Final Project

Project code:

github:*https://github.com/zhiangzz/507finalproject*
Kaggle API:*https://github.com/Kaggle/kaggle-api*
Step1-get the kaggle token from the account
Click on the profile picture and then get into the "My Account",and click on the "Create New API Token" button and download a kaggle.json file.
Step2-install the kaggle API and get the credentials

```
pip install kaggle
Place the kaggle file in ~/.kaggle/kaggle.json
chmod 600 ~/.kaggle/kaggle.json
export KAGGLE_USERNAME=datadinosaur
export KAGGLE_KEY=xxxxxxxxxxxxxx
```

Python Packages:pandas,ZipFile

This project is designed to recommend movies, TV series and TV shows for users. The project will ask users some questions and then recommend the optimal 5 movies, TV series and TV shows.

Data sources:

I applied for the kaggle API to download two datasets. Those datasets were in the zip files. So I unzip the files to get the dataset.(codes can be found in the data-access) What's more, because there are strings in the "Runtime" ,I process this column and strip "min" to keep the column as integers to compare conveniently.

https://www.kaggle.com/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows
It is the dataset of IMDB top 1000 movies and tv shows(csv), where there are 1000 rows and 16 columns.

Following is the columns of the imdb_top_1000.csv

- Poster_Link - Link of the poster that imdb using
- Series_Title = Name of the movie
- Released_Year - Year at which that movie released
- Certificate - Certificate earned by that movie
- Runtime - Total runtime of the movie
- Genre - Genre of the movie
- IMDB_Rating - Rating of the movie at IMDB site
- Overview - mini story/ summary
- Meta_score - Score earned by the movie
- Director - Name of the Director
- Star1,Star2,Star3,Star4 - Name of the Stars
- No_of_votes - Total number of votes

- Gross - Money earned by that movie

Release_Year, Runtime, Genre are principles of filtering movies or TV shows. As the dataset is huge, these 4 attributes will be grouped into different categories, which is convenient for users to choose. After users make choices, we will give 5 movies and TV shows of all attributes for users.

https://www.kaggle.com/unanimad/golden-globe-awards

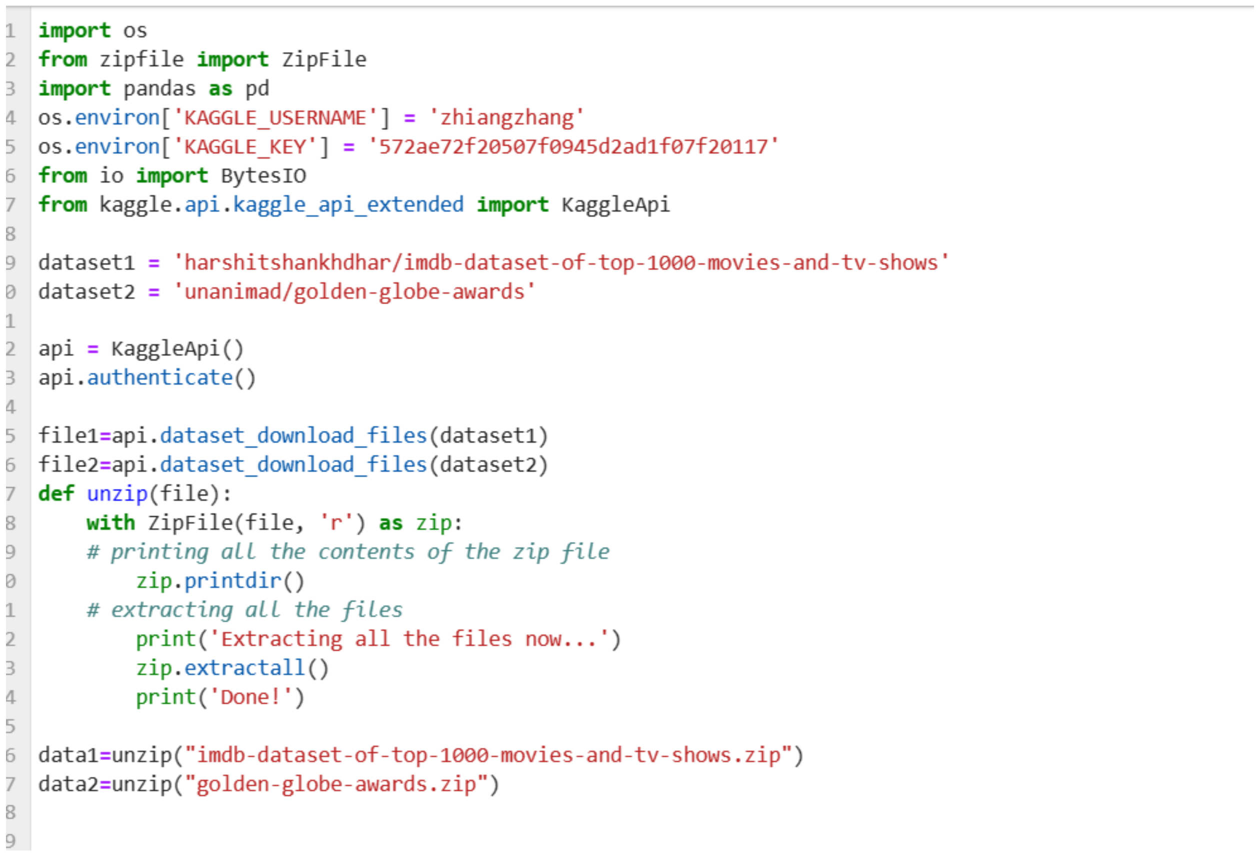
It is the dataset of the Golden Globe Awards list from 1944 to 2020(csv), where there are 7992 rows and 7 columns.

Following is the columns of golden_globe_awards.csv

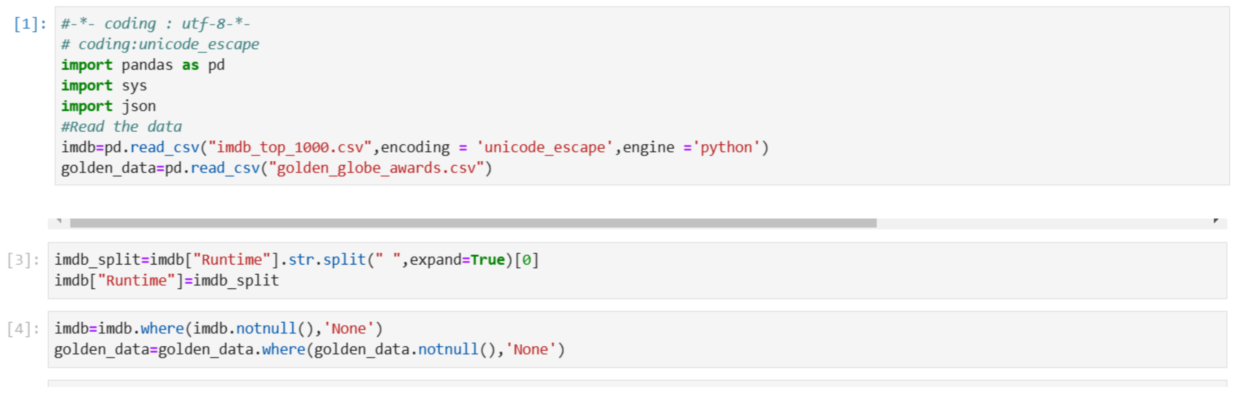
- Year_film - the year of the film
- Year_award - the year of award
- Ceremony - the ceremony of the award
- Category - the category of the award
- Nominee - the nominee name
- Film - the film name
- Win - Whether the film earn the award

In this dataset, year_award and category are the principles of filtering movies and TV series. As the dataset is huge, I categorize the “category” into two categories to choose conveniently. After users make choices, we will give 5 movies and TV shows of all attributes (except “win”) for users.

Following are the codes to get the API, download the data and preprocess the data.



```python
import os
from zipfile import ZipFile
import pandas as pd
os.environ['KAGGLE_USERNAME'] = 'zhiangzhang'
os.environ['KAGGLE_KEY'] = '572ae72f20507f0945d2ad1f07f20117'
from io import BytesIO
from kaggle.api.kaggle_api_extended import KaggleApi

dataset1 = 'harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows'
dataset2 = 'unanimad/golden-globe-awards'

api = KaggleApi()
api.authenticate()

file1=api.dataset_download_files(dataset1)
file2=api.dataset_download_files(dataset2)
def unzip(file):
    with ZipFile(file, 'r') as zip:
    # printing all the contents of the zip file
        zip.printdir()
    # extracting all the files
        print('Extracting all the files now...')
        zip.extractall()
        print('Done!')

data1=unzip("imdb-dataset-of-top-1000-movies-and-tv-shows.zip")
data2=unzip("golden-globe-awards.zip")


```



```
[1]: #-*- coding : utf-8-*-
     # coding:unicode_escape
     import pandas as pd
     import sys
     import json
     #Read the data
     imdb=pd.read_csv("imdb_top_1000.csv",encoding = 'unicode_escape',engine ='python')
     golden_data=pd.read_csv("golden_globe_awards.csv")
```

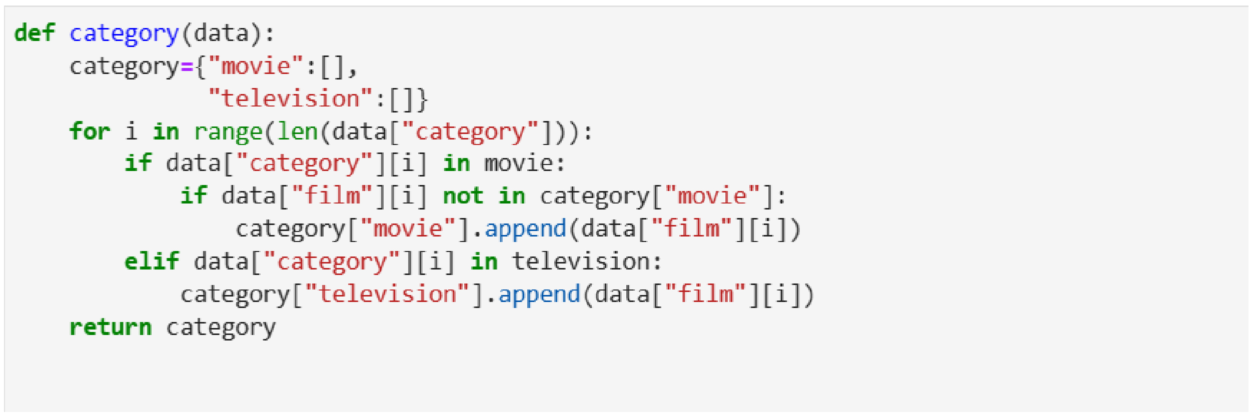
```
[3]: imdb_split=imdb["Runtime"].str.split(" ",expand=True)[0]
     imdb["Runtime"]=imdb_split
```

```
[4]: imdb=imdb.where(imdb.notnull(),'None')
     golden_data=golden_data.where(golden_data.notnull(),'None')
```

Following are the codes to categorize attributes.

```python
#data pre-processing:
def Release(data):
    release={
        "2010-2020":[],
        "2000-2010":[],
        "1990-2000":[],
        "1970-1990":[],
        "before 1970":[],
        "other":[]
    }
    for i in range(len(data["Released_Year"])):
        try:
            if 2010<int(data["Released_Year"][i])<=2020:
                release["2010-2020"].append(data["Series_Title"][i])
            elif 2000<int(data["Released_Year"][i])<=2010:
                release["2000-2010"].append(data["Series_Title"][i])
            elif 1990<int(data["Released_Year"][i])<=2000:
                release["1990-2000"].append(data["Series_Title"][i])
            elif 1970<int(data["Released_Year"][i])<=1990:
                release["1970-1990"].append(data["Series_Title"][i])
            elif int(data["Released_Year"][i])<=1970:
                release["before 1970"].append(data["Series_Title"][i])
        except:
            release["other"].append(data["Series_Title"][i])
    return release
```

```python
def Runtime(data):
    runtime={
        "<=60":[],
        "60-120":[],
        "120-180":[],
        ">180":[]
    }
    for i in range(len(data["Runtime"])):
        if int(data["Runtime"][i])<=60:
            runtime["<=60"].append(data["Series_Title"][i])
        elif 60<int(data["Runtime"][i])<=120:
            runtime["60-120"].append(data["Series_Title"][i])
        elif 120<int(data["Runtime"][i])<=180:
            runtime["120-180"].append(data["Series_Title"][i])
        elif int(data["Runtime"][i])>180:
            runtime[">180"].append(data["Series_Title"][i])
    return runtime
```

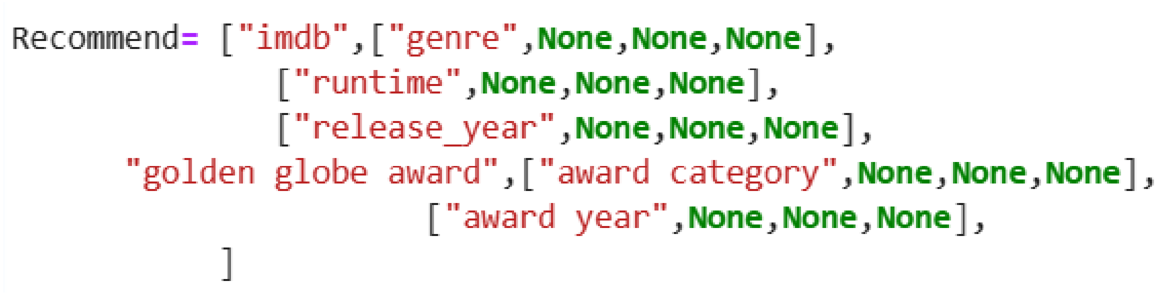


```python
def category(data):
    category={"movie":[],
              "television":[]}
    for i in range(len(data["category"])):
        if data["category"][i] in movie:
            if data["film"][i] not in category["movie"]:
                category["movie"].append(data["film"][i])
        elif data["category"][i] in television:
            category["television"].append(data["film"][i])
    return category
```

Data Structure:

I built a tree with two subtrees, imdb and golden globe award.”imdb” represents the imdb_top_1000.csv and “golden globe award” represents the golden_globe_awards.csv. In addition, under the “imdb” there are the 3 nodes “genre”, “runtime”, release_year”, and in the “golden globe award”, there are two nodes: “award category” and “award year”. However this is a dynamic tree, so users can choose what they all want to watch based on the nodes. And after users input answers based on those nodes, the list of names of movies, TV shows or TV series will be input after the answers, meanwhile answers will be inserted after the nodes in the Recommend Tree. After users get the recommendations, the codes can also create a recommend_tree.json file to see the detailed structure.

Following picture is the frame of the tree.



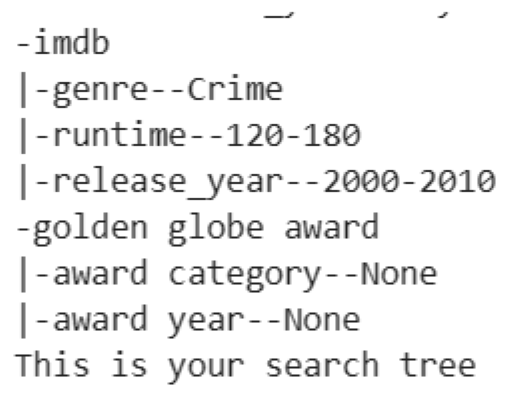
```python
Recommend= ["imdb",["genre",None,None,None],
                   ["runtime",None,None,None],
                   ["release_year",None,None,None],
      "golden globe award",["award category",None,None,None],
                           ["award year",None,None,None],
           ]
```

Data process is a main python file to process the data as a tree structure.(Detailed information can be found in the data_process.ipynb)

For example, I input answers into the tree.

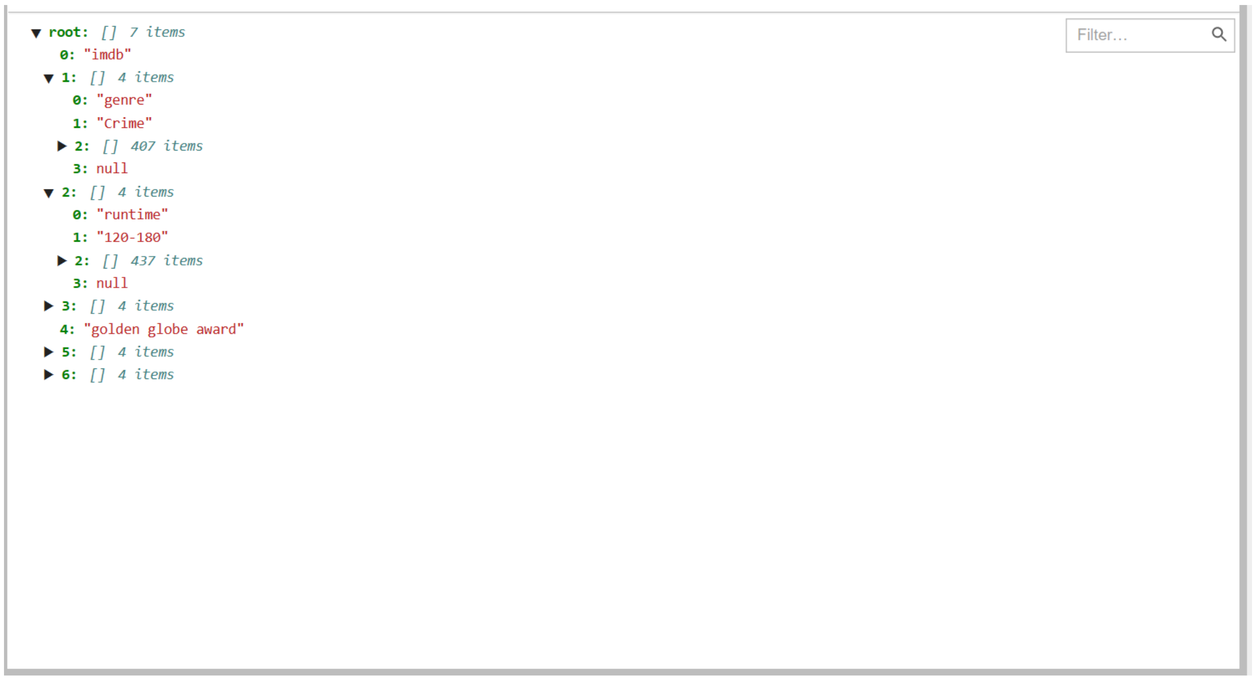
```
Which dataset do you want to use, imdb or golden globe award? imdb
The Genre contains Drama, Crime, Action, Adventure, Biography, History, Romance, Western, Thriller, Family, Animation, Comedy, Mystery,
Music, War, Horror.
Which genre do you want to watch?(hint:1 anwser) Crime
The Runtime contains <=60, 60-120, 120-180, >180 (min).
Which runtime(min) do you want to watch?(hint:1 anwser) 120-180
The Release year contains 2010-2020, 2000-2010, 1990-2000, 1970-1990, before 1970.
Which release_year do you want to watch?(hint:1 anwser) 2000-2010
```

Then this is the basic frame of the tree after I answer the question.

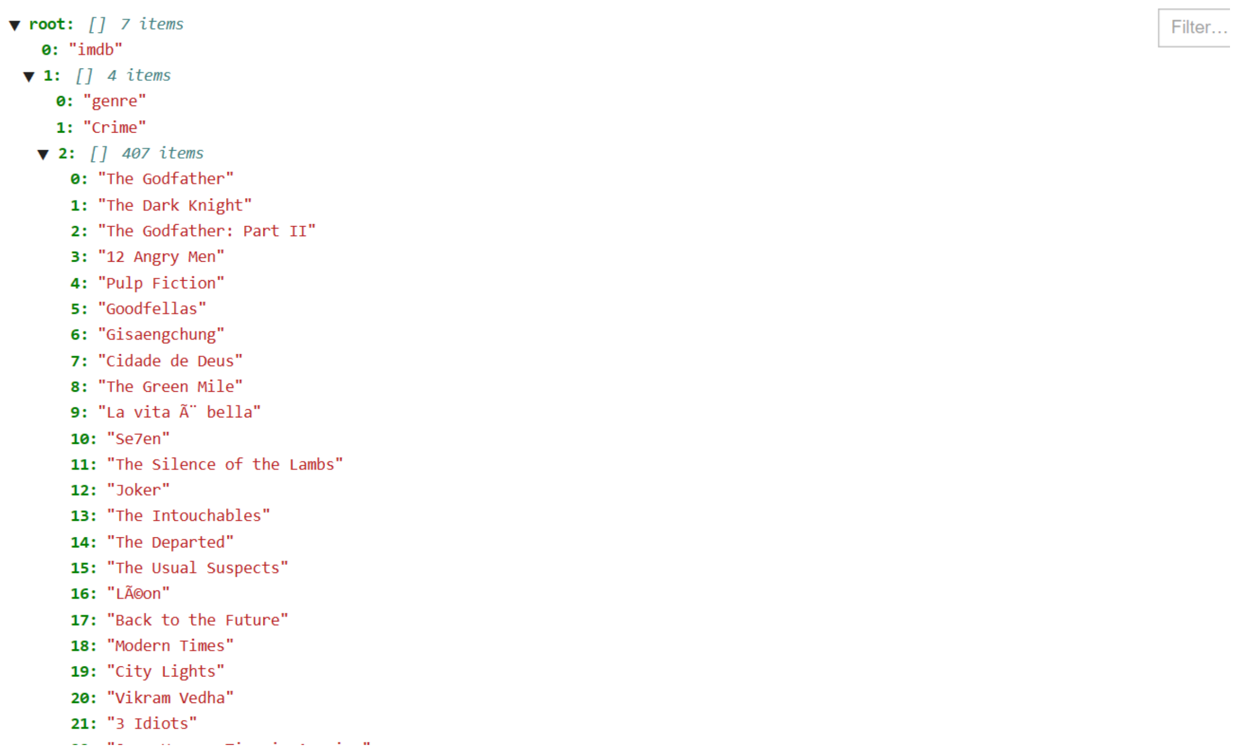


```
-imdb
|-genre--Crime
|-runtime--120-180
|-release_year--2000-2010
-golden globe award
|-award category--None
|-award year--None
This is your search tree
```

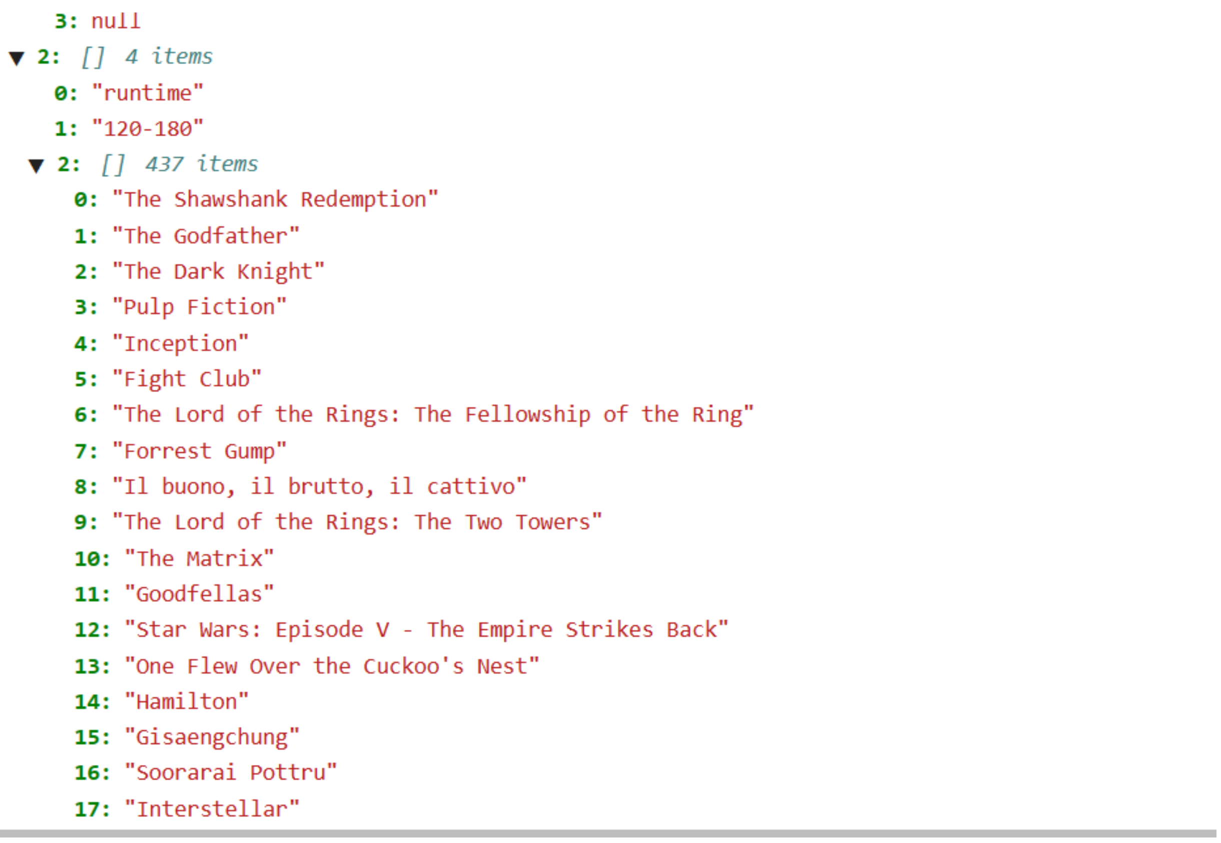
And these are the screenshots of the recommend_tree.json. (Details can be found in the reommend_tree.json.)



```
▼ root: [] 7 items
    0: "imdb"
  ▼ 1: [] 4 items
      0: "genre"
      1: "Crime"
    ▶ 2: [] 407 items
      3: null
  ▼ 2: [] 4 items
      0: "runtime"
      1: "120-180"
    ▶ 2: [] 437 items
      3: null
  ▶ 3: [] 4 items
    4: "golden globe award"
  ▶ 5: [] 4 items
  ▶ 6: [] 4 items
```

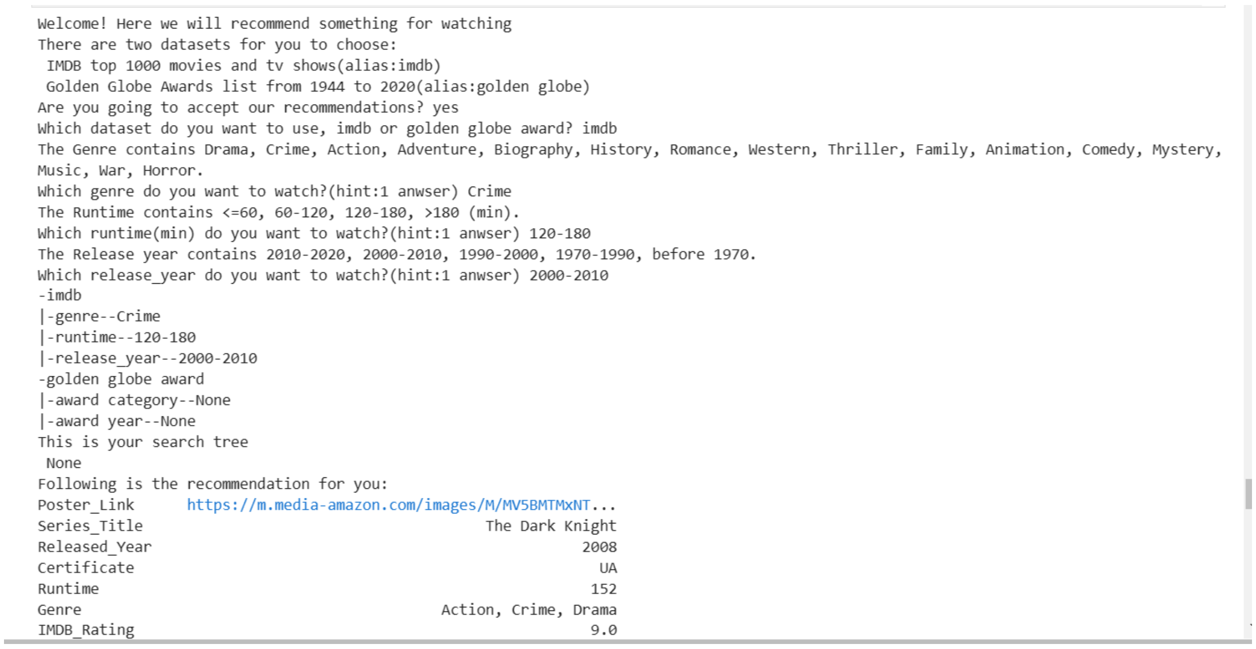


```
▼ root: [] 7 items
    0: "imdb"
  ▼ 1: [] 4 items
      0: "genre"
      1: "Crime"
    ▼ 2: [] 407 items
        0: "The Godfather"
        1: "The Dark Knight"
        2: "The Godfather: Part II"
        3: "12 Angry Men"
        4: "Pulp Fiction"
        5: "Goodfellas"
        6: "Gisaengchung"
        7: "Cidade de Deus"
        8: "The Green Mile"
        9: "La vita Ã¨ bella"
        10: "Se7en"
        11: "The Silence of the Lambs"
        12: "Joker"
        13: "The Intouchables"
        14: "The Departed"
        15: "The Usual Suspects"
        16: "LÃ©on"
        17: "Back to the Future"
        18: "Modern Times"
        19: "City Lights"
        20: "Vikram Vedha"
        21: "3 Idiots"
```

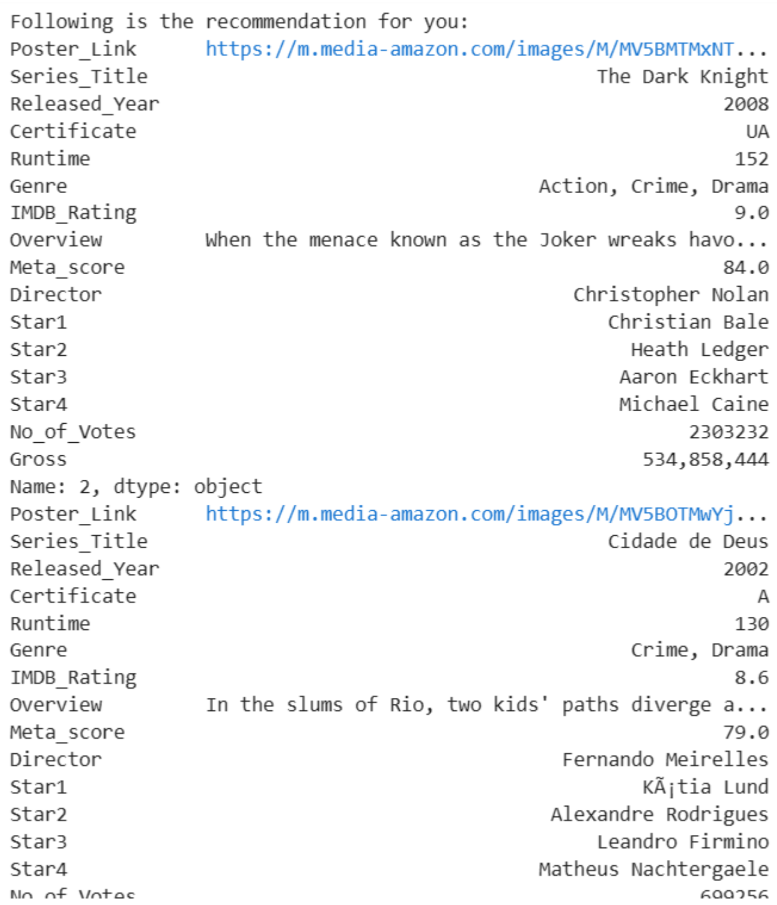


```
      3: null
  ▼ 2: [] 4 items
      0: "runtime"
      1: "120-180"
    ▼ 2: [] 437 items
        0: "The Shawshank Redemption"
        1: "The Godfather"
        2: "The Dark Knight"
        3: "Pulp Fiction"
        4: "Inception"
        5: "Fight Club"
        6: "The Lord of the Rings: The Fellowship of the Ring"
        7: "Forrest Gump"
        8: "Il buono, il brutto, il cattivo"
        9: "The Lord of the Rings: The Two Towers"
        10: "The Matrix"
        11: "Goodfellas"
        12: "Star Wars: Episode V - The Empire Strikes Back"
        13: "One Flew Over the Cuckoo's Nest"
        14: "Hamilton"
        15: "Gisaengchung"
        16: "Soorarai Pottru"
        17: "Interstellar"
```

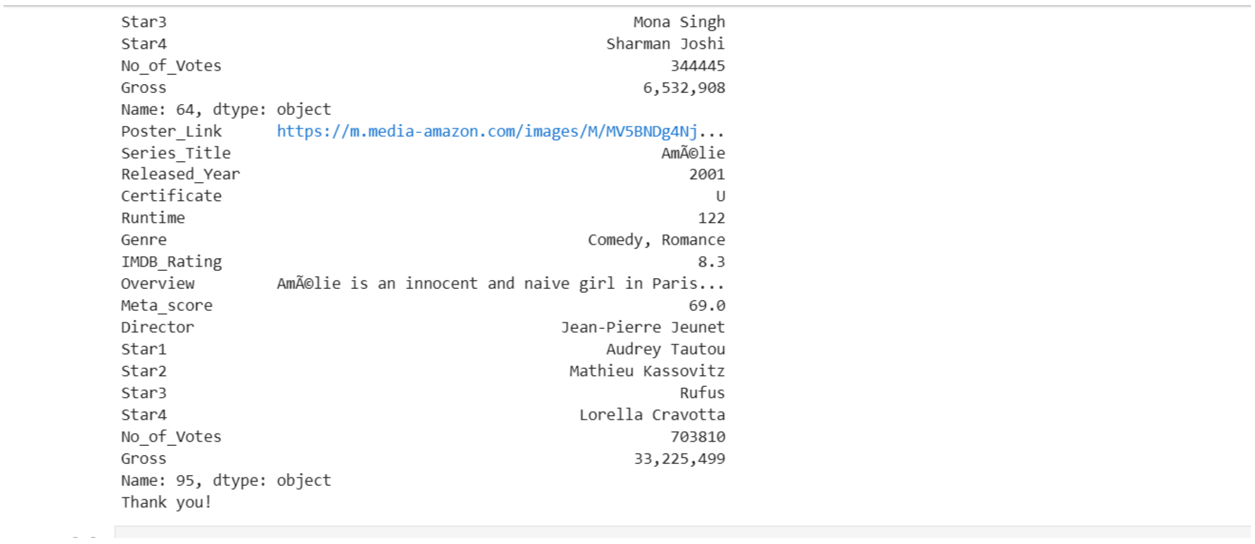
Interaction/Presentation:
Users can use command lines to input the dataset they want to choose.At first, they will be asked about whether they want to accept or reject the recommendations, if they reply “yes”, then they will be asked more detailed questions. However, if they reply, “no”,they will get the
 If they choose the imdb, they will be asked questions with options, like what genre they want to choose, what runtime they want to choose and what release year they want to choose. And then, We provide the recommended 5 movies or the TV shows with detailed information. If they choose the golden globe award, they will be asked questions with options, like which award category they want to choose and the which year they want to choose. And then, we provide the recommended 5 movies or TV series with detailed information. What’s more, if codes don’t find the recommended TV series, movies or TV shows, users could be asked questions again until they get their recommendations. The big difference is that recommendations are determined by all the principles, not just one principle like data structure.(Details can be found in the data_process.ipynb)
Following pictures are the screenshots of the demo of choosing imdb.



```
Welcome! Here we will recommend something for watching
There are two datasets for you to choose:
 IMDB top 1000 movies and tv shows(alias:imdb)
 Golden Globe Awards list from 1944 to 2020(alias:golden globe)
Are you going to accept our recommendations? yes
Which dataset do you want to use, imdb or golden globe award? imdb
The Genre contains Drama, Crime, Action, Adventure, Biography, History, Romance, Western, Thriller, Family, Animation, Comedy, Mystery,
Music, War, Horror.
Which genre do you want to watch?(hint:1 anwser) Crime
The Runtime contains <=60, 60-120, 120-180, >180 (min).
Which runtime(min) do you want to watch?(hint:1 anwser) 120-180
The Release year contains 2010-2020, 2000-2010, 1990-2000, 1970-1990, before 1970.
Which release_year do you want to watch?(hint:1 anwser) 2000-2010
-imdb
|-genre--Crime
|-runtime--120-180
|-release_year--2000-2010
-golden globe award
|-award category--None
|-award year--None
This is your search tree
 None
Following is the recommendation for you:
Poster_Link      https://m.media-amazon.com/images/M/MV5BMTMxNT...
Series_Title                                          The Dark Knight
Released_Year                                                    2008
Certificate                                                        UA
Runtime                                                           152
Genre                                             Action, Crime, Drama
IMDB_Rating                                                       9.0
```



```
Following is the recommendation for you:
Poster_Link      https://m.media-amazon.com/images/M/MV5BMTMxNT...
Series_Title                                      The Dark Knight
Released_Year                                                2008
Certificate                                                    UA
Runtime                                                       152
Genre                                        Action, Crime, Drama
IMDB_Rating                                                   9.0
Overview         When the menace known as the Joker wreaks havo...
Meta_score                                                   84.0
Director                                        Christopher Nolan
Star1                                              Christian Bale
Star2                                               Heath Ledger
Star3                                              Aaron Eckhart
Star4                                              Michael Caine
No_of_Votes                                               2303232
Gross                                                 534,858,444
Name: 2, dtype: object
Poster_Link      https://m.media-amazon.com/images/M/MV5BOTMwYj...
Series_Title                                       Cidade de Deus
Released_Year                                                2002
Certificate                                                     A
Runtime                                                       130
Genre                                                Crime, Drama
IMDB_Rating                                                   8.6
Overview         In the slums of Rio, two kids' paths diverge a...
Meta_score                                                   79.0
Director                                       Fernando Meirelles
Star1                                                 KÃ¡tia Lund
Star2                                         Alexandre Rodrigues
Star3                                             Leandro Firmino
Star4                                        Matheus Nachtergaele
No_of_Votes                                                699256
```



```
Star3                                             Mona Singh
Star4                                          Sharman Joshi
No_of_Votes                                           344445
Gross                                              6,532,908
Name: 64, dtype: object
Poster_Link      https://m.media-amazon.com/images/M/MV5BNDg4Nj...
Series_Title                                          AmÃ©lie
Released_Year                                            2001
Certificate                                                 U
Runtime                                                   122
Genre                                         Comedy, Romance
IMDB_Rating                                               8.3
Overview         AmÃ©lie is an innocent and naive girl in Paris...
Meta_score                                               69.0
Director                                    Jean-Pierre Jeunet
Star1                                            Audrey Tautou
Star2                                       Mathieu Kassovitz
Star3                                                    Rufus
Star4                                         Lorella Cravotta
No_of_Votes                                             703810
Gross                                               33,225,499
Name: 95, dtype: object
Thank you!
```

Demo:*https://github.com/zhiangzz/507finalproject/blob/main/demo.mp4*