TRƯỜNG ĐẠI HỌC VIỆT NHẬT
ĐẠI HỌC QUỐC GIA HÀ NỘI
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO CUỐI KÌ MÔN KHOA HỌC DỮ LIỆU

ĐỀ TÀI:
DỰ ĐOÁN DOANH THU PHIM SỬ DỤNG MÔ HÌNH LIGHTGBM

Sinh viên thực hiện	**:**	**Lê Hải Nam**
Lớp	**:**	**BCSE K1**
Khóa	**:**	**Khóa 1**
Chuyên ngành	**:**	**Khoa học và Kỹ thuật Máy tính**
Mã sinh viên	**:**	**21110096**
Giảng viên hướng dẫn	**:**	**TS. Lê Chí Ngọc**

Hà Nội, 2023

Chương 1: Giới thiệu

I. Lý do chọn đề tài

1. Tầm quan trọng của dự đoán doanh thu phim

a. Vai trò của ngành công nghiệp điện ảnh

Ngành công nghiệp điện ảnh không chỉ là một phần quan trọng của văn hóa đại chúng mà còn đóng góp tích cực vào nền kinh tế toàn cầu. Sự phát triển của điện ảnh đã tạo nên những tác phẩm nghệ thuật mang tính tượng trưng, lan tỏa giá trị văn hóa và góp phần thay đổi cách nhìn nhận của xã hội.

b. Sự ảnh hưởng của phim đối với xã hội và văn hóa

Giải trí và thông điệp: Phim không chỉ là một phương tiện giải trí mà còn là cầu nối giữa các văn hóa, mang theo thông điệp sâu sắc về đa dạng, tình yêu, sự phản kháng, và những chủ đề xã hội quan trọng khác.

Tác động tới ý thức và quan điểm: Các tác phẩm điện ảnh có thể thay đổi cách nhìn của người xem về một vấn đề cụ thể, thúc đẩy sự suy ngẫm và thảo luận xã hội.

c. Tầm quan trọng của dự đoán doanh thu phim

Khả năng dự đoán doanh thu phim trước khi phát hành có thể cung cấp thông tin quan trọng cho các nhà sản xuất, nhà đầu tư và nhà phân phối về tiềm năng thành công của một tác phẩm điện ảnh. Nó có thể giúp định hình chiến lược marketing, phân phối và kế hoạch tài chính cho các bộ phim. Việc hiểu và dự đoán doanh thu phim không chỉ giúp tối ưu hóa lợi nhuận mà còn giúp ngành công nghiệp điện ảnh phát triển bền vững thông qua việc sản xuất những tác phẩm có chất lượng và sức hấp dẫn đối với khán giả.

II. Mục tiêu nghiên cứu

1. Mục tiêu của dự án

a. Mục tiêu chính

Xây dựng một mô hình dự đoán doanh thu phim sử dụng thuật toán LightGBM trong môi trường học máy

b. Mục tiêu chi tiết

Sử dụng những kiến thức đã học được trong học phần Khoa học dữ liệu, qua đó thực hiện:

- Khám phá và tiền xử lý dữ liệu: Tiến hành phân tích dữ liệu, xử lý các giá trị thiếu, và chuẩn bị dữ liệu để huấn luyện mô hình.
- Trực quan hóa và phân tích mối quan hệ giữa các biến: Tạo các biểu đồ để hiểu rõ hơn về mối quan hệ giữa các yếu tố khác nhau và doanh thu phim.
- Xây dựng mô hình LightGBM: Thực hiện việc xây dựng mô hình dự đoán doanh thu phim sử dụng thuật toán LightGBM.
- Tinh chỉnh và đánh giá mô hình: Tinh chỉnh các tham số của mô hình để cải thiện độ chính xác và hiệu suất. Đánh giá hiệu suất của mô hình trên dữ liệu huấn luyện và kiểm tra.
- Dự đoán doanh thu phim: Sử dụng mô hình đã huấn luyện để dự đoán doanh thu của các bộ phim chưa được biết trước và đánh giá độ chính xác của dự đoán.

c. Ý nghĩa dự án

Nghiên cứu các mối quan hệ, các yếu tố ảnh hưởng đến doanh thu của một bộ phim.

Cung cấp thông tin dự đoán chính xác về doanh thu phim từ đó giúp các công ty sản xuất có thể hỗ trợ các quyết định liên quan đến đầu tư, sản xuất, và phân phối phim trong tương lai.

Giúp người học có thêm được hiểu biết về việc sử dụng mô hình học máy LightGBM vào trong các dự án thực tế nói chung và ngành điện ảnh nói riêng.

2. Lý do sử dụng LightGBM

a. Tổng quan về LightGBM

LightGBM là một thuật toán dựa trên cây quyết định, nổi tiếng với hiệu suất cao và khả năng xử lý tốt trên dữ liệu lớn và phức tạp. Một trong những điểm mạnh nhất của LightGBM là sự linh hoạt và khả năng tùy chỉnh cao. Thuật toán này có khả năng xây dựng các cây quyết định theo chiến lược tối ưu, tạo ra mô hình có khả năng dự đoán cao mà không cần tài nguyên tính toán lớn.

Một điểm đáng chú ý của LightGBM là việc chia dữ liệu theo các lá cây thay vì chia dữ liệu theo mức độ (level-wise), giúp tập trung vào các điểm dữ liệu quan trọng hơn và tạo ra các mô hình có hiệu suất tốt hơn. Khả năng tối ưu hóa hiệu suất và hạn chế overfitting thông qua việc kiểm soát các siêu tham số cũng là một điểm mạnh của LightGBM.

Với tốc độ huấn luyện nhanh và khả năng xử lý tốt trên dữ liệu lớn, LightGBM thường được ưu tiên trong việc dự báo và phân loại. Nó có thể được áp dụng rộng rãi trong nhiều lĩnh vực như tài chính, y tế, quảng cáo và nhiều ngành công nghiệp khác.

b. Lý do sử dụng LightGBM

Một số ưu điểm của LightGBM có thể kể đến như:

- Khả năng xử lý dữ liệu phi tuyến tính: Mô hình tuyến tính như Linear Regression hoặc Logistic Regression: Các mô hình này giả định mối quan hệ tuyến tính giữa biến đầu vào và đầu ra. Trong trường hợp dữ liệu phim có các tương tác phi tuyến tính phức tạp, mô hình tuyến tính có thể không thể mô tả mối quan hệ này một cách chính xác.
- Đa dạng và phức tạp của dữ liệu: Support Vector Machines (SVM): Mặc dù SVM có thể xử lý tốt dữ liệu phi tuyến tính, nhưng nó có thể không hiệu quả trên các tập dữ liệu lớn và phức tạp như dữ liệu phim với nhiều biến đa dạng và tương tác phức tạp.
- Chi phí tính toán và hiệu suất: Random Forests hoặc các thuật toán Gradient Boosting khác: Các thuật toán này cũng có khả năng xử lý dữ liệu phi tuyến tính và phức tạp. Tuy nhiên, LightGBM thường có hiệu suất tốt hơn với thời gian huấn luyện nhanh hơn và tốn ít tài nguyên tính toán hơn.
- Tính linh hoạt và tùy chỉnh mô hình: Neural Networks (NN): Mạng nơ-ron có khả năng học mô hình phức tạp, nhưng đôi khi cần nhiều dữ liệu huấn luyện và thời gian tính toán lớn. LightGBM có thể cung cấp hiệu suất tốt mà không cần nhiều tài nguyên tính toán và dữ liệu huấn luyện lớn.

III. Mô tả bộ dữ liệu

Để nói về TMDB, The Movie Database (TMDB) là một cộng đồng được xây dựng chứa dữ liệu về các bộ phim điện ảnh hoặc TV-series. Đây là một trang web nổi tiêng được xây dựng bởi một cộng đồng lớn nên lượng dữ liệu được cập nhật chính xác theo thời gian thực. Dữ liệu được sử dụng ở đây được lấy từ cuộc thi TMDB Box Office Prediction trên Kaggle. Đây là bộ dữ liệu bao gồm các thông tin về 7398 bộ phim thu được trên TMDB từ năm 1921 đến thời điểm diễn ra cuộc thi là 2017.

Dữ liệu gồm 22 trường dữ liệu bao gồm:

- belongs_to_collection: Dạng JSON chứa thông tin bộ phim có thuộc về một loạt phim nào đấy hay không. VD: Avenger, Mission Impossible, Fast & Furious... là những loạt phim. Nếu rỗng thì bộ phim là những phim lẻ.
- budget: Dạng số thực chứa thông tin về ngân sách sử dụng cho bộ phim, đơn vị $.
- genres: Dạng JSON chứa thông tin về các thể loại của bộ phim.
- homepage: Dạng string chứa đường dẫn đến trang chủ của bộ phim (nếu có).
- imdb_id: Dạng string chứa id của bộ phim trên trang đánh giá phim IMDB.
- original_language: Dạng string chứa thông tin ngôn ngữ gốc bộ phim.
- original_title: Dạng string chứa tên gốc bộ phim.
- overview: Dạng string chứa thông tin tổng quan giới thiệu bộ phim.
- popularity: dạng số thực chứa chỉ số đặc biệt của tmdb chấm điểm cho độ nổi tiếng của một bộ phim được đánh giá theo các yếu tố: số lượng đánh giá trong ngày, số lượt xem trong ngày, số lượt người dùng đánh dấu phim là sở thích trong ngày, số lượng người dùng thêm phim vào danh sách xem sau trong ngày, ngày ra mắt, tổng lượt đánh giá và những điểm số trước đó.
- poster_path: Dạng string chứa đường dẫn đến poster của phim
- production_companies: Dạng JSON chứa thông tin và ID của công ty sản xuất bộ phim.
- production_countries: Dạng JSON chứa thông tin và viết tắt quốc gia sản xuất theo chuẩn iso_3166_1.
- release_date: Dạng thời gian có chuẩn MM/DD/YYYY.
- runtime: Dạng số thực chứa thời lượng bộ phim.
- spoken_languages: Dạng JSON chứa thông tin và viết tắt của ngôn ngữ nói bộ phim theo chuẩn iso_639_1.
- status: Dạng string chứa 2 giá trị Released và Unreleased kiểm tra bộ phim đã được ra mắt chưa.
- Tagline: Dạng string chứa khẩu hiệu ngắn gọn thể hiện được nội dung bộ phim.
- title: Dạng string chứa tiêu đề của bộ phim sau chuyển sang tiếng Anh.
- keywords: Dạng JSON chứa thông tin về những keywords liên quan đến phim.

- cast: Dạng JSON chứa thông tin dàn diễn viên tham gia bộ phim.
- crew: Dạng JSON chứa thông tin đoàn sản xuất bộ phim.
- revenue: Dạng số thực chứa thông tin về doanh thu bộ phim.

Chương 2: Tiền xử lý dữ liệu

I. Công nghệ sử dụng

Dự án sẽ sử dụng ngôn ngữ lập trình Python cùng một số package khác của ngôn ngữ: Pandas, Numpy, Sklearn, Matplotlib, Seaborn, LightGBM,...

II. Đọc dữ liệu

Sử dụng thư viện Pandas để đọc dữ liệu từ hai file của dữ liệu:

- train = pd.read_csv("train.csv", delimiter=',')
- test = pd.read_csv("test.csv", delimiter=',')

III. Khám phá dữ liệu

1. Kiểm tra năm dữ liệu đầu tiên của dữ liệu

Sử dụng hàm head kiểm tra năm dữ liệu đầu tiên

	id	belongs_to_collection	budget	genres	homepage	imdb_id	original_language	original_title	overview	popularity	...	release_date
0	1	[{'id': 313576, 'name': 'Hot Tub Time Machine ...	14000000	[{'id': 35, 'name': 'Comedy'}]	NaN	tt2637294	en	Hot Tub Time Machine 2	When Lou, who has become the "father of the In...	6.575393	...	2/20/15
1	2	[{'id': 107674, 'name': 'The Princess Diaries ...	40000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	tt0368933	en	The Princess Diaries 2: Royal Engagement	Mia Thermopolis is now a college graduate and ...	8.248895	...	8/6/04
2	3	NaN	3300000	[{'id': 18, 'name': 'Drama'}]	http://sonyclassics.com/whiplash/	tt2582802	en	Whiplash	Under the direction of a ruthless instructor, ...	64.299990	...	10/10/14
3	4	NaN	1200000	[{'id': 53, 'name': 'Thriller'}, {'id': 18, 'n...	http://kahaanithefilm.com/	tt1821480	hi	Kahaani	Vidya Bagchi (Vidya Balan) arrives in Kolkata ...	3.174936	...	3/9/12
4	5	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 53, 'nam...	NaN	tt1380152	ko	마린보이	Marine Boy is the story of a former national s...	1.148070	...	2/5/09

5 rows × 23 columns

Hình 1: Năm dữ liệu đầu tiên tập dữ liệu

Sau khi kiểm tra tôi thấy không có những sự bất thường trong tập dữ liệu này.

2. Kiểm tra số lượng dữ liệu

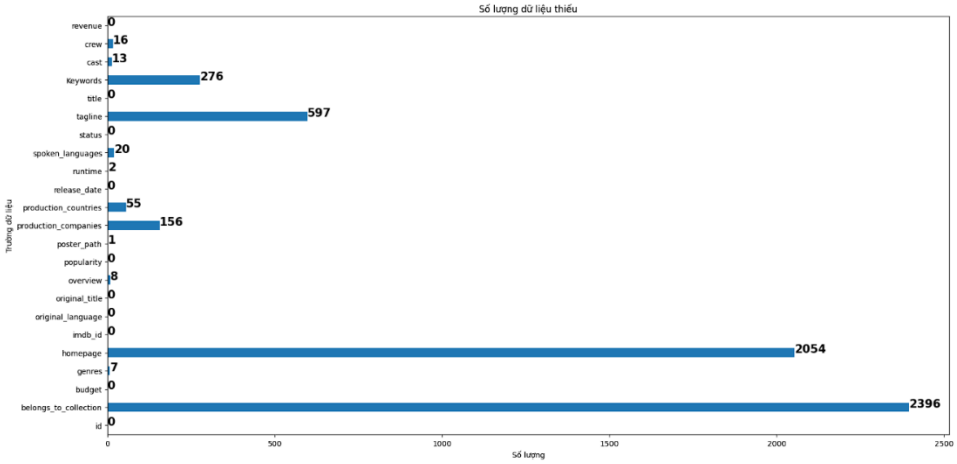
Sử dụng hàm shape để kiểm tra số lượng dữ liệu từ đó. Từ đó chúng ta thấy có 3000 dữ liệu ở tập train và 4398 dữ liệu ở tập test.

3. Kiểm tra dữ liệu thiếu

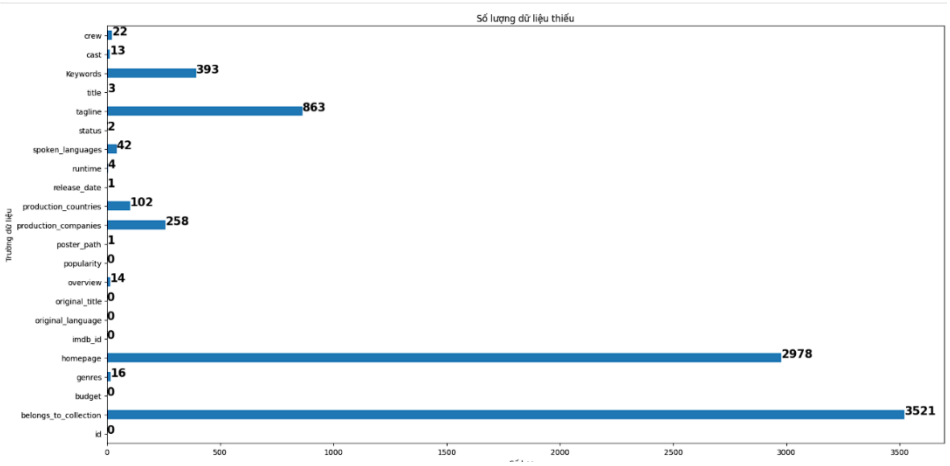
id	0
belongs_to_collection	2396
budget	0
genres	7
homepage	2054
imdb_id	0
original_language	0
original_title	0
overview	8
popularity	0
poster_path	1
production_companies	156
production_countries	55
release_date	0
runtime	2
spoken_languages	20
status	0
tagline	597
title	0
Keywords	276
cast	13
crew	16
revenue	0
dtype: int64	

id	0
belongs_to_collection	3521
budget	0
genres	16
homepage	2978
imdb_id	0
original_language	0
original_title	0
overview	14
popularity	0
poster_path	1
production_companies	258
production_countries	102
release_date	1
runtime	4
spoken_languages	42
status	2
tagline	863
title	3
Keywords	393
cast	13
crew	22
dtype: int64	

Hình 2: Các giá trị thiếu



Biểu đồ 1: Biểu đồ số lượng giá trị thiếu tập train



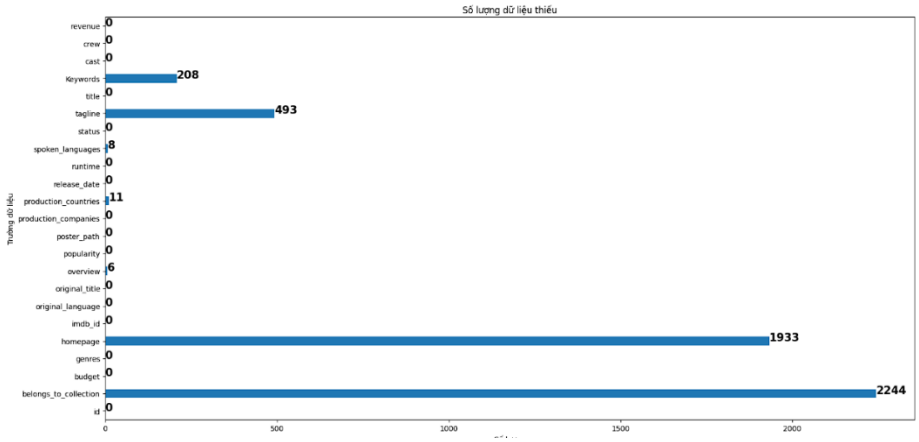
Biểu đồ 2: Biểu đồ số lượng giá trị thiếu tập test

Một số trường dữ liệu thiếu là do đặc tính của chúng ví dụ như: belongs_to_collection, homepage, overview, keywords, tagline. Một số trường dữ liệu khác mang tính quan trọng sẽ được xử lý trong phần sau.

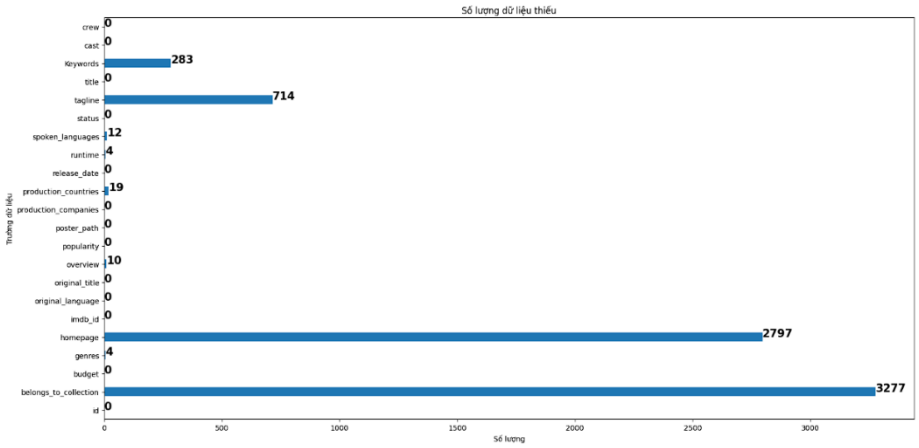
IV. Xử lý dữ liệu thiếu

1. Loại bỏ dữ liệu

Một số dữ liệu thiếu như ở cột crew, cast, production_companies đóng vai trò quan trọng trong việc phân tích, và khó để thu thập, ta sẽ tiến hành loại bỏ nó.



Biểu đồ 3: Biểu đồ số lượng giá trị thiếu tập train sau xử lý



Biểu đồ 4: Biểu đồ số lượng giá trị thiếu tập test sau xử lý

2. Bổ sung những dữ liệu thiếu

Có một số dữ liệu ở cột runtime, revenue hoặc budget bị lỗi dữ liệu. Ví dụ như:

Ở bộ phim có ID: 16, doanh thu của bộ phim bị lỗi và hiển thị là $198. Ở đây tôi sẽ thực hiện một số chỉnh sửa thông tin với dữ liệu được cập nhật trên Website.

```
train.loc[train['id'] == 16,'revenue'] = 192864          # Skinning
train.loc[train['id'] == 90,'budget'] = 30000000         # Sommersby
train.loc[train['id'] == 118,'budget'] = 60000000        # Wild Hogs
train.loc[train['id'] == 149,'budget'] = 18000000        # Beethoven
train.loc[train['id'] == 313,'revenue'] = 12000000       # The Cookout
train.loc[train['id'] == 451,'revenue'] = 12000000       # Chasing Liberty
train.loc[train['id'] == 464,'budget'] = 20000000        # Parenthood
train.loc[train['id'] == 470,'budget'] = 13000000        # The Karate Kid, Part II
train.loc[train['id'] == 513,'budget'] = 930000          # From Prada to Nada
train.loc[train['id'] == 797,'budget'] = 8000000         # Welcome to Dongmakgol
train.loc[train['id'] == 819,'budget'] = 90000000        # Alvin and the Chipmunks: The Road Chip
train.loc[train['id'] == 850,'budget'] = 90000000        # Modern Times
train.loc[train['id'] == 1112,'budget'] = 7500000        # An Officer and a Gentleman
train.loc[train['id'] == 1131,'budget'] = 4300000        # Smokey and the Bandit
train.loc[train['id'] == 1359,'budget'] = 10000000       # Stir Crazy
train.loc[train['id'] == 1542,'budget'] = 15800000       # All at once
train.loc[train['id'] == 1571,'budget'] = 4000000        # Lady and the Tramp
train.loc[train['id'] == 1714,'budget'] = 46000000       # The Recruit
train.loc[train['id'] == 1721,'budget'] = 17500000       # Cocoon
train.loc[train['id'] == 1865,'revenue'] = 25000000      # Scooby-Doo 2: Monsters Unleashed
train.loc[train['id'] == 2268,'budget'] = 17500000       # Madea Goes to Jail budget
train.loc[train['id'] == 2491,'revenue'] = 6800000       # Never Talk to Strangers
train.loc[train['id'] == 2602,'budget'] = 31000000       # Mr. Holland's Opus
train.loc[train['id'] == 2612,'budget'] = 15000000       # Field of Dreams
train.loc[train['id'] == 2696,'budget'] = 10000000       # Nurse 3-D
train.loc[train['id'] == 2801,'budget'] = 10000000       # Fracture
```

```
test.loc[test['id'] == 3889,'budget'] = 15000000       # Colossal
test.loc[test['id'] == 6733,'budget'] = 5000000        # The Big Sick
test.loc[test['id'] == 3197,'budget'] = 8000000        # High-Rise
test.loc[test['id'] == 6683,'budget'] = 50000000       # The Pink Panther 2
test.loc[test['id'] == 5704,'budget'] = 4300000        # French Connection II
test.loc[test['id'] == 6109,'budget'] = 281756         # Dogtooth
test.loc[test['id'] == 7242,'budget'] = 10000000       # Addams Family Values
test.loc[test['id'] == 7021,'budget'] = 17540562       #  Two Is a Family
test.loc[test['id'] == 5591,'budget'] = 4000000        # The Orphanage
test.loc[test['id'] == 4282,'budget'] = 20000000       # Big Top Pee-wee
```

```
# Update 'train' DataFrame
train.loc[train['id'] == 391, 'runtime'] = 86
train.loc[train['id'] == 592, 'runtime'] = 90
train.loc[train['id'] == 925, 'runtime'] = 95
train.loc[train['id'] == 978, 'runtime'] = 93
train.loc[train['id'] == 1256, 'runtime'] = 92
train.loc[train['id'] == 1542, 'runtime'] = 93
train.loc[train['id'] == 1875, 'runtime'] = 86
train.loc[train['id'] == 2151, 'runtime'] = 108
train.loc[train['id'] == 2499, 'runtime'] = 108
train.loc[train['id'] == 2646, 'runtime'] = 98
train.loc[train['id'] == 2786, 'runtime'] = 111
train.loc[train['id'] == 2866, 'runtime'] = 96

# Update 'test' DataFrame
test.loc[test['id'] == 4074, 'runtime'] = 103
test.loc[test['id'] == 4222, 'runtime'] = 93
test.loc[test['id'] == 4431, 'runtime'] = 100
test.loc[test['id'] == 5520, 'runtime'] = 86
test.loc[test['id'] == 5845, 'runtime'] = 83
test.loc[test['id'] == 5849, 'runtime'] = 140
test.loc[test['id'] == 6210, 'runtime'] = 104
test.loc[test['id'] == 6804, 'runtime'] = 145
test.loc[test['id'] == 7321, 'runtime'] = 87
```

Hình 3: Bổ sung dữ liệu của những dữ liệu lỗi (Nguồn: TMDB)

Ngoài ra còn có một số bộ phim có doanh thu bị sai so với thực tế theo tỉ lệ 1/1000000.

```
movie = train.id[train.budget > 1000][train.revenue < 100]

for k in movie :
    train.loc[train['id'] == k,'revenue'] =  train.loc[train['id'] == k,'revenue'] * 1000000
```

Hình 4: Scale những dữ liệu lỗi (Nguồn: TMDB)

V. Xử lý dữ liệu

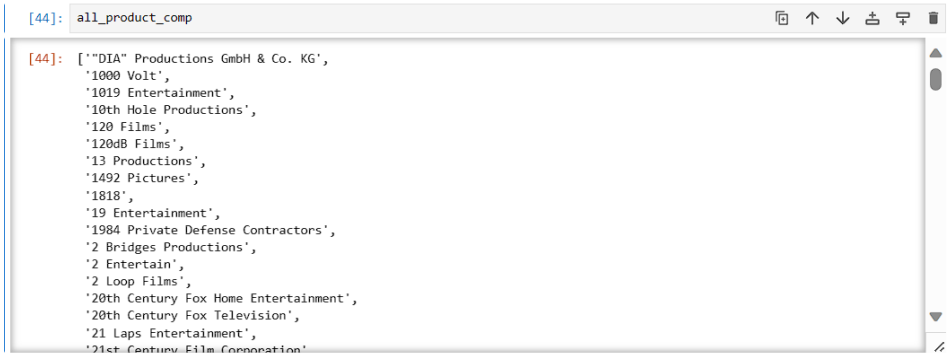
1. Định dạng lại kiểu dữ liệu string sang JSON/Dictionary

Sau khi đọc dữ liệu từ file csv vào, các dữ liệu dạng chữ được định dạng dưới kiểu string. Nhiệm vụ bây giờ sẽ là định dạng chuỗi string này thành kiểu dữ liệu JSON hay Dictionary để dễ dàng xử lý: Sử dụng hàm đối với mỗi cột, sử dụng hàm apply kết hợp với hàm lambda để thực hiện việc chuyển đổi: Nếu giá trị trong cột là NaN (giá trị thiếu), nó sẽ chuyển đổi

thành một từ điển trống ({}). Nếu không, nó sử dụng hàm literal_eval để đánh giá và chuyển đổi chuỗi đại diện của một từ điển thành đối tượng từ điển thực sự.

2. **Lấy danh sách các dữ liệu dạng chữ**

Dạng JSON gồm các cặp key-value, ở đây các dữ liệu dạng chữ có key là 'name' để lấy thông tin của trường dữ liệu đó. Ta sẽ sử dụng hàm for để đọc từng dữ liệu đó cho vào các list rỗng để lấy ra được danh sách những dữ liệu đã có. Điều này sẽ thuận tiện cho việc phân tích về những dữ liệu tiêu biểu sau này. Ví dụ sau khi thực hiện ta sẽ thu được dữ liệu về những công ty sản xuất phim có trong danh sách được sắp xếp theo thứ tự bảng chữ cái như sau:



Hình 5: Danh sách các công ty sản xuất trong dữ liệu

3. **Xử lý dữ liệu dạng dictionaries thành dạng lists chỉ chứa tên**

Tạo ra các cột tạm thời trong DataFrame để lưu trữ kết quả chuyển đổi từ dictionaries sang lists. Các cột tạm thời được đặt tên theo các tên cột gốc, nhưng được thêm phần "_temp" vào cuối tên.

Với mỗi cột tạm thời, hàm thực hiện việc chuyển đổi dictionaries thành lists chỉ chứa tên. Nếu giá trị trong cột dictionaries không rỗng ({}), hàm sẽ trích xuất tên từ các dictionaries và nối chúng lại thành một chuỗi ngăn cách bởi dấu phẩy. Kết quả này sẽ được gán vào các cột tạm thời tương ứng.

Cuối cùng, hàm trả về DataFrame đã được cập nhật giá trị sau khi chuyển đổi từ dictionaries sang lists cho các cột đã được chỉ định.

Một số dữ liệu sau khi được chuyển đổi:

belongs_to_collection_temp	genres_temp	spoken_languages_temp	production_companies_temp	production_countries_temp
Hot Tub Time Machine Collection	Comedy	English	Paramount Pictures,United Artists,Metro-Goldwy...	United States of America
The Princess Diaries Collection	Comedy,Drama,Family,Romance	English	Walt Disney Pictures	United States of America
	Drama	English	Bold Films,Blumhouse Productions,Right of Way ...	United States of America
	Horror,Thriller	English	Ghost House Pictures,North Box Productions	United States of America,Canada
The Muppet Collection	Action,Comedy,Music,Family,Adventure	English	Walt Disney Pictures,Jim Henson Productions,Ji...	United States of America
...	...	...	...	...
	Comedy,Romance	English	Warner Bros.,Morgan Creek Productions	United States of America

Hình 6: Dữ liệu sau khi được xử lý chuỗi

VI. Chuẩn hóa dữ liệu

1. Dữ liệu dạng chữ

a. Chọn ra những dữ liệu hàng đầu được xuất hiện nhiều nhất

Chọn ra những diễn viên, nhà sản xuất, công ty sản xuất, đất nước sản xuất, thể loại được yêu thích nhất để thêm vào cột dữ liệu. Những phim nào chứa các dữ liệu hàng đầu này sẽ được đánh số là 1, không sẽ kí hiệu là 0 (one-hot encode sử dụng hàm dummies).

b. Thêm các dữ liệu cần thiết

Thêm những cột dữ liệu cần thiết cho việc huấn luyện mô hình: Năm ra mắt, tháng ra mắt, các ngày ra mắt bộ phim. Ngoài ra thêm một cột là ‘budget_year_ratio’ đây là cột để lấy giá trị tỉ lệ giữa ngân sách với số năm bởi vì năm nào tỷ lệ lạm phát cũng sẽ xuất hiện nên việc chia với số năm tăng dần phần nào cũng phản ánh được con số thực của ngân sách. Sau cùng ta sẽ thu được một tập dữ liệu có dạng như sau:

dex	budget	popularity	runtime	title	revenue	Action	Adventure	Animation	Comedy	...	Sweden_name	South Africa_name	Hungary_name	United Arab Emirates_name
0	14000000	6.575393	93.0	Hot Tub Time Machine 2	12314651	0	0	0	1	...	0	0	0	0
1	40000000	8.248895	113.0	The Princess Diaries 2: Royal Engagement	95149435	0	0	0	1	...	0	0	0	0
2	3300000	64.299990	105.0	Whiplash	13092000	0	0	0	0	...	0	0	0	0
3	1200000	3.174936	122.0	Kahaani	16000000	0	0	0	0	...	0	0	0	0
5	8000000	0.743274	83.0	Pinocchio and the Emperor of the Night	3261638	0	1	1	0	...	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
992	1135654	3.878515	149.0	The Thief of Bagdad	1213880	1	1	0	0	...	0	0	0	0
993	60000000	14.092373	128.0	The Terminal	219417255	0	0	0	1	...	0	0	0	0
997	65000000	14.482345	120.0	The Long Kiss Goodnight	89456761	1	0	0	0	...	0	0	0	0
998	42000000	15.725542	90.0	Along Came Polly	171963386	0	0	0	1	...	0	0	0	0
999	35000000	10.512109	106.0	Abduction	82087155	1	0	0	0	...	0	0	0	0

× 205 columns

Hình 7: Bảng dữ liệu sau khi được chuẩn hóa các dữ liệu chữ

c. Xóa những trường dữ liệu không cần thiết

Xóa những cột dữ liệu không sử dụng, ở đây là cột ‘index’ và cột ‘title’ không phù hợp với việc đào tạo mô hình

2. Dữ liệu dạng số thực

Đối với những biến như ‘budget’ và ‘popularity’ ta sẽ thực hiện chuẩn hóa dữ liệu theo những phương pháp sau:

a. Chuyển đổi logarithm tự nhiên của dữ liệu

Việc sử dụng logarithm tự nhiên của giá trị dữ liệu thường được áp dụng trong xử lý dữ liệu khi dữ liệu ban đầu có phân phối không đồng đều hoặc có các giá trị lớn, dễ gây ảnh hưởng đáng kể đến các mô hình máy học. Dưới đây là một số lý do cụ thể:

- Giảm độ biến thiên: Logarithm tự nhiên thường được sử dụng để giảm độ biến thiên của dữ liệu. Điều này hữu ích khi dữ liệu ban đầu có độ lớn của các giá trị cực lớn so với các giá trị khác, làm cho chúng có ảnh hưởng lớn đến mô hình.
- Phân phối gần với phân phối chuẩn: Chuyển đổi logarithm tự nhiên có thể làm giảm độ lớn của các giá trị và khiến phân phối của dữ liệu gần hơn với phân phối chuẩn, giúp thuật toán máy học hoạt động hiệu quả hơn.

Sử dụng hàm ‘np.log(x + 1)’ để tính giá trí logarithm tự nhiên của dữ liệu sau đó cộng thêm 1 để tránh các giá trị x bằng 0.

b. Chuẩn hóa dữ liệu theo MinMaxScaler

MinMaxScaler là một phương pháp chuẩn hóa dữ liệu được sử dụng trong xử lý và chuẩn bị dữ liệu cho các mô hình máy học. Phương pháp này thường được sử dụng để đưa các giá trị của các biến về cùng một phạm vi, thường là từ 0 đến 1.

Công thức được tính như sau:

$$X_{\text{std}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Việc sử dụng MinMaxScaler giúp:

- Giữ nguyên tỷ lệ tương đối giữa các giá trị: Phương pháp này giúp giữ nguyên tỷ lệ tương đối giữa các giá trị, giúp mô hình dễ dàng học hơn và cải thiện hiệu suất của mô hình trong một số trường hợp.
- Giảm ảnh hưởng của giá trị ngoại lai: Khi chuẩn hóa dữ liệu về cùng một phạm vi, MinMaxScaler có thể giúp giảm ảnh hưởng của các giá trị ngoại lai đối với mô hình.
- Cải thiện tốc độ hội tụ của mô hình: Việc chuẩn hóa dữ liệu có thể giúp mô hình hội tụ nhanh hơn khi sử dụng các thuật toán dựa trên gradient descent.

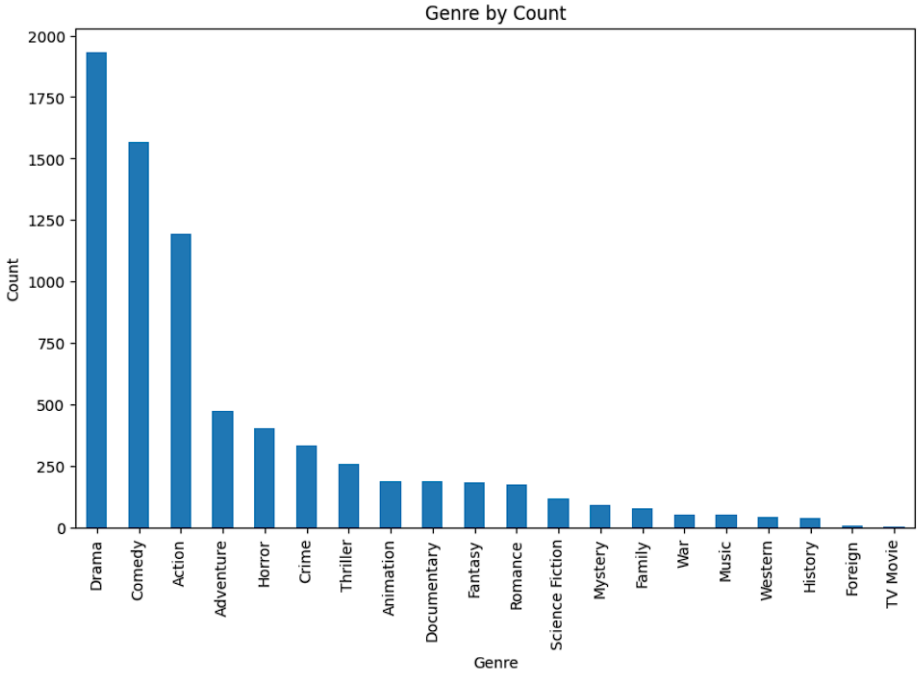
Chương 3: Trực quan hóa dữ liệu

I. Phân tích mối quan hệ giữa các biến

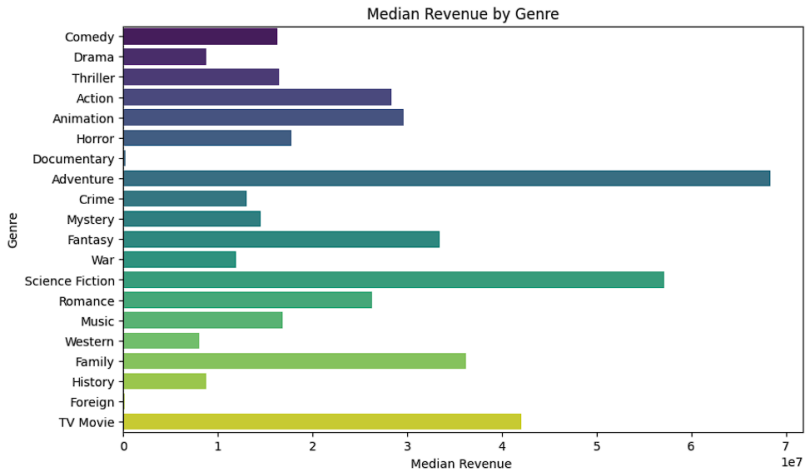
1. Đối với các biến chuỗi

a. Thể loại

Tìm ra những thể loại có nhiều bộ phim nhất



Biểu đồ 5: Danh sách số lượng phim theo thể loại

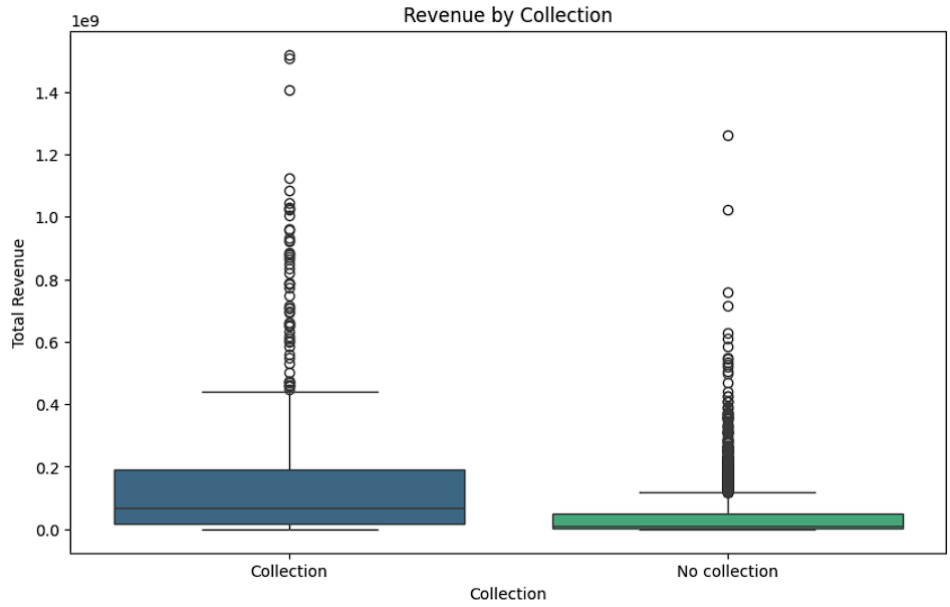


Biểu đồ 6: Biểu đồ mối quan hệ giữa một số thể loại phim với doanh thu

Qua hai biểu đồ trên chúng ta thấy, số lượng phim chính kịch được sản xuất nhiều nhất nhưng doanh thu của nó không hề cao. Tuy nhiên một số

phim thuộc thể loại phiêu lưu và khoa học viễn tượng thường có doanh thu cao hơn so với các phim khác. Điều này được lý giải bơi những phim thuộc thể loại này mang tính giải trí có được lượng khán giả cao hơn. Ngược lại thể loại phim tài liệu lại khá kén người xem.

b. Loạt phim



Biểu đồ 7: Biểu đồ so sánh doanh thu giữa phim thuộc loạt phim và những phim lẻ

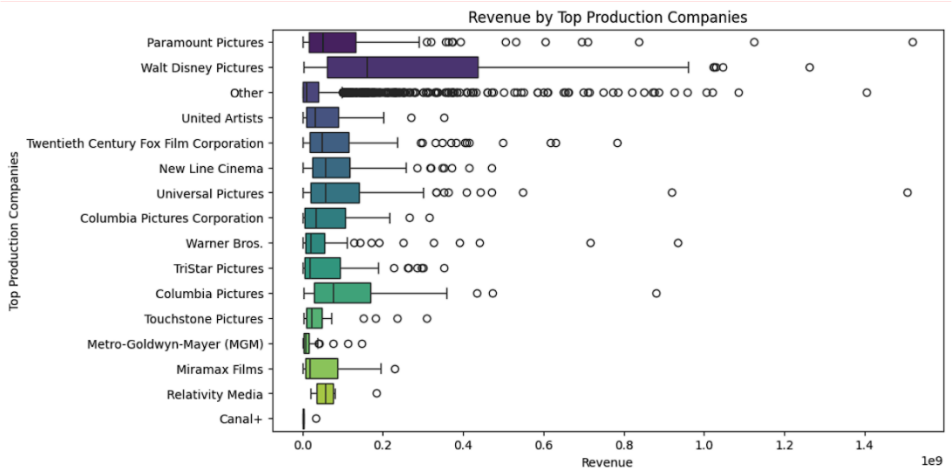
Dữ liệu cho thấy rằng doanh thu của phim collection movie thường cao hơn đáng kể so với doanh thu của phim no collection movie. Điều này có thể là do một số yếu tố, bao gồm:

- Sự quan tâm của người hâm mộ. Phim collection movie thường dựa trên các thương hiệu nổi tiếng hoặc các loạt phim ăn khách, vì vậy chúng thường thu hút nhiều người hâm mộ.
- Độ hấp dẫn của nội dung. Phim collection movie thường được đầu tư kỹ lưỡng hơn về mặt nội dung và hình ảnh, vì vậy chúng thường mang lại trải nghiệm xem hấp dẫn hơn.
- Chiến lược tiếp thị và phân phối. Phim collection movie thường được tiếp thị và phân phối rộng rãi hơn, vì vậy chúng có thể tiếp cận được với nhiều khán giả hơn.

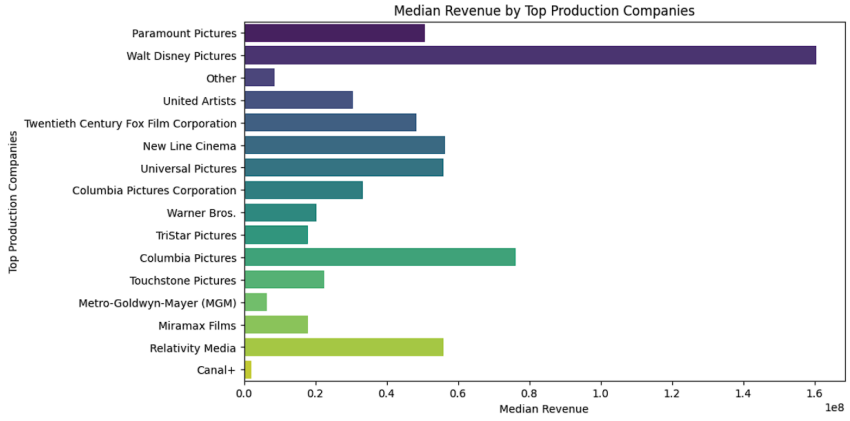
Tuy nhiên, cũng có một số trường hợp phim no collection movie có doanh thu cao hơn phim collection movie. Điều này có thể là do một số yếu tố, bao gồm:

- Sự nổi tiếng của đạo diễn hoặc diễn viên. Nếu phim có sự tham gia của đạo diễn hoặc diễn viên nổi tiếng, nó có thể thu hút nhiều khán giả dù không có thương hiệu hay nội dung hấp dẫn.
- Chiến lược tiếp thị sáng tạo. Nếu phim có chiến lược tiếp thị sáng tạo, nó có thể thu hút sự chú ý của khán giả và giúp tăng doanh thu. Xu hướng thị trường. Nếu thị trường có xu hướng ưa chuộng các thể loại phim khác nhau, phim no collection movie có thể được hưởng lợi từ xu hướng này.

c. Công ty sản xuất



Biểu đồ 8: Biểu đồ so sánh doanh thu của một số công ty lớn

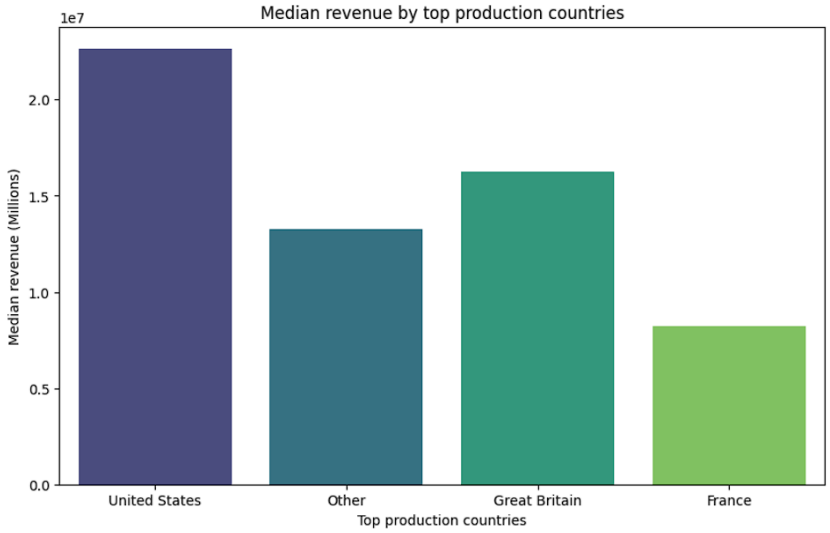


Biểu đồ 9: Biểu đồ so sánh giá trị trung vị doanh thu một số công ty lớn

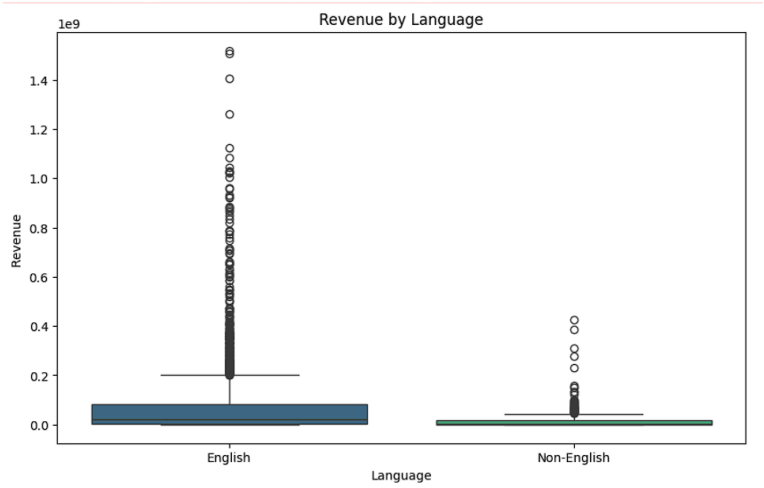
Nhìn vào biểu đồ chúng ta thấy số lượng bộ phim đến từ những công ty lớn có độ nhất diện cao thường có doanh thu trung bình cao hơn những công ty sản xuất nhỏ lẻ khác. Điều này được lí giải do các công ty lớn

thường có ngân sách và truyền thông mạnh hơn so với các công ty khác từ đó giúp doanh thu phim của họ cao hơn.

d. Quốc gia sản xuất và ngôn ngữ



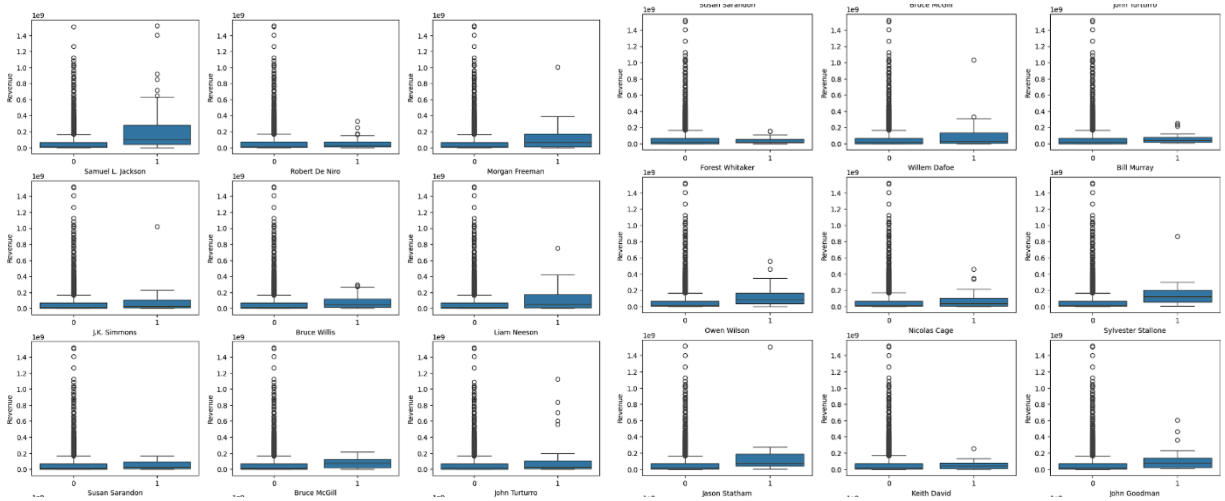
Biểu đồ 10: Biểu đồ so sánh doanh thu của một số quốc gia sản xuất lớn



Biểu đồ 11: Biểu đồ so sánh doanh thu của những phim sử dụng tiếng Anh và những phim khác

Dữ liệu có thể cho thấy sự đa dạng về ngôn ngữ trong ngành công nghiệp điện ảnh. Điều này phản ánh sự phong phú và đa dạng của văn hóa và ngôn ngữ trên toàn thế giới. Nếu tiếng Anh là ngôn ngữ gốc phổ biến nhất, điều này có thể phản ánh tầm ảnh hưởng của Hollywood và ngành công nghiệp điện ảnh Mỹ trên toàn cầu.

e. Diễn viên

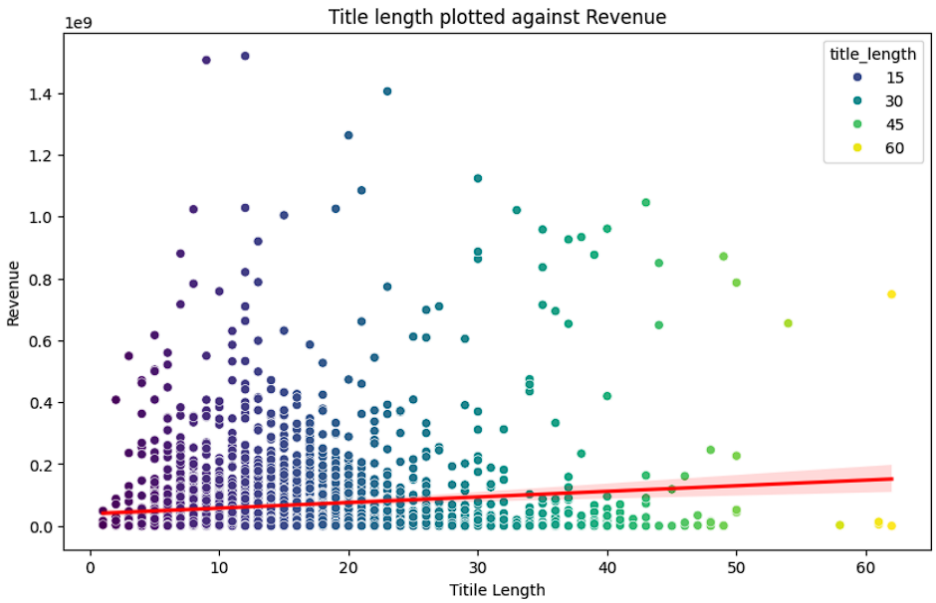


Biểu đồ 12: So sánh doanh thu phim của những phim có diễn viên hàng đầu tham gia với những phim khác

Đây là biểu đồ hộp thể hiện được doanh thu của những viên hàng đầu với những diễn viên còn lại. Qua đó chúng ta có thể thấy những phim có được sự tham gia của những diễn viên nổi tiếng sẽ có doanh thu cao hơn so với những phim khác.

f. Tiêu đề

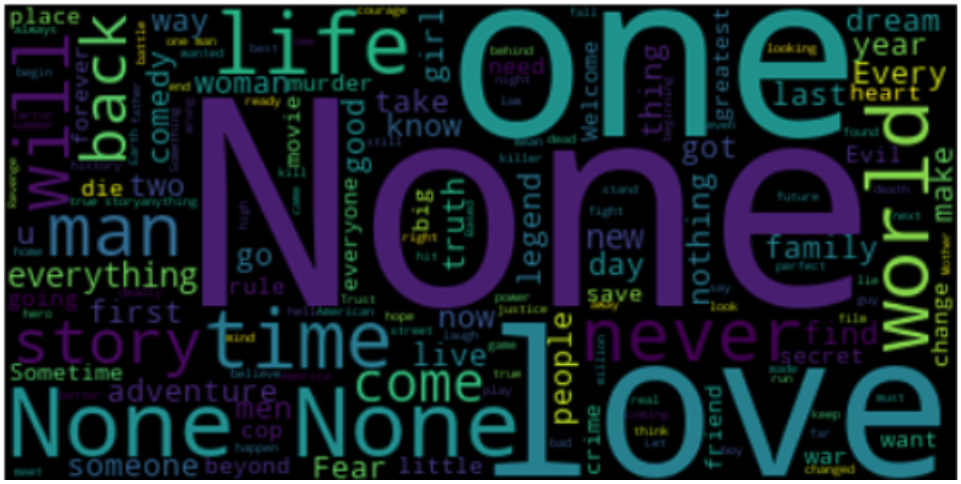
Hình 8: Những từ thường được chọn làm tiêu đề bộ phim



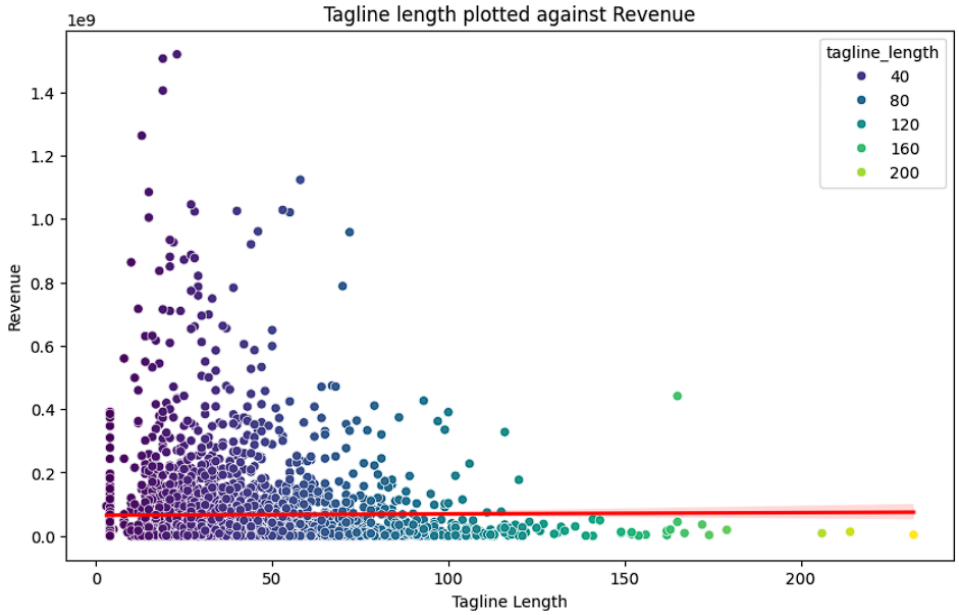
Biểu đồ 13: Biểu đồ mô tả mối quan hệ giữa độ dài tiêu đề phim với doanh thu

Từ biểu đồ chúng ta thấy được, một số từ thường được sử dụng để đặt làm tiêu đề của bộ phim như “Last”, “Man”, “Day”,... Một số từ này có thể sẽ gây thu hút hơn đối với khán giả. Ngoài ra qua biểu đồ tôi có thể thấy những phim có độ dài tiêu đề dài thường có doanh thu trung bình sẽ cao hơn.

g. Tagline



Hình 9: Những từ thường được chọn làm tagline bộ phim



Biểu đồ 14: Biểu đồ mô tả mối quan hệ giữa độ dài tagline phim với doanh thu

Tương tự, ta có thể thấy được một số từ thường được sử dụng làm tagline quảng bá phim, tuy nhiên mối quan hệ giữa độ dài tagline và doanh thu không được rõ ràng.

2. Đối với các biến số thực

a. Ngân sách

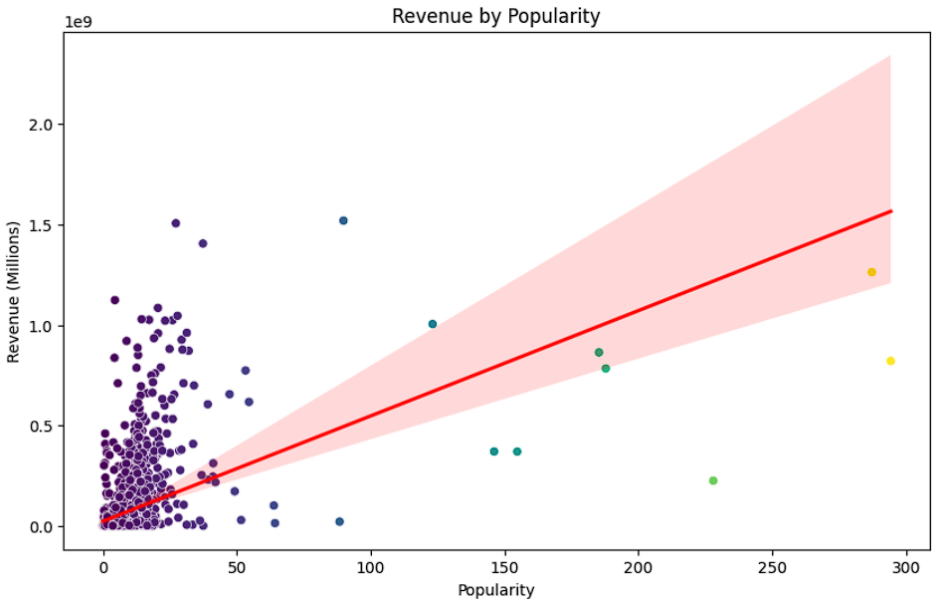
Biểu đồ 14: Biểu đồ mô tả mối quan hệ giữa ngân sách với doanh thu

Mối quan hệ: Đường hồi quy tuyến tính cho thấy mối quan hệ tích cực; khi ngân sách tăng lên, doanh thu cũng có xu hướng tăng.

Phân bố ngân sách: Hầu hết các điểm dữ liệu (màu tím) tập trung ở phía dưới cả hai trục, cho thấy nhiều bộ phim có ngân sách và doanh thu thấp.

Phim có ngân sách cao: Có ít điểm dữ liệu màu xanh lá cây và vàng, cho thấy có ít phim với ngân sách cao hơn. => Nhìn chung, biểu đồ cung cấp cái nhìn tổng quan về mối quan hệ giữa ngân sách sản xuất và doanh thu phòng vé, với xu hướng phim có ngân sách cao hơn có thể tạo ra doanh thu cao hơn. Điều này có thể hữu ích cho việc đánh giá chiến lược đầu tư và tiềm năng thu hồi vốn cho các dự án phim.

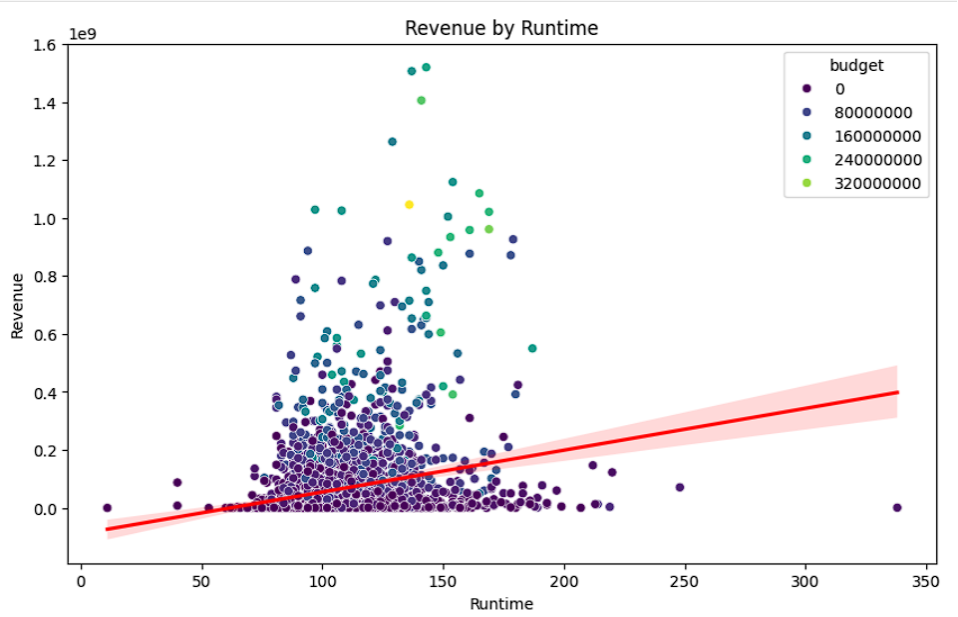
b. Popularity



Biểu đồ 15: Biểu đồ mô tả mối quan hệ giữa độ phổ biến phim với doanh thu

Popularity là biến đặc biệt của bộ dữ liệu này để thể hiện cho mức độ quan tâm phổ biến của khán giả đối với một bộ phim. Nhìn vào đường hồi quy, độ nổi tiếng phim có mối quan hệ tích cực, tỉ lệ thuận với nhau, những phim có chỉ số popularity cao thường đạt doanh thu cao.

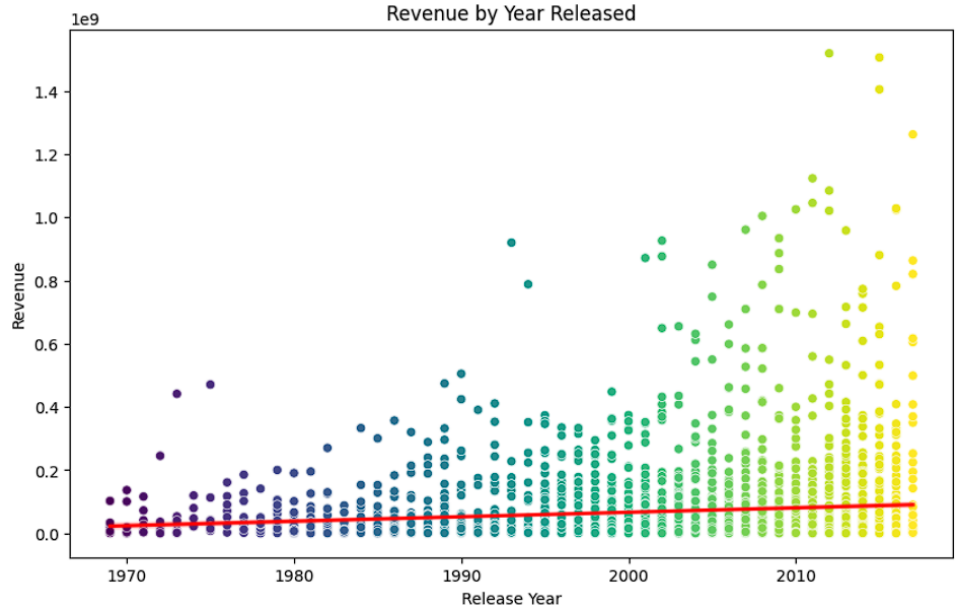
c. Runtime



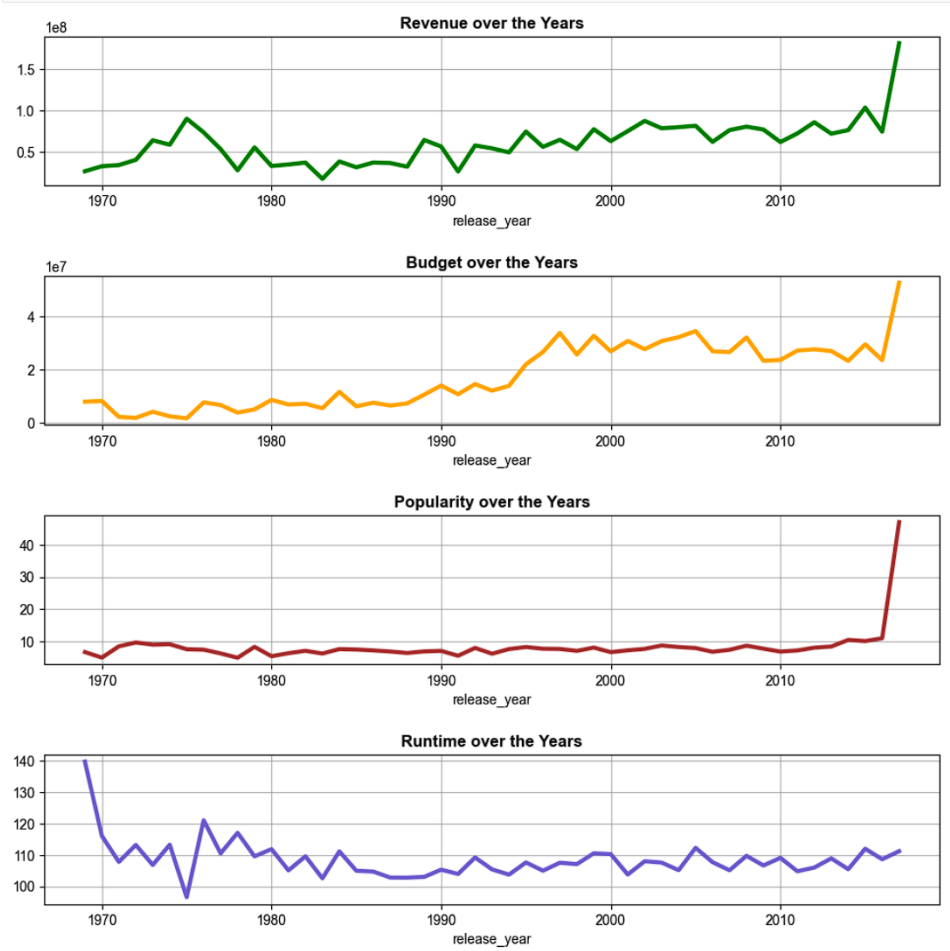
Biểu đồ 16: Biểu đồ mô tả mối quan hệ giữa thời gian bộ phim phim với doanh thu

Không phải những bộ phim có thời gian dài là có thể đạt được doanh thu cao, những bộ phim có khoảng thời gian trung bình từ hai đến ba tiếng sẽ có doanh thu cao hơn so với khoảng thời gian khác.

d. Năm ra mắt



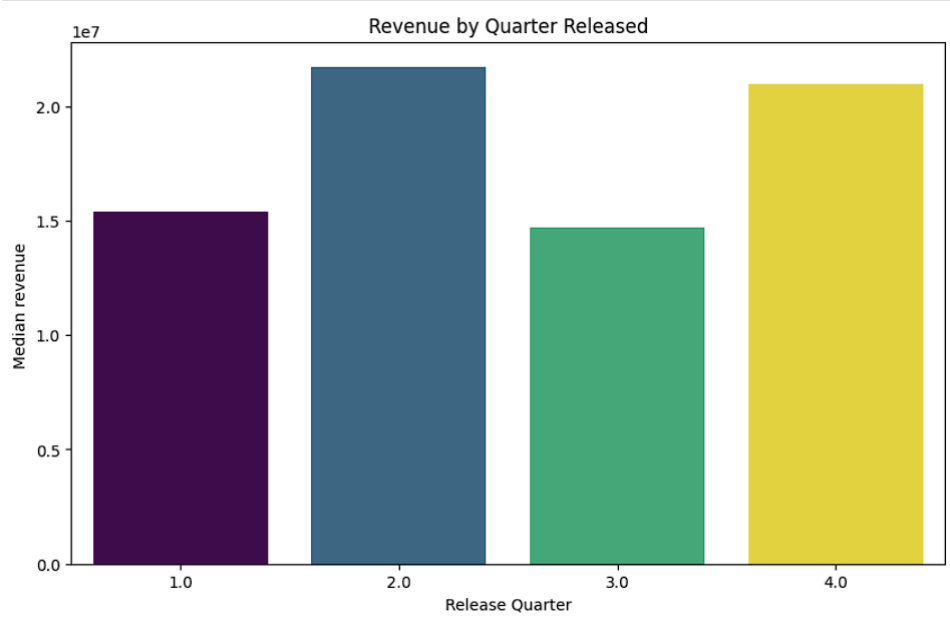
Biểu đồ 17: Biểu đồ mô tả mối quan hệ giữa năm ra mắt phim với doanh thu



Biểu đồ 18: Biểu đồ mô tả sự thay đổi của phim theo năm

Doanh thu phim đang ngày càng được tăng trưởng theo thời gian, những phim ra mắt sau có xu hướng có doanh thu cao hơn so với những phim thời kỳ trước. Điều này có thể xuất phát do tỉ giá, tuy nhiên cũng phản ánh được rằng mối quan tâm của mọi người đến điện ảnh đang ngày một tăng cao theo thời gian.

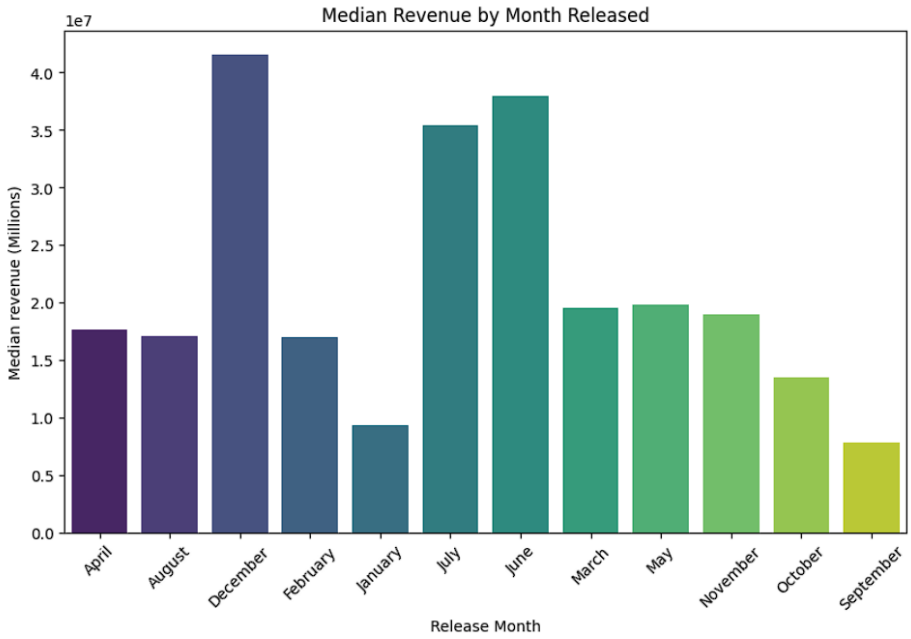
e. Quý ra mắt



Biểu đồ 19: Biểu đồ so sánh doanh thu của các quý ra mắt phim

Có sự chênh lệch doanh thu vào thời gian ra mắt tại các quý hàng năm. Mỹ là thị trường lớn nhất của phim chiếu rạp mà quý 2 với quý 4 là thời gian có dịp nghỉ lễ lớn của Mỹ vì vậy người dân sẽ có xu hướng đi xem phim nhiều hơn vào dịp này.

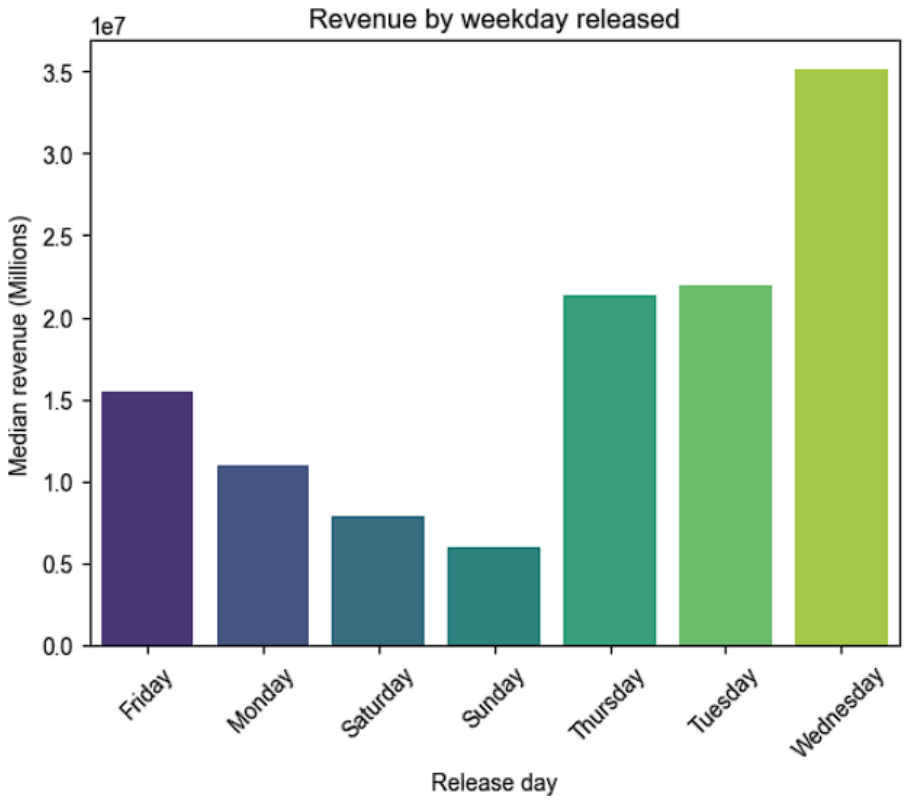
f. Tháng ra mắt



Biểu đồ 20: Biểu đồ so sánh doanh thu của các tháng ra mắt phim

Tháng 6, 7, thời gian nghỉ hè của sinh viên, cùng với tháng 12 là thời gian của những dịp lễ lớn của nước Mỹ như Giáng Sinh, lễ Tạ Ơn, Ngày Độc lập Mỹ nên xu hướng những phim ra mắt vào thời gian này sẽ có doanh thu cao hơn bởi người dân được nghỉ sẽ dành thời gian xem phim nhiều hơn.

g. Thứ ra mắt



Biểu đồ 20: Biểu đồ so sánh doanh thu của thời gian ra mắt phim (thứ)

Những bộ phim ra mắt vào giữa tuần thường có doanh thu cao hơn cuối tuần. Việc ra mắt phim vào giữa tuần giúp phim sẽ có thời gian để chạy truyền thông để lôi kéo người xem đến rạp vào cuối tuần.

II. Mô hình hóa dữ liệu với LightGBM

1. Chia dữ liệu

Chia dữ liệu train thành hai phần trainning set và validation set trong đó tập để huấn luyện chiếm 80% số lượng dữ liệu.

2. Cơ sở Toán học

a. Cây quyết định

Cây quyết định (decision tree) là một mô hình học máy dựa trên nguyên tắc của cấu trúc cây nhị phân để ra quyết định dựa trên dữ liệu đầu vào. Nó mô phỏng quá trình ra quyết định thông qua loạt các quyết định tuần tự được thực hiện từ nút gốc (root node) xuống đến các nút lá (leaf node). Thuật toán cốt lõi để xây dựng cây quyết định được gọi là ID3 của J. R. Quinlan, sử dụng tìm kiếm tham lam từ trên xuống trong không gian của các nhánh có thể mà không cần quay lại. Thuật toán ID3 có thể được sử dụng để xây dựng cây quyết định cho hồi quy bằng cách thay thế Information Gain và Standard Deviation Reduction. Standard deviation reduction được sử dụng để đo lường mức độ giảm của độ lệch chuẩn khi phân chia dữ liệu. Khi phân chia dữ liệu, ta cần các nhóm con có độ lệch chuẩn thấp nhất

Công thức tính Standard deviation reduction:

$$SDR(T,X) = S(T) - \sum_{c \in X} P(c)\, S(c)$$

Trong đó: SDR(T,X) là mức độ giảm độ lệch chuẩn của biến X đối với biến phụ thuộc T. S(T) là Độ lệch chuẩn của biến phụ thuộc T, Tổng xích-ma của P(c)S(c) là tổng độ lệch chuẩn của từng phần tử c trong tập dữ liệu X.

b. Gradient Boosting

Gradient boosting là một phương pháp học máy cơ bản được sử dụng cho cả bài toán regression và classification. Đây là một phương pháp ensemble learning, có nghĩa là nó kết hợp nhiều mô hình yếu để tạo thành một mô hình mạnh.

Quá trình của gradient boosting diễn ra theo các bước sau:

- Bước 1: Xây dựng một mô hình đơn giản: Bắt đầu bằng một mô hình rất đơn giản, thường là một cây quyết định để dự đoán giá trị ban đầu. Đối với bài toán hồi quy, mô hình ban đầu có thể là giá trị trung bình của hàm mục tiêu.
- Bước 2: Tìm sai lệch (residuals): Tính toán sai lệch giữa dự đoán và giá trị thực tế. Mục tiêu là tìm cách cập nhật mô hình để giảm sai lệch này. Sai số (residual) tại điểm dữ liệu i có thể được tính bằng cách lấy sự chênh lệch giữa giá trị thực tế và dự đoán:

  Sai số = yi - Fm(xi)

Trong đó: yi là giá trị thực tế tại điểm dữ liệu i, Fm(xi) là giá trị dự đoán của mô hình tại điểm dữ liệu i ở vòng lặp m

- Bước 3: Xây dựng mô hình mới để dự đoán sai lệch: Xây dựng một mô hình mới để dự đoán sai lệch được tìm thấy trong bước trước đó. ô hình mới này cố gắng dự đoán sai lệch còn lại, nghĩa là giảm thiểu khoảng cách giữa dự đoán của mô hình hiện tại và giá trị thực tế.
- Bước 4: Cập nhật mô hình tổng hợp: Cập nhật mô hình tổng hợp bằng cách thêm dự đoán từ mô hình mới vào dự đoán hiện tại, cố gắng giảm sai lệch giữa dự đoán và giá trị thực tế.
- Bước 5: Lặp lại quá trình: Quá trình cập nhật mô hình được lặp lại với mục tiêu là giảm thiểu sai lệch liên tục.

c. LightGBM

LightGBM (Light Gradient Boosting Machine) là một thư viện mã nguồn mở cho học máy, được phát triển bởi Microsoft Research. Nó là một thuật toán tối ưu hóa và xây dựng cây quyết định cho các bài toán regression, classification và ranking. LightGBM là một trong những công cụ mạnh mẽ và nhanh nhất trong các thuật toán gradient boosting.

Trong bài toán regression, hàm mất mát MSE có công thức: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$, đạo hàm bậc nhất của hàm mất mát MSE sẽ được tính theo công thức: $\text{Gradient} = \frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$ Trong đó: y^ là giá trị dự đoán, y là giá trị thực tế, L là hàm mất mát.

Thuật toán LightGBM hoạt động như sau: Xây dựng cây quyết định với quy tắc như đã nêu: Chọn Điểm Chia: Sử dụng histogram để tối ưu hóa việc chia dữ liệu. Dữ liệu được chia thành các bin (ngăn) dựa trên các giá trị của các đặc trưng. Các histogram đại diện cho phân phối của dữ liệu, giúp tối ưu hóa quá trình chia dữ liệu. Điểm chia được chọn dựa trên lượng giảm impurity (gain).

LightGBM sẽ xây dựng cây theo leaf-wise growth - một cách xây dựng cây quyết định mà thuật toán sẽ chọn node để chia dựa trên giá trị của độ lệch chuẩn tốt nhất. Công thức gain được tính dựa trên MSE (regression) để quyết định xem node có nên chia hay không. Sau đó cập nhật mô hình:

Xây dựng cây mới với điểm chia tốt nhất. Cập nhật mô hình tổng hợp bằng cách thêm dự đoán từ cây mới với learning rate vào dự đoán hiện tại.

Vì là một thuật toán gradient boosting, nên nó sẽ huấn luyện tập trung vào việc giảm sai số ở mỗi vòng lặp. LightGBM sử dụng thuật toán GOSS và EFB để tối ưu cây. Trong đó, GOSS là một kỹ thuật lấy mẫu dữ liệu dựa trên gradient trong quá trình huấn luyện mô hình. Kỹ thuật này chủ yếu tập trung vào việc lấy mẫu các điểm dữ liệu có gradient nhỏ một cách hiệu quả. Công thức của GOSS thường liên quan đến việc lấy mẫu các điểm dữ liệu có gradient lớn và giữ lại một tỷ lệ nhất định của các điểm dữ liệu có gradient nhỏ. Bằng cách này, GOSS giúp giảm thiểu số lượng dữ liệu cần tính toán trong quá trình huấn luyện mô hình mà vẫn giữ được sức mạnh của gradient để cập nhật các cây quyết định một cách hiệu quả.

EFB là một kỹ thuật mà LightGBM sử dụng để tối ưu hóa việc xử lý các đặc trưng (features) có mối quan hệ gần nhau hoặc có thể được gói gọn với nhau mà không ảnh hưởng đến hiệu suất của mô hình. Kỹ thuật này gom nhóm các đặc trưng có tương quan cao thành các gói đặc trưng. Bằng cách này, LightGBM có thể xử lý các gói đặc trưng này một cách hiệu quả hơn trong quá trình xây dựng cây quyết định, giảm chi phí tính toán và tăng tốc độ huấn luyện.

3. Xây dựng và tinh chỉnh mô hình

Thuật toán LightGBM được xây dựng gồm các tham số:

- max_depth: Độ sâu tối đa của cây. Nếu đặt là -1, thì độ sâu sẽ được quyết định bởi thuật toán dựa trên số lượng dữ liệu và num_leaves. Giá trị lớn hơn có thể dẫn đến overfitting.
- learning_rate: Tốc độ học, còn được gọi là shrinkage. Đây là tỷ lệ thu nhỏ mà mô hình áp dụng sau mỗi vòng tăng cường, giúp kiểm soát overfitting.
- max_bin: bin đề cập đến việc chia dữ liệu thành các ngăn (bin) để xây dựng histogram. Mỗi ngăn chứa một phạm vi giá trị của một features. Tham số max_bin trong LightGBM xác định số lượng ngăn tối đa mà LightGBM sẽ sử dụng để xây dựng histogram. Giá trị lớn hơn có thể tạo ra mô hình tốt hơn, nhưng cũng làm tăng thời gian huấn luyện và sử dụng nhiều bộ nhớ hơn. Ví dụ, nếu bạn có một đặc trưng với giá trị từ 1 đến 100 và đặt max_bin là 10, thì LightGBM sẽ tạo ra 10 ngăn, mỗi ngăn chứa 10 giá trị (1-10, 11-20, v.v.). LightGBM sau đó sẽ sử dụng các ngăn này để xây dựng histogram và tạo ra cây quyết định
- num_leaves: Số lượng lá tối đa trong mỗi cây. Giá trị lớn hơn có thể tạo ra mô hình tốt hơn, nhưng cũng có thể dẫn đến overfitting.
- feature_fraction: LightGBM sẽ thực hiện lấy ngẫu nhiên một tập hợp các feature trước khi mỗi lần lặp cây (ví dụ 0.6 LightGBM sẽ chọn 60% feature trước khi lặp cây)
- lambda_l1: Tham số này được sử dụng để áp dụng điều chuẩn L1 (Lasso Regularization), đây là thành phần được thêm vào hàm mất mát có công thức: $L1\ \text{regularization} = \lambda \sum_{i=1}^{n} |w_i|$ Trong đó: L1 là phương pháp chuẩn hóa L1. λ là tham số điều chỉnh, wi là trọng số của mô hình, n là số lượng trọng số trong mô hình. Chuẩn L1 giúp feature selection tự động, giúp loại bỏ hoặc giảm thiểu ảnh hưởng của các đặc trưng không quan trọng trong quá trình huấn luyện mô hình. Điều này giúp mô hình trở nên đơn giản hơn và ngăn chặn hiện tượng overfitting.

Xây dựng hàm tinh chỉnh mô hình: Ở đây tôi sẽ xây dựng một hàm tìm kiếm lưới với hai tham số mỗi tham số ba giá trị mẫu (9 kết hợp thử nghiệm từ đó tăng tốc thời gian chạy trong khi nếu sử dụng 6 tham số thì sẽ

cần 3^6 lần kết hợp) để tìm ra sự kết hợp tối ưu của các tham số. Hàm bắt đầu bằng việc khởi tạo các biến để lưu trữ giá trị RMSE thấp nhất và các tham số tốt nhất. Hàm sử dụng vòng lặp lồng nhau để duyệt qua tất cả các kết hợp giá trị cho hai tham số đầu vào.

Bên trong vòng lặp, hàm thiết lập các giá trị tham số hiện tại. Sau đó, nó thực hiện cross-validation (lgb.cv) sử dụng LightGBM với các giá trị tham số được cung cấp và in ra giá trị RMSE và số vòng boosting tương ứng.

Nếu giá trị RMSE thu được thấp hơn so với giá trị RMSE tối ưu hiện tại, nó cập nhật giá trị RMSE tối ưu và ghi nhận các siêu tham số tốt nhất.

Cuối cùng, sau khi duyệt qua tất cả các kết hợp, hàm in ra cặp siêu tham số tốt nhất cùng với giá trị RMSE tương ứng.

Sau khi tối ưu, tôi thu được những giá trị tham số như sau:

- parameters_lgb['learning_rate'] = 0.05
- parameters_lgb['max_depth'] = 5
- parameters_lgb['max_bin'] = 255
- parameters_lgb['num_leaves'] = 31
- parameters_lgb['feature_fraction'] = 0.5
- parameters_lgb['lambda_l1'] = 0.2

4. Thực hiện đào tạo mô hình

```
clf_lgb = lgb.train(params = parameters_lgb,
                    train_set = lgb.Dataset(X_train_part, y_train_part),
                    num_boost_round = 10000,
                    valid_sets = [lgb.Dataset(X_val, y_val)],
                    early_stopping_rounds = 500,
                    verbose_eval = 10)

y_pred = clf_lgb.predict(X_cross)
```

Hình 10: Mô hình huấn luyện LightGBM

params = parameters_lgb: Đây là tham số chính để truyền vào các cài đặt của mô hình LightGBM. parameters_lgb chứa một từ điển các cài đặt cho mô hình đã nêu ở trên.

train_set = lgb.Dataset(X_train_part, y_train_part): Đây là tập dữ liệu huấn luyện, sử dụng lgb.Dataset từ LightGBM. X_train_part là tập dữ liệu đặc trưng và y_train_part là tập dữ liệu target (biến cần dự đoán) tương ứng.

num_boost_round = 10000: Đây là số lượng vòng lặp huấn luyện (boosting rounds) tối đa mà mô hình sẽ thực hiện. Điều này xác định số lượng cây cây mà mô hình Gradient Boosting sẽ sử dụng.

valid_sets = [lgb.Dataset(X_val, y_val)]: Đây là tập dữ liệu validation được sử dụng để đánh giá mô hình trong quá trình huấn luyện. X_val là tập dữ liệu đặc trưng validation và y_val là tập dữ liệu target tương ứng.

early_stopping_rounds = 500: Tham số này cho phép dừng sớm quá trình huấn luyện nếu không có cải thiện đáng kể trong số liệu validation sau một số vòng lặp nhất định (500 trong trường hợp này). Nếu không có cải thiện trong 500 vòng lặp liên tiếp, huấn luyện sẽ dừng lại để tránh overfitting và tiết kiệm thời gian huấn luyện.

erbose_eval = 10: Thông số này quy định cách thông tin về quá trình huấn luyện được hiển thị. Trong trường hợp này, thông tin sẽ được hiển thị sau mỗi 10 vòng lặp.

5. Đánh giá hiệu suất mô hình

a. Độ lệch bình phương trung bình

Độ lệch bình phương trung bình của mô hình là: $93844149.71351561

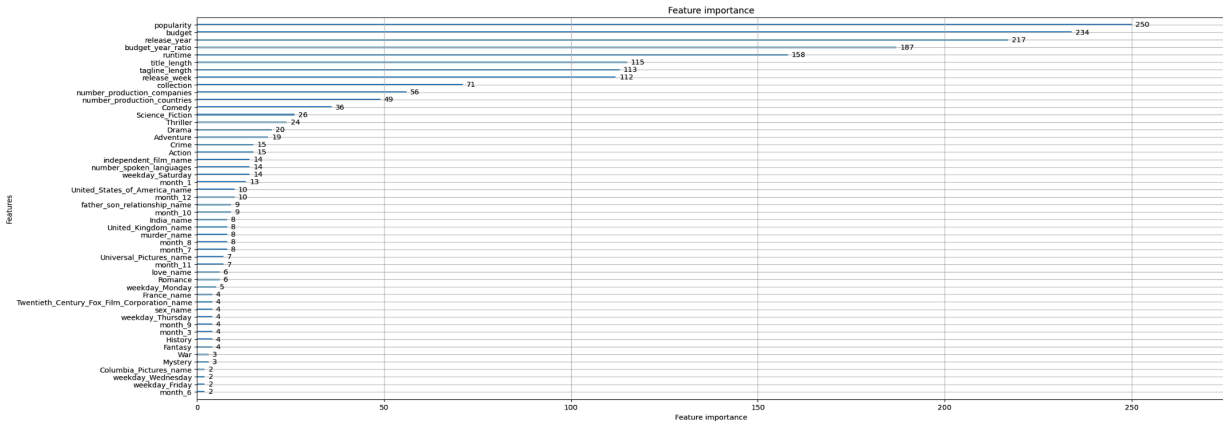
b. Giá trị dự đoán

	predictions	actual	difference
0	8.819892e+07	1.202071e+08	-3.200820e+07
1	8.846433e+06	5.897865e+07	-5.013222e+07
2	4.028911e+07	9.571488e+07	-5.542576e+07
3	1.324165e+07	1.990536e+07	-6.663705e+06
4	3.083126e+07	9.996575e+07	-6.913449e+07
5	1.260267e+07	1.876740e+05	1.241500e+07
6	8.286576e+06	2.804874e+06	5.481702e+06
7	4.477689e+06	1.000000e+06	3.477689e+06
8	7.424153e+07	8.500000e+06	6.574153e+07
9	6.252413e+07	1.197722e+08	-5.724810e+07
10	1.833429e+08	1.941687e+08	-1.082580e+07
11	3.465262e+07	5.320818e+07	-1.855556e+07
12	1.740744e+07	3.769990e+06	1.363745e+07
13	7.286098e+07	1.400000e+06	7.146098e+07
14	2.200527e+06	2.166547e+07	-1.946494e+07
15	1.235475e+08	8.665856e+07	3.688895e+07
16	1.591535e+08	1.166433e+08	4.251013e+07
17	8.490490e+06	4.900000e+07	-4.050951e+07
18	1.782203e+07	5.600000e+06	1.222203e+07
19	3.551768e+03	1.367400e+04	-1.012223e+04
20	1.255790e+07	2.000000e+07	-7.442099e+06
21	4.730478e+06	5.000000e+06	-2.695219e+05
22	1.260017e+08	3.730629e+08	-2.470611e+08
23	7.963003e+06	7.514300e+04	7.887860e+06
24	4.213927e+07	1.947811e+07	2.266116e+07
25	2.791102e+08	7.098275e+08	-4.307172e+08
26	5.308285e+06	7.900000e+06	-2.591715e+06
27	3.891844e+06	1.165738e+07	-7.765541e+06
28	8.140493e+06	9.902115e+06	-1.761622e+06
29	7.666770e+06	1.020921e+06	6.645849e+06
30	6.824273e+06	2.200000e+06	4.624273e+06
31	2.367739e+07	1.345903e+06	2.233149e+07
32	4.288460e+07	2.305712e+07	1.982748e+07
33	6.182618e+06	4.233330e+07	-3.615068e+07
34	1.030597e+07	7.774230e+05	9.528545e+06

Hình 11: Doanh thu dự đoán sau khi sử dụng LightGBM

Sai số doanh thu phim so với giá trị thực tế khá lớn, tuy nhiên nhìn vào việc đây là bộ dữ liệu chứa những bộ phim nổi tiếng từ trước đến nay, doanh thu của nó tương đối cao lên đến hàng trăm triệu đô nên sai số vậy là thực tế so với một mô hình học máy cơ bản.

c. Các dữ liệu quan trọng



Hình 12: Các trường dữ liệu quan trọng được sử dụng trong thuật toán

Các trường dữ liệu quan trọng được sử dụng trong thuật toán là: popularity, budget, release_year, budget_year_ratio,..

Chương 4: Kết luận

Trong bài viết trên tôi đã xây dựng một mô hình dự đoán doanh thu phòng vé toàn cầu của các bộ phim dựa trên dữ liệu của hơn 7.000 bộ phim từ The Movie Database. Mô hình sử dụng thuật toán LightGBM và đạt được kết quả sai số bình phương trung bình tối thiểu là: $93844149.71351561 trên tập dữ liệu thử nghiệm.

Dựa trên kết quả phân tích dữ liệu ban đầu, chúng ta có thể thấy rằng các yếu tố ảnh hưởng đến doanh thu phòng vé của một bộ phim bao gồm:

- Popularity: Độ nhận diện bộ phim đối với người xem đóng vai trò quan trọng đến với doanh thu. Các công ty cần có những chiến lược truyền thông sáng suốt để đưa bộ phim đến khách hàng.
- Ngân sách: Cần dành lượng ngân sách phù hợp cho bộ phim. Con số đã cho chúng ta thấy không phải phim ngân sách cao sẽ có doanh thu cao nhưng phim có ngân sách thấp thì doanh thu trung bình sẽ không thể cao bằng những phim khác.
- Thể loại: Các bộ phim thuộc thể loại hành động, phiêu lưu, khoa học viễn tưởng, và siêu anh hùng thường có doanh thu cao hơn.
- Diễn viên: Các bộ phim có sự tham gia của các diễn viên nổi tiếng thường có doanh thu cao hơn.
- Ngày phát hành: Các bộ phim phát hành trong mùa hè thường có doanh thu cao hơn.
- Nước sản xuất: Các bộ phim của Mỹ thường có doanh thu cao hơn.

Mô hình dự đoán của tôi đã sử dụng một số yếu tố này để đưa ra dự đoán về doanh thu phòng vé của các bộ phim. Tuy nhiên mô hình còn đối mặt với nhiều hạn chế ví dụ như:

- Tập dữ liễu vẫn còn hạn chế. Dữ liệu thu được khi train chỉ dừng lại ở 1409 dữ liệu và 353 dữ liệu làm validation nên kết quả sẽ còn nhiều hạn chế.
- Tốn tài nguyên: LightGBM có thể sử dụng lượng bộ nhớ lớn hơn so với một số thuật toán khác, đặc biệt là khi xử lý các tập dữ liệu lớn với các cấu hình mô hình phức tạp bởi nó phải xây dựng một lượng lớn cây quyết định trong quá trình huấn luyện.
- Thời gian huấn luyện với tham số: Quá trình tìm kiếm tham số tốt nhất có thể tốn thời gian nếu có một số lượng lớn tham số cần điều chỉnh. Trong dự án của mình, tôi chỉ sử dụng ba giá trị để thử nghiệm, nên có thể sẽ không được tối ưu, tuy nhiên sau này khi thử nghiệm với nhiều giá trị hơn để tìm giá trị thực sự tối ưu thì đấy là thời gian để tìm chúng là rất lớn.

- Độ nhạy cảm với tham số: Các tham số để huấn luyện đóng vai trò quan trọng trong quá trình huấn luyện. Trong quá trình thực hiện project, tôi đã một lần truyền sai một tham số đầu vào và cho ra được kết quả output với kết quả rất tệ. Điều này thể hiện khi sử dụng LightGBM chúng ta cần cẩn thận với các tham số đầu vào.
- Khó tiếp cận: Mô hình không cung cấp các giải thích rõ ràng về cách nó đưa ra dự đoán của mình vì vậy trong quá trình tìm hiểu có thể gặp nhiều khó khăn trong việc tối ưu mô hình.

Tuy nhiên, cần lưu ý rằng doanh thu phòng vé của một bộ phim cũng chịu ảnh hưởng của nhiều yếu tố khác, chẳng hạn như chất lượng của bộ phim, chiến lược tiếp thị, và tình hình kinh tế xã hội. Do đó, kết quả dự đoán của mô hình chỉ mang tính chất tham khảo.

Một số hướng phát triển cho mô hình dự đoán trong tương lai bao gồm:

- Sử dụng dữ liệu bổ sung: Ngoài dữ liệu từ dataset đã tạo, chúng ta có thể sử dụng thêm các dữ liệu khác như dữ liệu về các đánh giá của khán giả, dữ liệu về doanh thu phòng vé của các bộ phim tương tự, để cải thiện độ chính xác của mô hình.
- Tìm các tham số tối ưu cho thuật toán. Đây là một bước rất quan trọng nhưng tốn kém nhiều tài nguyên.
- Sử dụng các thuật toán khác: Ngoài LightGBM, chúng ta có thể thử nghiệm các thuật toán khác như XGBoost, CatBoost, để tìm ra thuật toán phù hợp nhất với dữ liệu.
- Phát triển ứng dụng để các nhà dự đoán phim có thể tham khảo nhận được dự đoán doanh thu của bộ phim ra mắt tương lai.

Tài liệu tham khảo và Link source code

1. “Decision Tree Regression.” Data Mining Map, https://saedsayad.com/decision_tree_reg.htm. Accessed 4 January 2024.
2. “Welcome to LightGBM’s documentation! — LightGBM 4.2.0.99 documentation.” LightGBM's documentation!, https://lightgbm.readthedocs.io/en/latest/index.html. Accessed 4 January 2024.
3. Link source code: https://github.com/namkjs/DataSciencePrj.git

Lời cảm ơn

Em xin cảm ơn thầy Lê Chí Ngọc đã tham gia giảng dạy và giúp đỡ em trong học phần Khoa học dữ liệu này. Kỳ học vừa qua là một trải nghiệm mới đối với em, không chỉ về kiến thức mà thầy còn đã dạy em về cả tư duy và kỹ năng mềm trong nghề. Những kiến thức của thầy dạy đã giúp em rất nhiều trong chặng đường tương lai sắp tới.

Trân trọng,

Nam
Lê Hải Nam