University of Mumbai
2020-2021

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING
8, Saboo Siddik Road, Byculla, Mumbai - 400 008
Department of Computer Engineering

Project Report

On

“GAME RECOMMENDATION SYSTEM”

By

BOOTWALA YOUHAAN KARIM
GAUTAM HRITHIK VINAY
JUWALEY AMAAN SAJID

Guided By
Asst. Prof SAIQA KHAN

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING
8, Saboo Siddik Road, Byculla, Mumbai -400 008.
This is to certify that,

Roll No	Name
3117007	Youhaan Karim Bootwala
3117010	Hrithik Vinay Gautam
3117016	Amaan Sajid Juwaley

Of Third Year (T.E Semester VI) degree course in Computer Engineering, have completed the specified project report on:

GAME RECOMMENDATION SYSTEM

As a partial fulfilment of the project work in a satisfactory manner as per the rules of the curriculum laid by the University of Mumbai, during the Academic Year July 2020 — June 2021.

Internal Guide

External Examiner

Internal Examiner

Head Of Department

Principal

This project report entitled

GAME RECOMMENDATION SYSTEM

By

Youhaan Bootwala

Hrithik Gautam

Amaan Juwaley

is approved for the degree of Computer Engineering.

EXAMINERS

1.

2.

SUPERVISORS

1.

2.

Date:

Place:

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to our parents for motivating and encouraging us throughout the career.

We wish to express our sincere thanks to our principal Dr. Ganesh Kame, M.H. Saboo Siddik College of Engineering for providing us all the facilities, support and wonderful environment to meet our project requirements.

We would also take the opportunity to express our humble gratitude to our Head of Department of Computer Engineering **Dr. Zainab Pirani** for supporting us in all aspects and for encouraging us with her valuable suggestions to make our project success.

We our highly thankful to our internal project guide **Prof. Saiqa Khan** whose valuable guidance helped us understand the project better, their constant guidance and willingness to share their vast knowledge us understand this project and its manifestations in great depths and helped us to complete the project successfully.

We would also like to acknowledge with much appreciation the role of the staff of the Computer Department, especially the Laboratory staff, who gave the permission to use the labs when needed and the necessary material to complete the project.

We would like to express our gratitude and appreciate the guidance given by other supervisors and project guides, their comments and tips helped us in improving our presentation skills. Although there may be many who remain unacknowledged in this humble note of appreciation but there are none who remain unappreciated.

Bootwala Youhaan

Gautam Hrithik

Juwaley Amaan

Table of Contents

List of Figures

Abstract

The gaming industry as a whole generates more than 100 billion dollars every year and it has been on an increase from 100 billion since 2017. It makes more than the movie industry, online streaming industry and the music industry combined. Thus, making the industry much more lucrative and much more data filled than any other. So, to process that much amount of data there are techniques which can be used such as content based filtering and collaborative based filtering, we are using content-based filtering to make our game recommendation system. Our system evaluates the number of hours the game has been played by the user, the tags it has been categorized with, the genre it belongs to and finally the publisher of the game; then recommends them games based on those criteria to make it easier for them to know which games to buy or play based on their own preferences and interests. The data sets are processed using pandas data frame.

Chapter 1: Introduction and Motivation

1.1 Introduction

In modern society, gaming has become a part of popular culture that everyone enjoys regardless of gender or age. With the advent of various gaming platforms, such as mobile, high-end PC, and console games, users can enjoy playing games as per convenience; thus, their needs (or wishes) regarding games have diversified. Recently, a new genre that combines several existing game genres has been researched and developed to meet the needs of users.

Owing to such efforts, the gaming industry has grown and will continue to expand steadily. It has been analysed that the total size of the gaming industry will grow too approximately $196 billion by 2022, if the mobile and online markets are combined. Owing to the rapid growth of the gaming industry, a large number of games are being released in an increasingly short period of time and game development has become faster. Indeed, statistics, show that a total of 9050 games have been released in the year 2018. This shows a 28% increase over the year 2017.

Thus, many games have been released in a short period of time, making it difficult for people to find the games they want. To address such difficulties, game recommendation systems and services have emerged.

Game recommendation systems in the past tended to focus on the classification of games rather than on recommendation. In this system, it was possible to search without specifying a keyword, and detailed results were provided according to category. However, detailed analysis and information about the game were insufficient, and the absence of self-assessment and review and recommendation systems resulted in a failure to meet the needs of users. In addition, existing game recommendation systems were developed for gamers who played a variety of games. In this process, additional reviews were utilized to encourage users to review and evaluate games based on additional criteria. However, with such recommendation systems, effective recommendations were only available to users who had sufficient gaming experience. For users having little or no knowledge or experience of games, appropriate game recommendation was almost impossible.

This project has been made to promote different types of games, provide children as well as teenagers with the type of games they love the most as well as explore other games which they might become interested to play. As we all know that sometimes there are particular games, we might love but can't find it anywhere or able to find it after an intensive search on the Internet.

So, through this system, the process of finding that particular game becomes a lot easier and thus, it saves the energy and time of the gamer. Everyone can access this system if they have a good net connection during the day and night which means that this system is active for 24 hours so gamers won't have a problem and can engage in gaming whenever they want. For this project we have used content-based filtering algorithm for the rating of the game and for users to register themselves in the game.

Content-based recommender systems work with profiles of users that are created at the beginning. A profile has information about a user and his taste. Taste is based on how the user rated items. Generally, when creating a profile, recommender systems make a survey, to get initial information about a user in order to avoid the new-user problem. In the recommendation process, the engine compares the items that were already positively rated by the user with the items he didn’t rate and looks for similarities. Those items that are mostly similar to the positively rated ones, will be recommended to the user.

The Content based algorithm works as follows:

- Step 1: Registered users create their user profile, where a login and password are created by the user.
- Step 2: Information is stored in the database which is queried to display a list of available games to the user.
- Step 3: Once the user selects the games displayed to them all the details of the games are retrieved for database
- Step 4: Now the rating given by the user for the games are considered and average of rating is calculated according to the genre of the game.
- Step 5: Now all the games from the database are retrieved with threshold value as average of rating taken from users’ profile and games from database are retrieved.
- Step 6: Final Recommendations are displayed to user based on step 5.

As the content-based model links with these content texts, it has some inner links with the field of Natural Language Processing, we will focus on a basic Natural Language Processing model TF-IDF here, which is concentrating on the weight of each word in the content text. Each word’s weight will be weighted by not only its frequency on this content text, but by also the frequency of all the text set, and the algorithm of TF-IDF is: W (i, j) =tf (i, j) X log(N/df(i)) tf (i, j) =no of occurrences of i in j df(i)=no of documents containing I N=total no of documents After we got each words’ TF-IDF values, linear-kernel used here to compute this game’s descriptions TF-IDF weights with other game’s descriptions TF-IDF weights.

The ground-truth about linear-kernel is: if there are two matrices that have the same dimensions, some big values in the same position of these matrices will make their multiply still big in the result matrix calculated by linear-kernel. Then we will make a sequence of these results calculated by the linear kernel, and if the results are high, we will put them as the most relative recommendations for this game. We have also used cosine similarity which involves us taking our User x Item matrix and cantering our “ratings” by subtracting the mean “rating” for the entire row. Note, we will not actually be using ratings for this, but substituting that idea for play time. As always, we will be using Python to make this happen.

Additionally, we’ll need to import a few libraries to help facilitate and make the processing a bit easier like Pandas, Sklearn, Numpy, and Matplotlib

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

1.2 Aim

The aim is to promote and recommend games that match the user preferences
and then user-based collaborative filtering is applied on the individual ratings of
the games for a particular gamer to find similarity between those gamers and also
making it more accessible to the gamers based on their preferences.

1.3 Objectives

Games are one of the most famous ways through which children and teenagers
entertain themselves. Goals of this system are:

- To make the games more accessible to the general public
- To promote the games so public can access more categories of games
- To increase the number of gamers
- To increase the revenue for developers and staff involved with that game
- To recommend the particular category of games that the gamers are interested in
- Increase the interest in other categories of games
- To learn problem-solving, strategy, trust, calculated risk-taking, how to adapt to unforeseen issues and how to share.
- To practice fair play and having respect for other players as well regardless of winning or losing
- To teach the gamers to participate in order to work harder and smarter to achieve a goal
- To contribute with the integral part of a child's development as playing with joy nourishes the soul

1.4 Motivation

Our motivation to make this system is to recommend different kinds of games based on different categories thus providing them with an ample number of games to choose from thus contributing to their integral development, increasing sportsmanship with respect to different game

Chapter 2: Proposed method and design Implementation

2.1 Proposed Method

The method used by us to recommend games to the user is by using Cosine Similarity to find the closest game with the keywords combined by using Tf-idf. The detailed method used is described below:

2.1.1 Using Content Based Filtering

Content-based Filtering is a Machine Learning technique that uses similarities in features to make decisions. This technique is often used in recommender systems, which are algorithms designed to advertise or recommend things to users based on knowledge accumulated about the user.

A Content-Based Recommender works by the data that we take from the user, either explicitly (rating) or implicitly (clicking on a link). By the data we have about the user's play-taste by which we suggest the user the games that they might like. As the user provides more input or take more actions on the recommendation, the engine becomes more accurate.

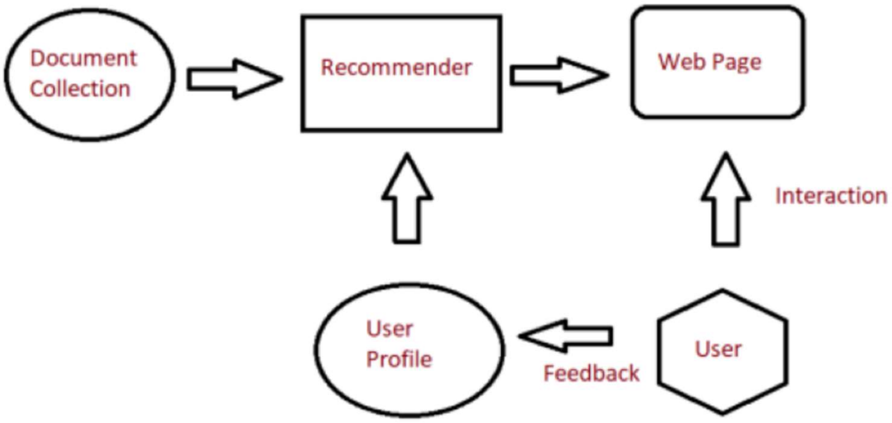


Fig. 2.1 General Game Recommender

It relies on similarities between features of the items. It recommends items to a customer based on previously rated highest items by the same customer. List of features about these items needs to be generated.

- Each item will have an item profile
- A table structure will list these properties
- Comparing what and how many features match and collect scores
- Recommend highest scored item
- Code will be based on an algorithm, by given some item, the most similar item will be found
- Best scoring match will be provided to the user
- This method relies on item features only, and not the user preferences.

Content-based evaluation item profile can be seen as vector. Measured 0 or 1, depends on YES or NO of containing a feature inside that item.

2.1.2 TF-IDF

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

To pick important words, TF-IDF method was used. It will calculate the total words of the data frame column which we will process using Pandas Dataframe.

2.1.3 Cosine Similarity

We will use the Cosine Similarity from sklearn, as the metric to compute the similarity between two movies. Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1.

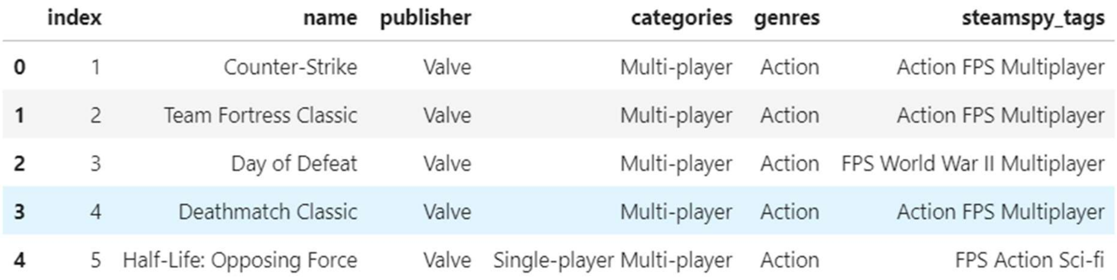
0 means no similarity, whereas 1 means that both the items are 100% similar.

2.2 Implementation

We have used two datasets in order to recommend games. The first dataset comprises of the Game Data which has all the Game related data including tags such as Genre, Publisher, Categories, etc.

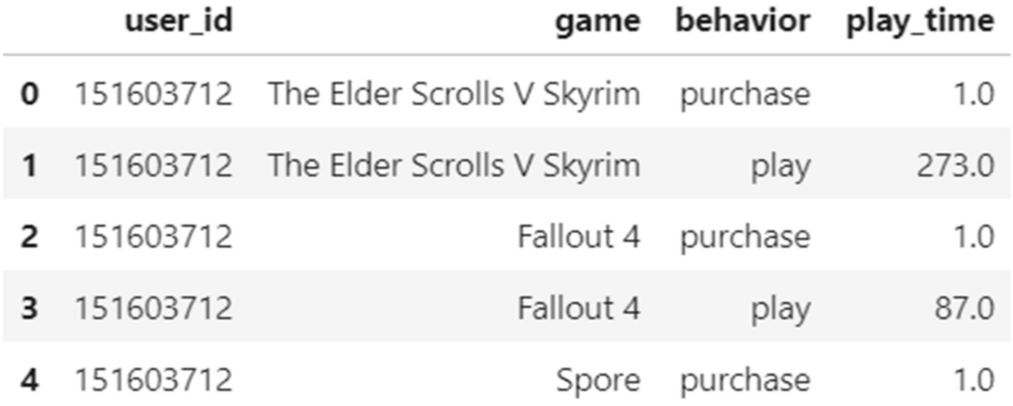
The second dataset comprises of the User data which has all the user related data such as User ID, Games that he plays with its playtime and its playing behaviour.

The datasets are shown below:



	index	name	publisher	categories	genres	steamspy_tags
0	1	Counter-Strike	Valve	Multi-player	Action	Action FPS Multiplayer
1	2	Team Fortress Classic	Valve	Multi-player	Action	Action FPS Multiplayer
2	3	Day of Defeat	Valve	Multi-player	Action	FPS World War II Multiplayer
3	4	Deathmatch Classic	Valve	Multi-player	Action	Action FPS Multiplayer
4	5	Half-Life: Opposing Force	Valve	Single-player Multi-player	Action	FPS Action Sci-fi

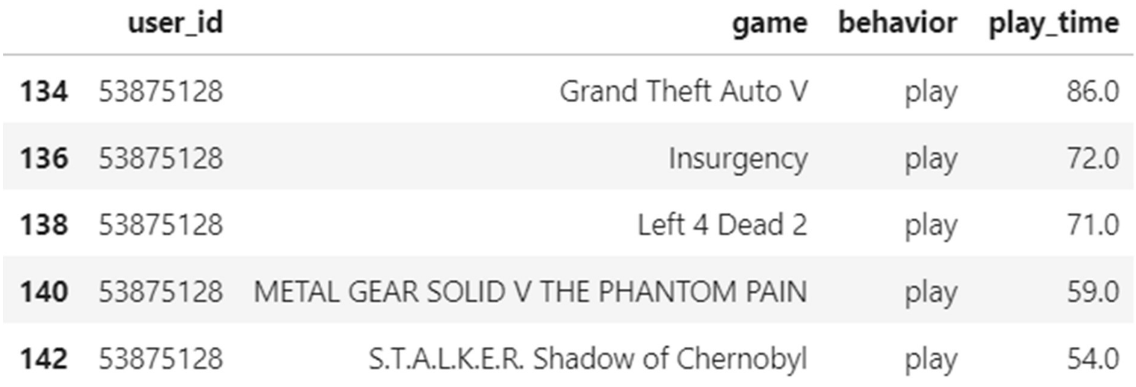
Fig. 2.2 Game Data



	user_id	game	behavior	play_time
0	151603712	The Elder Scrolls V Skyrim	purchase	1.0
1	151603712	The Elder Scrolls V Skyrim	play	273.0
2	151603712	Fallout 4	purchase	1.0
3	151603712	Fallout 4	play	87.0
4	151603712	Spore	purchase	1.0

Fig. 2.3 User Data

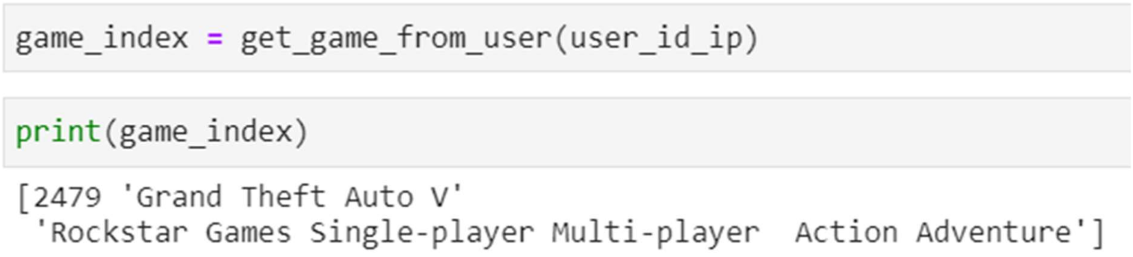
First, we process the dataset by using Pandas Dataframes import the dataset for further processing.
After importing, we ask the user to enter their user_id so as to get the games that they play along with the playtime for that particular game.
Then we categorised the resulting dataframe where the behaviour is ‘play’.



	user_id	game	behavior	play_time
134	53875128	Grand Theft Auto V	play	86.0
136	53875128	Insurgency	play	72.0
138	53875128	Left 4 Dead 2	play	71.0
140	53875128	METAL GEAR SOLID V THE PHANTOM PAIN	play	59.0
142	53875128	S.T.A.L.K.E.R. Shadow of Chernobyl	play	54.0

Fig. 2.4 User Games

Then we get the game with the maximum playtime from the above dataframe as



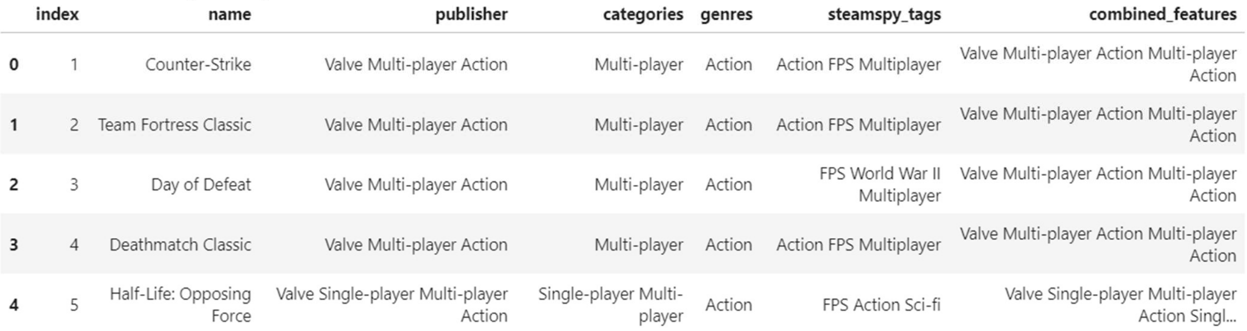
```
game_index = get_game_from_user(user_id_ip)
```

```
print(game_index)
```

```
[2479 'Grand Theft Auto V'
 'Rockstar Games Single-player Multi-player  Action Adventure']
```

Fig. 2.5 Game Index

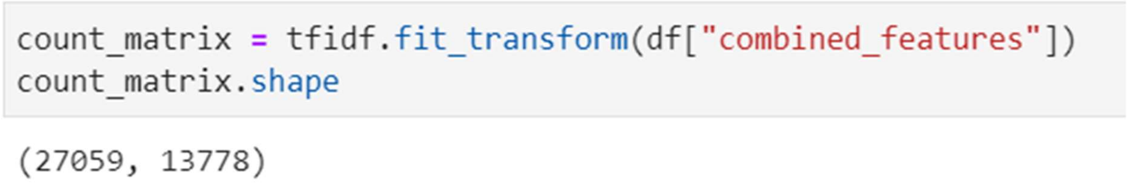
Now we use a combine function to combine the tags which will be necessary for recommending the games from the Game Dataset



	index	name	publisher	categories	genres	steamspy_tags	combined_features
0	1	Counter-Strike	Valve Multi-player Action	Multi-player	Action	Action FPS Multiplayer	Valve Multi-player Action Multi-player Action
1	2	Team Fortress Classic	Valve Multi-player Action	Multi-player	Action	Action FPS Multiplayer	Valve Multi-player Action Multi-player Action
2	3	Day of Defeat	Valve Multi-player Action	Multi-player	Action	FPS World War II Multiplayer	Valve Multi-player Action Multi-player Action
3	4	Deathmatch Classic	Valve Multi-player Action	Multi-player	Action	Action FPS Multiplayer	Valve Multi-player Action Multi-player Action
4	5	Half-Life: Opposing Force	Valve Single-player Multi-player Action	Single-player Multi-player	Action	FPS Action Sci-fi	Valve Single-player Multi-player Action Singl...

Fig. 2.6 Combined Features

After this we find out the count matrix using Tf-idf Fit Transform

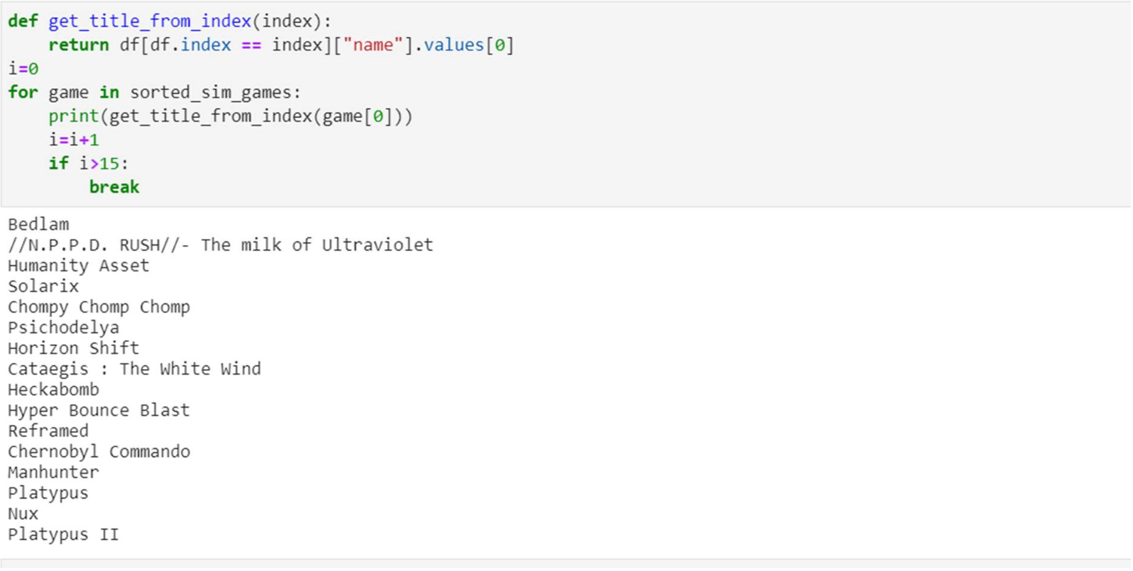


```
count_matrix = tfidf.fit_transform(df["combined_features"])
count_matrix.shape
```

```
(27059, 13778)
```

Fig. 2.7 Using Tf-idf Fit Transform

Then we find the cosine similarity for the count matrix and recommend the games having the closest distance from the list of enumerated cosine similarity. Then we display the 15 game recommendations that we get via the sorted list.

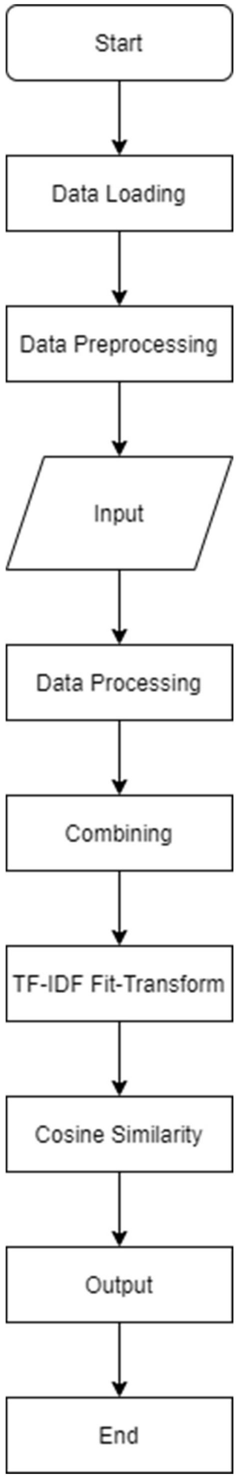


```
def get_title_from_index(index):
    return df[df.index == index]["name"].values[0]
i=0
for game in sorted_sim_games:
    print(get_title_from_index(game[0]))
    i=i+1
    if i>15:
        break
```

```
Bedlam
//N.P.P.D. RUSH//- The milk of Ultraviolet
Humanity Asset
Solarix
Chompy Chomp Chomp
Psichodelya
Horizon Shift
Cataegis : The White Wind
Heckabomb
Hyper Bounce Blast
Reframed
Chernobyl Commando
Manhunter
Platypus
Nux
Platypus II
```

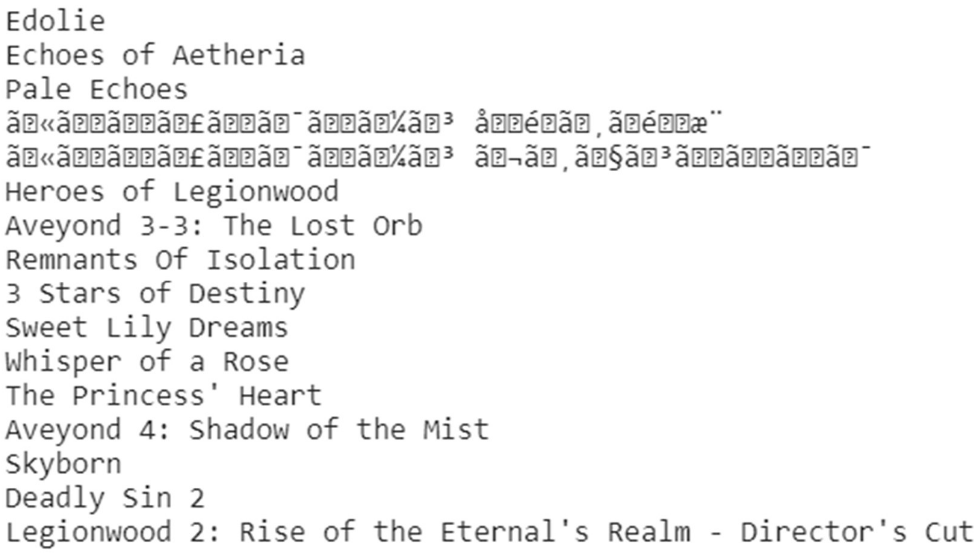
Fig. 2.8 Recommended Games

Chapter 3: Work Flow



Chapter 4: Results

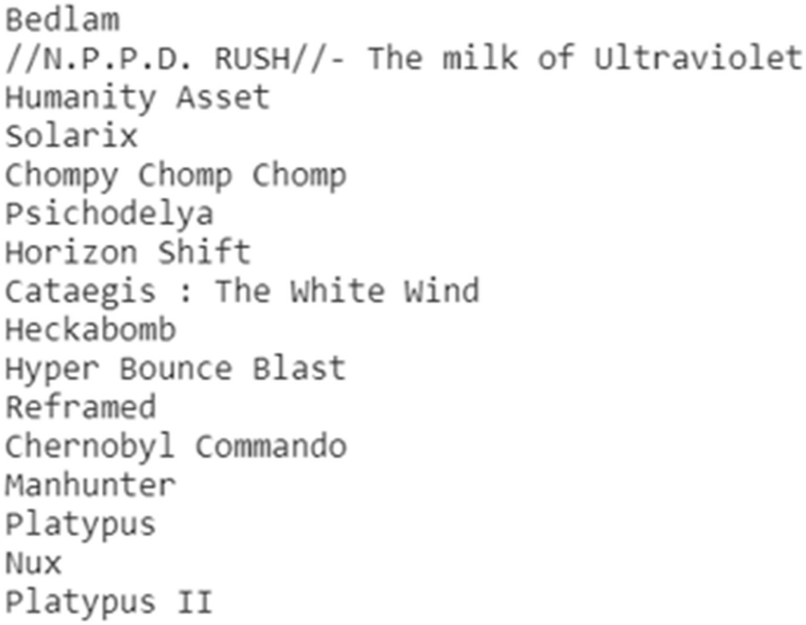
1. For user id = 151603712
   Recommended Games are:



```
Edolie
Echoes of Aetheria
Pale Echoes
ã⍰«ã⍰⍰ã⍰⍰ã⍰£ã⍰⍰ã⍰¯ã⍰⍰ã⍰¼ã⍰³ å⍰⍰é⍰ã⍰¸ã⍰é⍰⍰æ¨
ã⍰«ã⍰⍰ã⍰⍰ã⍰£ã⍰⍰ã⍰¯ã⍰⍰ã⍰¼ã⍰³ ã⍰¬ã⍰¸ã⍰§ã⍰³ã⍰⍰ã⍰⍰ã⍰⍰ã⍰¯
Heroes of Legionwood
Aveyond 3-3: The Lost Orb
Remnants Of Isolation
3 Stars of Destiny
Sweet Lily Dreams
Whisper of a Rose
The Princess' Heart
Aveyond 4: Shadow of the Mist
Skyborn
Deadly Sin 2
Legionwood 2: Rise of the Eternal's Realm - Director's Cut
```

Fig. 4.1 Game for user 151603712

2. For user id = 53875128
   Recommended Games are:



```
Bedlam
//N.P.P.D. RUSH//- The milk of Ultraviolet
Humanity Asset
Solarix
Chompy Chomp Chomp
Psichodelya
Horizon Shift
Cataegis : The White Wind
Heckabomb
Hyper Bounce Blast
Reframed
Chernobyl Commando
Manhunter
Platypus
Nux
Platypus II
```

Fig. 4.2 Game for user 53875128

Chapter 5: Conclusion and Future Scope

5.1 Conclusion

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems.

We came up with a strategy that focuses on dealing with user's personal interests and based on their games played, games are recommended to them. This strategy helps in improving accuracy of the recommendations. A personal profile was already created for each user, where each user has access to their own history of the games they have played or purchased. Based on the tags of those games with the highest playtime, games are being recommended. Using two datasets (one for each i.e., User Data and Game Data) improved the performance of the recommender and it thus helped us to get the most relevant games to recommend.

5.2 Future Scope

The scope of this project is to provide an easy option for the Gamers so that they can easily access to the particular type of games they are fond with. It also saves the time of the gamers as all different categories of games can be found on this one system so they don't have to go through the pain for searching on different websites on the Internet

This system can be accessed anywhere who has net connection at any time of day or night, thus benefitting the gamers. It also plays a major role in promoting different types of games thus generating a huge revenue for the developers and the staff involved in making that game

As we all know, that playing games is one of the most loved hobbies of children and even adults, the scope of the Game Recommendation System is wide thus becoming the most important factor for the promotion of the games

The Asia Pacific region is at the heart of the global video gaming industry. According to estimates, there were over 1.5 billion video gamers in the region in 2020, generating a combined revenue of 78.3 billion U.S. dollars. This represents almost double the revenue generated in the second largest region, North America. Given how much money the average gamer devotes to their hobby, it is no wonder that the gaming industry is worth billions – gamers worldwide spent an average of over 123 U.S. dollars on gaming over a three-month period in 2018, which included purchases on full games, downloads, and supporting live streamers. PayPal seemed to be the payment method of choice for many gamers, while four percent gained access to the latest gaming releases via a bank transfer. Since games are played by a wide number of players around the world there is no doubt that the Gaming Industry is one of the largest industries around the world

so therefor it also plays a huge role for making career choices. It makes up for a large segment on the employment front. According to statistics, game industry jobs provide employment to as many as 1.7 million individuals, with the employment rate growing by 62,000 jobs (on average) every year.
So, with an industry, so vast and growing so fast among the public, Game Recommendation System provides a great environment to explore with its services for gamers as well as employment for the Game Staff.

References

https://www.onely.com/blog/what-is-tf-idf/

https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

https://towardsdatascience.com/understanding-cosine-similarity-and-its-application-fd42f585296a

https://pandas.pydata.org/docs/user_guide/

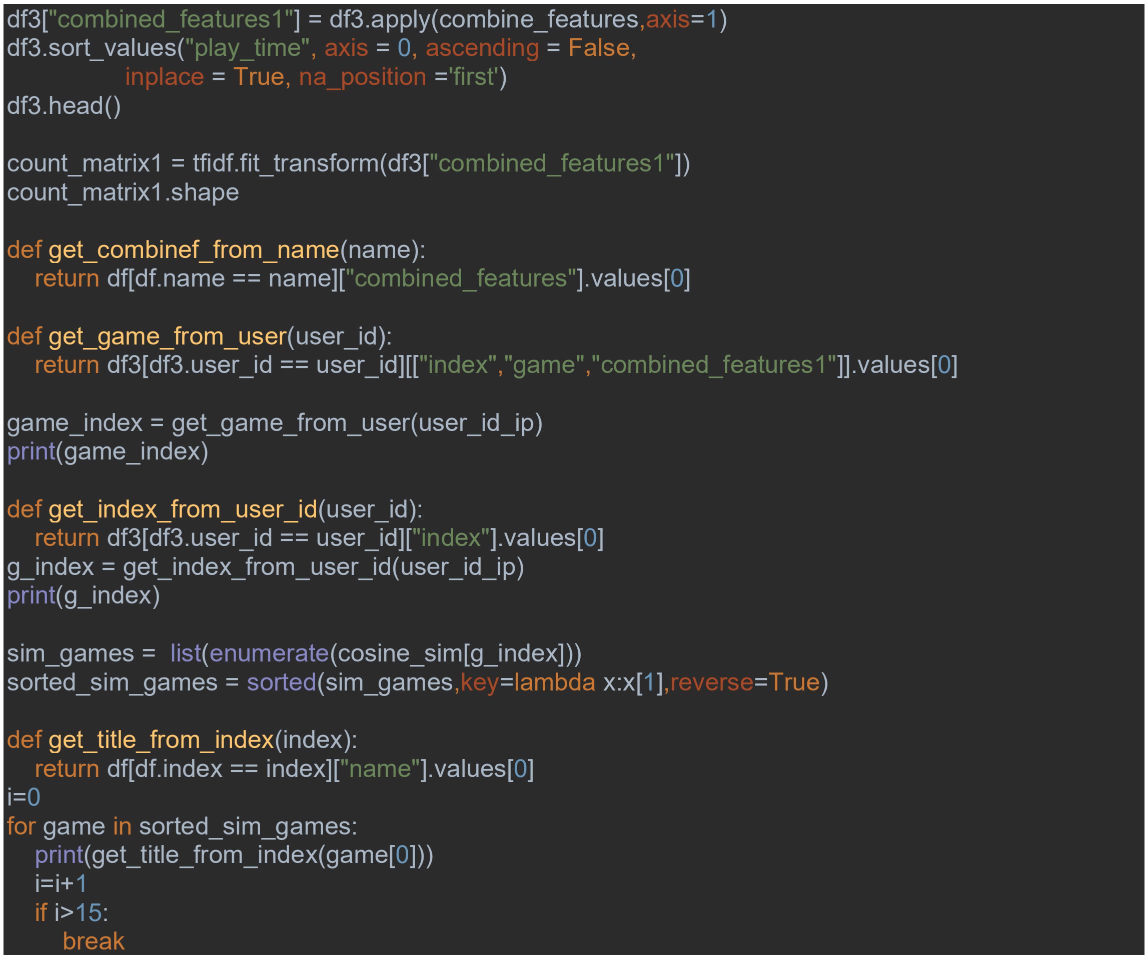
https://stackoverflow.com/questions/57507832/unable-to-allocate-array-with-shape-and-data-type

https://www.kaggle.com/nikdavis/steam-store-games?select=steamspy_tag_data.csv

https://www.kaggle.com/tamber/steam-video-games

Source Code

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

def combine_features(row):
    return row["publisher"]+" "+row["categories"]+" "+row["genres"]
df = pd.read_csv("steam_games.csv", encoding = "ISO-8859-1")
df.columns

features = ['genres','steamspy_tags','categories','publisher']
for feature in features:
    df[feature] = df[feature].fillna('')
df.head()

df["combined_features"] = df.apply(combine_features,axis=1)
df.head()

df4 = df
df4.head()

tfidf = TfidfVectorizer(stop_words='english')
count_matrix = tfidf.fit_transform(df["combined_features"])
count_matrix.shape
count_matrix

cosine_sim = cosine_similarity(count_matrix)

user_tags = ['user_id', 'game', 'behavior', 'play_time', '0']
df2 = pd.read_csv("steam_user.csv", names=user_tags)
df2 = df2.drop(['0'],axis=1)
df2.head()

user_id_ip = int(input("Enter your user id"))

user_data_ipr = df2[(df2.user_id == user_id_ip) & (df2.behavior == 'play') & (df2.play_time >= 10)]
user_data_ipr.head()

df3 = pd.merge(df, user_data_ipr, left_on="name", right_on="game")
df3.head()

print(df4['categories'].dtypes)

temp = df4["categories"].copy()
df4["publisher"]= df4["publisher"].str.cat(temp, sep =" ")
df4.head()

temp2 = df4["genres"].copy()
df4["publisher"]= df4["publisher"].str.cat(temp2, sep =" ")
df4.head()

df4 = df4.drop(["categories", "genres"], axis=1)
df4.head()

df4 = df4.rename(columns = {"publisher":"tags"}, inplace = False)
df4.head()
```



```
df3["combined_features1"] = df3.apply(combine_features,axis=1)
df3.sort_values("play_time", axis = 0, ascending = False,
                inplace = True, na_position ='first')
df3.head()

count_matrix1 = tfidf.fit_transform(df3["combined_features1"])
count_matrix1.shape

def get_combinef_from_name(name):
    return df[df.name == name]["combined_features"].values[0]

def get_game_from_user(user_id):
    return df3[df3.user_id == user_id][["index","game","combined_features1"]].values[0]

game_index = get_game_from_user(user_id_ip)
print(game_index)

def get_index_from_user_id(user_id):
    return df3[df3.user_id == user_id]["index"].values[0]
g_index = get_index_from_user_id(user_id_ip)
print(g_index)

sim_games =  list(enumerate(cosine_sim[g_index]))
sorted_sim_games = sorted(sim_games,key=lambda x:x[1],reverse=True)

def get_title_from_index(index):
    return df[df.index == index]["name"].values[0]
i=0
for game in sorted_sim_games:
    print(get_title_from_index(game[0]))
    i=i+1
    if i>15:
        break
```