ĐỒ ÁN CUỐI KỲ

ĐỀ TÀI : PHÂN TÍCH DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH VỚI DỮ LIỆU TRÊN TRANG WEB IMDB

GIẢNG VIÊN HƯỚNG DẪN : THẦY TRẦN TRUNG KIÊN

THÀNH VIÊN THỰC HIỆN:

NGUYỄN HỮU THẮNG – 1712756

PHẠM NHƯ QUYỀN – 1712714

GIỚI THIỆU CHỦ ĐỀ

- IMDb (Internet Movie Database) là một trang web trực tuyến lưu trữ thông tin chi tiết nhất về các bộ phim cũng như các chủ đề về điện ảnh, truyền hình và video game
- Ở đây, đồ án này thực hiện về việc thu thập dữ liệu của 1000 bộ phim trên trang web IMDb và khám phá, khai thác dữ liệu này, tiền xử lý và xây dựng mô hình dự đoán điểm đánh giá của các bộ phim trong dữ liệu

CÁC MỤC TRÌNH BÀY:

- Thu thập dữ liệu từ trang web www.imdb.com bằng python
- Tiền xử lý dữ liệu
- Khám phá dữ liệu
- Tiếp tục tiền xử lý dữ liệu và Xây dựng mô hình

THU THẬP DỮ LIỆU

- Dữ liệu cần thu thập bao gồm 1000 bộ phim từ trang web với các thuộc tính :
  - tiêu đề phim
  - nội dung tóm tắt của phim
  - thời lượng chiếu phim
  - đạo diễn phim
  - các ngôi sao trong phim
  - thể loại phim
  - chứng chỉ của phim
  - điểm đánh giá phim
  - điểm đánh giá trung bình thang 100
  - điểm đánh giá
  - năm sản xuất phim
  - doanh thu của phim

THU THẬP DỮ LIỆU

1. **Soul** (2020)

PG | 100 min | Animation, Adventure, Comedy

8.1 Rate this 83 Metascore

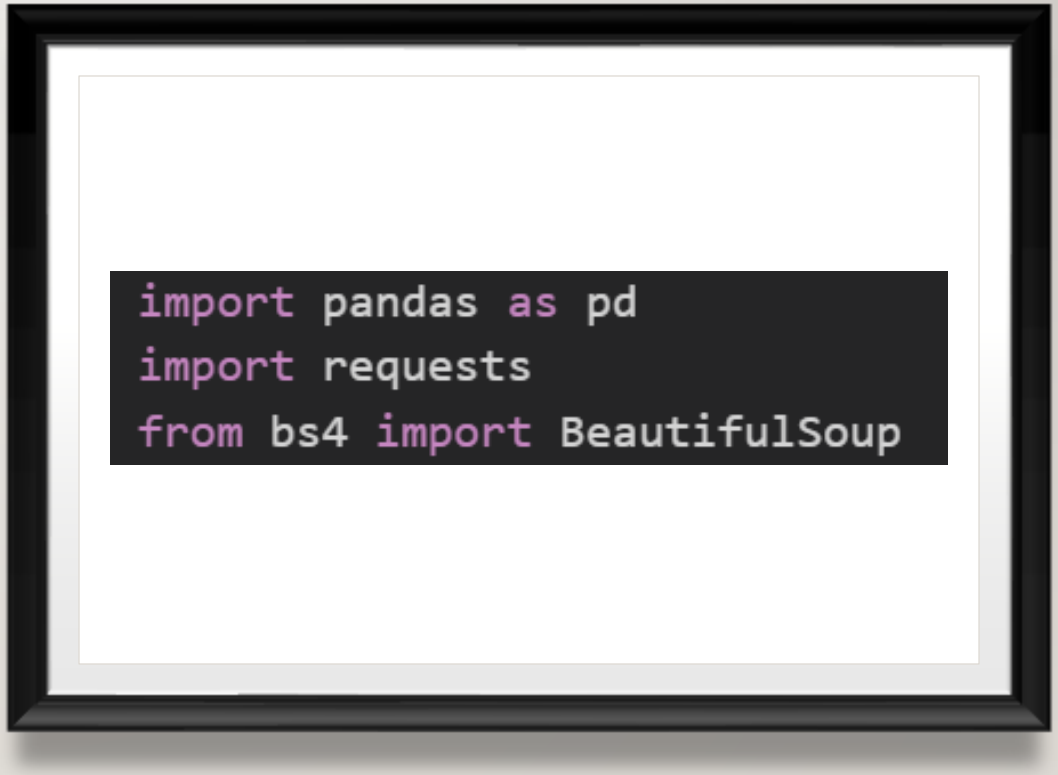
A musician who has lost his passion for music is transported out of his body and must find his way back with the help of an infant soul learning about herself.

Directors: Pete Docter, Kemp Powers | Stars: Jamie Foxx, Tina Fey, Graham Norton, Rachel House

Votes: 126,286

THU THẬP DỮ LIỆU

- Việc thu thập dữ liệu bằng ngôn ngữ Python được thực hiện trên notebook với việc sử dụng các thư viện hỗ trợ bao gồm pandas, bs4, requests
-



THU THẬP DỮ LIỆU

- Ta sử dụng request để lấy toàn bộ html của trang web như sau:

```
url = 'https://www.imdb.com/search/title/?groups=top_1000&count=100&start='+ str(page)+'&ref_=adv_nxt'
data_page = requests.get(url,headers={"Accept-Language": "en-US"})
```

THU THẬP DỮ LIỆU

- Sau đó khởi tạo các danh sách sau để lưu các thuộc tính của bộ phim thu thập được

```
titles = [] # List chứa các tiêu đề
certificates = [] # List chứa các chứng chỉ của phim
ratings = [] # List chứa điểm đánh giá của phim
genres = [] # List chứa các thể loại của phim
votes =[] # List chứa số lượng vote cho phim
metascores = [] # List chứa điểm đánh giá trung bình của phim từ 0 - 100
runtime_list = [] # List chứa thời lượng phim
directors_list = [] # List chứa các đạo diễn của phim
stars_list = [] # List chứa các ngôi sao của phim
grosses_list =[] # List chứa doanh thu của phim
introduction_list = [] # List chứa giới thiệu nội dung của phim
produce_year_list = [] # List chứa năm sản xuất của phim
```

THU THẬP DỮ LIỆU

```
soup = BeautifulSoup(data_page.text)
list_item_content = soup.findAll('div',class_ = 'lister-item-content')
print(len(list_item_content))
for movie_content in list_item_content:
    list_muted_text = movie_content.findAll('p',class_='text-muted')
    introduction_list.append(list_muted_text[1].text.replace("\n ",""))
    item_header = movie_content.find('h3',class_='lister-item-header')
    title = item_header.find('a')
    rating = movie_content.find('strong')
    certificate = movie_content.find('span',class_ = 'certificate')
    vote = movie_content.find('p',class_ ='sort-num_votes-visible')
    genre = movie_content.find('span',class_='genre')
    metascore = movie_content.find('span',class_='metascore favorable')
    runtime = movie_content.find('span',class_='runtime')
    p_content = movie_content.find('p',class_='')
    produce_year = movie_content.find('span',class_='lister-item-year')
```

- Sau đó ta sử dụng bs4 để lấy từng thuộc tính của từng bộ phim như sau

THU THẬP DỮ LIỆU

```
df = pd.DataFrame({'runtime':runtime_list,
                   'genres':genres,
                   'titles':titles,
                   'certificates':certificates,
                   'votes':votes,
                   'metascores':metascores,
                   'ratings':ratings,
                   'directors':directors_list,
                   'stars':stars_list,
                   "gross":grosses_list,
                  'introduction':introduction_list,
                  'produce_year':produce_year_list})
movie_df = movie_df.append(df)
```

- Cuối cùng ta lưu hết các thuộc tính vào một dataframe và viết vào file movie_IMDB.csv

TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Các bước khám phá:
  - Đọc dữ liệu từ file movie_IMDB.csv
  - Tiền xử lý dữ liệu
  - Phân tích dữ liệu
  - Khám phá dữ liệu và trả lời câu hỏi

TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Đọc dữ liệu từ file movie_IMDB.csv

ĐỌC DỮ LIỆU TỪ TỆP ĐÃ THU THẬP movie_IMDB.csv

```
df = pd.read_csv("movie_IMDB.csv")
```

```
df.head()
```

	runtime	genres	titles	certificates	votes	metascores	ratings	directors	stars	gross	introduction	produce_year
0	100	Animation,Adventure,Comedy	Soul	PG	127241	83.0	8.1	Pete Docter, Kemp Powers	Jamie Foxx, Tina Fey, Graham Norton, Rach...	NaN	A musician who has lost his passion for mus...	2020
1	117	Comedy,Drama	Another Round	Not Rated	26265	81.0	7.8	Thomas Vinterberg	Mads Mikkelsen, Thomas Bo Larsen, Magnus ...	NaN	Four friends, all high school teachers, tes...	2020
2	181	Action,Adventure,Drama	Avengers: Endgame	PG-13	800195	78.0	8.4	Anthony Russo, Joe Russo	Robert Downey Jr., Chris Evans, Mark Ruff...	858.37	After the devastating events of Avengers: I...	2019
3	113	Action,Comedy,Crime	The Gentlemen	R	230234	NaN	7.8	Guy Ritchie	Matthew McConaughey, Charlie Hunnam, Mich...	NaN	An American expat tries to sell off his hig...	2019
4	132	Comedy,Drama,Thriller	Parasite	R	541127	96.0	8.6	Bong Joon Ho	Song Kang-Ho, Lee Sun-kyun, Cho Yeo-jeong...	53.37	Greed and class discrimination threaten the...	2019

Ý nghĩa của các cột trong bộ dữ liệu

```
with open('description.txt', 'r') as f:
    print(f.read())
```

```
VARIABLE DESCRIPTIONS:
titles          The title of the movie
introduction    The inntroduction of the movie
genres          The genres of the movie
certificates    The license type of the movie
votes           the number of Votes
metascores      Metascore Based on critic reviews provided by Metacritic.com
directors       The name of directors
gross           The Gross of the movie
runtime         the duration of the movie(min)
ratings         A weighted average vote for user ratings
```

TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Ta tiếp tục tiền xử lý với cột produce_year
- Ở đây ta thấy produce_year có type là object không phù hợp với thuộc tính này nên ta chuyển thuộc tính về dạng int64
- Ta xóa các giá trị quý I, II, III, IV của produce_year và dung pd.to_numeric để chuyển produce_year về dạng số

```
df.dtypes
```

```
runtime          int64
genres          object
titles          object
certificates    object
votes            int64
metascores     float64
ratings        float64
directors       object
stars           object
gross          float64
introduction    object
produce_year    object
dtype: object
```

```
# Thay đổi xoá giá trị quí của produce_year và chuyển cột này sang kiểu numeric
df['produce_year'] = df['produce_year'].str.replace("I ","")
df['produce_year'] = df['produce_year'].str.replace("I","")
df['produce_year'] = df['produce_year'].str.replace("II ","")
df['produce_year'] = df['produce_year'].str.replace("II","")
df['produce_year'] = df['produce_year'].str.replace("III ","")
df['produce_year'] = df['produce_year'].str.replace("III","")
df['produce_year'] = df['produce_year'].str.replace("IV ","")
df['produce_year'] = df['produce_year'].str.replace("IV","")
df['produce_year']=pd.to_numeric(df['produce_year'], errors='coerce')
df['produce_year'].unique()
```

```
array([2020, 2019, 1972, 2001, 1994, 2013, 2014, 2015, 2018, 2008, 2010,
       2017, 1990, 2005, 2002, 2009, 1999, 2006, 2000, 2016, 1977, 2003,
       1997, 2011, 1995, 1985, 1965, 2007, 2012, 1998, 1988, 1974, 1981,
       1993, 1979, 1991, 1987, 1983, 2004, 1980, 1984, 1976, 1946, 1982,
       1975, 1941, 1986, 1971, 1996, 1957, 1989, 1966, 1992, 1958, 1962,
       1968, 1942, 1964, 1960, 1939, 1967, 1954, 1973, 1978, 1963, 1959,
       1952, 1940, 1961, 1969, 1950, 1931, 1956, 1927, 1938, 1937, 1951,
       1934, 1970, 1936, 1949, 1933, 1955, 1944, 1953, 1948, 1947, 1922,
       1930, 1921, 1943, 1932, 1920, 1935, 1925, 1928, 1926, 1945, 1924],
      dtype=int64)
```

TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Bây giờ các cột dữ liệu có hai loại type là categorical và numeric
- Với các cột categorical ta tìm các thông tin bao gồm tỉ lệ phần tử thiếu, số lượng giá trị trong mỗi cột và chúng phân bổ như thế nào
- Với các cột numeric ta tìm thông tin về tỉ lệ thiếu phần tử, giá trị nhỏ nhất, lớn nhất và các giá trị phân vị

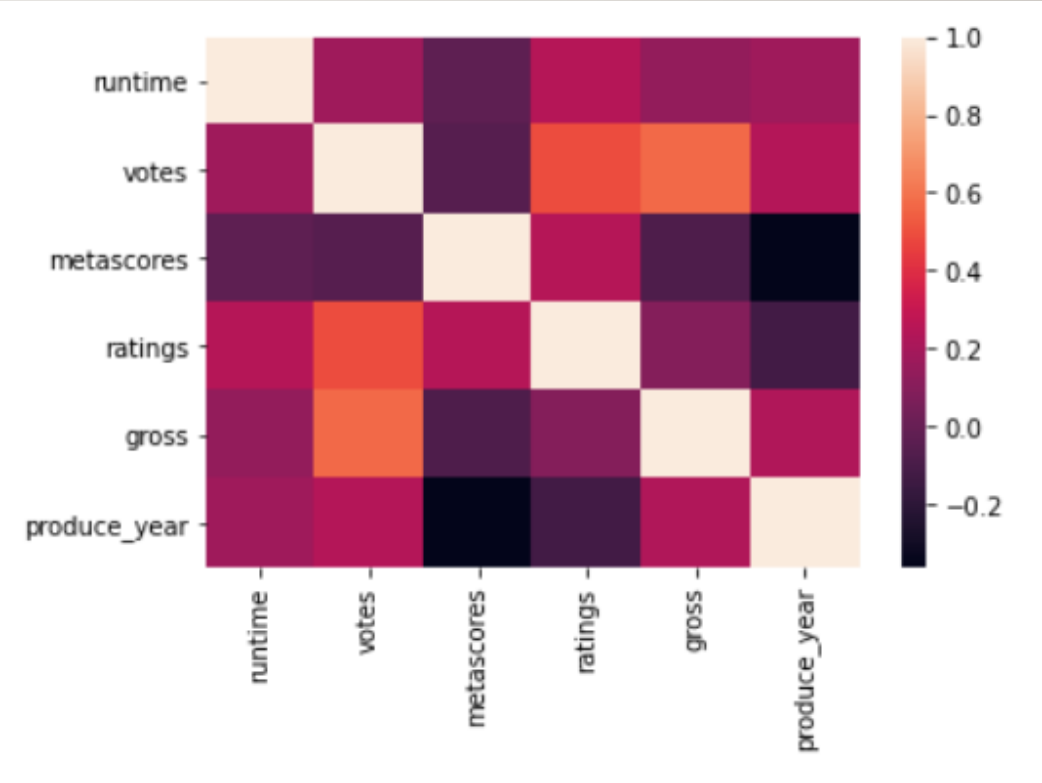
TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

	certificates	directors	genres	introduction	stars	titles
missing_ratio	1.1	0	0	0	0	0
num_values	14	642	21	1000	2962	997
value_ratios	{'R': 38.5, 'Not Rated': 17.2, 'PG-13': 17.0, 'PG': 14.4, 'G': 4.6, 'Passed': 3.9, 'Approved': 2...	{' Steven Spielberg': 1.3, ' Alfred Hitchcock': 1.3, ' Hayao Miyazaki': 1.0, ' Marti...	{'Drama': 28.4, 'Comedy': 9.2, 'Crime': 8.2, 'Adventure': 7.8, 'Action': 7.5, 'Thriller': 5.4, '...	{' Inspired by a true story, a comedy centered on a 27-year-old guy who learns of his cancer d...	{' Tom Hanks': 0.3, ' Robert De Niro': 0.3, ' Clint Eastwood': 0.2, ' Al Pacino'...	{'Scarface': 0.2, 'Drishyam': 0.2, 'The Girl with the Dragon Tattoo': 0.2, 'Black Hawk Down': 0....

	runtime	votes	metascores	ratings	gross	produce_year
missing_ratio	0.0	0.00	23.0	0.0	16.8000	0.0
min	45.0	25171.00	61.0	7.6	0.0000	1920.0
lower_quartile	102.0	56430.25	73.0	7.7	3.3250	1976.0
median	119.0	138174.00	80.0	7.9	24.2650	1998.5
upper_quartile	136.0	372540.75	88.0	8.1	82.5675	2009.0
max	321.0	2333702.00	100.0	9.3	936.6600	2020.0

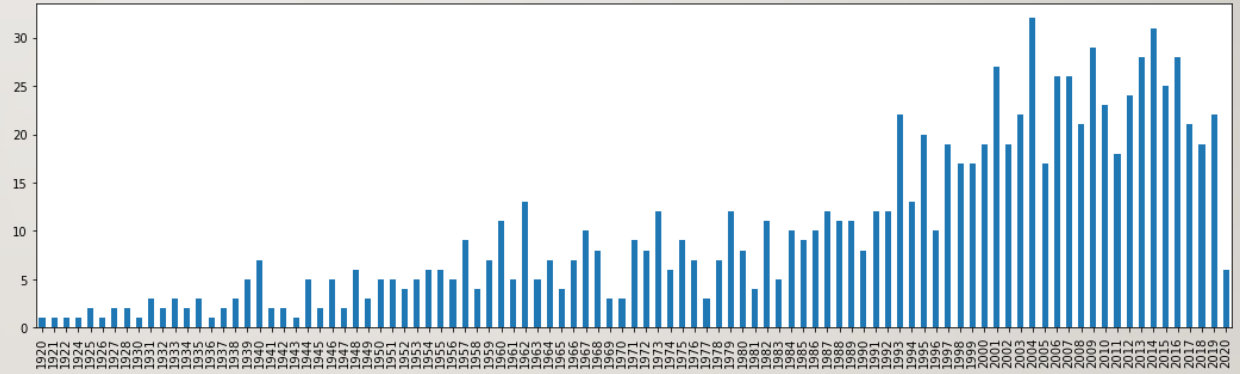
TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Tiếp theo ta tính toán và trực quan sự tương quan của các thuộc tính numeric
- Ta thấy ở đây mối tương quan giữa votes, ratings, Gross là rõ ràng nhất



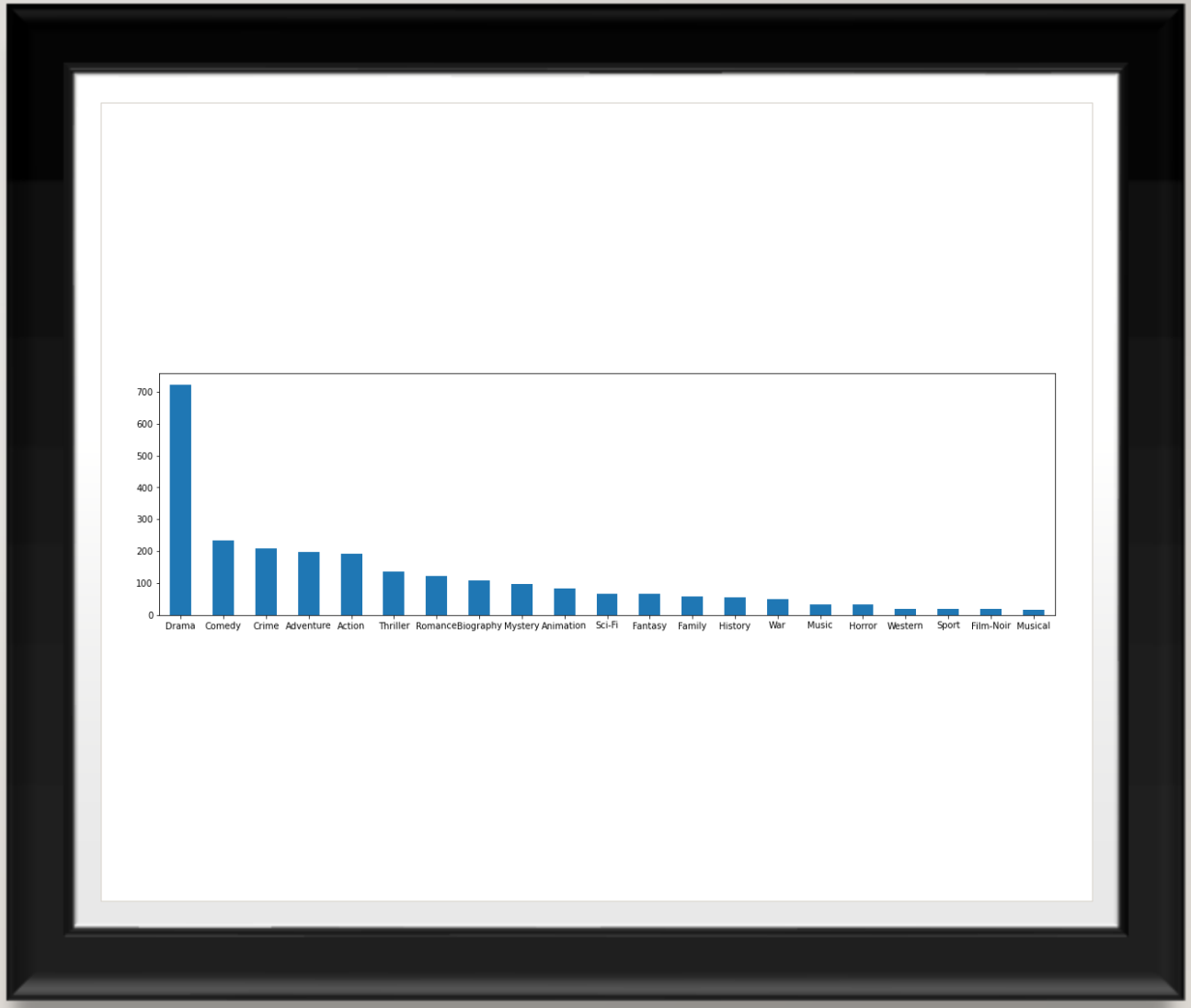
TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Ta tiếp tục trực quan số lượng phim mỗi năm trong 1000 phim trong dữ liệu
- Nhận xét : có thể thấy trong top 1000 phim thì các phim xuất hiện khoảng 40 năm trở lại đây xuất hiện nhiều nhất trong đó nhiều nhất là 20 năm đầu của thế kỷ XXI



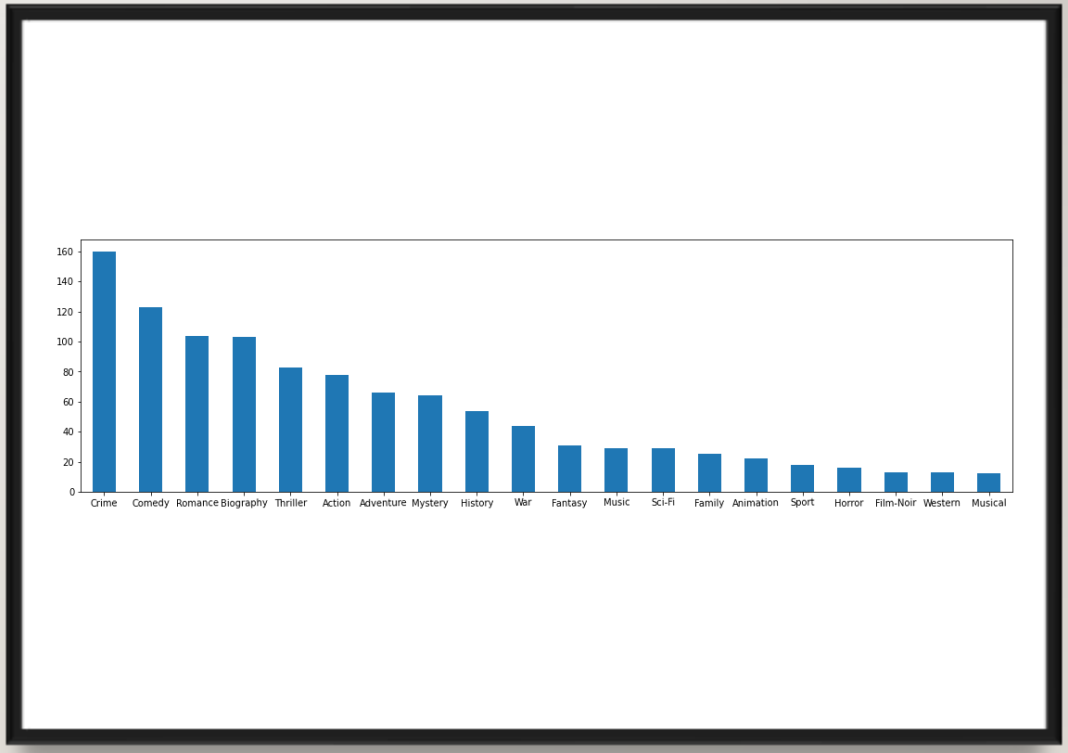
TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Biểu đồ đây thể hiện độ phổ biến của thể loại phim trong top 1000 phim IMDb
- Qua biểu đồ ta thấy Thể loại Drama xuất hiện hơn 70% trong 1000 bộ phim của IMDb cách biệt hẳn so với độ phổ biến của các thể loại khác, đồng thời thấy được các thể loại phim nào đang thịnh hành nhất trên IMDb



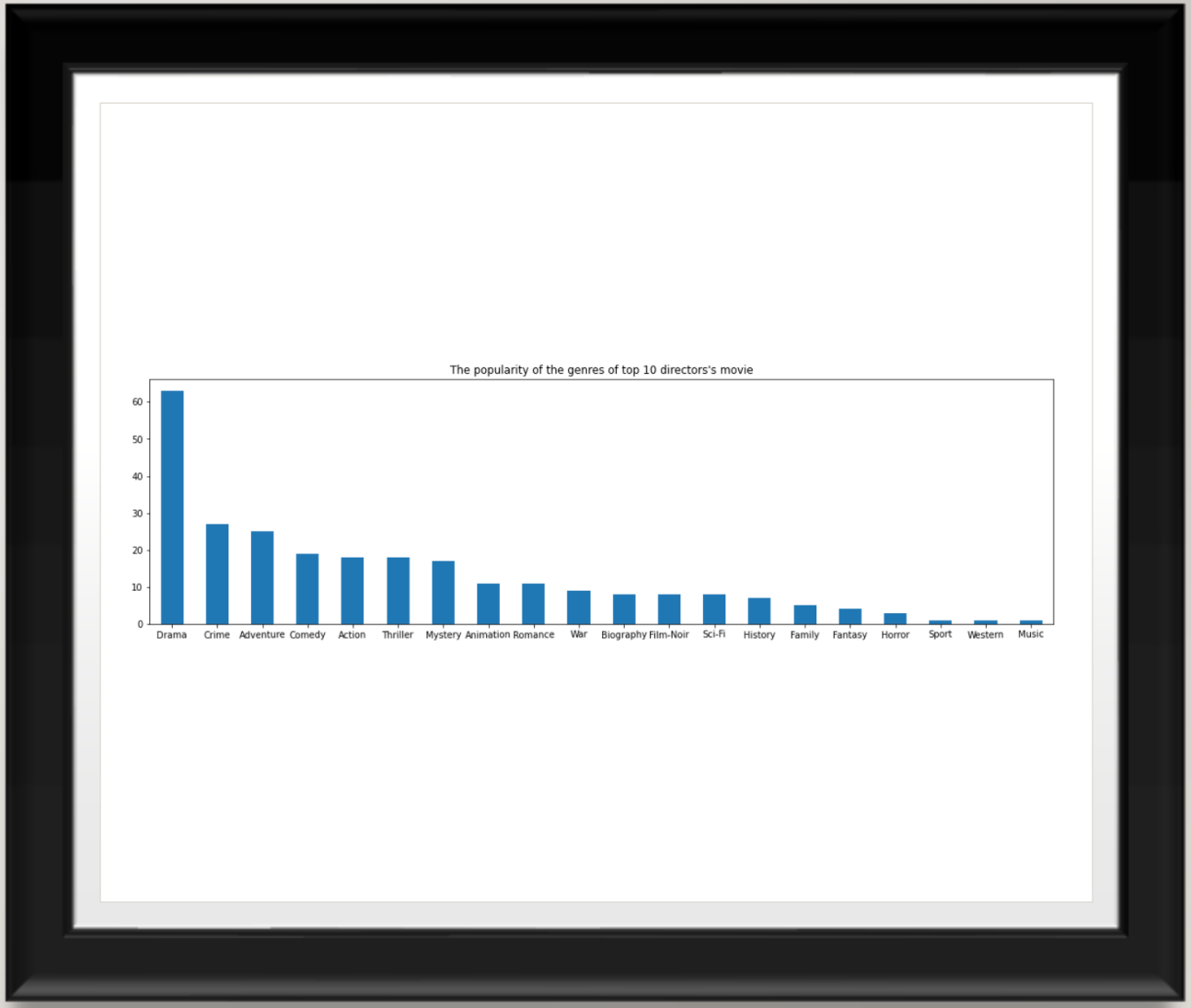
TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Biểu đồ thể hiện độ phổ biến của các thể loại phim đi cùng với thể loại Drama
- Nhận xét: ta thấy rõ ở đây thể loại phim Crime là thể loại đi cùng phổ biến nhất với thể loại Drama so với các thể loại khác(gần 160 phim trên) và thứ tự độ phổ biến của mỗi thể loại đi cùng với thể loại Drama khác nhiều so với độ phổ biến của chúng trên toàn bộ dữ liệu cho thấy độ phổ biến xuất hiện cùng của các thể loại trong một bộ phim



TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Biểu đồ thể hiện độ phổ biến của các thể loại phim mà top 10 đạo diễn trong dữ liệu
- Nhận xét: Các thể loại phim được khai thác của top 10 đạo diễn nhiều phim nhất được thể hiện ở đây giúp cho ta biết được thể loại phim nào đang được các đạo diễn khai thác và trở nên thịnh hành nhất bao gồm Drama, Advanture, Crime, Comedy, Mystery, ...



TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

- Trả lời câu hỏi : Các bộ phim nào đạt ratings cao nhất mỗi năm ?
- Các bước thực hiện:
  - Gom nhóm các năm sản xuất sử dụng groupby của thư viện pandas
  - Lấy index của giá trị ratings lớn nhất trong mỗi nhóm
  - Lấy index gọi về dataframe cũ để lấy kết quả cuối cùng

titles	certificates	votes	metascores	ratings	directors	stars	gross	introduction	produce_year
The Cabinet of Dr. Caligari	Not Rated	57212	NaN	8.1	[Robert Wiene]	Werner Krauss, Conrad Veidt, Friedrich Feher, Lil Dagover	NaN	Hypnotist Dr. Caligari uses a somnambulist, Cesare, to commit murders.	1920
The Kid	Passed	112713	NaN	8.3	[Charles Chaplin]	Charles Chaplin, Edna Purviance, Jackie Coogan, Carl Miller	5.45	The Tramp cares for an abandoned child, but events put that relationship in jeopardy.	1921
osferatu	Not Rated	88582	NaN	7.9	[F.W. Murnau]	Max Schreck, Alexander Granach, Gustav von Wangenheim, Greta Schröder	NaN	Vampire Count Orlok expresses interest in a new residence and real estate agent Hutter's wife.	1922
Sherlock Jr.	Passed	41700	NaN	8.2	[Buster Keaton]	Buster Keaton, Kathryn McGuire, Joe Keaton, Erwin Connelly	0.98	A film projectionist longs to be a detective, and puts his meagre skills to work when he is f...	1924
he Gold Rush	Passed	100617	NaN	8.2	[Charles Chaplin]	Charles Chaplin, Mack Swain, Tom Murray, Henry Bergman	5.45	A prospector goes to the Klondike in search of gold and finds it and more.	1925

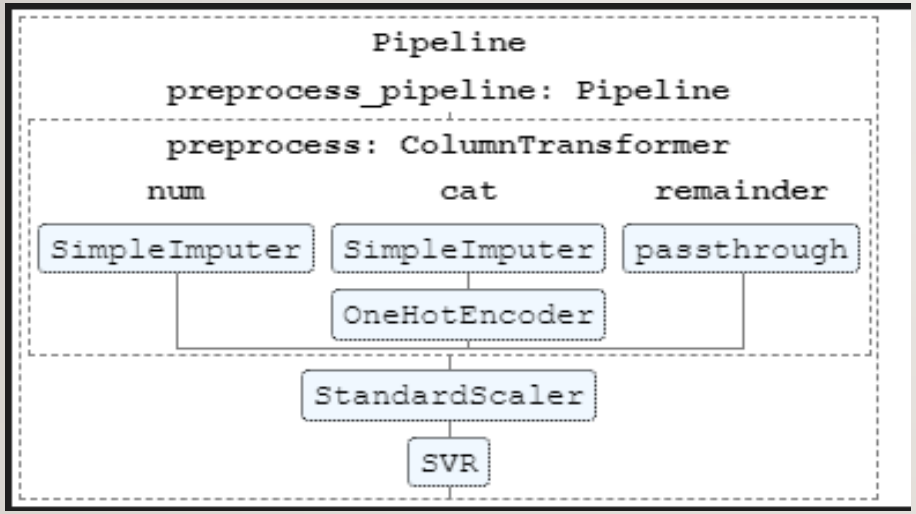
TIỀN XỬ LÝ VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

- Các bước thực hiện:
  - Tiền xử lý và tách dữ liệu
  - Xây dựng mô hình sử dụng thư viện sklearn

TIỀN XỬ LÝ VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

- Nhóm xây dựng process pipeline để tiền xử lý các cột có kiểu numeric
- Nhóm sử dụng mô hình SVM(Support vector machine) để hồi qui dự đoán kết quả và kiểm tra 1 số tham số ảnh hưởng tới độ chính xác của mô hình như kernel, C, gamma,…

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN




```
param_grid = {
    'SVR__kernel': ['poly','rbf','sigmoid','linear'],
    'SVR__C': [0.001,0.01,0.1,1,10,100],
    'SVR__gamma': ['scale', 'auto']
    }

search = GridSearchCV(full_pipeline, param_grid, cv=3)
search.fit(train_X_df, train_y_sr)
print("score :",search.score(train_X_df, train_y_sr))
print("Best parameter (CV score=%0.3f):" % search.best_score_)
print('best param: ',search.best_params_)
```

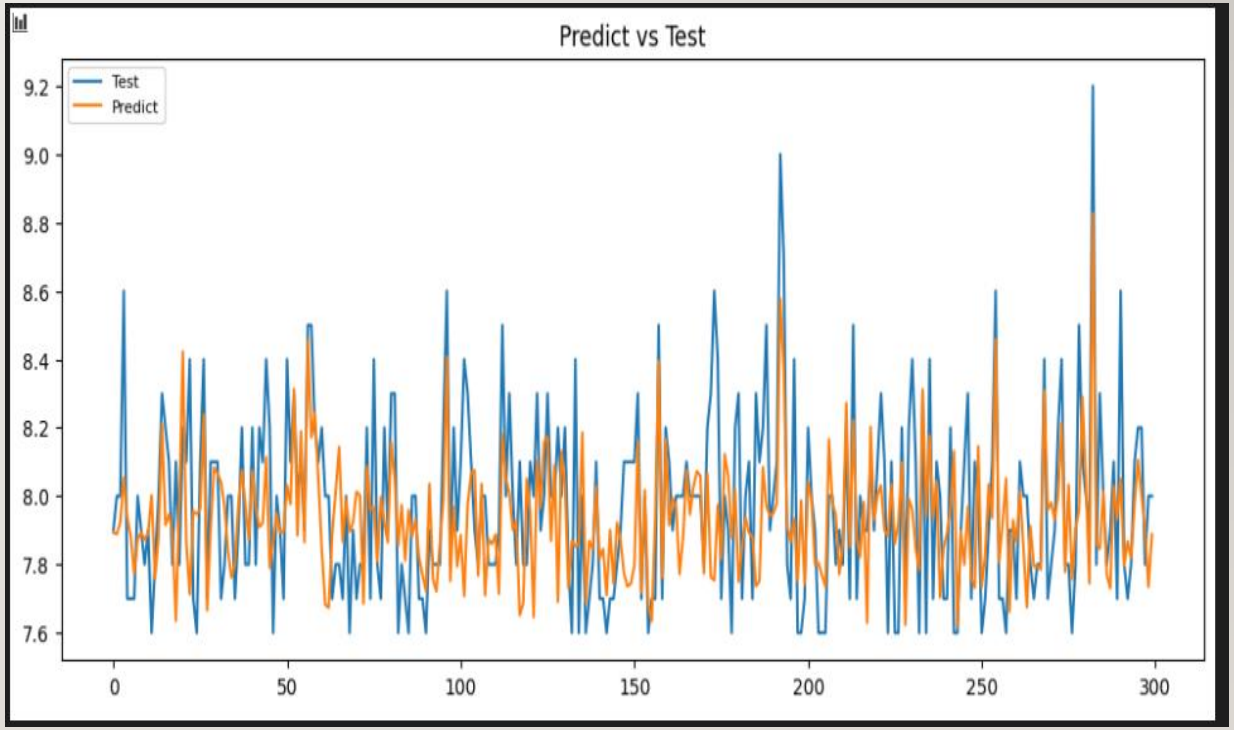
```
score : 0.4839579422461685
Best parameter (CV score=0.443):
best param:  {'SVR__C': 0.1, 'SVR__gamma': 'scale', 'SVR__kernel': 'linear'}
```

ẢNH HƯỞNG CỦA CÁC SIÊU THAM SỐ TỚI MÔ HÌNH

- Hàm kernel ảnh hưởng tới miền không gian có thể phân tách dữ liệu
- Siêu tham số C: là tham số xác định mức độ phạt đối với các lỗi
- Siêu tham số gamma là một tham số có thể được coi là 'độ lan truyền' của kernel và là vùng quyết định

KẾT QUẢ VÀ SO SÁNH VỚI BỘ TEST

- Kết quả dự đoán còn chưa chính xác với bộ test nhưng độ lệch giữa giá trị test và predict không nhiều
- Nhờ đó mô hình có thể dự đoán rating cho các bộ phim mới ra rạp 1 cách tương đối và có thể hỗ trợ tác vụ cho recommend system



KHÓ KHĂN KHI LÀM ĐỒ ÁN

- Khó khăn khi chọn 1 đề tài trả lời 1 câu hỏi có ý nghĩa và việc tự raw data khi tìm nguồn trên mạng.
- Khó khăn khi tiền xử lý dữ liệu mình đã tự raw. Nhóm vẫn chưa xử lý được các cột categorical data có nhiều giá trị như "genres","directors","stars" chuyển về dạng số và các cột bao gồm dữ liệu văn bản như "titles","introduction" cũng chưa có cách xử lý.
- Sẽ làm thêm nếu có thêm thời gian: Tìm kiếm cách xử lý các cột "titles","introduction","genres","directors","stars" chuyển đổi thành dạng có thể đưa vào mô hình học máy.

NHỮNG GÌ HỌC ĐƯỢC SAU KHI LÀM ĐỒ ÁN

- Cách tự raw data.
- Cách khám phá dữ liệu cho nhiều kiểu dữ liệu khác nhau.
- Xây dựng pipeline cho tiền xử lý.
- Xây dựng mô hình và kiểm tra mô hình thích hợp với nhiều giá trị tham số.

TÀI LIỆU THAM KHẢO

- https://www.imdb.com/
- https://en.wikipedia.org/wiki/IMDb