PROJECT BASED LEARNING REPORT

ON

# MOVIE RECOMMENDER SYSTEM USING MACHINE LEARNING

Submitted To

**SAVITARIBAI PHULE PUNE UNIVERSITY, PUNE**

Submitted By

1.Aman Rajjak Sayyad S190514204

2.Arman Rajjak Sayyad S190514206

Under the Guidance of

Prof. Dr. V.S Wadne

**DEPARTMENT OF COMPUTER ENGINEERING**

JSPM's IMPERIAL COLLEGE OF ENGINEERING &

RESEARCH

WAGHOLI, PUNE- 412207

2022-23

JSPM's Imperial College of Engineering & Research,

Wagholi Pune – 412207

## CERTIFICATE

This is to certify that the PBL Project entitled Movie Reccomender System Using Machine Learning Submitted by Aman Rajjak Sayyad is a bonafide student and work carried by him, under my guidance as a part of Project Based Learning course.

| | | |
|---|---|---|
| Guide | Head of Department | Principal |
| Computer Department | Computer Department | I.C.O.E.R, Pune |

# CONTENT

# List Of Figures

# 1.ABSTRACT

A movie recommendation system, or a movie recommender system, is an ML-based approach to filtering or predicting the users' film preferences based on their past choices and behavior. It's an advanced filtration mechanism that predicts the possible movie choices of the concerned user and their preferences towards a domain-specific item, aka movie.The basic concept behind a movie recommendation system is quite simple. In particular, there are two main elements in every recommender system: users and items. The system generates movie predictions for its users, while items are the movies themselves.

The primary goal of movie recommendation systems is to filter and predict only those movies that a corresponding user is most likely to want to watch. The ML algorithms for these recommendation systems use the data about this user from the system's database. This data is used to predict the future behavior of the user concerned based on the information from the past. Movie recommendation systems use a set of different filtration strategies and algorithms to help users find the most relevant films. The most popular categories of the ML algorithms used for movie recommendations include content-based filtering and collaborative filtering systems.

# 2.INTRODUCTION

Machine Learning is an AI technique that teaches computers to learn from experience. Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. Deep learning is a specialized form of machine learning. Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.

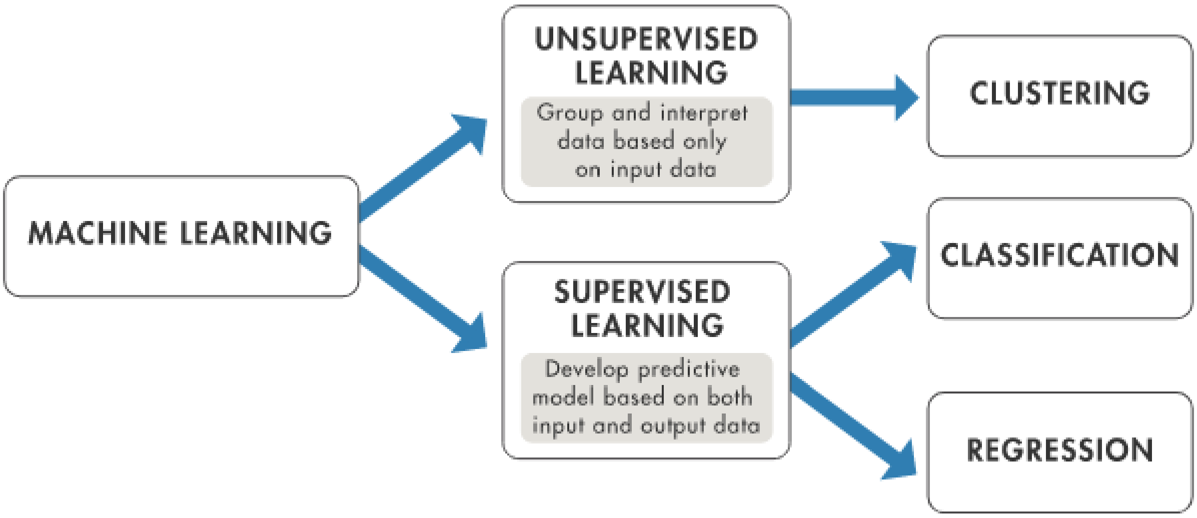

MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLUSTERING
CLASSIFICATION
REGRESSION

Fig.2.1 Machine Learning

Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. [Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, agriculture, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers, but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning.

# 3.PROBLEM STATEMENT

To create a movie recommender system using machine learning.

# 4.LITERATURE SURVEY

The original K-means algorithm was proposed by MacQueen .The ISODATA algorithm by Ball and Hall was an early but sophisticated version of k-means. Clustering divides the objects into meaningful groups. Clustering is unsupervised learning. Document clustering is automatic document organization. In K-means clustering technique we choose K initial centroids, where K is the desired number of clusters. Each point is then assigned to the cluster with nearest mean i.e. the centroid of the cluster. Then we update the centroid of each cluster based on the points that are assigned to the cluster. We repeat the process until there is no change in the cluster center (centroid). Therefore we have used ratios of 1:1, 1:2, and 1:3 depending on total number of votes received by a movie. we have also found that the movies which have rating less than 5 are the ones which are least suitable for recommendation, and are least desirable by users. Users generally want to see a good movie and higher rating ensures that our predicted movie set are of those movies which are liked by a large number of users. Weights assigned to other attributes are generally based on the average of total movies associated with that particular attribute to the total number of movies in our data set.

When any user enters our system MOVREC he has a couple of options. He /she can search a particular movie or see upcoming movies list or can go to our recommendation page. On recommendation page he is given the choice to select/input values for different attributes. On the basis of these input values, we search our search our database and prepare an array of suitable movies. Movies included in the array are those whose even one attribute value matches with the input value of the user. We then calculate the number of movies in our array with the help of a counter. If the counter value is less than or equal to twenty we display the movie list sorted according to ratings associated with the movies. If number of movies is greater than twenty then we apply a pre filter and select top twenty movies according to rating. If two movies have same rating then priority is given to the movie having a large number of votes. After filtering the movie list we match the attributes value to their respective weights and compute the total weight of each movie. Once we have calculated the total weight of each movie we apply K-means clustering algorithm on these group of movies. In our research we have also found that generally a user prefer a list with five movies so we assume K equal to be 4 so that an average every K has five movies, where K is the number of cluster to be formed.

For each cluster k1, k2 , k3, k4 we assume initial centroid c1, c2, c3, c4 which corresponds to the first, sixth, eleventh, and sixteenth movie in the movie array. After defining the initial centroid we compute the distance of all the other data points from each centroid and assign the remaining data points (movies) to closest centroid and form clusters. The distance measure we have used to calculate the distance between data points and centroid is the Euclidean Distance. After forming initial clusters we take one cluster at a time. We again calculate centroids but this time each centroid corresponds to mean of the points in that cluster. After recalculating centroids we compute the distance of all data points with respect to these newly formed centroids and reassign them to form clusters. We repeat this process till there is no change in centroids. This ensures that the clusters finally formed are optimized and no further grouping is possible. Once final cluster are formed we compute the average rating of all points belonging to that cluster i.e. cluster rating, then according to the input user query we display the cluster having highest cluster rating.

In proposed model we use a pre filter before applying K-means algorithm. The attributes used to calculate distance of each point from centroid are 1. Genre 2. Actor 3. Director 4. Year 5. Rating Different attributes have different weights. In our research we have found that the most appropriate recommendations that can be generated should be based on the ratings given to the movies by previous users, therefore we have given more importance to the rating attribute than other attributes. These ratings have been taken from www.imdb.com because perhaps it has the largest collection of movies along with the rating given to these movies by a large number of different users from different parts of the world. Another important parameter in our proposed model is total number of votes received by a particular movie. We have divided number of votes in to three categories that is less than or equal to 1000, more than 1000 but less than or equal to 10,000 and greater than 10,000.

# 5.PROPOSED METHODOLOGY

## 5.1SOFTWARE REQUIREMENT SPECIFICATION

MORES

Software Requirement

Specification

29:05:2023

Introduction

### Purpose

The purpose of the Software Requirements Specification (SRS) document is to provide a detailed overview of our software product, its parameters, and goals. This document aims to gather and analyze and give an in-depth insight into the Hybrid Movie Recommender system by defining the problem statement in detail. It concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The straightforward user interface, hardware, and software requirements of the Hybrid Movie Recommender system are provided in this document.

### Scope of project

This software helps users of the customer platform explore content quickly with our recommendation system's help. The software we are developing is a Hybrid Recommendation System for Movies, which uses the combination of collaborative and content-based filtering in the context of web-based recommender systems. In particular, we will link the well-known TMDB data set. The content filtering part of the system is based on trained neural networks representing individual user preferences. Using various experiments, we will demonstrate the influence of supplementary user and item features on our proposed hybrid recommender's prediction accuracy. To decrease system runtime and reveal latent user and item relations, we will factorize our hybrid model via singular value decomposition (SVD). Due to the enormous amount of information available online, the need for highly developed personalization and filtering systems is growing permanently. Recommendation systems constitute a specific type of information filtering that attempts to present items according to the interests expressed by a user.

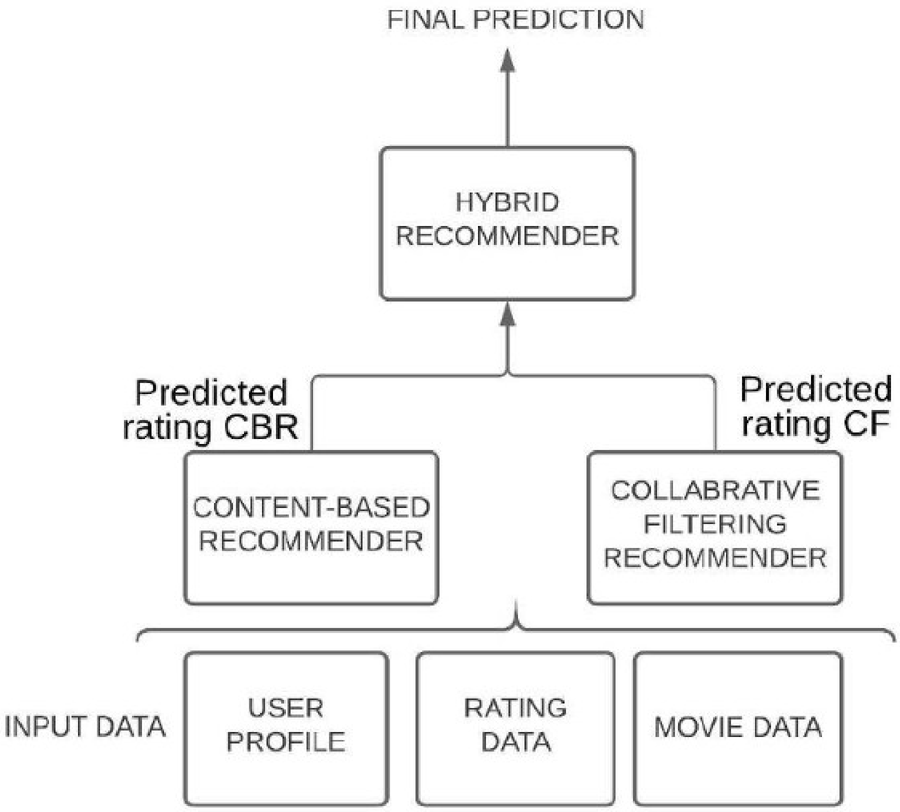


Fig.5.1.1 overall description

User Use Case Diagram

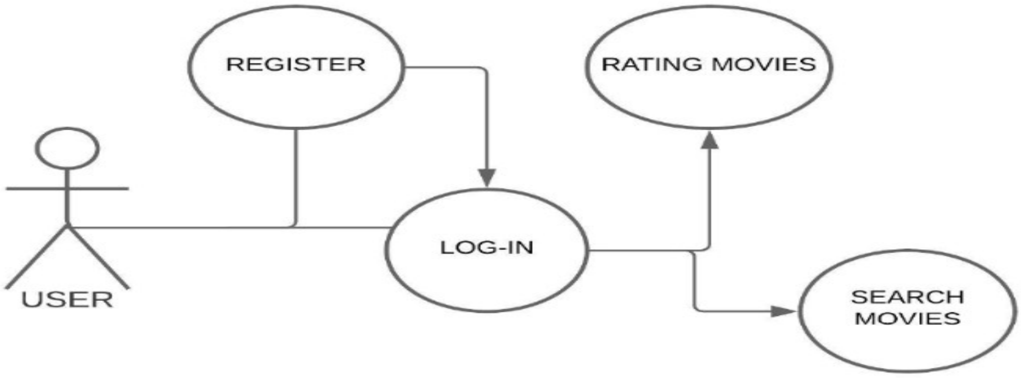


Fig.5.1.2 use case diagram

Functional Requirements Specification

1. INPUT: “REGISTER” option selected by the user OUTPUT: user prompted to enter personal data for registration.
2. INPUT: the user enters registration data OUTPUT: user redirected to the home page with the search engine PROCESSING: check if the input values are in the correct format as needed.

3. INPUT: “LOG-IN” option selected by the user OUTPUT: user prompted to enter USER NAME and PASSWORD.
4. INPUT: enter USER NAME and PASSWORD OUTPUT: If the credentials are correct, then redirect to the home page with the search engine. If the credentials are wrong, it shows a prompt of wrong credentials and asks to re-enter the credentials.
5. INPUT: “search” option, OUTPUT: user prompted to enter “MOVIE” name.
6. INPUT: “MOVIE NAME” OUTPUT: Details about the movie if present in dataset and related recommendations of other movies. PROCESSING: if the user is logged in, then based on collaborative and content filtering, new movies are recommended if not based only on content filtering, movies are recommended.
7. INPUT: LIKE / DISLIKE MOVIE OUTPUT: The movie rating is added to the user profile.

## Non-Functional Requirements

1. The online Recommendation system will be hosted on Heroku, which provides free servers for hoisting our sites.
2. The users and content-providers need to have access to the necessary bandwidth internet to access the website.

## PROCESSES AND TECHNOLOGY

1. The recommender system uses supervised approaches such as classification and regression and unsupervised approaches such as dimensionality reduction and clustering/compression using topic modeling.
2. Matrix factorization, Singular Value Decomposition, factorization machines, connections to probabilistic graphical models, and methods that can be easily expanded to be tailored for different problems.

## Performance

The product shall be based on the web and has to be run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon the hardware components of the client/customer.If the Internet connection is good then The application should update the interface on interaction within 2-4 seconds.

## Security

## Data Transfer

The system will use secure sockets in all transactions that include any confidential customer information. The system will automatically log out all customers after a period of inactivity. The system shall confirm all transactions with the customer's web browser. The system will not leave any cookies on the customer's computer containing the user's password or any of the user's confidential information.

## Data Storage

The customer's web browser and the system's back-end servers will never display a customer's password. The system's back-end databases will be encrypted.

## Safety

Databases should  be redundant to prevent loss of data.Backups of the databases should be done hourly.

## Software Quality Attributes

1. Availability: All the movies may not be present,we should contain as many movies as possible based on priority and ratings etc.If a user searches for a movie it should be available for the user.
2. Correctness: If a user searches for a movie with a keyword like genres,movie name etc,the correct and related movies should be recommended in case the movie searched is not present.
3. Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
4. Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.

## External Interface Requirements

1. User Interfaces The user interface for the software shall be compatible with any browser such as Internet Explorer, Mozilla, and Google Chrome by which users can access the system.
2. Hardware Interfaces Since the application must run over the internet, all the hardware required to be connected to the internet will be a hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.
3. Software Interfaces We have chosen Windows operating system for its best support and user-friendliness.To save the movie details, users preferences etc, we have chosen SQL database.To implement the project we will use Python libraries,Javascript,Html etc.
4. Anacondas navigator , Jupyter , ipython

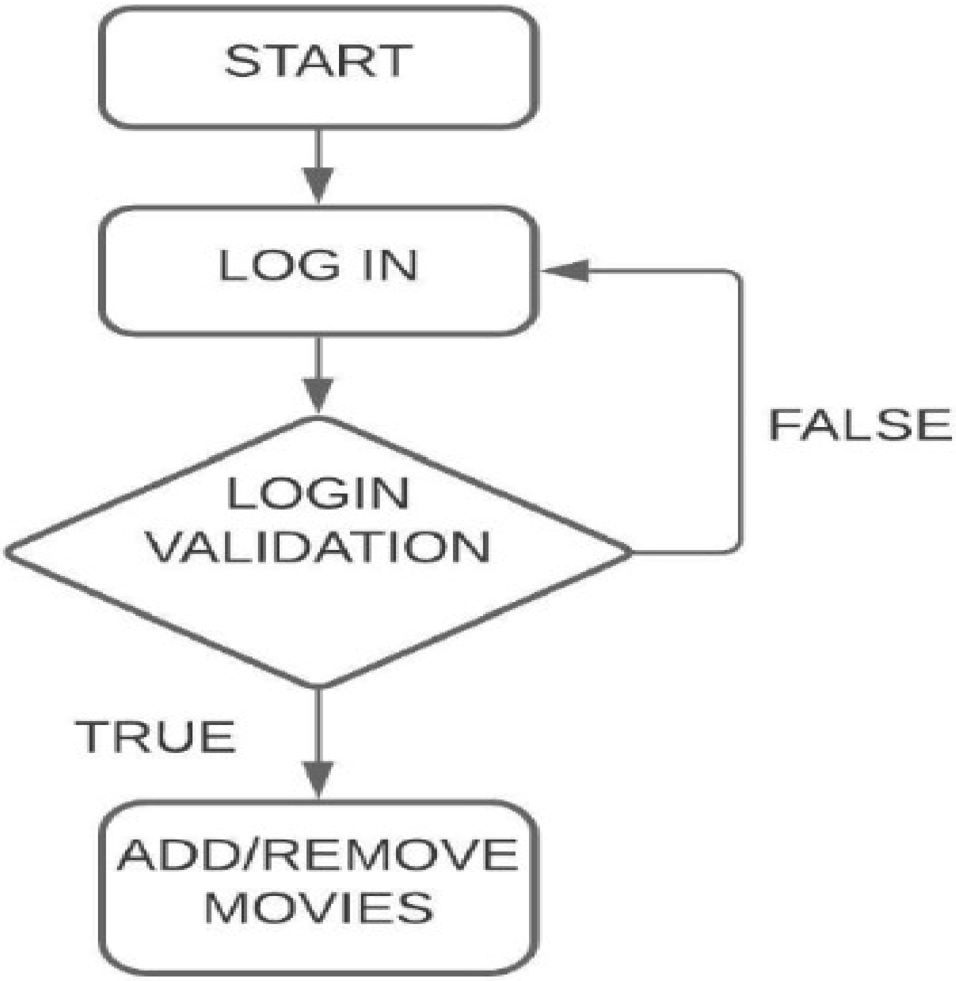


Fig.5.1.3 flow chart for login

## CONSTRAINTS

The system must be completed within a budget of 2000Rs and launched within two months.

## 5.2 DESCRIPTION

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch Reinforcement machine learning is a machine learning model that is similar to supervised learning, but the algorithm isn't trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.

## 5.3 ARCHITECTURE

### Content-Based Filtering

A filtration strategy for movie recommendation systems, which uses the data provided about the items (movies). This data plays a crucial role here and is extracted from only one user. An ML algorithm used for this strategy recommends motion pictures that are similar to the user's preferences in the past. Therefore, the similarity in content-based filtering is generated by the data about the past film selections and likes by only one user.

### Collaborative Filtering

As the name suggests, this filtering strategy is based on the combination of the relevant user's and other users' behaviors. The system compares and contrasts these behaviors for the most optimal results. It's a collaboration of the multiple users' film preferences and behaviors. What's the mechanism behind this strategy? The core element in this movie recommendation system and the ML algorithm it's built on is the history of all users in the database. Basically, collaborative filtering is based on the interaction of all users in the system with the items (movies).

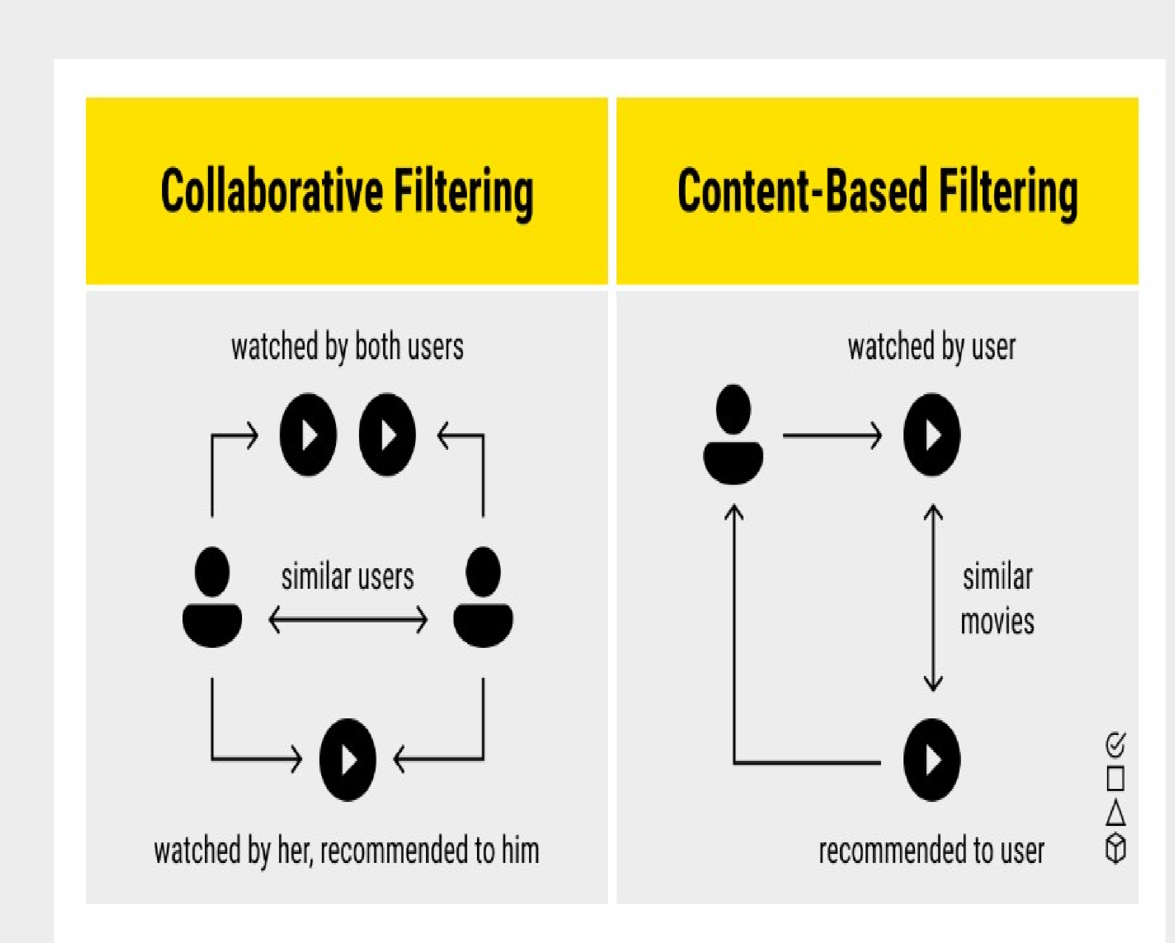


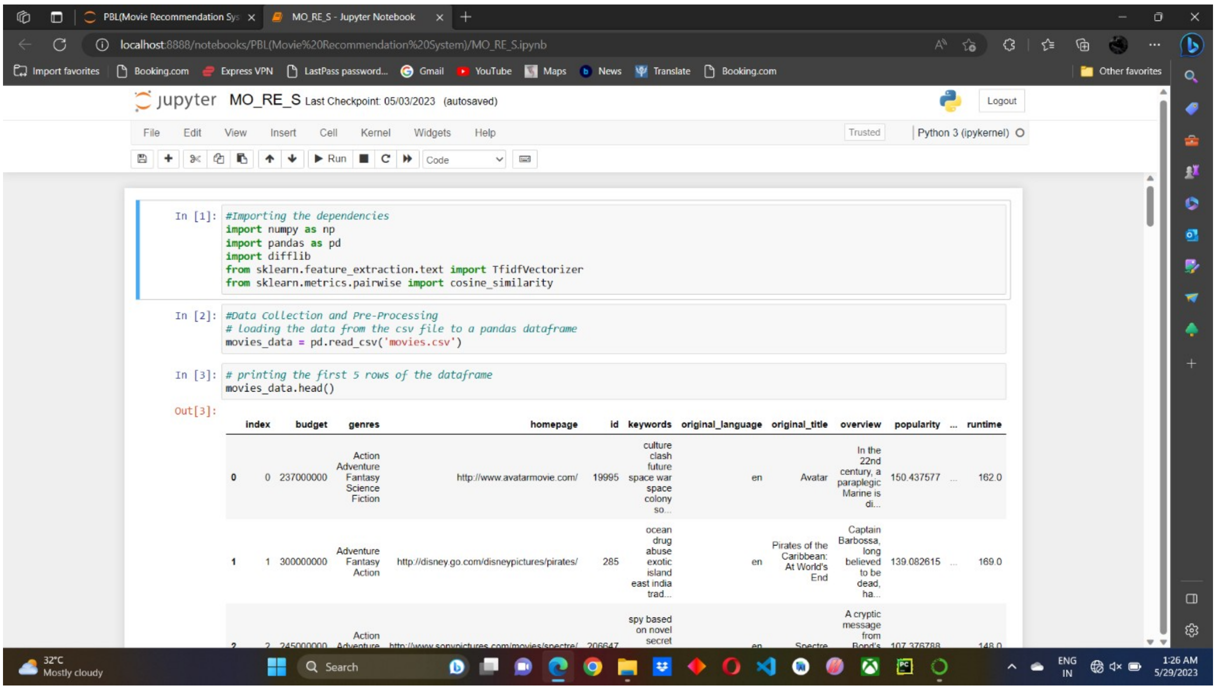
Fig.5.3.1 Architecture

## 5.4 ALGORITHMS

A number of machine learning algorithms are commonly used. These include:

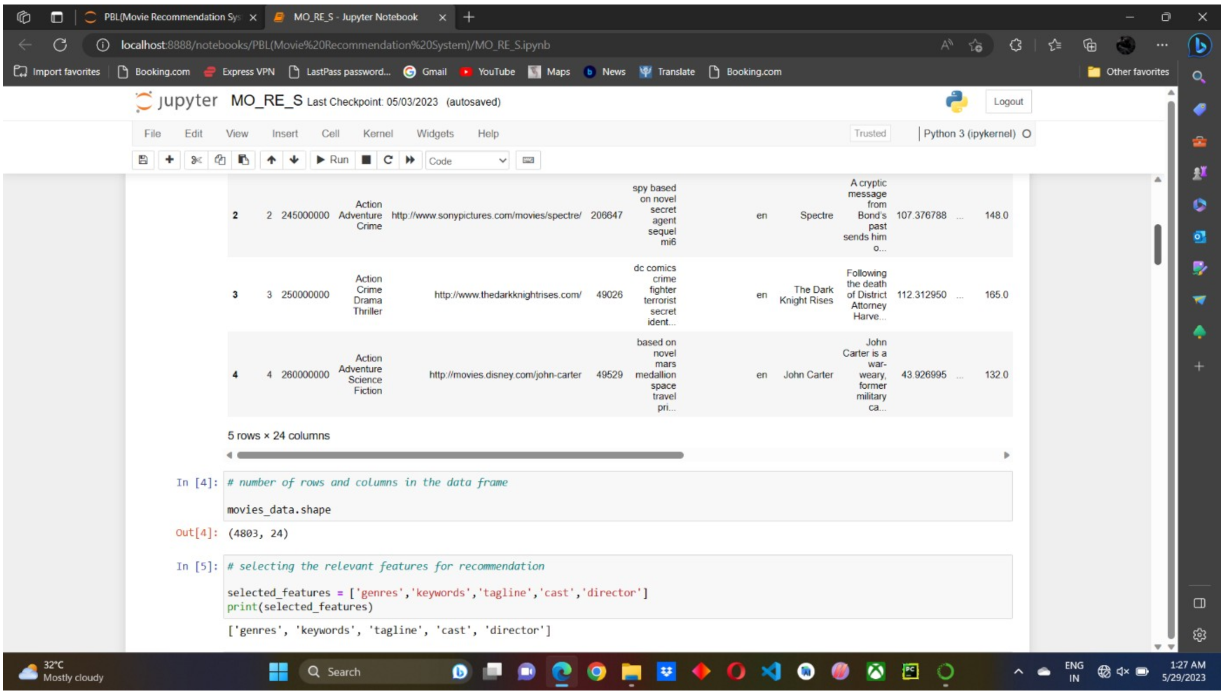
1. Neural networks: Neural networks simulate the way the human brain works, with a huge number of linked processing nodes. Neural networks are good at recognizing patterns and play an important role in applications including natural language translation, image recognition, speech recognition, and image creation.
2. Linear regression: This algorithm is used to predict numerical values, based on a linear relationship between different values. For example, the technique could be used to predict house prices based on historical data for the area.
3. Logistic regression: This supervised learning algorithm makes predictions for categorical response variables, such as“yes/no” answers to questions. It can be used for applications such as classifying spam and quality control on a production line.
4. Clustering: Using unsupervised learning, clustering algorithms can identify patterns in data so that it can be grouped. Computers can help data scientists by identifying differences between data items that humans have overlooked.
5. Decision trees: Decision trees can be used for both predicting numerical values (regression) and classifying data into categories. Decision trees use a branching sequence of linked decisions that can be represented with a tree diagram. One of the advantages of decision trees is that they are easy to validate and audit, unlike the black box of the neural network.
6. Random forests: In a random forest, the machine learning algorithm predicts a value or category by combining the results from a number of decision trees.

# 6.RESULT



```
In [1]: #Importing the dependencies
        import numpy as np
        import pandas as pd
        import difflib
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.metrics.pairwise import cosine_similarity

In [2]: #Data Collection and Pre-Processing
        # loading the data from the csv file to a pandas dataframe
        movies_data = pd.read_csv('movies.csv')

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

Out[3]:
```

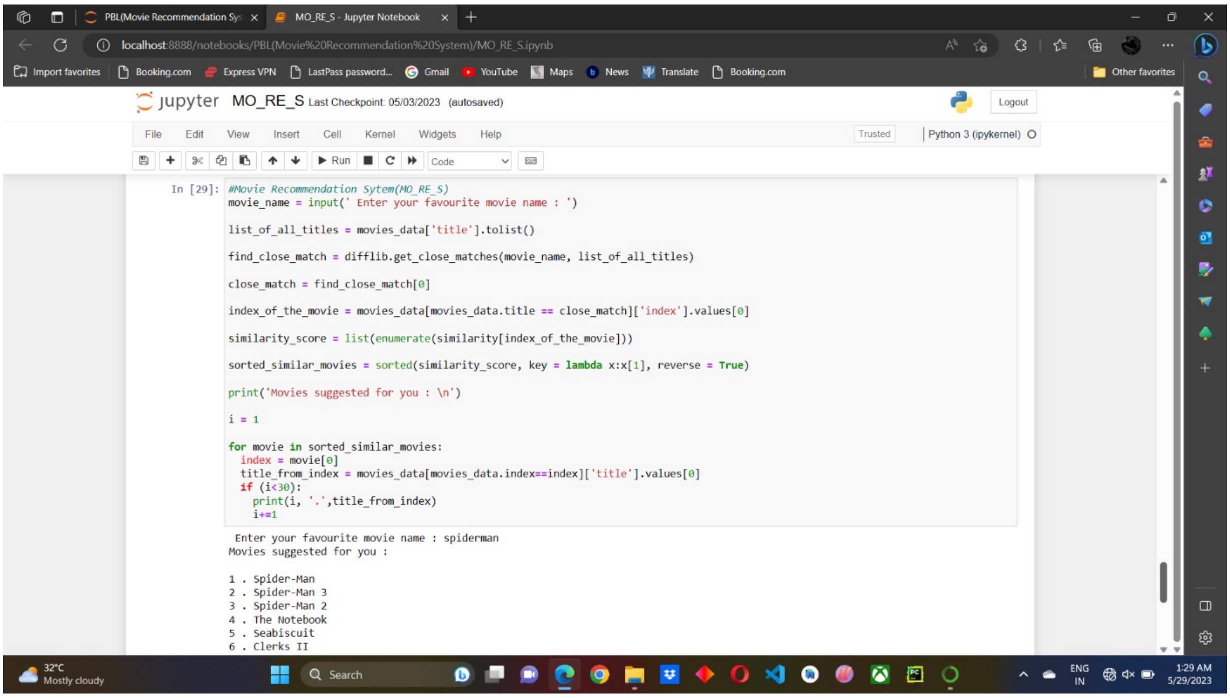
| | index | budget | genres | homepage | id | keywords | original_language | original_title | overview | popularity | ... | runtime |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 237000000 | Action<br>Adventure<br>Fantasy<br>Science<br>Fiction | http://www.avatarmovie.com/ | 19995 | culture<br>clash<br>future<br>space war<br>space<br>colony<br>so... | en | Avatar | In the<br>22nd<br>century, a<br>paraplegic<br>Marine is<br>di... | 150.437577 | ... | 162.0 |
| 1 | 1 | 300000000 | Adventure<br>Fantasy<br>Action | http://disney.go.com/disneypictures/pirates/ | 285 | ocean<br>drug<br>abuse<br>exotic<br>island<br>east india<br>trad... | en | Pirates of the<br>Caribbean:<br>At World's<br>End | Captain<br>Barbossa,<br>long<br>believed<br>to be<br>dead,<br>ha... | 139.082615 | ... | 169.0 |
| 2 | 2 | 245000000 | Action<br>Adventure | http://www.sonypictures.com/movies/spectre/ | 206647 | spy based<br>on novel<br>secret | en | Spectre | A cryptic<br>message<br>from<br>Bond's | 107.376788 | ... | 148.0 |



| | index | budget | genres | homepage | id | keywords | original_language | original_title | overview | popularity | ... | runtime |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 245000000 | Action<br>Adventure<br>Crime | http://www.sonypictures.com/movies/spectre/ | 206647 | spy based<br>on novel<br>secret<br>agent<br>sequel<br>mi6 | en | Spectre | A cryptic<br>message<br>from<br>Bond's<br>past<br>sends him<br>o... | 107.376788 | ... | 148.0 |
| 3 | 3 | 250000000 | Action<br>Crime<br>Drama<br>Thriller | http://www.thedarkknightrises.com/ | 49026 | dc comics<br>crime<br>fighter<br>terrorist<br>secret<br>ident... | en | The Dark<br>Knight Rises | Following<br>the death<br>of District<br>Attorney<br>Harve... | 112.312950 | ... | 165.0 |
| 4 | 4 | 260000000 | Action<br>Adventure<br>Science<br>Fiction | http://movies.disney.com/john-carter | 49529 | based on<br>novel<br>mars<br>medallion<br>space<br>travel<br>pri... | en | John Carter | John<br>Carter is a<br>war-<br>weary,<br>former<br>military<br>ca... | 43.926995 | ... | 132.0 |

5 rows × 24 columns

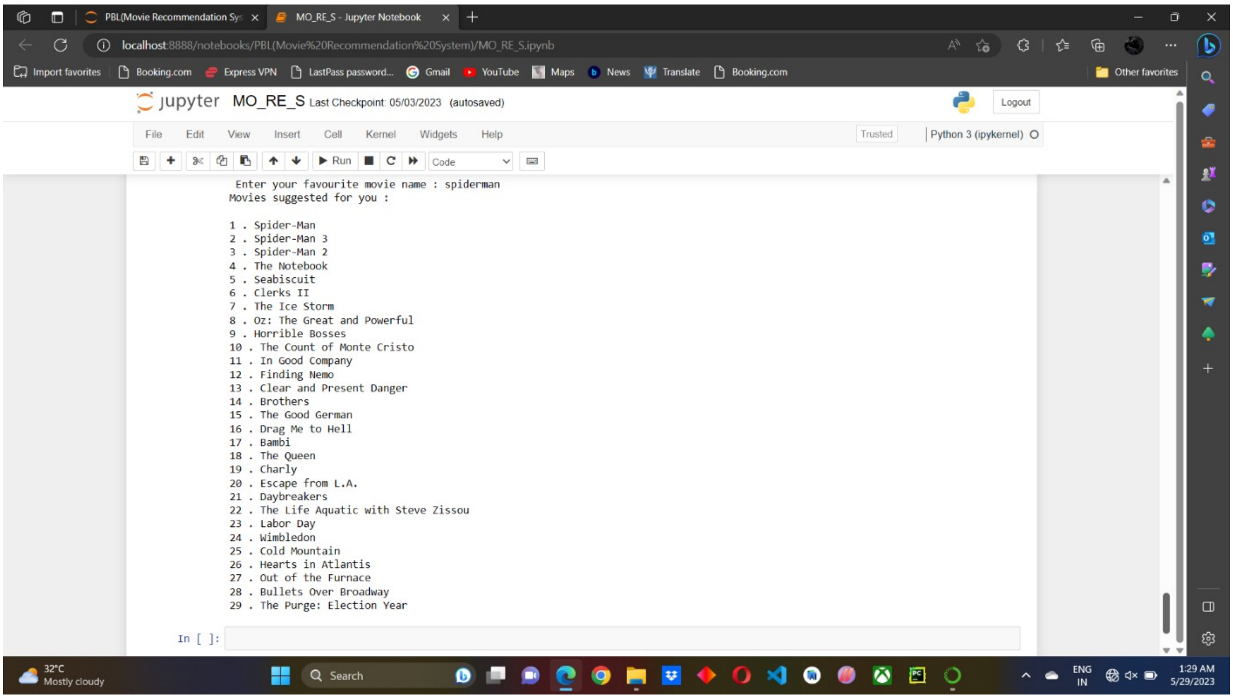
```
In [4]: # number of rows and columns in the data frame

        movies_data.shape

Out[4]: (4803, 24)

In [5]: # selecting the relevant features for recommendation

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

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



jupyter MO_RE_S Last Checkpoint: 05/03/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code
Trusted
Python 3 (ipykernel)
Logout

```
In [29]: #Movie Recommendation Sytem(MO_RE_S)
movie_name = input(' Enter your favourite movie name : ')

list_of_all_titles = movies_data['title'].tolist()

find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)

close_match = find_close_match[0]

index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]

similarity_score = list(enumerate(similarity[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<30):
    print(i, '.',title_from_index)
    i+=1
```

```
 Enter your favourite movie name : spiderman
Movies suggested for you :

1 . Spider-Man
2 . Spider-Man 3
3 . Spider-Man 2
4 . The Notebook
5 . Seabiscuit
6 . Clerks II
```



jupyter MO_RE_S Last Checkpoint: 05/03/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code
Trusted
Python 3 (ipykernel)
Logout

```
 Enter your favourite movie name : spiderman
Movies suggested for you :

1 . Spider-Man
2 . Spider-Man 3
3 . Spider-Man 2
4 . The Notebook
5 . Seabiscuit
6 . Clerks II
7 . The Ice Storm
8 . Oz: The Great and Powerful
9 . Horrible Bosses
10 . The Count of Monte Cristo
11 . In Good Company
12 . Finding Nemo
13 . Clear and Present Danger
14 . Brothers
15 . The Good German
16 . Drag Me to Hell
17 . Bambi
18 . The Queen
19 . Charly
20 . Escape from L.A.
21 . Daybreakers
22 . The Life Aquatic with Steve Zissou
23 . Labor Day
24 . Wimbledon
25 . Cold Mountain
26 . Hearts in Atlantis
27 . Out of the Furnace
28 . Bullets Over Broadway
29 . The Purge: Election Year

In [ ]:
```

# 7. CONCLUSION

In this project we have introduced MovieREC, a recommender system for movie recommendation. It allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes and using K-means algorithm. By the nature of our system, it is not an easy task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinions. Based on informal evaluations that we carried out over a small set of users we got a positive response from them. We would like to have a larger data set that will enable more meaningful results using our system. Additionally we would like to incorporate different machine learning and clustering algorithms and study the comparative results. Eventually we would like to implement a web based user interface that has a user database, and has the learning model tailored to each user.

# REFERENCES

1. Han J., Kamber M., “Data Mining: Concepts and Techniques”, Morgan Kaufmann (Elsevier), 2006.
2. Ricci and F. Del Missier, “Supporting Travel Decision making Through Personalized Recommendation,” Design Personalized User Experience for e-commerce, pp. 221-251, 2004.
3. Steinbach M., P Tan, Kumar V., “Introduction to Data Mining.” Pearson, 2007.
4. Jha N K, Kumar M, Kumar A, Gupta V K “Customer classification in retail marketing by data mining” International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014 ISSN 2229-5518.