Module Code & Module Title

CU6051NI - Artificial Intelligence

Assessment Weight & Type

75% Individual coursework

Year and Semester

2022-2023

Student Name: Keshab Acharya

London Met ID: 19031062

College ID: NP01CP4A190202

Assignment Due Date: 26 Jan 2022

Assignment Submission Date: 26 Jan 2022

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

The topic of this paper is movie suggestions. Because of its ability to provide improved entertainment, a movie recommendation is vital in our social lives. Users can be recommended a set of movies based on their interests or the popularity of the films. A recommendation system is used to make suggestions for things to buy or see. They employ a big collection of information to steer consumers to the things that will best match their needs. They're mostly employed for commercial purposes. MOVREC also assists users in efficiently and effectively locating movies of their choosing based on the movie experiences of other users, without wasting time in pointless searching.m

ACKNOWLEDGEMENT

I'd want to use this opportunity to express my heartfelt gratitude and admiration to Islington college, Kamal pokhari, for providing me with a place to study and complete my project work. I'd want to use this opportunity to offer my heartfelt gratitude to Dr. Sunil Raut Chetrii, for his unwavering support and encouragement throughout this endeavor. I'd also like to express my gratitude to every one of the members who have supported me in some way. Finally, I'd like to express my gratitude to my parents and friends for their unwavering moral support throughout the execution of this project.

Contents

List of Figures

1)Introduction

Explanation of topic/AI Concepts used

AI sometimes called machine intelligence is intelligence demonstrated by machines which also refers to the simulation of human intelligence in the machine which is programmed to think like humans and mimics their actions. Some of the action that is designed to do is speech recognition, learning, planning, and problem-solving. It addressed the crucial question of what knowledge is required in any aspect of thinking, how should that knowledge be represented and how should it be used. (Saleh, April 2019)

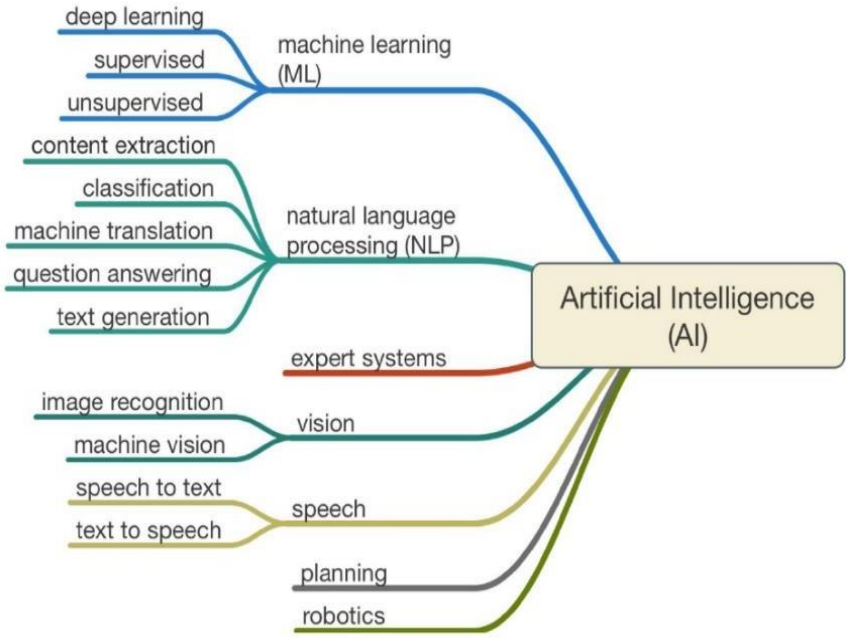


Figure 1: AI branches

Keshab Acharya
19031062

According to John McCarthy (father of artificial intelligence), it is “The science and engineering of making intelligent machines, especially intelligent computers programs” (sciencedaily, January 23, 2022). It is accomplished by studying how the human brains think. It attempts to understand intelligence entities. The term AI is frequently applied to developing system projects with the intellectual processes characteristics of humans. There are different ways of learning AI, the easy and simplest is learning by trial and error. Siri, Alexa, and other smart assistants, self-driving cars, Robo-advisors, email spam filters, and Netflix’s recommendation system are some examples of AI. (Copeland, Dec 14, 2021)

The intelligence is imperceptible. It contains

1) Reasoning
2) Learning
3) Problem-solving
4) Perception
5) Linguistic intelligent

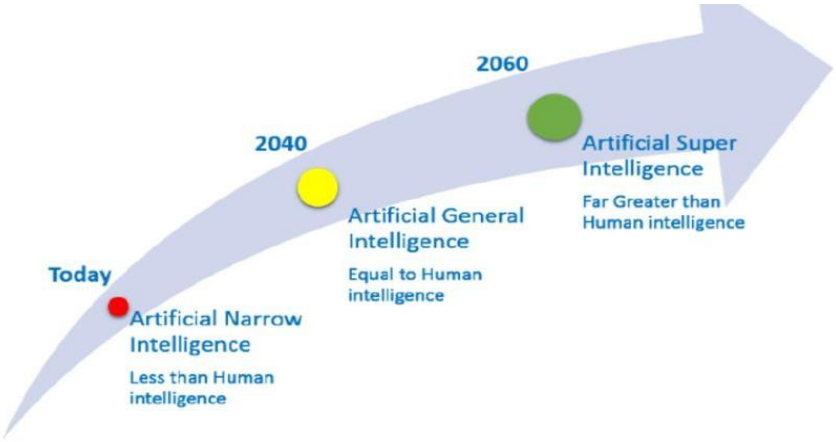


Figure 2: Future of AI

An RS (Recommendation System) is an information filtering system that tries to forecast the user's preferences and provide recommendations based on those choices. It is also a simple algorithm that aims to provide the most relevant information to the user by discovering a pattern in the database. (Agrawal, July 13, 2021) It forecasts the most likely product that

consumers will purchase and is of interest to them. It mainly deals with the large volume of information present by filtering the most important information best on the data provided by a user (Zakir Hussain, 2021)

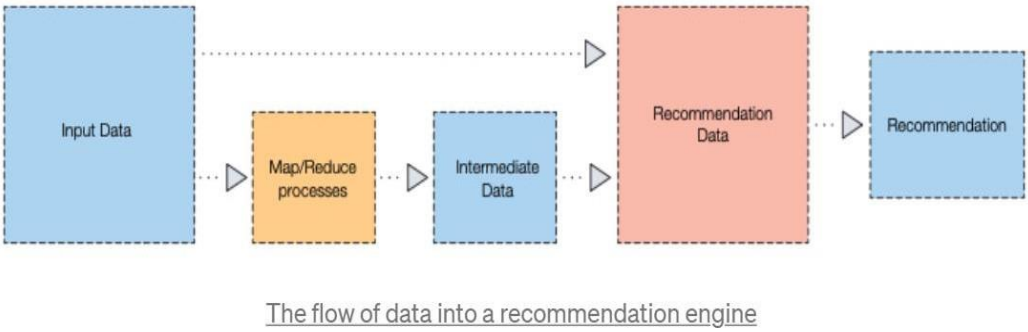


Figure 3: Flow of data in recommendation system

Recommendation system has become increasingly popular over the last few years and nowadays they are utilized in most of the online platform that we used such as Netflix, Amazon, YouTube, and many others. The content of such platforms varies from movies, music, video, books, to people on professional and dating websites, to research results returned on Google. They can collect information about users' choices and use that information to improve their suggestions for the future. (F.O.Isinkaye, November 2015,).

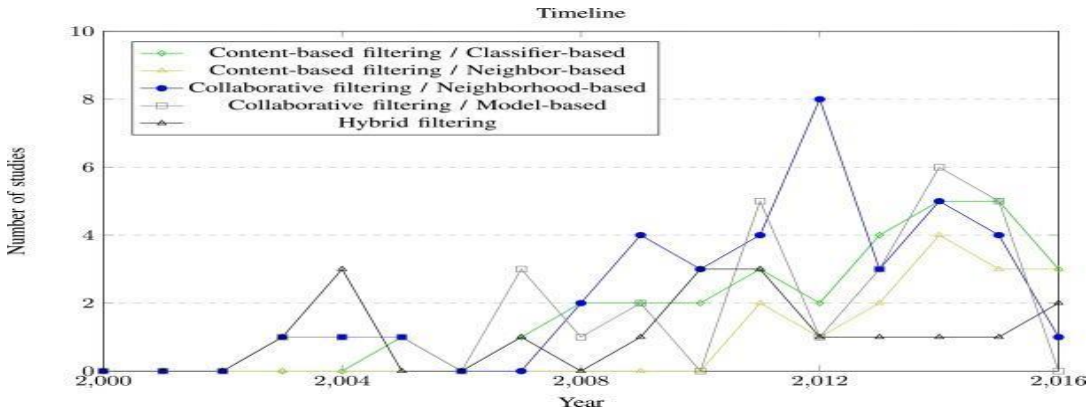


Figure 4: Graph representation of algorithm

Below is a very simple illustration of how a recommender system works in the context of e-commerce sites

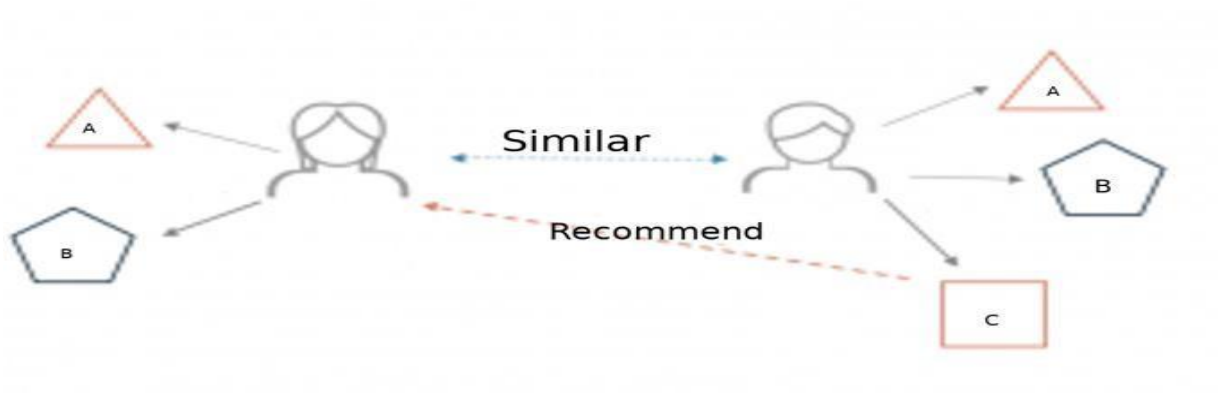


Figure 5:Example of recommendation system

In the above diagram, it is shown that the two users buy the same items A and B from one of the e-commerce stores. When both users buy the same item the similarity index of both of the users is computed, depending on which the system will recommend another item C to both of the users because it detects that both the user are similar in terms of the items they purchase.

Explanation/Introduction of the chosen problem topic

The project I will be working on is a movie recommender system. This project is important in our social life due to its strength in providing enhanced entertainment. Such a recommendation system can suggest a set of movies to users based on their interest or the popularity of movies. People may face difficulties finding the best movie for them and to overcome such difficulties I am building the system, so this movie can recommend the movie that they might like. for this project, we will use collaborative filtering, which is entirely based on a behavior and not on the context. More especially it is based on the similarity in preferences, taste, and choice of two users. It will recommend the movie to

the user based on the ratings given by the user. People might not find the ideal movie that they are searching so this can be the perfect project for those users.

The MRS (movie recommendation system) gives a framework to assist users in classifying users with similar interests, making recommender systems a vital component of websites and e-commerce applications. The major goal of this research is to suggest a recommender system for a movie recommendation system.

Problem Domain

We have hundreds of alternatives to select from for every product on the market as we are in the "age of abundance". Social networking and internet shopping are the best examples of it. The recommendation system aids in personalizing stage and assisting users in finding what they enjoy.

Selecting a movie on an online platform to watch with friends on the occasion of holidays may be difficult. Online users may view movies y simply looking at their cover or narrative. The movie suggested by the friends or family may not interest you because interests differ from person to person. So, I am trying to solve this problem by developing a movie recommendation system

1) We will predict their rating for movies they haven't watched yet to obtain the recommendations for our users but in the real world, we will likely encounter new users or movies without a history its worth mentioning such problems consider as **cold start problems** which can be handled by recommendations based on meta-information.

2) If the user does not submit ratings for the things they have purchased, the rating model will become highly sparse, potentially causing **data sparsity issues**.

3) When the single item is represented with two or more different names then the recommendation system doesn't recognize whether the term shows various items or the same which rise to **Synonymy problems.** (analyticssteps, 2022)

4) Another biggest issue is the **Scalability** of the algorithm having a real-world database under a recommendation system

5) Many products are added in the database of the recommendation system daily but only already existing products are recommended to users as newly added products are not rated so an issue of **Latency arises**

   Overspecialization problems, lack of data, changing data, changing user preferences, and unpredictable items are other problems that are found in recommendation systems

Figure 6: Recommendation system

2) Background

Research on the chosen topic

Everyone enjoys watching movies, regardless of their age, gender, race, or color. The most fascinating revelation is how distinctive our movie preferences are in terms of selections and mixes. The term "recommendation system" refers to a filtration procedure that. The purpose is to forecast a user's ratings of a domain-specific product or item. In our case, the domain-specific object is a movie, which is why the main focus of A recommendation system's purpose is to predict which movies a user would appreciate based on their tastes and other user data.

Recommendation system has become increasingly popular over the last few years and nowadays they are utilized in most of the online platforms that we used such as Netflix, Amazon, YouTube, and many others. The content of such platforms varies from movies, music, video, books, to people on professional and dating websites, to research results returned on Google. They can collect information about users' choices and use that information to improve their suggestions for the future.

Research on similar Topic

Flick Metrix Recommendation system

You'll have the best chance of finding the movie you want if you are using both Netflix and Amazon Prime. It has tons of different filters to sort the list of recommended movies. On this site, movies can be filtered by using movies rating, Rotten Tomatoes, Metacritic. Other options like genres, the year the film was released and you can also choose which streaming service you want to watch on it. Setting all the above information you will get a list of the best-recommended film with essential information, like a short description, the director and the cast, and a link to the trailer (makeuseof, 2022).

Netflix recommendation system

Netflix's machine learning-based suggestions are based on user feedback. Data is collected and refreshed in the machine learning algorithm behind the scenes every time a viewer spends time watching a film or television show. The algorithm becomes more up-to-date and accurate as a viewer views more videos. These titles are used as the first step for personalized recommendations. (Springboard India, 2019, Nov 5)

Spotify Recommendation system

Spotify uses three main types of models to recommend music. The first is Natural Language Processing (NLP) models. These compare songs using terms found in places like articles on the internet to describe them. The following list is designed to suggest lesser-known music. These Content-Based models listen to the audio and use similarities to propose songs that

are similar to the one you're listening to. The final method is Collaborative Filtering. It basically creates a user vector for each user as well as a song vector for each song. It then compares them to find music that is similar to each other as well as music that similar individuals appreciate. I'll spend the majority of this essay talking about Collaborative Filtering because it's the most common. (KHARWALMARCH, 3, 2021)

3)Solution

Explanation of proposed solution/ approach to solve the problem

For this project, we are using the collaborative filtering-based recommender system for movie recommendations. Pandas, NumPy e.t.c libraries are used while building this application using the python programing language. The solution of this project can be extended to any application or device. Kaggle datasets will be used for developing this system.

The primary objective of this solution is to determine how the recommendation system makes it easy for users to choose items they like and more enjoyable. For developing this project, I looked into AI and machine learning for a collaborative filtering algorithm system. Python programing language and collaborative filtering were used to create this system.

Used Tools and Technology

Some of the technology that has been used in this project are:

1) Anaconda Navigator

   It's a desktop graphical user interface included in the Anaconda distribution that lets you run the program and manage the conda package, environment, and channel without having to use command-line commands. In order to run, many scientific packages require a specific version of other packages. Data scientists use various environments and multiple versions of many software to distinguish between these two versions. Conda, a command-line programmer, can manage both packages and environments. Navigator is a point-and-click interface for working with packages and environments that eliminates the need to type conda commands in a terminal window.. (anaconda, 2022)

2) NumPy

It is a python library used for working with an array. Travis Oliphant created NumPy in 2005 which is an open-source project and can be used freely. It is partially written in python, but most of the part that requires fast computation is written in c or c++.
It can also be used as an efficient multi-dimensional container of generic data. Arbitrary data types can be defined which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

3) Pandas

Pandas is a fast, powerful, flexible, and easy-to-use, open-source data analysis and manipulation tool, built on top of the python programming language.

Explanation of algorithm used

It is one of the most important things which need to be considered while developing an intelligent recommender system that improves recommendations as more data about users is acquired. A website like Amazon, YouTube, and Netflix also uses the same filtering-based recommender system as a part of their sophisticated recommendation systems as it provides a recommendation to users in the basic of likes and dislikes of similar users. It works by searching a large group of people and finding smaller sets of users with tastes similar to particular users.

Content-based filtering has some limitations, to address those limitations collaborative filtering is used. There are various techniques to locate similar people or objects in collaborative filtering, as well as multiple ways to calculate ratings based on the ratings of similar users. The similarity isn't based on age, genre, or any other information about the users. It is calculated only one of the basic of the rating given by the users. (Pandey, May 9, 2019)

Collaborative filtering recommender system doesn’t rely on the machine analyzable contents so, it is capable of accurate recommendation which is the main and key advantage of the CF recommender system. In this system, users who had similar choices in the past will have similar choices in the future as well. (builtin, july 21, 2021)

There are two types of Collaborative filtering

1) User-based

Products are recommended to a user in this model based on the fact that they have been enjoyed by users who are similar to the user. For example, if Derrick and Dennis both enjoy the same films and a new film that Derick enjoys is out, we can recommend it to Dennis because they both tend to enjoy the same films (geeksforgeeks, 16 Jul, 2020)

1) Item-based

Based on past evaluations, these systems find things that are comparable. For example, if users A, B, and C provided books X and Y 5-star ratings, then when user D buys book Y, the system also recommends that they buy book X since the algorithm classifies book X and Y as similar based on user A, B, and C's ratings. (geeksforgeeks, 16 Jul, 2020)

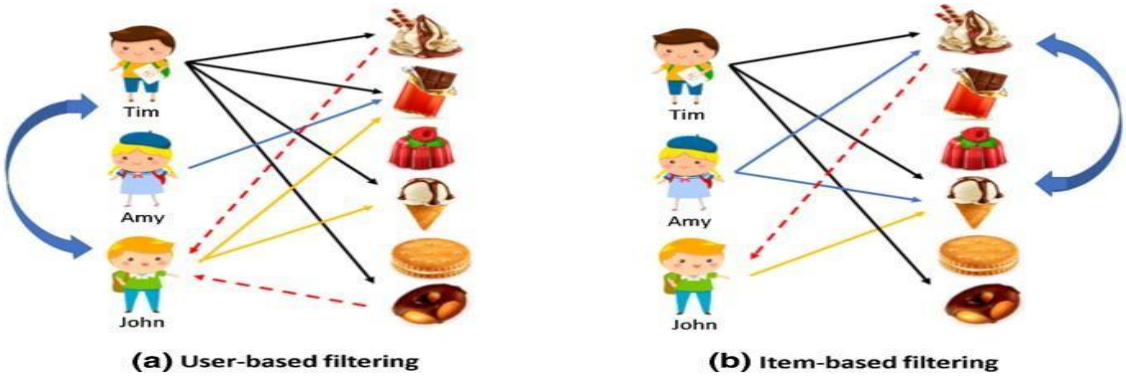


Figure 7: collaborative filtering

KNN Algorithm (K nearest neighbor)

It is a sort of supervised machine learning method that may be used to solve both classification and regression predicting problems. It is mostly utilized to solve classification and prediction problems in the industry. The following two characters might be a good way to describe KNN. It is one of the simplest machine learning algorithms based on supervised learning techniques. (tutorialspoint, 2022)

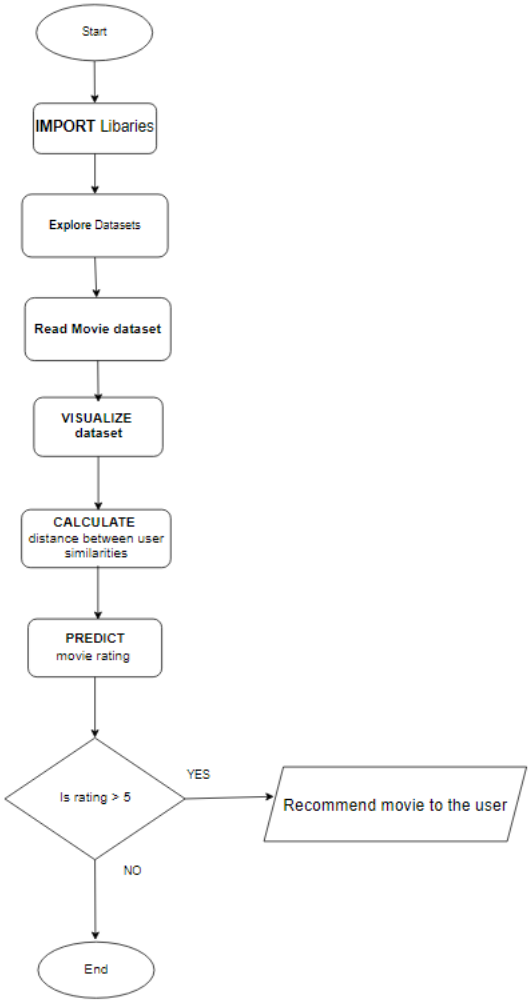
- **Lazy learning algorithm**
  It is considered a lazy learning algorithm because it doesn't have a specialized training phase and used all the data for training while classification.

- **Non-parametric learning algorithm**
  It doesn't assume anything about the underlying data so it is also considered a Non-parametric learning algorithm.

Pseudocode of the solution

```
START
IMPORT dataset
EXPLORE dataset
PREPARE dataset for training
SPLIT dataset into training and testing
TRAIN the dataset with train split
CALCULATE distance between user similarities
TEST the model
GET viewer ratings
IF ratings>5 and viewer>100:
RECOMMEND movie to similar user
END
```

Diagrammatical representation of the solution



(asq., 2022)

Figure 8: Flowchart

Explanation of development process

The system built in this project was developed by using python programing language. This project required the completion of a recommendation engine that is used for a movie recommendation system with the use of an anaconda. Anaconda is free software that runs on Windows, Linux, Mac OS X, it is also a python and R programing language for data science, Machine learning, and others which makes package management and development easier.

Writing the code for the recommendation engine Jupyter Notebook IDE is used. To install the libraries required for the project anaconda prompt is used.

All the data of the user data set that is bound to use in this project was taken from Kaggle. The raw data received were converted into usable data. The data contain the movie history of the user along with the rating given by the user after watching a movie.

To begin, a class was created using a Python script. After that, the class's focus changes to making recommendations from the implicit library. All of the user's data has been trained, data and models have been imported, and the suggestion has been generated using all of the user's data sets

Achieved Result

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import numpy as np

In [4]: cinemaname_df = pd.read_csv('movies.csv',usecols=['movieId','title','tag'],dtype={'movieId': 'int32', 'title': 'str'})
        cinemarating_df=pd.read_csv('ratings.csv',usecols=['userId', 'movieId', 'rating'],
            dtype={'userId': 'int32', 'movieId': 'int32', 'rating': 'float32'})

In [5]: cinemaname_df.head()
Out[5]:
```

	movieId	title	tag
0	1	Toy Story (1995)	funny
1	2	Jumanji (1995)	Highly quotable
2	3	Grumpier Old Men (1995)	will ferrell
3	4	Waiting to Exhale (1995)	Boxing story
4	5	Father of the Bride Part II (1995)	MMA

```
In [6]: df = pd.merge(cinemarating_df,cinemaname_df,on='movieId')
        df.head()
Out[6]:
```

	userId	movieId	rating	title	tag
0	1	1	4.0	Toy Story (1995)	funny
1	5	1	4.0	Toy Story (1995)	funny
2	7	1	4.5	Toy Story (1995)	funny
3	15	1	2.5	Toy Story (1995)	funny
4	17	1	4.5	Toy Story (1995)	funny

```
In [7]: combine_cinema_rating = df.dropna(axis = 0, subset = ['title'])
        movie_rating_count = (combine_cinema_rating.
             groupby(by = ['title'])['rating'].
```

Figure 9: SS of coding -1

Keshab Acharya
19031062

In [7]:
```
combine_cinema_rating = df.dropna(axis = 0, subset = ['title'])
movie_rating_Count = (combine_cinema_rating.
     groupby(by = ['title'])['rating'].
     count().
     reset_index().
     rename(columns = {'rating': 'totalRatingCount'})
     [['title', 'totalRatingCount']]
    )
movie_rating_Count.head()
```

Out[7]:

	title	totalRatingCount
0	'71 (2014)	1
1	'Hellboy': The Seeds of Creation (2004)	1
2	'Round Midnight (1986)	2
3	'Salem's Lot (2004)	1
4	'Til There Was You (1997)	2

In [8]:
```
rating_with_totalCinemaRatingCount = combine_cinema_rating.merge(movie_rating_Count, left_on = 'title', right_on = 'title', how =
rating_with_totalCinemaRatingCount.head()
```

Out[8]:

	userId	movieId	rating	title	tag	totalRatingCount
0	1	1	4.0	Toy Story (1995)	funny	215
1	5	1	4.0	Toy Story (1995)	funny	215
2	7	1	4.5	Toy Story (1995)	funny	215
3	15	1	2.5	Toy Story (1995)	funny	215
4	17	1	4.5	Toy Story (1995)	funny	215

In [9]:
```
pd.set_option('display.float_format', lambda x: '%.3f' % x)
print(movie_rating_Count['totalRatingCount'].describe())
```

```
count   9719.000
```

Figure 10: SS of coding-2

Keshab Acharya
19031062

In [9]:
```
pd.set_option('display.float_format', lambda x: '%.3f' % x)
print(movie_rating_count['totalRatingCount'].describe())
```

```
count   9719.000
mean      10.375
std       22.406
min        1.000
25%        1.000
50%        3.000
75%        9.000
max      329.000
Name: totalRatingCount, dtype: float64
```

In [10]:
```
popularity_threshold = 50
rating_popular_cinema= rating_with_totalCinemaRatingCount.query('totalRatingCount >= @popularity_threshold')
rating_popular_cinema.head()
```

Out[10]:

	userId	movieId	rating	title	tag	totalRatingCount
0	1	1	4.000	Toy Story (1995)	funny	215
1	5	1	4.000	Toy Story (1995)	funny	215
2	7	1	4.500	Toy Story (1995)	funny	215
3	15	1	2.500	Toy Story (1995)	funny	215
4	17	1	4.500	Toy Story (1995)	funny	215

In [11]:
```
rating_popular_cinema.shape
```

Out[11]:
```
(41362, 6)
```

In [12]:
```
## creating a Pivot matrix

cinema_features_df=rating_popular_cinema.pivot_table(index='title',columns='userId',values='rating').fillna(0)
cinema_features_df.head()
```

Out[12]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
title																					
10 Things I Hate About You (1999)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	3.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000
12 Angry Men (1957)	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	0.000	...	5.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2001: A Space Odyssey (1968)	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	...	0.000	0.000	5.000	0.000	0.000	5.000	0.000	3.000	0.000	4.500
28 Days Later (2002)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.500	0.000	5.000
300 (2007)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.000	...	0.000	0.000	0.000	0.000	3.000	0.000	0.000	5.000	0.000	4.000

5 rows × 606 columns

Figure 11: SS of coding-3

Keshab Acharya
19031062

```
In [13]: from scipy.sparse import csr_matrix

         cinema_features_df_matrix = csr_matrix(cinema_features_df.values)

         from sklearn.neighbors import NearestNeighbors


         model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
         model_knn.fit(cinema_features_df_matrix)

Out[13]: NearestNeighbors(algorithm='brute', metric='cosine')

In [14]: cinema_features_df.shape

Out[14]: (450, 606)

In [15]: query_index = np.random.choice(cinema_features_df.shape[0])
         print(query_index)
         distances, indices = model_knn.kneighbors(cinema_features_df.iloc[query_index,:].values.reshape(1, -1), n_neighbors = 6)

         311

In [16]: cinema_features_df.head()

Out[16]:
```

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
title																					
10 Things I Hate About You (1999)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	3.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000
12 Angry Men (1957)	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	0.000	...	5.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2001: A Space Odyssey (1968)	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	...	0.000	0.000	5.000	0.000	0.000	5.000	0.000	3.000	0.000	4.500
28 Days Later (2002)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.500	0.000	5.000
300 (2007)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.000	...	0.000	0.000	0.000	0.000	3.000	0.000	0.000	5.000	0.000	4.000

5 rows × 606 columns

```
In [17]: for i in range(0, len(distances.flatten())):
             if i == 0:
                 print('Recommendations for {0}:\n'.format(cinema_features_df.index[query_index]))
             else:
                 print('{0}: {1}, with distance of {2}:'.format(i, cinema_features_df.index[indices.flatten()[i]], distances.flatten()[i]))

         Recommendations for Pleasantville (1998):

         1: My Cousin Vinny (1992), with distance of 0.47539228200912476:
         2: Big (1988), with distance of 0.5209939479827881:
         3: Who Framed Roger Rabbit? (1988), with distance of 0.5248650908470154:
         4: Wayne's World (1992), with distance of 0.5319476127624512:
         5: Truman Show, The (1998), with distance of 0.5350018739700317:
```

Figure 12: SS of coding- 4

conclusion

Analysis of the work done

The table below shows the analysis of work done in this project. This project includes all the research and work I have done to complete a recommendation system and collaborative filtering system on the AI topic.

S.NO	Topic	Status
1	Research on AI Topic	Completed
2	Choose and confirm AI Topic	Completed
3	Research on Recommendation system	Completed
4	Research on problem domain and solution	Completed
5	Research on collaborating filtering method	Completed
6	Choose and confirm AI algorithm	Completed
7	Research on a similar project	Completed
8	Pseudocode and flowchart	Completed
9	Finalize research document	Completed
10	Start of the project	Completed
11	Train datasets	Completed
12	Project outcome	Completed
13	Finalization of the project	Completed
14	Finalization of the document	Completed

How solution addresses the real world Problems

The system can have based on a variety of approaches such as content-based, collaborative approaches, hybrid but furthermost the movie recommendation system is based on collaborative filtering. in this system,

1) The user is asked to rate movies that the user has already seen and these ratings are applied to the recommended other movie to the users which users haven't perceived by utilizing collaborative filtering which is based on similar ratings which help all individuals to get the best genre of the movie they want**.**

2) People don’t have to waste time searching for the best movie as they can easily find the movie in the recommendation system. It satisfied all the users by providing the best movie which fits them. The system will be able to recommend to the user items outside their preferences.

We can find several movie recommendation systems today, but they all lack certain features. They are yet to be improvised and get better recommendations with better selections. They recommend movies based on user choice but the rating should be considered as well.

Future work

Following the completion of this project, we studied and learned about AI and recommendation engines and how knn algorithms work. We also developed a simple movie recommendation system. we may be also engaged in doing certain works given below in the future.

1) Developing the Content-based and hybrid-based program in the future to gain more knowledge on the recommendation system
2) Make a program more reliable and faster.

Bibliography

Agrawal, S. K., July 13, 2021. *analyticsvidhya.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2021/07/recommendation-system-understanding-the-basic-concepts/#:~:text=A%20recommendation%20system%20is%20a,suggests%20relevant%20items%20to%20users.
[Accessed 25 01 2022].
anaconda, 2022. *anaconda.* [Online]
Available at: https://docs.anaconda.com/anaconda/navigator/index.html#:~:text=Anaconda%20Navigator%20is%20a%20desktop,without%20using%20command%2Dline%20commands.
[Accessed 25 01 2022].
asq., 2022. *asq..* [Online]
Available at: https://asq.org/quality-resources/flowchart
[Accessed 25 01 2022].
builtin, july 21, 2021. *builtin.* [Online]
Available at: https://builtin.com/data-science/collaborative-filtering-recommender-system
[Accessed 25 01 2022].
Copeland, B., Dec 14, 2021. *britannica.* [Online]
Available at: https://www.britannica.com/technology/artificial-intelligence
[Accessed 25 01 2022].
F.O.Isinkaye, Y., November 2015,. *Recommendation systems:,* Egypt: Egyptian Informatics Journal.
geeksforgeeks, 16 Jul, 2020. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/item-to-item-based-collaborative-filtering/
[Accessed 25 01 2022].
July 21, 2021, July 21, 2021. *July 21, 2021.* [Online]
Available at: https://c.com/data-science/collaborative-filtering-recommender-system
[Accessed 25 01 2022].
KHARWALMARCH, A., 3, 2021. *thecleverprogrammer.* [Online]
Available at: https://thecleverprogrammer.com/2021/03/03/spotify-recommendation-system-with-machine-learning/#:~:text=The%20Spotify%20recommendation%20system%20uses,provide%20a%20better%20user%20experience.
[Accessed 25 01 2022].
makeuseof, 2022. *makeuseof.* [Online]
Available at: https://www.makeuseof.com/tag/film-recommendation-sites/
[Accessed 25 01 2022].
Pandey, P., May 9, 2019. *towardsdatascience.* [Online]
Available at: https://towardsdatascience.com/the-remarkable-world-of-recommender-systems-bff4b9cbe6a7
[Accessed 25 01 2022].
Saleh, Z., April 2019. *Artificial Intelligence ,* Egypt: The British university in Egypt.
sciencedaily, January 23, 2022. *sciencedaily.* [Online]
Available at: https://www.sciencedaily.com/terms/artificial_intelligence.htm
[Accessed 25 01 2022].
Springboard India, 2019, Nov 5. *Springboard India.* [Online]
Available at: https://medium.com/@springboard_ind/how-netflixs-recommendation-engine-

works-bd1ee381bf81
[Accessed 05 01 2022].
tutorialspoint, 2022. *tutorialspoint.* [Online]
Available at:
https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm
[Accessed 26 01 2022].
Zakir Hussain, A. B., 2021. *sciencedirect.* [Online]
Available at: https://www.sciencedirect.com/topics/computer-science/recommender-systems
[Accessed 25 01 2022].