

MOVIE RECOMMENDATION SYSTEM APPLYING CONTENT-BASED METHODS



Database Management

GVHD

Group

Member

CAO THI NHAM

01

1. Phan Cong Hieu (Leader) – 20%
2. Nguyen Quoc Manh – 20%
3. Tran Trinh Thanh Ngan – 20%
4. Nguyen Quang Tu – 20%
5. Nguyen Thi Ngoc Xuan – 20%

MỤC LỤC

I. Giới thiệu đề tài.....	2
1. Mục đích đề tài.....	2
2. Bố cục xây dựng đề tài.....	2
II. Quy trình cào dữ liệu từ rottentomatoes.com	2
1. Scrape URL của các bộ phim.....	2
2. Scrape thông tin của phim.....	7
3. Thông tin về bộ dữ liệu đã scrape	10
III. Trực quan hoá dữ liệu gốc	13
1. Tổng quan về dữ liệu.....	13
2. EDA.....	22
IV. Tiền xử lý và đổ dữ liệu lên MS Azure	26
V. Xây dựng mô hình học máy với Python.....	31
VI. Ứng dụng Web app với thư viện Streamlit trên Python.....	37

I. Giới thiệu đề tài

1. Mục đích đề tài.

Đề tài của nhóm là xây dựng một mô hình đề xuất phim đơn giản ứng dụng phương pháp Content-based Filtering.

2. Bố cục xây dựng đề tài

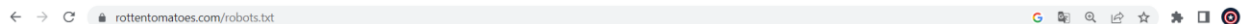
- 1) Tiến hành cào dữ liệu từ trang [rottentomatoes.com](https://www.rottentomatoes.com), sau khi cào, tiến hành trực quan mô tả và khai phá dữ liệu.
- 2) Tiền xử lý với SQL trên Azure cloud.
- 3) Xây dựng mô hình đề xuất phim bằng phương pháp Content-based với 3 thuộc tính phim là: thể loại, đạo diễn, diễn viên.
- 4) Ứng dụng thư viện [Streamlit](#) để tạo ra một giao diện website đơn giản, deploy mô hình qua tên miền của [Render](#).

II. Quy trình cào dữ liệu từ rottentomatoes.com

Để Scrape dữ liệu trên website [rottentomatoes.com](https://www.rottentomatoes.com) ta sẽ sử dụng python kết hợp với thư viện BeautifulSoup4 và selenium

1. Scrape URL của các bộ phim

Đầu tiên, truy cập vào file robots.txt của rottentomatoes.com ở link <https://www.rottentomatoes.com/robots.txt> để biết link nào không cho phép scrape (Disallow) và sitemap của trang. Sitemap là một tệp liệt kê các trang và tệp tin trên website, trong đó có liệt kê các link sitemap URL movie cần scrape.



```
User-agent: *  
Disallow: /search  
Disallow: /user/id/  
Sitemap:  
https://www.rottentomatoes.com/sitemaps/sitemap.xml
```

Sau đó, vào python và import các thư viện cần dùng

```

from selenium import webdriver
from requests import get
from bs4 import BeautifulSoup
import bs4
import pandas as pd
import numpy as np
from time import sleep
from random import randint
from time import time
from warnings import warn
from datetime import datetime
from datetime import timedelta
from pytz import timezone
from IPython.display import clear_output
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.common.exceptions import TimeoutException
from selenium.common.exceptions import StaleElementReferenceException
import re
import os
import sys
import glob
import shutil
from pathlib import Path
import re

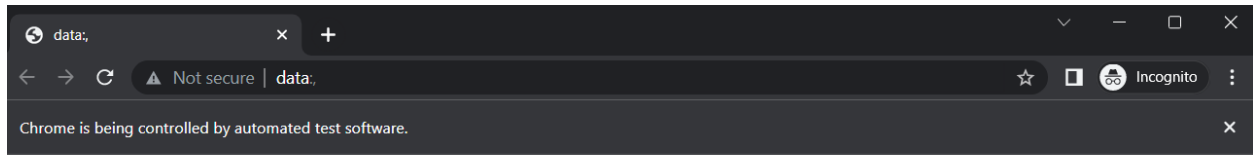
```

Mở trình duyệt chrome ở trạng thái ẩn danh bằng webdriver trong thư viện selenium để Scrape dữ liệu

```

# Mở trình duyệt Google Chrome ở trạng thái Tab ẩn danh
options = webdriver.ChromeOptions()
options.add_argument('--ignore-certificate-errors')
options.add_argument('--incognito')
driver = webdriver.Chrome(executable_path='C:\chromedriver', options=options)
driver.implicitly_wait(0)

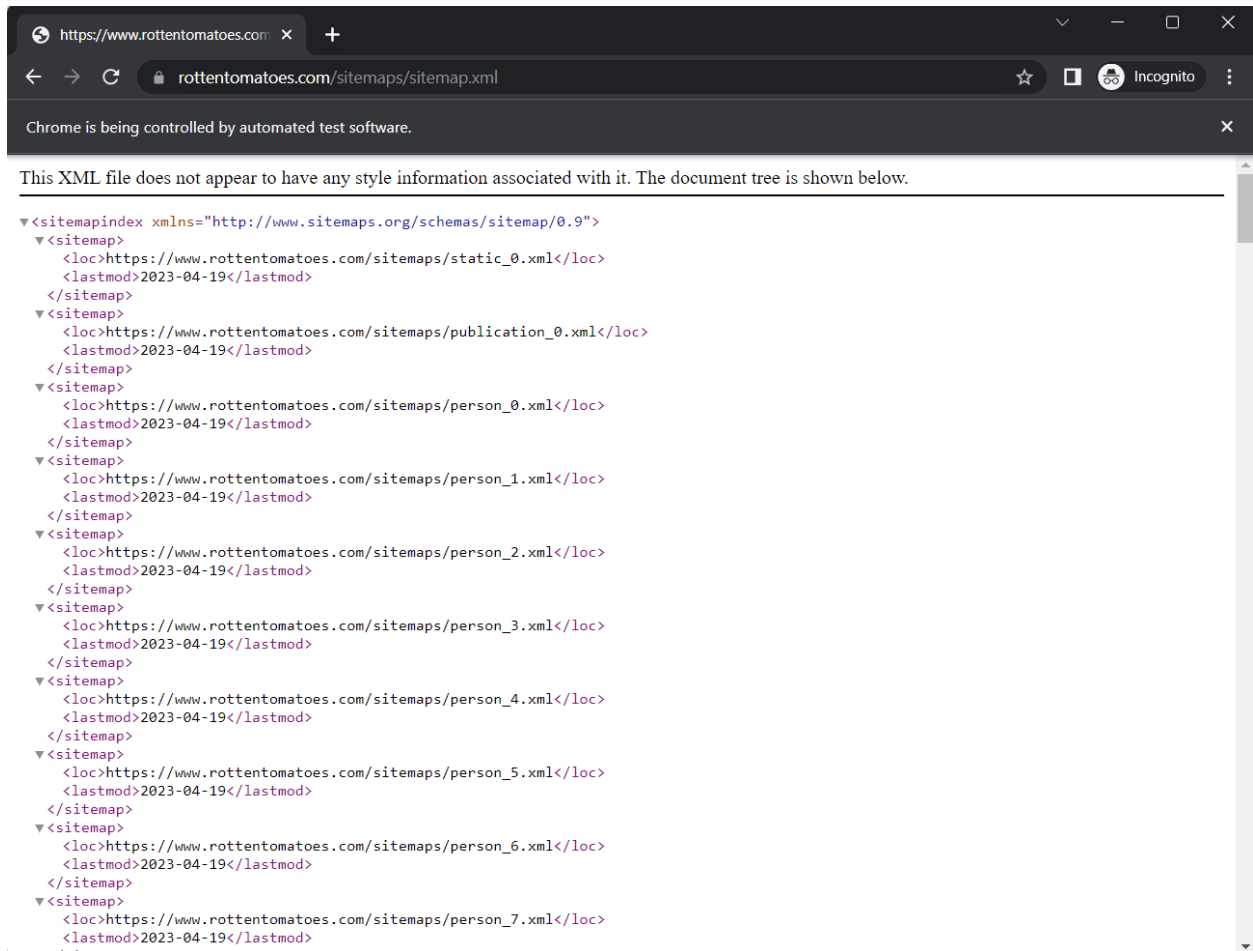
```



Mở link sitemap

Mở sitemap của Rotten Tomatoes

```
url = "https://www.rottentomatoes.com/sitemaps/sitemap.xml"
driver.get(url)
```



Đợi 30 giây cho sitemap load xong và lấy dữ liệu HTML của trang gán vào biến “soup”

```
# Đợi cho đến khi sitemap load xong
element = WebDriverWait(driver, 30).until(EC.presence_of_element_located((By.CSS_SELECTOR, 'div[class="pretty-print"]')))
html_of_interest = driver.execute_script('return arguments[0].innerHTML', element)
soup = BeautifulSoup(html_of_interest, 'lxml')
```

Tạo list chứa tất cả các sitemap scrape được thông qua biến soup

```
# Tạo list sitemap chung
list_all_sitemap = []

all_sitemap = soup.select('div[class="folder"] > div[class="opened"] > div[class="line"] > span:not([class])')

for sm in all_sitemap:
    sm = sm.text
    if 'http' in sm:
        list_all_sitemap.append(sm)
```

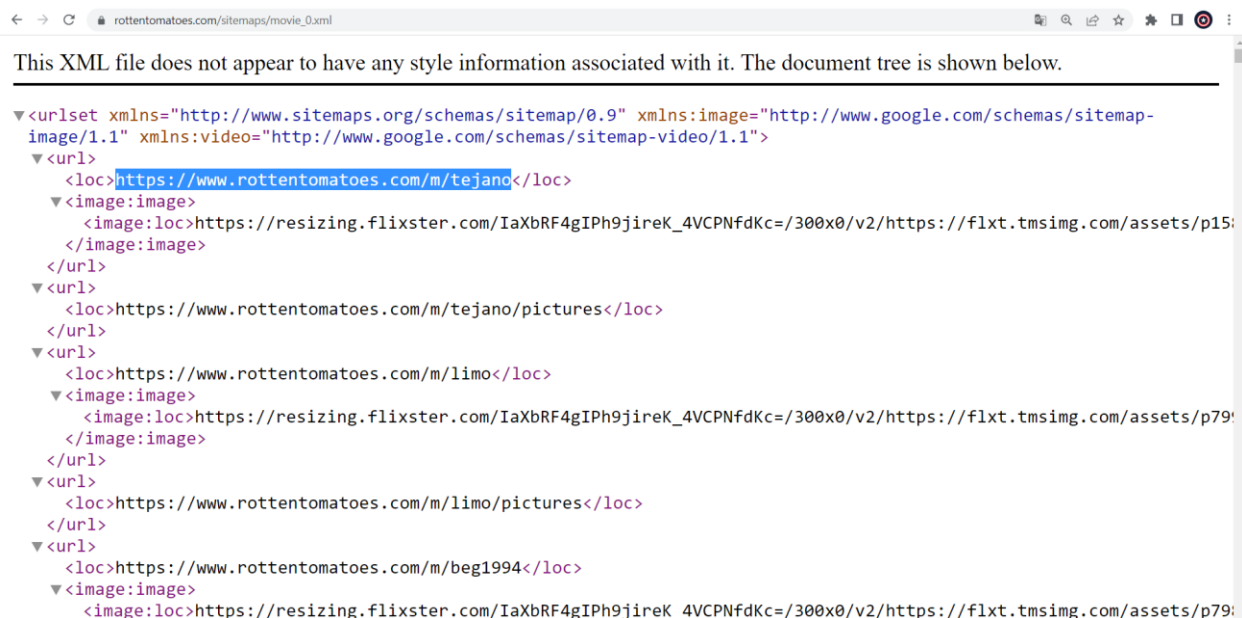
Scrape các sitemap URL movie và lưu vào list (để lấy URL movie)

```
</sitemap>
▼ <sitemap>
  <loc>https://www.rottentomatoes.com/sitemaps/movie_0.xml</loc>
  <lastmod>2023-04-19</lastmod>
</sitemap>
▼ <sitemap>
  <loc>https://www.rottentomatoes.com/sitemaps/movie_1.xml</loc>
  <lastmod>2023-04-19</lastmod>
</sitemap>
▼ <sitemap>
  <loc>https://www.rottentomatoes.com/sitemaps/movie_2.xml</loc>
  <lastmod>2023-04-19</lastmod>
</sitemap>
```

```
# Tạo list sitemap movie
list_sitemap_movie = []

for sm_movie in list_all_sitemap:
    sm_movie = re.findall(r'https://www.rottentomatoes.com/sitemaps/movie_[0-9]{1,3}.xml', sm_movie)
    if sm_movie != []:
        list_sitemap_movie.append(sm_movie[0])
```

Trong các Sitemap URL movie sẽ chứa tất cả các URL movie phim cần scrape



Tạo list chứa URL movies để lưu data được khi scrape

```
# Tạo list chứa url movie
list_url_movie = []

for sm_movie in list_sitemap_movie:
    # Mở sitemap
    driver.get(sm_movie)

    # Đợi 30s cho trang load
    element_movie = WebDriverWait(driver, 30).until(EC.presence_of_element_located((By.CSS_SELECTOR, 'div[class="pretty-print"]')))
    html_of_interest = driver.execute_script('return arguments[0].innerHTML', element_movie)
    soup_movie = BeautifulSoup(html_of_interest, 'lxml')

    # Lưu url movie
    url_movie = soup_movie.select('div[class="folder"] > div[class="opened"] > div[class="folder"] > div[class="opened"] > div[class="line"]')

    for url in url_movie:
        url = url.text
        if 'https://www.rottentomatoes.com' in url and 'pictures' not in url and 'trailer' not in url:
            list_url_movie.append(url)
```

Cuối cùng, chuyển list URL movie thành DataFrame và lưu thành file URL_movie.csv

```
# Save file URL_movie.csv
df_movie = pd.DataFrame(list_url_movie, columns=['URL movie'])
df_movie.to_csv('URL_movie.csv', index=False)
```

URL movie
https://www.rottentomatoes.com/m/beg1994
https://www.rottentomatoes.com/m/limo
https://www.rottentomatoes.com/m/tejano
https://www.rottentomatoes.com/m/outpatient
https://www.rottentomatoes.com/m/1221483-paa
https://www.rottentomatoes.com/m/cold_room
https://www.rottentomatoes.com/m/disciple_of_death
https://www.rottentomatoes.com/m/genesis1998

2. Scrape thông tin của phim

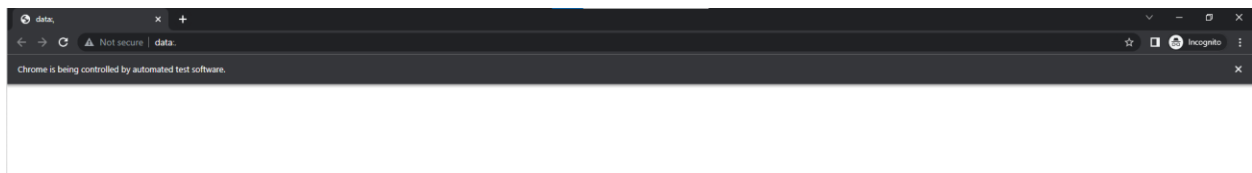
Dùng URL của các movie từ file “URL_movie.csv” trên để scrape thông tin của từng movie:

Import các thư viện

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from requests import get
from bs4 import BeautifulSoup
import pandas as pd
from time import sleep
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import traceback
```

Mở chrome bằng thư viện selenium ở trạng thái ẩn danh

```
# Mở trình duyệt Google Chrome ở trạng thái Tab ẩn danh
options = webdriver.ChromeOptions()
options.add_argument('--ignore-certificate-errors')
options.add_argument('--incognito')
options.add_argument("--start-maximized")
s = Service('C:\chromedriver')
driver = webdriver.Chrome(service=s, options=options)
driver.implicitly_wait(0)
```

Sau đó, tạo DataFrame chứa các URL movie từ file URL_movie.csv đã thu thập

```
# List tất cả url movie
list_all_url_movie = pd.read_csv('E:/Crawl_Web/Rotten Tomatoes/Crawl URL/URL_movie.csv').squeeze('columns').tolist()
```

Tạo 1 vòng lặp for để scrape từng URL movie trong DataFrame.

+ Mở URL và đợi tối đa 30 giây cho đến khi trang load xong. Sau đó tạo một List rỗng có tên là df_movie

```
df_movie = []

for url_movie in list_all_url_movie:
    try:
        # Mở url phim
        driver.get(url_movie)

        # Đợi cho đến khi url Load xong
        element = WebDriverWait(driver, 30).until(EC.presence_of_element_located((By.CSS_SELECTOR, 'main[id="main_container"]')))
        html_of_interest = driver.execute_script('return arguments[0].innerHTML', element)
        soup = BeautifulSoup(html_of_interest, 'lxml')
```

+ Tạo một Dict chứa thông tin Scrape được

```
dict_info_movie = {}
```

+ Scrape tiêu đề phim và cho vào Dict đã tạo với key tương ứng là 'Title'

```
# Tiêu đề phim
dict_info_movie['Title'] = soup.select('h1[class="scoreboard__title"]')[0].text
```

+ Scrape điểm đánh giá của nhà phê bình và cho vào Dict với key là 'Tomatometer score'

```
# Điểm đánh giá của nhà phê bình
tomatometer = soup.select('score-board')[0].attrs['tomatometerscore']
dict_info_movie['Tomatometer score'] = tomatometer
```

+ Scrape điểm đánh giá của khán giả và cho vào Dict với key là 'Audience score'

```
# Điểm đánh giá của khán giả
dict_info_movie['Audience score'] = soup.select('score-board')[0].attrs['audiencescore']
```

+ Scrape số lượt đánh giá của nhà phê bình và cho vào Dict với key là 'Tomatometer count'

```
# Số lượt đánh giá của nhà phê bình
dict_info_movie['Tomatometer count'] = soup.select('a[slot="critics-count"]')[0].text.strip()
```

+ Scrape số lượt đánh giá của khán giả và cho vào Dict với key là 'Audience count'

```
# Số lượt đánh giá của khán giả
dict_info_movie['Audience count'] = soup.select('a[slot="audience-count"]')[0].text.strip()
```

+ Scrape trạng thái đánh giá của nhà phê bình và cho vào Dict với key là ‘Tomatometer state’

```
# Trạng thái đánh giá phim của nhà phê bình
dict_info_movie['Tomatometer state'] = soup.select('score-board')[0].attrs['tomatometerstate']
```

+ Scrape trạng thái đánh giá của khán giả và cho vào Dict với key là ‘Audience state’

```
# Trạng thái đánh giá phim của khán giả
dict_info_movie['Audience state'] = soup.select('score-board')[0].attrs['audiencestate']
```

+ Scrape tất cả thông tin của phim(Rating, thể loại, thời lượng, đạo diễn,...)

```
list_name_info = soup.select('b[class="info-item-label"]')
list_info = soup.select('span[data-qa="movie-info-item-value"]')

for num_info in range(len(list_name_info)):
    name_info = list_name_info[num_info].text.replace(' ', '').replace('\n', '').replace(':', '')
    info = list_info[num_info].text.replace(' ', '').replace('\n', '').replace(':', '')
    dict_info_movie[name_info] = info
```

+ Scrape tóm tắt nội dung phim và cho vào Dict với key là Synopsis”

```
dict_info_movie['Synopsis'] = soup.select('p[data-qa="movie-info-synopsis"]')[0].text.strip()
```

+ Scrape tên diễn viên tham gia đóng phim và cho vào Dict với key là ‘Cast’

```
list_raw_cast = soup.select('a[data-qa="cast-crew-item-link"] > p')
list_cast = []
```

```
for c in range(len(list_raw_cast)):
    list_cast.append(list_raw_cast[c].text)
```

```
dict_info_movie['Cast'] = ', '.join(list_cast)
```

+ Từ Dict chứa dữ liệu Scrape được ta sẽ cho vào List rồi df_movie ta tạo ban đầu

```
df_movie.append(dict_info_movie)
```

Tạo DataFrame từ List df_movie

```
df_movie = pd.DataFrame(df_movie)
```

Title ▼	Tomatometer score	Audience score	Tomatometer count	Audience count	Tomatometer state	Audience state	Genre ▼
Paa	50	67	8 Reviews	1,000+ Ratings	rotten	upright	Drama
Small Town Wisconsin	83	88	12 Reviews	Fewer than 50 Ratings	fresh	upright	Comedy, Drama
The 100-Year-Old Man Who	68	69	81 Reviews	2,500+ Ratings	fresh	upright	Comedy, Adventure
Escape From Alcatraz	97	85	30 Reviews	50,000+ Ratings	fresh	upright	Mystery & thriller
Adrift	69	65	204 Reviews	1,000+ Ratings	fresh	upright	Adventure, Drama
Born to Kill	83	74	6 Reviews	500+ Ratings	fresh	upright	Crime, Drama
My Giant	21	25	28 Reviews	5,000+ Ratings	rotten	spilled	Comedy, Drama
A State of Mind	89	92	28 Reviews	1,000+ Ratings	fresh	upright	Documentary
What to Expect When You're	23	47	136 Reviews	100,000+ Ratings	rotten	spilled	Comedy, Drama
Dangerous Men	50	42	10 Reviews	Fewer than 50 Ratings	rotten	spilled	
La Sapienza	88	50	26 Reviews	100+ Ratings	fresh	spilled	Drama
Jerusalem	54	52	24 Reviews	500+ Ratings	rotten	spilled	Horror

Cuối cùng, lưu DataFrame trên thành file movie_info.csv

```
df_movie.to_csv('movie_info.csv', index=False)
```

3. Thông tin về bộ dữ liệu đã scrape

Tên cột	Giá trị có thể nhận	Ý nghĩa
Title	"Paa", "Small Town Wisconsin",...	Tên phim
Tomatometer score	0 -> 100	Điểm đánh giá của nhà phê bình
Audience score	0 -> 100	Điểm đánh giá của khán giả
Tomatometer count	"8 Reviews", "12 Reviews", "81 Reviews",...	Số lượt đánh giá của nhà phê bình
Audience count	"1,000+ Ratings", "Fewer than 50 Ratings", "2,500+ Ratings",...	Số lượt đánh giá của khán giả
Tomatometer state	"rotten", "fresh", "certified-fresh"	Thể hiện % số lượt đánh giá tích cực của nhà phê bình - rotten: Dưới 60% - fresh: Hơn 60%

		- certified-fresh: Hơn 75%, có ít nhất 80 đánh giá và 5 nhà phê bình hàng đầu trong đó
Audience state	“upright”, “spilled”	Thể hiện % số lượt đánh giá 3.5 sao trở lên của khán giả - upright: Hơn 60% - spilled: Dưới 60%
Genre	“Drama”, "Comedy, Drama", "Comedy, Adventure",...	Thể loại phim
Original Language	"Hindi", "English", "Swedish", "Korean",...	Ngôn ngữ trong phim
Director	"R. Balki", "Niels Mueller", "Felix Herngren",...	Tên đạo diễn
Producer	"Sunil Manchanda", "Scott K. Foley, Hongtao Liu, Niels Mueller, Josh Rosenberg",...	Tên nhà sản xuất
Writer	"R. Balki", "Jason Naczek", "Felix Herngren, Hans Ingemansson",...	Tên nhà biên kịch
Release Date (Theaters)	“Dec 4, 2009 limited”, “Jun 3, 2022 limited”,...	Ngày ra mắt tại rạp
Box Office (Gross USA)	"\$199.2K", "\$923.9K",...	Tổng doanh thu phòng vé (USD)
Runtime	"2h 13m", "1h 49m", "1h 54m",...	Thời lượng phim
Distributor	'Big Pictures', 'Quiver Distribution', 'Music Box Films',...	Đơn vị phát hành phim
Synopsis		Tóm tắt nội dung phim
Cast		Tên diễn viên đóng phim

Release Date (Streaming)	'Jun 10, 2022', 'Aug 18, 2015',...	Ngày phát trực tuyến
Rating	'R (Some Violence Language)', 'PG',...	Xếp hạng MPAA (nhằm giới hạn độ tuổi xem phim)
Aspect Ratio	'Scope (2.351)', 'Flat (1.851)', '35mm',...	Tỉ lệ khung hình
Sound Mix	'SDDS, Dolby SR, Dolby Digital, Surround, Dolby A, Dolby Stereo', 'Stereo',...	Các kĩ thuật phối âm áp dụng trong phim
View the collection	'Jurassic Park', 'DC Universe', 'Star Trek',...	Phim thuộc franchise nào?

III. Trục quan hoá dữ liệu gốc

1. Tổng quan về dữ liệu

```
RangeIndex: 33838 entries, 0 to 33837
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Title                                33838 non-null  object
1   Tomatometer score                    33838 non-null  int64
2   Audience score                       30302 non-null  float64
3   Tomatometer count                    33838 non-null  object
4   Audience count                       33838 non-null  object
5   Tomatometer state                    33838 non-null  object
6   Audience state                       30302 non-null  object
7   Genre                                33016 non-null  object
8   Original Language                    32682 non-null  object
9   Director                             33678 non-null  object
10  Producer                             28287 non-null  object
11  Writer                               26653 non-null  object
12  Release Date (Theaters)              19694 non-null  object
13  Box Office (Gross USA)               13216 non-null  object
14  Runtime                              32608 non-null  object
15  Distributor                          17399 non-null  object
16  Synopsis                             32569 non-null  object
17  Cast                                 33689 non-null  object
18  Release Date (Streaming)             28635 non-null  object
19  Rating                               15667 non-null  object
20  Aspect Ratio                         5400 non-null   object
21  Sound Mix                            9746 non-null   object
22  View the collection                  505 non-null    object
dtypes: float64(1), int64(1), object(21)
memory usage: 5.9+ MB
```

Tỷ lệ % null của các cột:

Title	0.000000
Tomatometer score	0.000000
Audience score	10.449790
Tomatometer count	0.000000
Audience count	0.000000
Tomatometer state	0.000000
Audience state	0.000000
Genre	0.000000
Original Language	0.000000
Director	0.000000
Producer	0.000000
Writer	0.000000
Release Date (Theaters)	0.000000
Box Office (Gross USA)	60.943318
Runtime	0.000000
Distributor	0.000000
Synopsis	0.000000
Cast	0.000000
Release Date (Streaming)	15.376204
Rating	0.000000
Aspect Ratio	0.000000
Sound Mix	0.000000
View the collection	0.000000
text_length	0.000000
Year	0.000000

- Dữ liệu bao gồm 33838 dòng và 23 cột. Trong đó các cột 'View the collection', 'Aspect Ratio', 'Sound Mix' có lượng dữ liệu null rất lớn. Bên cạnh đó các cột Box Office (Gross USA) và Rating dữ liệu null cũng trên 50%. Có vài lý do dẫn đến điều này, bao gồm:

- + Tính sẵn có của dữ liệu: Một số phim có thể chưa được phát hành ở Hoa Kỳ và do đó không có sẵn dữ liệu cho 'Box Office (Gross USA)'.

- + “ Aspect Ratio ”: Cột này thể hiện tỷ lệ giữa chiều rộng và chiều cao của màn hình phim. Tuy nhiên, không phải tất cả các phim đều có cùng tỷ lệ khung hình và một số phim có thể không có sẵn thông tin này. Do đó, các giá trị null trong cột này có thể là do dữ liệu không đầy đủ hoặc bị thiếu đối với một số phim.

- + 'Sound Mix': Cột này đại diện cho loại định dạng âm thanh được sử dụng trong phim. Tuy nhiên, không phải tất cả các phim đều có định dạng âm thanh giống nhau và một số phim có thể không có sẵn thông tin này.

- + Rating: cột này thể hiện yêu cầu độ tuổi được phép xem phim đối với từng bộ phim. Có thể có nhiều bộ phim không có yêu cầu nên dẫn đến dữ liệu null cho cột này.

	Category	Count
0	Number of Movies	33838
1	Number of Genres	36
2	Number of Original Language	100
3	Number of Cast Members	213743
4	Number of Director	17659
5	Number of Producer	29526
6	Number of Writer	24808
7	Number of Distributor	2669
8	Number of Ratings	6555
9	Number of Tomatometer state	3
10	Number of Audience state	2
11	Number of Synopsis	32551

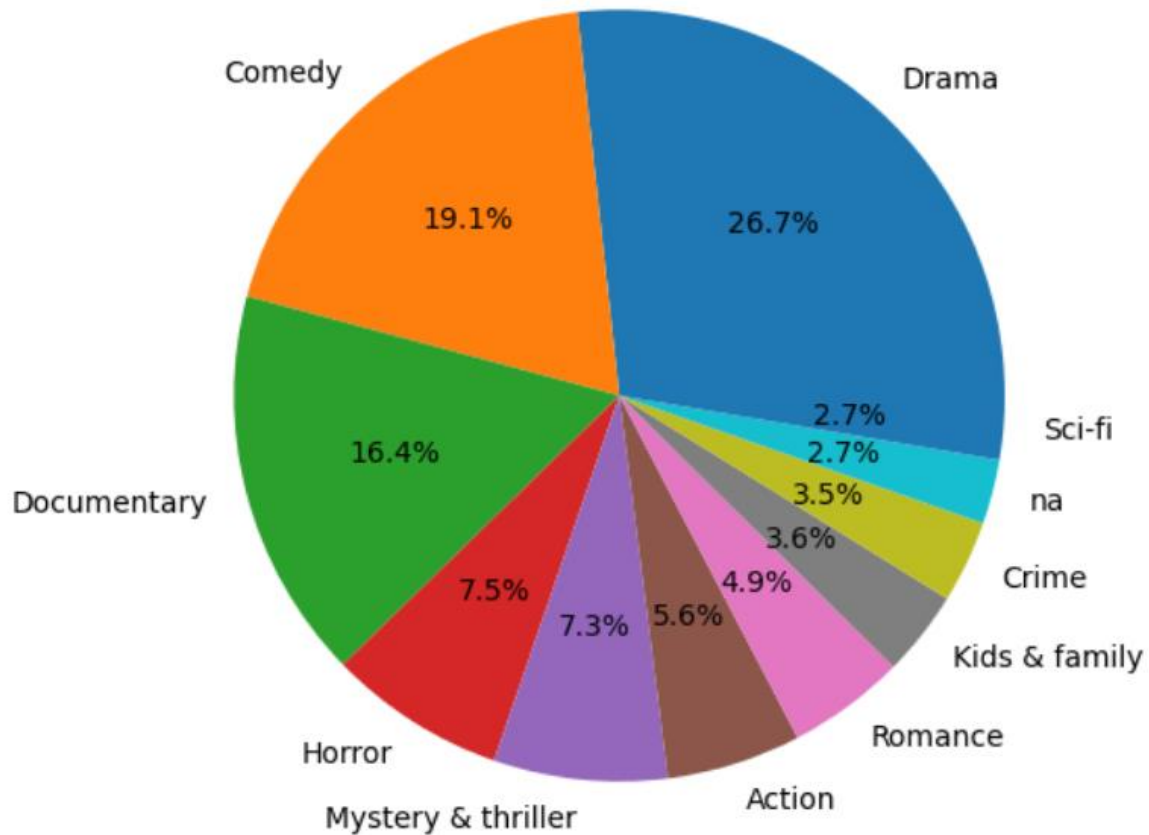
- Có tổng cộng 33838 bộ phim.
- Có tất cả 36 thể loại phim.
- Số lượng ngôn ngữ gốc của bộ phim là 100.
- Số lượng diễn viên trong dữ liệu là 213743 người.
- Tổng số đạo diễn là 17659 người.
- Số lượng nhà sản xuất phim là 29526 người
- Số lượng các nhà biên kịch tham gia là 24808 người.
- Số lượng các nhà phân phối phim (mua quyền phát hành và phân phối) là 2669.
- Tổng số lượng Ratings là 6555, tuy nhiên nếu rút gọn lại sẽ là 10 loại chính (NR, PG-13, TV14, ...)
- Bao gồm 3 cấp độ chỉ trạng thái xếp hạng của một bộ phim hoặc chương trình truyền hình dựa trên tỷ lệ phần trăm đánh giá tích cực của các nhà phê bình điện ảnh.
 - + Tomatometer state "Certified Fresh" (Màu sắc: xanh lá cây): Đạt được tỷ lệ đánh giá tích cực 75% hoặc cao hơn từ các nhà phê bình điện ảnh được Rotten Tomatoes
 - + Tomatometer state "Rotten" (Màu sắc: đỏ): Đạt được tỷ lệ đánh giá tích cực dưới 60% từ các nhà phê bình điện ảnh.
 - + Tomatometer state "Fresh" (Màu sắc: xanh lam): Đạt được tỷ lệ đánh giá tích cực từ 60% đến dưới 75% từ các nhà phê bình điện ảnh.
- Có 2 cấp độ xếp hạng của một bộ phim hoặc chương trình truyền hình dựa trên đánh giá của khán giả. Bao gồm:
 - + Audience state "Upright" (Màu sắc: xanh lam): Audience Score từ 60% đến dưới 75%.
 - + Audience state "Spilled" (Màu sắc: đỏ): Audience Score dưới 60%.
- Có tổng cộng 32551 bộ phim có tài liệu tóm tắt nội dung của bộ phim.

Number of Original Language 101

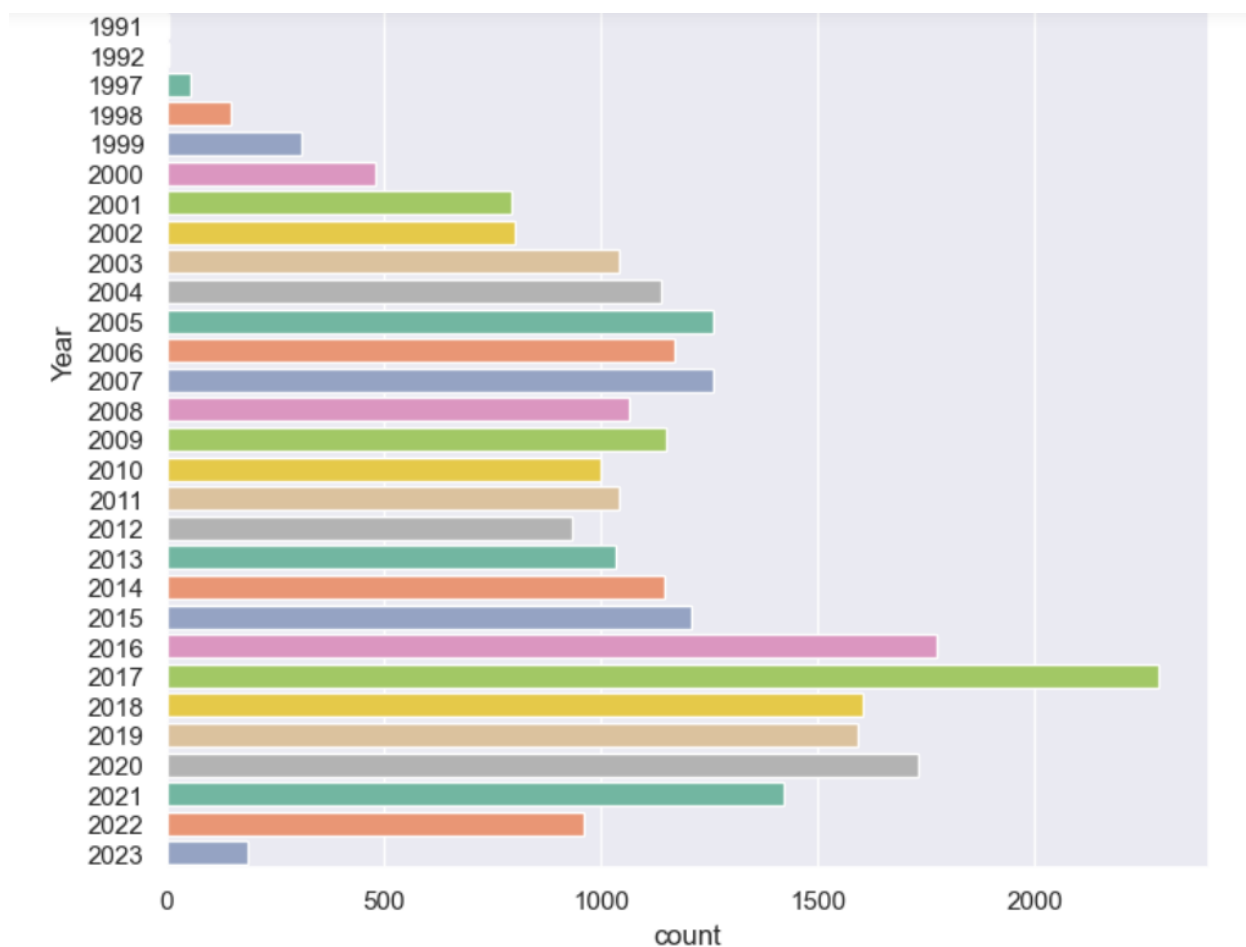
Rank	Original Language	Frequency
1	English	24448
2	na	1156
3	English (United Kingdom)	844
4	French (France)	819
5	French (Canada)	741
6	Hindi	665
7	Spanish	665
8	Japanese	557
9	Chinese	529
10	German	415
11	Italian	373
12	Korean	245
13	Spanish (Spain)	212
14	Arabic	197
15	English (Australia)	167
16	Russian	151
17	Portuguese (Brazil)	146
18	Swedish	111
19	Hebrew	109
20	Danish	102
21	Persian	100

Nhìn vào hình trên có thể thấy, số lượng ngôn ngữ trong dữ liệu lên đến 100, hay dữ liệu phim của tomato thuộc 100 thứ tiếng khác nhau (1 cột là giá trị na được điền vào cho các dòng có dữ liệu null). Ngôn ngữ chiếm số lượng lớn nhất là ‘English’ (phim Tiếng Anh) với 24448 bộ phim, chiếm gần 73% tổng số dữ liệu. Điều này dễ hiểu bởi Tiếng Anh là ngôn ngữ phổ biến nhất trên thế giới, được sử dụng như một ngôn ngữ thứ hai hoặc thứ ba tại nhiều quốc gia khác nhau, đặc biệt là ở các nước có nền công nghiệp phát triển. Do đó, các bộ phim bằng tiếng Anh có thể đáp ứng được nhu cầu của một khán giả toàn cầu.

Genre Count



- Có thể dễ dàng nhận thấy từ biểu đồ rằng thể loại phim chiếu nhiều nhất trong dữ liệu là "Drama" với 26,7% và không khó để giải thích việc này vì Drama là một thể loại phim rộng, có thể bao gồm nhiều chủ đề khác nhau, như tình yêu, gia đình, tội phạm, chiến tranh, v.v. Vì vậy, thể loại này có thể phù hợp với một đối tượng khán giả rộng lớn hơn so với những thể loại phim khác.



Có thể nhìn thấy trong dữ liệu, các phim trong tập dữ liệu hầu hết nằm trong giai đoạn năm 1997-2023. Trong đó, số lượng phim của năm 2017 chiếm cao nhất với hơn 2000 bộ phim được phát hành. Điều này cho thấy năm 2017 là một năm đặc biệt với sự phát triển mạnh mẽ của ngành công nghiệp điện ảnh hoặc có thể là do một số yếu tố khác như chiến lược tiếp thị, chất lượng phim, ... được triển khai hiệu quả.

	Title	Release Date (Streaming)
9599	My Man	1928-12-18
22947	Cinema Sabaya	2023-06-13

Trong dữ liệu bộ phim được phát hành với thời gian lâu nhất là “My Man” ra đời vào cuối năm 1928 (cách đây gần 1 thế kỷ). Và bộ phim được phát hành gần đây nhất là “Cinema Sabaya” sẽ được phát trên nền tảng trực tuyến vào tháng 6 năm 2023.

Top 20 đạo diễn xuất hiện nhiều nhất:

Woody Allen	51
Alfred Hitchcock	49
John Ford	40
Clint Eastwood	39
Werner Herzog	37
Ingmar Bergman	37
Steven Soderbergh	36
Martin Scorsese	36
Sidney Lumet	35
Spike Lee	35
Fritz Lang	35
Jean-Luc Godard	34
Michael Curtiz	34
Steven Spielberg	34
Ron Howard	32
John Huston	32
Raoul Walsh	31
Blake Edwards	30
Howard Hawks	30
Robert Altman	30

Bảng trên là kết quả của top 20 đạo diễn có số lượng phim nhiều nhất. Nổi bật nhất là đạo diễn Woody Allen chiếm tổng số phim nhiều nhất là 51 bộ phim - từng giành 3 Giải Oscar cho các hạng mục đạo diễn xuất sắc và kịch bản gốc với phong cách làm phim trí tuệ và rất nhiều tác phẩm xuất sắc. (thông tin tra ngoài)

Top 10 nhà sản xuất có số lượng phim nhiều nhất:

Jason Blum	89
Tim Bevan	89
Scott Rudin	88
Eric Fellner	85
Brian Grazer	79
Arnon Milchan	63
Joel Silver	59
Christine Vachon	58
Randall Emmett	54
John Davis	54

Bảng kết quả là top 10 nhà sản xuất có số lượng phim nhiều nhất với ≥ 54 bộ phim. Nổi bật là 2 nhà sản xuất “Jason Blum” và “Tim Bevan” với số lượng phim lên đến 89 bộ phim.

Top 10 Writer:

	7232
Woody Allen	49
Mark Monroe	41
Luc Besson	41
Ingmar Bergman	36
Werner Herzog	33
Ben Hecht	28
Jean-Luc Godard	28
John Hughes	28
William Shakespeare	27

Name: Writer, dtype: int64

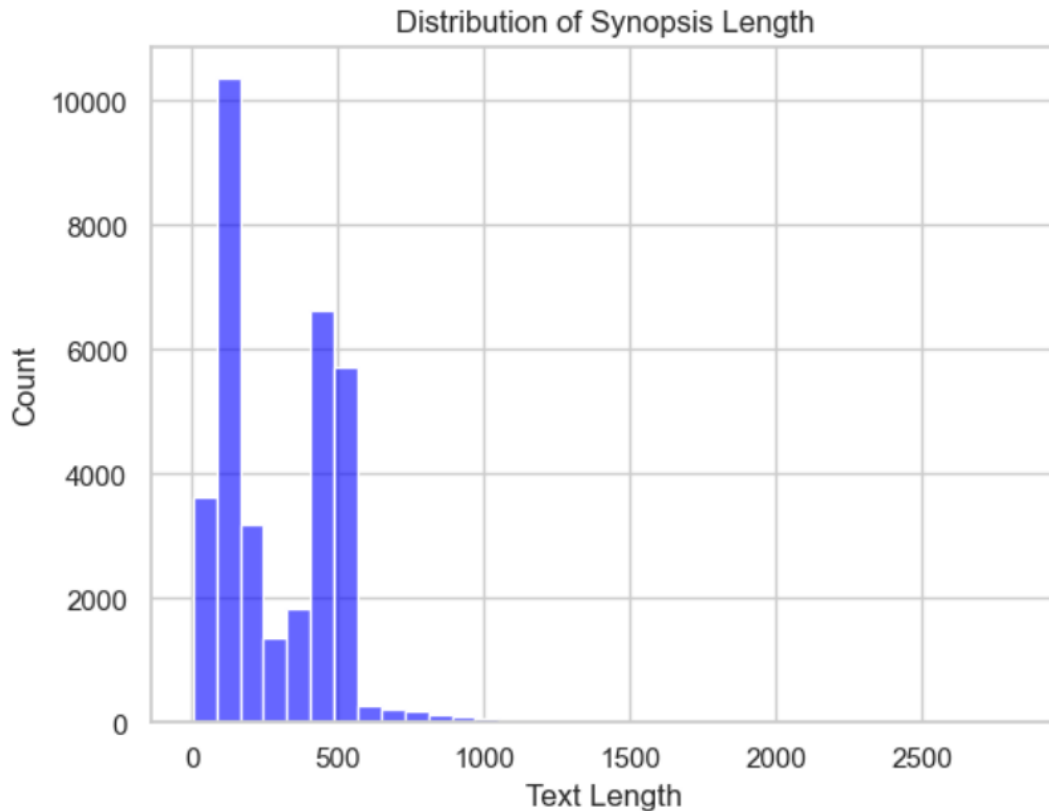
Dữ liệu sau khi phân tích thì có thể thấy số đứng vị trí cao nhất với số lượng phim 49 bộ là biên kịch Woody Allen. Và ông cũng chính là người đạo diễn duy nhất có số lượng phim trên 50 bộ trong tổng số dữ liệu. (thông tin từ: Trang web IMDb)

Top 10 Distributor:

	16439
Paramount Pictures	848
Universal Pictures	701
20th Century Fox	679
Warner Bros. Pictures	617
IFC Films	548
Columbia Pictures	471
Metro-Goldwyn-Mayer	431
Sony Pictures Classics	416
Lionsgate Films	408

Name: Distributor, dtype: int64

Nhìn hình có thể thấy, có 5 nhà phân phối phim đã phân phối hơn 500 bộ phim. Và không khó để hiểu điều này khi các nhà phân phối trên đều là những công ty lớn, có sự hiện diện mạnh mẽ trên toàn cầu và là những nhà phân phối phim hàng đầu của Hollywood. Tuy nhiên, nổi bật nhất vẫn là công ty Paramount Pictures đứng vị trí cao nhất với 848 bộ phim.



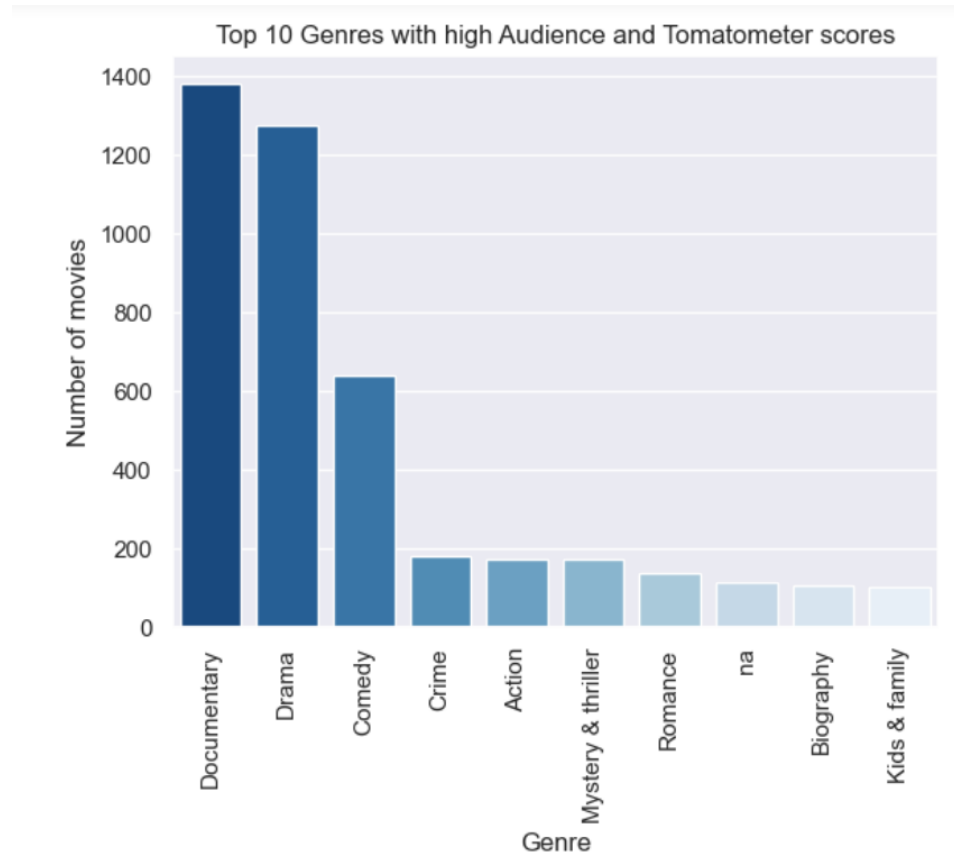
Nhìn vào biểu đồ, ta có thể thấy rằng hầu hết các bộ phim có độ dài tóm tắt dưới 500 từ. Điều này có thể được hiểu là phần lớn các bộ phim có nội dung tóm tắt ngắn gọn và súc tích, có thể do sở thích của khán giả hiện nay đang tập trung vào các bộ phim có nội dung đơn giản, dễ hiểu và nhanh chóng. Trong đó hơn 10000 bộ phim có độ dài tóm tắt nằm trong khoảng 80-160 từ.

Tổng quan về dữ liệu gồm có:

- Ngôn ngữ phim chiếm số lượng nhiều nhất là 'English', gần 73% bộ phim của dữ liệu.
- Thể loại phim chiếu nhiều nhất trong dữ liệu là "Drama" với 26,7%.
- Số lượng bộ phim của năm 2017 hơn 2000 phim, đứng vị trí thứ nhất so với các năm khác.
- Nổi bật nhất trong số 17659 đạo diễn là Woody Allen với tổng số phim nhiều nhất là 51 bộ phim.
- 2 nhà sản xuất "Jason Blum" và "Tim Bevan" nổi bật với số lượng phim lên đến 89 bộ phim.
- Nhà biên kịch đứng vị trí cao nhất với số lượng phim 49 bộ là biên kịch Woody Allen.
- Có 5 nhà phân phối phim đã phân phối hơn 500 bộ phim, nổi bật nhất vẫn là

công ty Paramount Pictures đứng vị trí cao nhất với 848 bộ phim.
- Hầu hết các bộ phim có độ dài tóm tắt dưới 500 từ.

2. EDA



- Một bộ phim thật sự đáng xem và có chất lượng tốt là bộ phim mà Tomatometer score - được tính dựa trên tỷ lệ phần trăm các đánh giá tích cực trên tổng số các đánh giá - cao hơn 80%. Tương tự Audience score cao hơn 80% thường được coi là một bộ phim có độ phổ biến và đáng xem.

- Phân tích ở những biểu đồ trước đó cho thấy những bộ phim “Drama” chiếm phần lớn dữ liệu, tuy nhiên xét về bộ phim phổ biến, chất lượng cao và có hơn 80% đánh giá tích cực thì thể loại phim “Document” chiếm lớn hơn với gần 1400 bộ phim. Điều này có thể vì các bộ phim tài liệu thường được đánh giá cao bởi các nhà phê bình vì tính chân thực, sự thật và nội dung tương tác với khán giả.

- Các bộ phim này thường được tạo ra để thử nghiệm và thách thức các quan điểm định kiến của khán giả, đồng thời mang lại cho họ sự hiểu biết mới về các vấn đề xã hội và thế giới. Trong khi đó, với các bộ phim Drama, những yếu tố như kịch bản, diễn xuất, đạo diễn và sản xuất đều ảnh hưởng đến chất lượng của bộ phim và điều này có thể dẫn đến sự đánh giá khác nhau từ khán giả và nhà phê bình.

Top 10 bộ phim có doanh thu phòng vé cao nhất:

	Title	Box Office	Release Date
25323	Avengers: Endgame	858400000.0	2019-07-30
33677	Spider-Man: No Way Home	814100000.0	NaT
16758	Top Gun: Maverick	718500000.0	2022-08-22
4408	Confess, Fletch	711600000.0	2022-09-16
9738	Black Panther	700200000.0	2018-05-02
24608	Avengers: Infinity War	678800000.0	2018-08-14
20198	Avatar: The Way of Water	678300000.0	2023-03-28
31735	Titanic	658800000.0	2002-01-08
14338	Jurassic World	652600000.0	2015-10-20
19585	Star Wars: The Last Jedi	620200000.0	2018-03-27

- Kết quả trả về cho thấy những bộ phim nổi tiếng và thành công nhất của Hollywood trong thời gian gần đây. Trong số này, Avengers: Endgame và Black Panther đều là những bộ phim thuộc vũ trụ điện ảnh Marvel, đạt doanh thu rất lớn trong thị trường toàn cầu. Titanic và Jurassic World cũng là những bộ phim thành công lớn và thu hút được đông đảo khán giả.

- Hơn nữa, trong dữ liệu có thể thấy rằng các bộ phim của Avengers đã thu hút được sự quan tâm của khán giả liên tục qua các năm từ 2019-2021. Bộ phim Spider-Man: No Way Home cũng được đề cập, đây là một bộ phim mới được phát hành vào năm 2021 và thuộc vũ trụ điện ảnh Marvel. Thông tin một lần nữa chứng minh vũ trụ điện ảnh Marvel đã trở thành một trong những thương hiệu phim lớn và thu hút được sự quan tâm của đông đảo khán giả trong suốt nhiều năm.

No.	Title	Runtime	Tomatometer score	Audience score	Genre
0 10335	Jerrod Carmichael: Rothaniel	60	100	83.0	Comedy
1 11210	The Adventures of Prince Achmed	60	100	90.0	Kids & family
2 12933	Sound and Fury	60	97	88.0	Documentary
3 3983	Stories of Our Lives	60	100	100.0	Drama
4 2147	Bo Burnham: Make Happy	60	100	90.0	Stand-up
5 5933	The Weight of Gold	60	94	94.0	Documentary
6 25480	Anthony Jeselnik: Thoughts and Prayers	60	100	90.0	Comedy
7 24537	The Kid	60	100	95.0	Comedy
8 24240	Charlie Is My Darling	60	100	82.0	Documentary
9 14484	Saudi Women's Driving School	60	100	86.0	Documentary

- Bảng trên là kết quả của top 10 bộ phim có Runtime ngắn nhất nhưng Tomatometer score và Audience score trên 80 , đặc điểm chung là thời lượng phim 60 phút và các bộ phim thuộc nhiều thể loại khác nhau. Điều này cho thấy rằng không những chất lượng của bộ

phim, thể loại phim quan trọng mà thời lượng cũng có thể ảnh hưởng đến sự đánh giá của khán giả và giới phê bình. Hay có thể nói những bộ phim này là những bộ phim có nội dung cô đọng, súc tích nhưng vẫn đảm bảo chất lượng phim.

- Thời lượng phim có thể ảnh hưởng đến trải nghiệm xem phim của khán giả. Một bộ phim quá dài có thể khiến khán giả mệt mỏi, không tập trung vào nội dung của phim và không thể cảm nhận được thông điệp mà bộ phim muốn truyền tải. Ngược lại, một bộ phim quá ngắn có thể khiến khán giả cảm thấy thiếu sự trọn vẹn của câu chuyện và không đủ thời gian để phát triển các nhân vật và tình tiết. Vì vậy thời lượng phim phải đủ để phát triển các nhân vật và tình tiết, nhưng cũng không quá dài để không khiến khán giả mệt mỏi.

	No.	Title	Release Date (Streaming)
0	762	The Fugitive	1997-03-25
1	3838	Cool Hand Luke	1997-06-24
2	1255	Shine	1997-08-22
3	33094	Mad Max 2: The Road Warrior	1997-08-22
4	32142	Goodfellas	1997-08-22
5	12517	Mask	1997-08-22
6	25689	The Right Stuff	1997-08-22
7	29711	The Bridges of Madison County	1997-08-22
8	7620	The Birdcage	1997-08-27
9	27235	The Chinese Connection	1997-10-10

- Bảng trên là kết quả của top 10 bộ phim theo thời gian phát hành tại rạp lâu đời nhất nhưng đồng thời nhận được số điểm từ khán giả và các nhà phê bình phim trên 80, hay có thể gọi là những bộ phim bất hủ. Năm 1997 được xem là một năm đánh dấu sự phát triển của ngành công nghiệp điện ảnh với sự ra mắt của nhiều bộ phim đình đám, và các bộ phim trong top 10 đều là những tác phẩm nổi bật trong năm đó. Sự lâu đời của các bộ phim này cũng cho thấy rằng việc đầu tư vào chất lượng và nội dung của bộ phim là rất quan trọng. Những bộ phim được sản xuất với tâm huyết và chất lượng cao sẽ luôn có giá trị, ảnh hưởng lớn đến khán giả và luôn được khán giả nhớ đến.

```

avg_runtime_by_genre:
  Genre
  History          123.505241
  War              120.423237
  Biography        115.707358
  Gay & lesbian    113.000000
  Action           106.811047
  Musical          106.447727
  Western          106.267477
  Other            106.000000
  Crime            105.107312
  Romance          103.973422
  Adventure        103.896926
  Drama            102.293057
  Sci-fi           101.643735
  Mystery & thriller 101.081458
  Fantasy          100.450673
  Comedy           96.764837
  Holiday          95.339623
  Horror           92.375715
  Kids & family    91.494516
  Documentary      88.284002
  Lgbtq+           85.000000
  Music            83.000000
  Stand-up         49.833333
  Animation        36.086957

```

- Kết quả thời lượng trung bình của các bộ phim trong thể loại 'History', 'War' khá dài (hơn 2 tiếng). Điều này không khó để giải thích bởi việc sản xuất các bộ phim trong thể loại này đòi hỏi sự tập trung cao độ và sự chuẩn bị kỹ lưỡng. Các bộ phim trong thể loại này thường có nội dung phức tạp và chi tiết lịch sử phải được trình bày một cách cặn kẽ, điều này đòi hỏi sự nghiên cứu và tìm hiểu kỹ lưỡng từ đội ngũ sản xuất để đảm bảo tính chính xác và độ tin cậy của bộ phim. Do đó, thời lượng trung bình của các bộ phim trong thể loại 'History', 'War' cao hơn so với các thể loại khác.
- Bên cạnh đó các thể loại phim như 'Animation', 'Stand-up' và 'Variety' thường có nội dung đơn giản và thiết kế để giải trí nhanh và dễ tiếp cận cho khán giả. Do đó, thời lượng của các bộ phim trong các thể loại này thường ngắn hơn so với các thể loại khác.
- Qua đó, cho thấy rằng thời lượng của một bộ phim cần phải phù hợp với nội dung và thể loại của nó để đáp ứng nhu cầu giải trí của khán giả và thu hút sự quan tâm của họ.

IV. Tiền xử lý và đổ dữ liệu lên MS Azure

Tạo database trên Azure:

The screenshot displays the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information. The main content area shows the deployment progress for a new SQL database server. The deployment is currently in progress, with a status of 'Accepted'. The deployment details table lists the resources being deployed, including the server, firewall rules, and databases. The deployment is expected to be completed by 10:22:38 AM on 5/4/2023.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Microsoft.SQLDatabase.newDatabaseNewServer_2a07038402444c1dac2cb | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Overview

Inputs

Outputs

Template

Deployment is in progress

Deployment name: Microsoft.SQLDatabase.newDatabaseNewServer... Start time: 5/4/2023, 10:22:38 AM
Subscription: Azure for Students Correlation ID: 3f691f7f-b3a0-47ee-be6c-97673d48abd2

Resource group: QT_CSDL

Deployment details

Resource	Type	Status	Operation details
tomatoesrotten	Microsoft.Sql/servers	Accepted	Operation details

Give feedback

Tell us about your experience with deployment

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

Free Microsoft tutorials

Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Microsoft.SQLDatabase.newDatabaseNewServer_2a07038402444c1dac2cb | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Overview

Inputs

Outputs

Template

Your deployment is complete

Deployment name: Microsoft.SQLDatabase.newDatabaseNewServer... Start time: 5/4/2023, 10:22:38 AM
Subscription: Azure for Students Correlation ID: 3f691f7f-b3a0-47ee-be6c-97673d48abd2

Resource group: QT_CSDL

Deployment details

Resource	Type	Status	Operation details
tomatoesrotten/Clientip-2023-5-4_1	Microsoft.Sql/servers/firewallrules	OK	Operation details
tomatoesrotten/AllowAllWindowsAz	Microsoft.Sql/servers/firewallrules	OK	Operation details
tomatoesrotten/rotten_tomatoes	Microsoft.Sql/servers/databases	Created	Operation details
tomatoesrotten/Default	Microsoft.Sql/servers/connectionP...	OK	Operation details
tomatoesrotten	Microsoft.Sql/servers	OK	Operation details
tomatoesrotten	Microsoft.Sql/servers	Created	Operation details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

Cost Management

Get notified to stay within your budget and prevent unexpected charges on your bill.

Set up cost alerts >

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

Free Microsoft tutorials

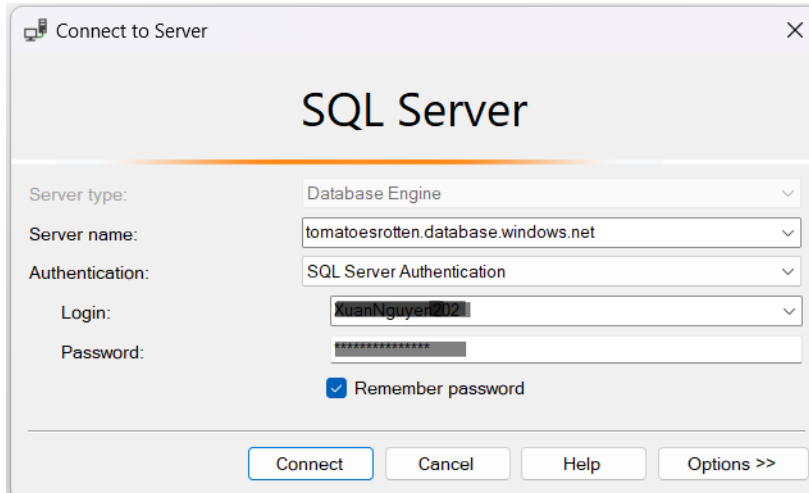
Start learning today >

Work with an expert

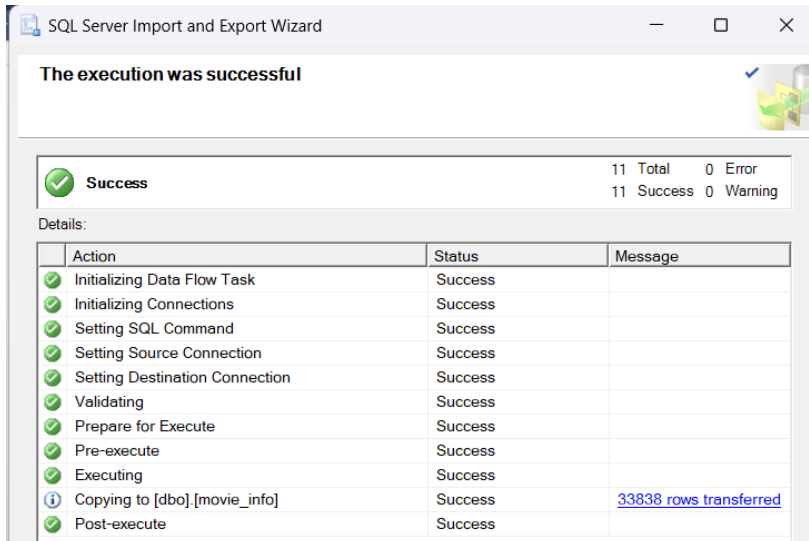
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

Connect SQL server đã tạo:

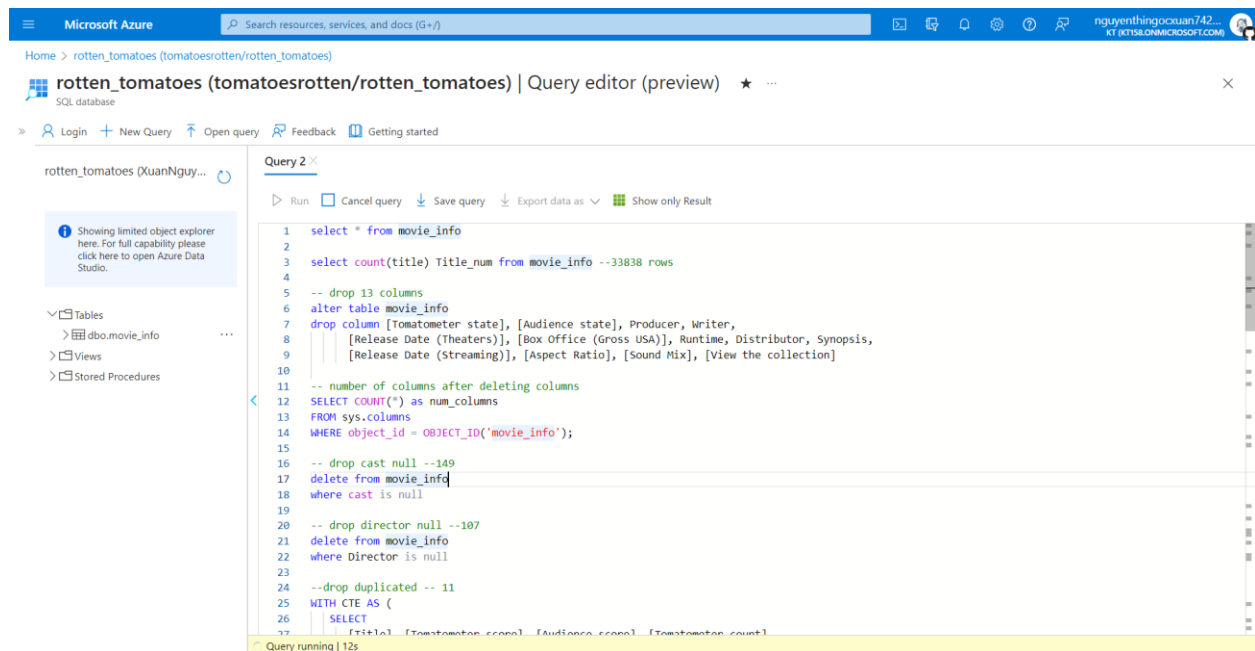


Import data (data gốc thu thập từ rotten tomatoes):



Action	Status	Message
Initializing Data Flow Task	Success	
Initializing Connections	Success	
Setting SQL Command	Success	
Setting Source Connection	Success	
Setting Destination Connection	Success	
Validating	Success	
Prepare for Execute	Success	
Pre-execute	Success	
Executing	Success	
Copying to [dbo].[movie_info]	Success	33838 rows transferred
Post-execute	Success	

Sau đó, tiến hành tiền xử lý trực tiếp trên azure:



Tiền xử lý theo trình tự trong file Preprocessing.sql

Gồm: **(Dữ liệu gồm 23 cột, 33838 dòng)**

- Xóa các cột không quan trọng (13 cột: [Tomatometer state], [Audience state], [Producer], [Writer], [Release Date (Theaters)], [Box Office (Gross USA)], [Runtime], [Distributor], [Synopsis], [Release Date (Streaming)], [Aspect Ratio], [Sound Mix], [View the collection])

```
-- drop 13 columns
alter table movie_info
drop column [Tomatometer state], [Audience state], [Producer], [Writer],
[Release Date (Theaters)], [Box Office (Gross USA)], [Runtime], [Distributor], [Synopsis],
[Release Date (Streaming)], [Aspect Ratio], [Sound Mix], [View the collection]
```

- Xóa những dòng null của cột Cast (149 dòng), cột Director (107 dòng)

```
-- drop cast null --149
delete from movie_info
where cast is null
```

```
-- drop director null --107
delete from movie_info
where Director is null
```

- Xóa duplicated (11 dòng)

```
--drop duplicated -- 11
WITH CTE AS (
    SELECT
        [Title], [Tomatometer score], [Audience score], [Tomatometer count],
        [Audience count], [Genre], [Original Language], [Cast], [Rating],
        RN = ROW_NUMBER() OVER(PARTITION BY [Title], [Tomatometer score],
        [Audience score], [Tomatometer count], [Audience count], [Genre],
        [Original Language], [Cast], [Rating] ORDER BY [Title])
    FROM dbo.movie_info
)
DELETE FROM CTE WHERE RN > 1;
```

- Thay thế những dòng null của cột [Original Language] bằng giá trị mode (996 dòng)

```
--replace original language by mode (996 rows effected)
```

```
UPDATE movie_info
SET [Original Language] = (
    SELECT TOP 1 [Original Language]
    FROM movie_info
    WHERE [Original Language] IS NOT NULL
    GROUP BY [Original Language]
    ORDER BY COUNT(*) DESC
)
WHERE [Original Language] IS NULL;
```

- Thay những giá trị null ở cột [Audience Score] = [Tomatometer Score] (3433 dòng)

```
-- fill audience score with tomatoes score (3433 rows effected)
```

```
update movie_info
set [Audience score] = [Tomatometer score]
where [Audience score] is null
```

- Chỉnh sửa dạng dữ liệu của cột Rating (xóa các mô tả chi tiết, chỉ giữ lại kí hiệu chính: R, PG,..)

```
--Rating
SELECT rating FROM movie_info
--movie --33571 --> 11 distinct values
update movie_info
set rating =
CASE
    WHEN Rating LIKE 'R%' THEN 'R'
    WHEN Rating LIKE 'PG-13%' THEN 'PG-13'
    WHEN Rating LIKE 'PG %' THEN 'PG'
    WHEN Rating LIKE 'G%' THEN 'G'
    when rating is null then 'NR'
    when rating like 'NC-17%' then 'NC-17'
    when rating like 'TVMA%' then 'TVMA'
    when rating like 'TVPG%' then 'TVPG'
    when rating like 'TV14%' then 'TV14'
    when rating like 'TVY7%' then 'TVY7'
    ELSE Rating
END
FROM movie_info
```

- Xóa giá trị null ở cột [Genre] (672 dòng)
- Xóa các ký tự alphabet ở cột [Tomatometer count], chỉ giữ lại số
- Sửa định dạng giá trị cột [Audience count]

```
--audience count
SELECT DISTINCT [Audience count] FROM movie_info
update movie_info
set [Audience count] =
    case
    when [Audience count] like '%+%' then SUBSTRING([Audience count],1, PATINDEX('% [a-zA-Z]%', [Audience count]))
    when [Audience count] like '%Fewer%' then SUBSTRING([Audience count], PATINDEX('%[0-9]%', [Audience count]),2)+'-'
    when [Audience count] like 'Ratings' or [Audience count] like '0 Ratings' then '0'
    else [Audience count]
    end
```

Sau khi tiền xử lý, dữ liệu còn 32899 dòng và 10 cột.

rotten_tomatoes (tomatoesrotten/rotten_tomatoes) | Query editor (preview) ✱

SQL database

Login + New Query Open query Feedback Getting started

rotten_tomatoes (XuanNguy...) 🔄

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

Tables

- dbo.movie_info
 - Title (nvarchar, null)
 - Tomatometer score (float, null)
 - Audience score (float, null)
 - Tomatometer count (nvarchar, null)
 - Audience count (nvarchar, null)
 - Genre (nvarchar, null)
 - Original Language (nvarchar, null)
 - Director (nvarchar, null)
 - Cast (nvarchar, null)
 - Rating (nvarchar, null)
- Views
- Stored Procedures

Query 2 ✕

Run Cancel query Save query Export data as Show all

Results Messages

Search to filter items...

Title	Tomatometer score	Audience score	Tomatometer count	Audience count	Genre
Following	82	85	33	10,000+	Mystery & thriller
Bad Genius	100	93	21	2,500+	Mystery & thriller
Vivere	44	58	16	250+	Drama, Romance, Lgbtq+
Corpus Callosum	63	63	8	50-	Fantasy, Comedy
Enlighten Up!	56	50	39	5,000+	Documentary
10 Years	60	40	58	10,000+	Comedy
The Sitter	22	39	113	25,000+	Comedy
The Rewrite	66	42	62	5,000+	Romance, Comedy
How to Rob	100	100	6	100+	Drama, Crime
Chacun Son Cinema	100	71	6	1,000+	Comedy, Drama
The House of the Spirits	32	72	38	5,000+	Drama

Query succeeded | 52s

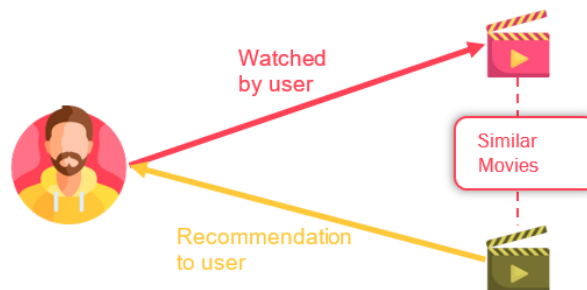
Export file để xây model.

V. Xây dựng mô hình học máy với Python

Tiến hành xây dựng mô hình đề xuất phim dựa trên phương pháp Content-based.

Giới thiệu sơ lược về phương pháp Content-based:

Phương pháp Content-based trong đề xuất phim là một phương pháp giúp đề xuất phim cho người dùng dựa trên một số thuộc tính quan trọng của phim, như thể loại, diễn viên, đạo diễn, năm sản xuất, đánh giá, và mô tả của phim. Phương pháp Content-based sử dụng các thuộc tính này để tìm các phim tương tự hoặc liên quan đến phim mà người dùng đã xem hoặc đang quan tâm.



Mô hình content-based nhóm áp dụng sẽ chủ yếu dựa trên ý kiến của các chuyên gia phê bình phim. (Điểm đánh giá của các nhà phê bình, số lượt đánh giá từ các nhà phê bình)

Import thư viện cần thiết và kiểm tra một số thông tin cơ bản của dataset.

```
import pandas as pd
```

```
path = '/kaggle/input/rotten-tomato-metadata-scraping/movie_info_cleaned.xlsx'
# path = '/content/drive/My Drive/DBmana/Dataset/movie_info_cleaned.xlsx'
film_data = pd.read_excel(path)
```

```
film_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32899 entries, 0 to 32898
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Title                  32899 non-null  object
1   Tomatometer score      32899 non-null  int64
2   Audience score         32899 non-null  int64
3   Tomatometer count      32899 non-null  int64
4   Audience count         32899 non-null  object
5   Genre                  32899 non-null  object
```



```
6  Original Language  32899 non-null  object
7  Director          32899 non-null  object
8  Cast              32899 non-null  object
9  Rating            32899 non-null  object
```

```
dtypes: int64(3), object(7)
```

```
memory usage: 2.5+ MB
```

```
film_data.shape
```

```
(32899, 10)
```

```
film_data.duplicated().value_counts()
```

```
False    32899
```

```
dtype: int64
```

Công thức tương tự công thức tính trọng số phim của IMDB

$$\text{Weighted Rating (WR)} = \left(\frac{v}{v+m} \cdot R \right) + \left(\frac{m}{v+m} \cdot C \right)$$

where:

- v is the number of reviews by critics for the movie;
- m is the minimum number of reviews by critics required to be listed;
- R is the average score of the movie by critics;
- And C is the mean number of reviews by critics across the whole dataset

Tiến hành xây dựng hàm tính trọng số cho phim dựa trên công thức của IMDB.

Với m là số lượt đánh giá tối thiểu từ các nhà phê bình, ở đây m được chọn là ngưỡng 90% ($m=120$). Hay nói cách khác, 90% các bộ phim trong dataset có dưới 120 lượt đánh giá từ các nhà phê bình, vậy nên để một bộ phim được đề xuất thì bộ phim đó phải có ít nhất 120 lượt đánh giá từ các nhà phê bình.

```
C = film_data['Tomatometer count'].mean()
C
```

42.744521110064134

```
m= film_data['Tomatometer count'].quantile(0.90)
m
# (film_data['Tomatometer count'] >= 120).sum()
```

120.0

```
def weighted_rating(x, m=m, C=C):
    v = x['Tomatometer count']
    R = x['Tomatometer score']
    # Calculation based on the WR formula
    return (v/(v+m) * R) + (m/(m+v) * C)
```

Mô hình content-based nhóm xây dựng sẽ dựa trên 3 thuộc tính chính của phim là: Thể loại, đạo diễn và diễn viên.

```
# These three columns will be used to create a content-based model.
film_data[['Genre', 'Director', 'Cast']].head(2)
```

	Genre	Director	Cast
0	Action, Adventure, Fantasy, Sci-fi	Ruben Fleischer	Tom Hardy, Michelle Williams, Riz Ahmed, Scott...
1	Horror, Mystery & thriller	James Ashcroft	Daniel Gillies, Matthias Luafutu, Miriama McDo...

Ở 2 cột thể loại và diễn viên chỉ giữ lại 5 giá trị đầu tiên nếu có nhiều hơn 5 giá trị. (Nhiều hơn 5 thể loại phim, dàn diễn viên nhiều hơn 5 người)

```
# Select the first five elements in the 'Genre' and 'Cast' columns.
features = ['Genre', 'Cast']
for feature in features:
    film_data[feature] = film_data[feature].apply(lambda x: ', '.join(x.split(
        ', ')[0:5]))
```

Chuyển dữ liệu văn bản về dạng chữ viết thường và bỏ khoảng trắng thừa.

```
# Function to convert all strings to lower case and strip names of spaces
features = ['Genre', 'Director', 'Cast']
for feature in features:
    film_data[feature] = film_data[feature].apply( lambda s: ', '.join(i.strip().lower() for i in s.split(',')))
```

Tạo thêm cột tổng hợp (“Soup”) nhằm gộp giá trị ở 3 cột thể loại, đạo diễn và diễn viên về thành 1 cột duy nhất. Sau cột thể loại và đạo diễn, để tránh hiện tượng văn bản dính nhau khi gộp, ta cần nối thêm một khoảng trắng (‘ ’).

```
# Create soup
film_data['soup'] = film_data['Genre'] + ' ' + film_data['Director'] + ' ' + film_data['Cast']
```

```
film_data['soup'][0:3]
```

```
0    action, adventure, fantasy, sci-fi ruben fleis...
1    horror, mystery & thriller james ashcroft dani...
2    comedy, drama peter hedges katie holmes, patri...
```

Bỏ dấu phẩy giữa các giá trị và nối lại bằng khoảng trắng (hay thay ký tự “,” thành “ ”)

```
film_data['soup'] = film_data['soup'].apply(lambda x: ' '.join(x.split(',')))
```

```
film_data['soup'][0:3]
```

```
0    action adventure fantasy sci-fi ruben fleische...
1    horror mystery & thriller james ashcroft danie...
2    comedy drama peter hedges katie holmes patrici...
Name: soup, dtype: object
```

Xây dựng ma trận CountVectorizer

Ta có bộ từ vựng là tập hợp các từ duy nhất xuất hiện trong cột ‘soup’ của dataframe film_data.

Ta sẽ có một ma trận mà mỗi cột đại diện cho một từ xuất hiện trong bộ từ vựng, và mỗi hàng đại diện cho một bộ phim. Giá trị trong mỗi ô của ma trận là tần suất xuất hiện của từ tương ứng trong bộ phim đó.

Mô đun CountVectorizer của thư viện sklearn được dùng để tính toán ma trận này.

```
# Import CountVectorizer and create the count matrix
from sklearn.feature_extraction.text import CountVectorizer

count = CountVectorizer(stop_words='english')
count_matrix = count.fit_transform(film_data['soup'])
```

Tính giá trị cosine (độ đo tương tự) cho toàn bộ dữ liệu phim với mô đun cosine_similarity.

```
# Compute the Cosine Similarity matrix based on the count_matrix
from sklearn.metrics.pairwise import cosine_similarity
cosine_sim = cosine_similarity(count_matrix, count_matrix)
```

Reset index của data frame film_data và tạo ra 1 series với tên phim và index tương ứng của phim.

```
# Reset index of our main DataFrame and construct reverse mapping as before
film_data = film_data.reset_index()
indices = pd.Series(film_data.index, index=film_data['Title'])
```

Xây dựng hàm recommend phim:

Lấy ra index phim cần đề xuất, lấy ra các giá trị cosine và chọn lấy 15 bộ phim có giá trị cosine (độ tương tự) cao nhất.

```
def get_recommendations(title):
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:16]
    movie_indices = [i[0] for i in sim_scores]
```

```

movies = film_data.iloc[movie_indices][['Title', 'Genre', 'Tomatometer count', 'Tomatometer score']]
C = film_data['Tomatometer count'].mean()
m = film_data['Tomatometer count'].quantile(0.90)
t = film_data['Tomatometer score'].quantile(0.50)
qualified = movies[(movies['Tomatometer count'] >= m) & (movies['Tomatometer score'] >= t)]
qualified['WR_score'] = qualified.apply(weighted_rating, axis=1)
qualified = qualified.sort_values('WR_score', ascending=False).head(15)
return qualified

```

Kết quả trả về sẽ là một data frame với các bộ phim tương tự được đề xuất, data frame đầu ra sẽ có các cột như: Tên phim, thể loại, số lượt đánh giá từ các nhà phê bình, điểm số của nhà phê bình.

Điều kiện để một bộ phim được đề xuất là có tối thiểu 120 lượt đánh giá từ các nhà phê bình (ngưỡng 90%) và có điểm số từ nhà phê bình ít nhất là 72 (ngưỡng 50%). Các bộ phim sẽ được sắp xếp giảm dần theo trọng số phim.

Ví dụ sau khi chạy mô hình đề xuất:

```

film_data.loc[film_data['Title'] == 'Black Panther', film_data.columns[:7]]

```

	index	Title	Tomatometer score	Audience score	Tomatometer count	Audience count	Genre
29917	29917	Black Panther	96	79	531	50,000+	action, adventure, fantasy

Black Panther có thể loại là hành động, phiêu lưu, hư cấu.

```

get_recommendations('Black Panther')

```

Hệ thống đề xuất ra được các bộ phim có thể loại tương tự và đáp ứng đủ các điều kiện về số lượt đánh giá từ các nhà phê bình (≥ 120), điểm số từ nhà phê bình (≥ 72) và trọng số (WR) giảm dần, có 2 phim là Creed và Fruitvale Station tuy không cùng thể loại nhưng có chung đặc điểm là có nam diễn viên Michael B. Jordan thủ vai.

	Title	Genre	Tomatometer count	Tomatometer score	WR_score
10402	Creed	drama	315	95	80.584695
18530	The Jungle Book	kids & family, adventure, action, fantasy	330	94	80.331872
8569	Ant-Man and The Wasp	action, adventure, fantasy, comedy	445	87	77.600606
11004	Avengers: Infinity War	action, adventure, fantasy, sci-fi	490	85	76.687447
10444	Fruitvale Station	drama	216	94	75.694472
14375	Black Panther: Wakanda Forever	action, adventure, fantasy	430	84	74.998805
11774	The Hobbit: The Desolation of Smaug	fantasy, adventure	254	74	63.971504

```
film_data.loc[film_data['Title'] == 'Toy Story 4', film_data.columns[:7]]
```

	index	Title	Tomatometer score	Audience score	Tomatometer count	Audience count	Genre
7644	7644	Toy Story 4	97	94	460	50,000+	kids & family, comedy, adventure, fantasy, ani...

	Title	Genre	Tomatometer count	Tomatometer score	WR_score
5802	Toy Story 3	kids & family, comedy, adventure, fantasy, ani...	311	98	82.615644
19557	Toy Story 2	kids & family, comedy, adventure, fantasy, ani...	171	100	76.389493
8493	Onward	kids & family, comedy, adventure, fantasy, ani...	345	88	76.321167
9706	How to Train Your Dragon 2	kids & family, fantasy, adventure, comedy, ani...	189	92	72.871659

VI. Ứng dụng Web app với thư viện Streamlit trên Python.

Một số câu lệnh cơ bản để triển khai Web app với Streamlit:

```
st.set_page_config(
    page_title="RECOMMEND SYSTEM",
    page_icon="🔥",
    layout="wide")
st.title('**:blue[FILM RECOMMENDATION SYSTEM]**')
with st.form("Thông tin"):
    options = data["Title"]
    name = st.selectbox('**:red[Typing the film title]**', options=options)
    submit = st.form_submit_button('**:Get films**')
```

```
# Nút tìm kiếm
if submit:
    with st.spinner("Loading..."):
        time.sleep(0.25)
    try:
        a = recommendations(name)
        st.success('**Success**', icon="✅")
        st.write('**:orange[Here are movies similar to]**', name)
        for i in range(len(a)):
            with st.form('' + str(i) + ''):
                st.markdown(f':green[**🎬Title**:] {a.iloc[i, 0]}')
                st.markdown(f':green[**🎬Genre**:] {a.iloc[i, 1]}')
                st.markdown(f':green[**📊Tomatometer count**:] {a.iloc[i, 2]}')
                st.markdown(f':green[**📊Tomatometer score**:] 🍅 {a.iloc[i, 3]} 🍅')
                st.markdown(f':green[**📊WR_score**:] {a.iloc[i, 4]}')
                submit = st.form_submit_button(str(i + 1), disabled=True)
    except:
        st.error('There are no movies that are similar to ' + name, icon="❌")
```

Vì tài nguyên, khả năng tính toán của bên thứ 3 là có hạn (phần tính toán chỉ số cosine cho hơn 32000 phim cần đến hơn 13GB RAM) nên để đảm bảo mô hình có thể chạy được trên website và đảm bảo một tốc độ không quá chậm, nhóm quyết định demo với dữ liệu khoảng 5000 dòng đầu tiên của dataset (< 6.5 lần so với dataset ban đầu), tuy nhiên mô hình vẫn có thể chạy được với dataset gốc trên google colab.

Upload tài nguyên cần thiết lên github:

The screenshot shows a GitHub repository interface. At the top, there are navigation buttons: 'main', 'Go to file', 'Add file', 'Code', and 'About'. The repository name is 'TopdevVN Update RecommendSystem.py', with a green checkmark and '2 hours ago' indicating a recent update. Below the repository name is a table listing files and their update status:

File Name	Update Description	Update Time
CNAME	Create CNAME	yesterday
README.md	Update README.md	yesterday
RecommendSystem.py	Update RecommendSystem.py	2 hours ago
film_data.csv	Add files via upload	11 hours ago
requirements.txt	Rename Resources/requirements.txt to requi...	yesterday

Below the file list, the 'README.md' content is visible, starting with the title 'FilmRecommendation'. On the right side of the repository page, there is a description: 'Basic Web App Movie Recommendation using Streamlit library.' and a link to 'basicfilmrecommendationsystem.o...'. There are also tags for 'movie' and 'recommendation-system', and statistics showing 0 stars, 1 watching, and 0 forks.

Deploy mô hình với Render:

render	Dashboard	Blueprints	Env Groups	Docs	New +	Phan Cong Hieu	▼
Overview							
🔍 Search services							
NAME	STATUS		TYPE	RUNTIME	REGION		
🌐 BasicFilmRecommendationSystem	● Deploy succeeded		Web Service	Python 3	Singapore		
🌐 BasicMovieRecommendationSystem	● Deploy failed		Web Service	Python 3	Singapore		
🌐 BasicFilmRecom	● Deploy failed		Web Service	Python 3	Singapore		

Kết quả sau khi deploy: [Link Web app](#)

