors before it goes on to a data analyst to be analyzed.



Recommender systems produce a list of recommendations in any of the two ways –

- **Collaborative filtering:** Collaborative filtering approaches build a model from user's past behavior (i.e. items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that user may have an interest in.
- **Content-based filtering:** Content-based filtering approaches uses a series of discrete characteristics of an item in order to recommend additional items with similar properties. Content-based filtering methods are totally based on a description of the item and a profile of the user's preferences. It recommends items based on user's past preferences.

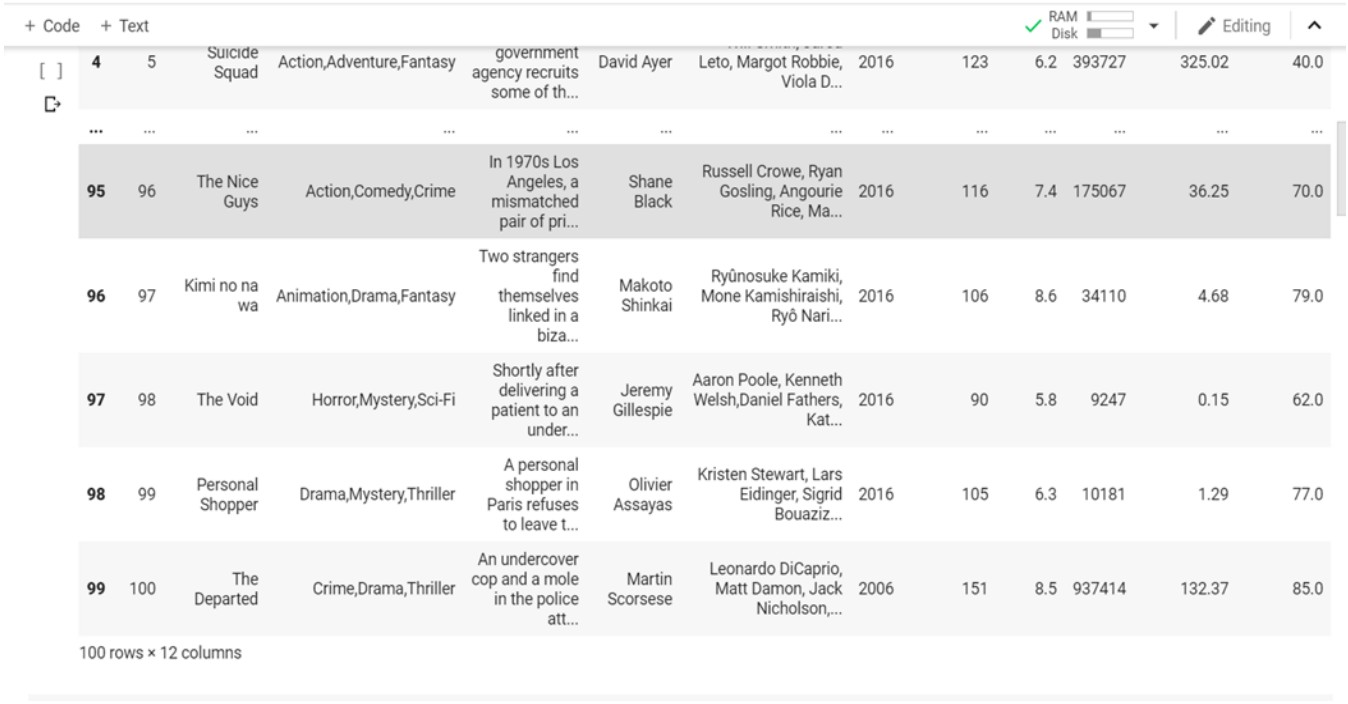


Coding

1)Loading data

```
[ ] import numpy as np
    import pandas as pd
    movie=pd.read_csv('movie_data.csv')
    movie.head(100)
```

	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
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
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	59.0
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0
...	...	...	...	...	...	...	...	...	...	...	...	...
95	96	The Nice	Action,Comedy,Crime	In 1970s Los Angeles, a	Shane	Russell Crowe, Ryan Gosling, Angourie	2016	116	7.4	175067	36.25	70.0



4	5	Suicide Squad	Action,Adventure,Fantasy	government agency recruits some of th...	David Ayer	Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0
...	...	...	...	...	...	...	...	...	...	...	...	...
95	96	The Nice Guys	Action,Comedy,Crime	In 1970s Los Angeles, a mismatched pair of pri...	Shane Black	Russell Crowe, Ryan Gosling, Angourie Rice, Ma...	2016	116	7.4	175067	36.25	70.0
96	97	Kimi no na wa	Animation,Drama,Fantasy	Two strangers find themselves linked in a biza...	Makoto Shinkai	Ryûnosuke Kamiki, Mone Kamishiraishi, Ryô Nari...	2016	106	8.6	34110	4.68	79.0
97	98	The Void	Horror,Mystery,Sci-Fi	Shortly after delivering a patient to an under...	Jeremy Gillespie	Aaron Poole, Kenneth Welsh,Daniel Fathers, Kat...	2016	90	5.8	9247	0.15	62.0
98	99	Personal Shopper	Drama,Mystery,Thriller	A personal shopper in Paris refuses to leave t...	Olivier Assayas	Kristen Stewart, Lars Eidinger, Sigrid Bouaziz...	2016	105	6.3	10181	1.29	77.0
99	100	The Departed	Crime,Drama,Thriller	An undercover cop and a mole in the police att...	Martin Scorsese	Leonardo DiCaprio, Matt Damon, Jack Nicholson,...	2006	151	8.5	937414	132.37	85.0

100 rows × 12 columns

2)calculating average rating of movies

```
[ ]  import pandas as pd
     Average_ratings=pd.DataFrame(movie.groupby('Title')['Rating'].mean())
     Average_ratings.head(100)
```

Title	Rating
(500) Days of Summer	7.7
10 Cloverfield Lane	7.2
10 Years	6.1
12 Years a Slave	8.1
127 Hours	7.6
...	...
Black Swan	8.0
Blackhat	5.4
Blair Witch	5.1
Bleed for This	6.8
Blended	6.5

100 rows × 1 columns

3)calculating total rating of movies

```
[ ] Total_ratings=pd.DataFrame(movie.groupby('Title')['Rating'].count())
    Total_ratings.head(100)
```

Title	Rating
(500) Days of Summer	1
10 Cloverfield Lane	1
10 Years	1
12 Years a Slave	1
127 Hours	1
...	...
Black Swan	1
Blackhat	1
Blair Witch	1
Bleed for This	1
Blended	1

100 rows × 1 columns

4)calculating correlation

```
[ ] movie_user=movie.pivot_table(index='Rank',columns='Title',values='Rating')
    movie_user.head(100)
```

Title	(500) Days of Summer	10 Cloverfield Lane	10 Years	12 Years a Slave	127 Hours	13 Hours	1408	17 Again	2012	20th Century Women	21	21 Jump Street	22 Jump Street	2307: Winter's Dream	28 Weeks Later	3 Days to Kill	3 Idiots	300	30 Ri... of ... Empi
Rank																			
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
96	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
97	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
98	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
99	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
100	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

100 rows × 999 columns

5) Now we need to select a movie to test our recommender system. Choose any movie title from the data. Here, I chose ‘10 years’

```
+ Code + Text                                                    RAM Disk   Editing

[ ] correlations = movie_user.corrwith(movie_user['10 Years'])
    correlations.head(100)

/usr/local/lib/python3.6/dist-packages/numpy/lib/function_base.py:2526: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar)
/usr/local/lib/python3.6/dist-packages/numpy/lib/function_base.py:2455: RuntimeWarning: divide by zero encountered in true_divide
  c *= np.true_divide(1, fact)
Title
(500) Days of Summer   NaN
10 Cloverfield Lane    NaN
10 Years               NaN
12 Years a Slave       NaN
127 Hours              NaN
                        ..
Black Swan             NaN
Blackhat               NaN
Blair Witch            NaN
Bleed for This         NaN
Blended                NaN
Length: 100, dtype: float64
```

6) creating dataframe with 'rating' count values

```
+ Code + Text                                                    Disk   Editing

[ ] ratings = pd.DataFrame(movie.groupby('Title')['Rating'].mean())

    ratings['num of ratings'] = pd.DataFrame(movie.groupby('Title')['Rating'].count())

    ratings.head()
```

Title	Rating	num of ratings
(500) Days of Summer	7.7	1
10 Cloverfield Lane	7.2	1
10 Years	6.1	1
12 Years a Slave	8.1	1
127 Hours	7.6	1

7) Visualization imports:

plot graph of 'num of ratings column'

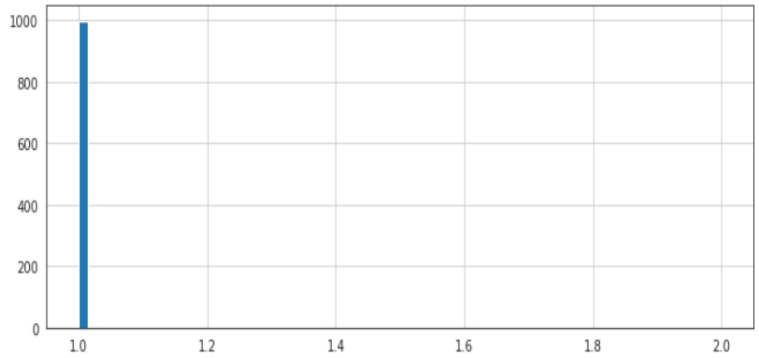
```
[ ] import matplotlib.pyplot as plt
    import seaborn as sns

    sns.set_style('white')
    %matplotlib inline
```

```
plt.figure(figsize =(11, 4))

ratings['num of ratings'].hist(bins = 70)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad7e268518>
```

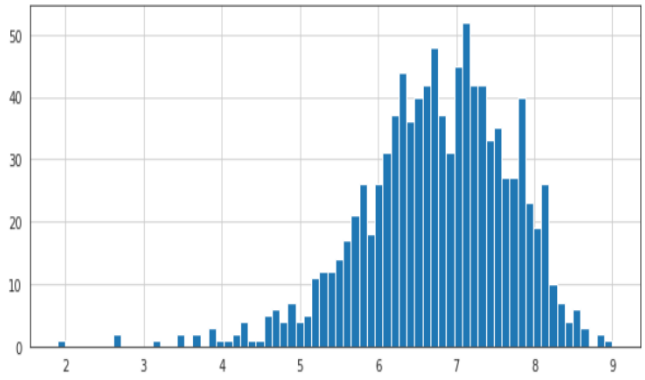


plot graph of 'ratings' column

```
[ ] plt.figure(figsize =(10, 4))

    ratings['Rating'].hist(bins = 70)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad7dd23cf8>
```



8) Sorting values according to the 'num of rating column'

```
[ ]  #Sorting values according to
    # the 'num of rating column'
    moviemat = movie.pivot_table(index ='Rank',
                  columns ='Title', values ='Rating')

    moviemat.head()

    ratings.sort_values('num of ratings', ascending = False).head(10)
```

Title	Rating	num of ratings
The Host	6.45	2
(500) Days of Summer	7.70	1
Straight Outta Compton	7.90	1
Srpski film	5.20	1
Stake Land	6.50	1
Star Trek	8.00	1
Star Trek Beyond	7.10	1
Star Trek Into Darkness	7.80	1
Star Wars: Episode VII - The Force Awakens	8.10	1
Stardust	7.70	1

9) Now we will remove all the empty values and merge the total ratings to the correlation table.

```
[ ] recommendation = pd.DataFrame(correlations,columns=['Correlation'])
    recommendation.dropna(inplace=True)
    recommendation = recommendation.join(Average_ratings['Total Ratings'])
    recommendation.head(5)
```

title	Correlation	Total Ratings
'burbs, The (1989)	0.240563	17
(500) Days of Summer (2009)	0.353833	42
*batteries not included (1987)	-0.427425	7
10 Cent Pistol (2015)	1.000000	2
10 Cloverfield Lane (2016)	-0.285732	14

```
[ ] recc = recommendation[recommendation['Total Ratings']>100].sort_values('Correlation',ascending=False).reset_index()

    recc = recc.merge(movie_titles_genre,on='title', how='left')
    recc.head(10)
```

	title	Correlation	Total Ratings	movieId	genres
0	Toy Story (1995)	1.000000	215	1	Adventure|Animation|Children|Comedy|Fantasy
1	Incredibles, The (2004)	0.643301	125	8961	Action|Adventure|Animation|Children|Comedy
2	Finding Nemo (2003)	0.618701	141	6377	Adventure|Animation|Children|Comedy
3	Aladdin (1992)	0.611892	183	588	Adventure|Animation|Children|Comedy|Musical
4	Monsters, Inc. (2001)	0.490231	132	4886	Adventure|Animation|Children|Comedy|Fantasy
5	Mrs. Doubtfire (1993)	0.446261	144	500	Comedy|Drama
6	Amelie (Fabuleux destin d'Amélie Poulain, Le) ...	0.438237	120	4973	Comedy|Romance
7	American Pie (1999)	0.420117	103	2706	Comedy|Romance
8	Die Hard: With a Vengeance (1995)	0.410939	144	165	Action|Crime|Thriller
9	E.T. the Extra-Terrestrial (1982)	0.409216	122	1097	Children|Drama|Sci-Fi

CONCLUSION

In this model, we studied what a recommender system is and how we can create it in Python using the Pandas and Numpy library. It is important to mention that the recommender system I created is very simple using the concepts of data analytics . Real-life recommender systems use very complex algorithms

REFERENCES:

- https://www.kaggle.com