Thông tin nhóm:

Tên thành viên	MSSV	Nhiệm vụ	Hoàn thành
Nguyễn Minh Đạt	21127592	- Phân tích - Đặt câu hỏi	100%
Nguyễn Nhật Truyền	21127191	- Tiền xử lý - Khám phá dữ liệu	100%

Mục lục:

1. Thông tin bộ dữ liệu

- **Tên bộ dữ liệu**: Netflix Movie Rating Dataset
- **Nguồn**: Kaggle
- **Mô tả**:
    - Đây là tập dữ liệu đủ lớn để xây dựng một mô hình đề xuất tốt và được điều chỉnh từ "tập dữ liệu giải thưởng Netflix" rất lớn, và bạn có thể gặp vấn đề về bộ nhớ khi đào tạo mô hình sử dụng tập dữ liệu đó.
    - Netflix đã tổ chức cuộc thi mở Netflix Prize để tìm ra thuật toán tốt nhất để dự đoán xếp hạng của người dùng đối với các bộ phim.
- **Cấu trúc**:
    - Tệp phim chứa Movie_ID, Tên, Năm.
    - Tệp xếp hạng chứa Movie_ID, User_ID, Xếp hạng.
- **Quy mô**:
    - 17,770 bộ phim.
    - 17,337,458 đánh giá.
- **Cách thức xử lý dữ liệu**:
    - Môi trường:
        - Google Colab.
        - Ngôn ngữ Python 3.0.
    - Đầu tiên, chúng em tải dữ liệu từ Kaggle về máy tính cá nhân.
    - Sau đó, chúng em đẩy dữ liệu lên Google Drive.
    - Từ tập tin .ipynb, chúng em thực hiện lệnh “mount" đến Google Drive.
    - Cuối cùng, chúng em truy xuất và sử dụng dữ liệu.
    - Lưu ý:
        - Chúng em dự định đẩy dữ liệu lên Github, sau đó thực hiện lệnh “git clone" để lấy dữ liệu.
        - Tuy nhiên, chúng em không thể thực hiện được vì kích thước tập tin quá lớn.
        - Do đó, chúng em chấp nhận làm theo hướng bất tiện hơn là mount Google Drive.
    - Cài đặt:

```
from google.colab import drive
drive.mount('/content/drive')
# Set the path to the data folder
folder_path = '/content/drive/MyDrive/Lab02_Visualization'
# Set the current working directory to the data folder
```

```
os.chdir(folder_path)

# Set the path to the CSV files
csv_movie = 'Netflix_Dataset_Movie.csv'
csv_rating = 'Netflix_Dataset_Rating.csv'
# Read the CSV file into a pandas DataFrame
df_movie = pd.read_csv(csv_movie)
df_rating = pd.read_csv(csv_rating)
```

2. Phân tích

- **Chất lượng dữ liệu**: Tốt.
  - Không trùng lặp.
  - Không mất dữ liệu.
- **Thuộc tính thời gian**:
  - Đơn vị: Năm
  - Được tính từ năm 1915 đến năm 2005.
  - Số lượng năm lớn → Cần được gom nhóm theo thập kỉ.
- **Nhu cầu thêm cột dữ liệu**:
  - Thuộc tính cần thêm: Thập kỉ.
  - Được tính theo năm tương ứng.
  - Ví dụ: Các năm 1991, 1994, 1997 đều thuộc thập kỉ 1990.
  - Cài đặt:

```
# Create a new column 'Decade' that represents the decade for each movie
df_movie['Decade'] = (df_movie['Year'] // 10) * 10
```

- **Nhu cầu liên kết dữ liệu**:
  - Lý do: Thông tin về bộ phim và thông tin về đánh giá nằm ở hai file khác nhau.
  - Sau khi đọc file và lưu dữ liệu vào dataframe, chúng em cần kết hai dataframe lại với nhau theo thuộc tính “Movie_ID".
  - Từ đó, chúng em có thể khai thác được các thông tin quan trọng về mối quan hệ giữa bộ phim và đánh giá.
  - Cài đặt:

```
# Merge df_movie and df_rating on 'Movie_ID'
df_merged = pd.merge(df_movie, df_rating, on='Movie_ID')
```

3. Đặt câu hỏi

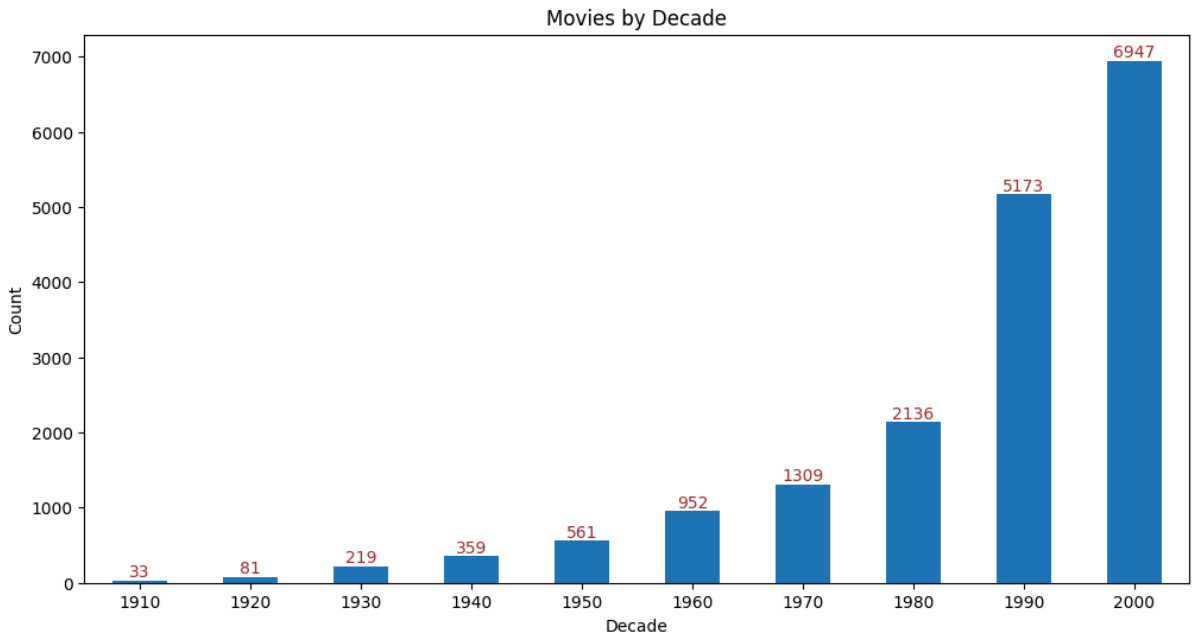
- Trước hết, chúng em đặt các câu hỏi đơn giản để làm quen với bộ dữ liệu:
    - Phân bố số lượng bộ phim theo thập kỉ.
    - Phân bố số lượng đánh giá.
- Sau đó, chúng em tìm hiểu mối tương quan giữa các biến định lượng.
- Cuối cùng, chúng em đặt câu hỏi về giá trị trung bình của các đánh giá theo từng thập kỉ.

4. Trả lời câu hỏi

- Phân bố số lượng bộ phim theo thập kỉ.
    - Để trả lời cho câu hỏi này, chúng em lựa chọn sử dụng biểu đồ cột.
    - Lý do: biểu đồ cột cho phép trực quan dữ liệu một cách rõ ràng và thuận lợi cho việc so sánh thông tin.
    - Cài đặt:

```
# Get the decade distribution
decade_distribution = df_movie['Decade'].value_counts().sort_index()
# Create the bar chart
plt.figure(figsize=(12, 6))
decade_distribution.plot(kind='bar')
plt.title('Movies by Decade')
plt.xlabel('Decade')
plt.ylabel('Count')
plt.xticks(rotation=0)
# Add the value above each bar
for i, v in enumerate(decade_distribution):
   plt.text(i, v, str(v), color='brown', fontweight='light', ha='center',
va='bottom')
plt.show()
```

    - Kết quả:



- Nhận xét:
  - Biểu đồ lệch trái, với phần lớn bộ phim nằm bên phải. Điều đó cho thấy số lượng bộ phim tăng mạnh qua từng thập kỉ.
  - Cụ thể, số lượng bộ phim gần như tăng gấp đôi qua mỗi thập kỉ. Tiêu biểu như:
    - Tăng từ **81** (1920) lên **219** (1930).
    - Tăng từ **561** (1950) lên **952** (1960).
    - Tăng từ **2136** (1980) lên **5173** (1990).

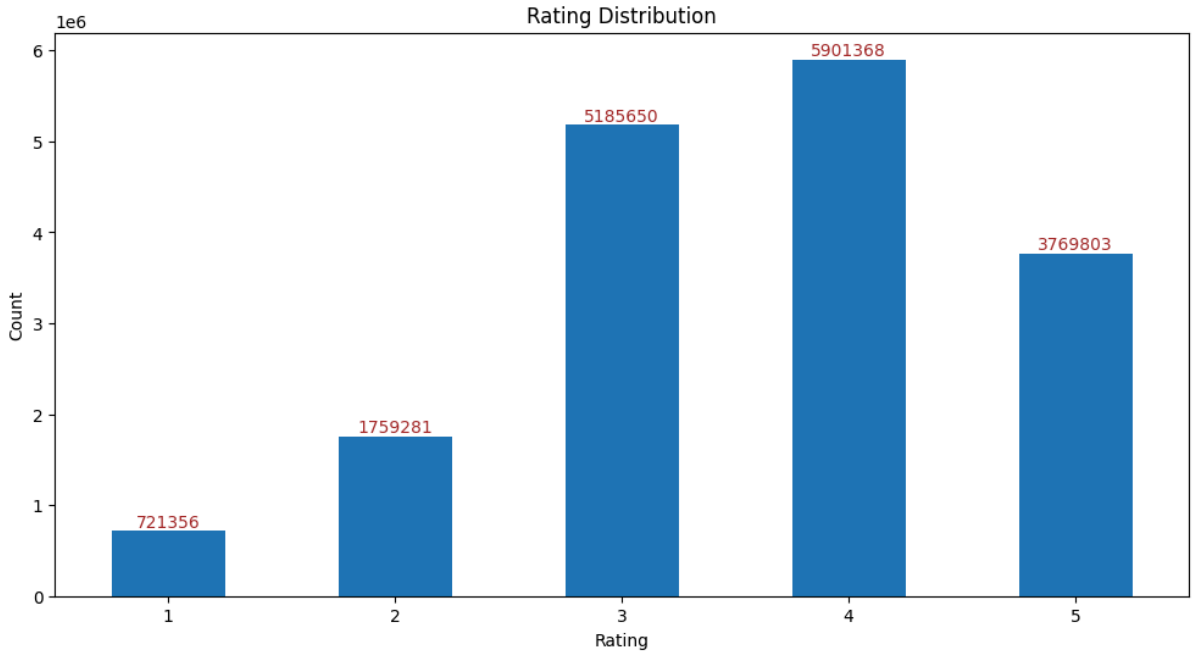
- Phân bố đánh giá.

- Để trả lời câu hỏi này, chúng em tiếp tục sử dụng biểu đồ cột.
- Cài đặt:

```
# Get the rating distribution
rating_distribution = df_rating['Rating'].value_counts().sort_index()
# Create the bar chart
plt.figure(figsize=(12, 6))
rating_distribution.plot(kind='bar')
plt.title('Rating Distribution')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=0)
# Add the value above each bar
for i, v in enumerate(rating_distribution):
   plt.text(i, v, str(v), color='brown', fontweight='light', ha='center',
va='bottom')
```

```
plt.show()
```

- Kết quả:



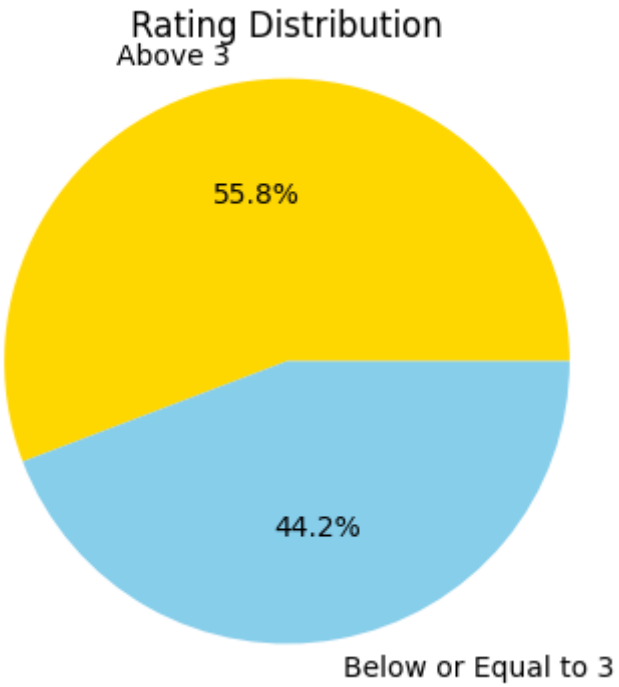
- Nhận xét:
  - Biểu đồ lệch trái, với phần lớn đánh giá nằm về phía bên phải. Điều đó cho thấy số lượng đánh giá tích cực (từ 3 sao trở lên) hoàn toàn vượt trội so với số lượng đánh giá dưới 3 sao.
  - Cụ thể, các đánh giá 4 sao chiếm số lượng nhiều nhất (gần 6 triệu đánh giá), tiếp theo đến các đánh giá 3 sao (trên 5 triệu đánh giá).
- Giả sử:
  - Chúng em chia các đánh giá thành 2 nhóm:
    - Nhóm trên 3.
    - Nhóm bé hơn hoặc bằng 3.
  - Sau đó, chúng em thực hiện trực quan tỉ lệ của 2 nhóm sử dụng biểu đồ tròn.
  - Cài đặt:

```
# Divide the ratings into two groups
above_3 = (df_rating['Rating'] > 3).sum()
below_equal_3 = (df_rating['Rating'] <= 3).sum()

# Create the pie chart
labels = ['Above 3', 'Below or Equal to 3']
sizes = [above_3, below_equal_3]
```

```
colors = ['gold', 'skyblue']
plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('Rating Distribution')
plt.axis('equal')  # Equal aspect ratio ensures that pie is circular.
plt.show()
```

- Kết quả:



- Nhận xét:
  - Số lượng các đánh giá trên 3 sao chiếm hơn một nửa tổng số lượng đánh giá.
  - Điều này có thể cho thấy chất lượng các bộ phim hoặc sự tích cực trong cách đánh giá của khán giả.

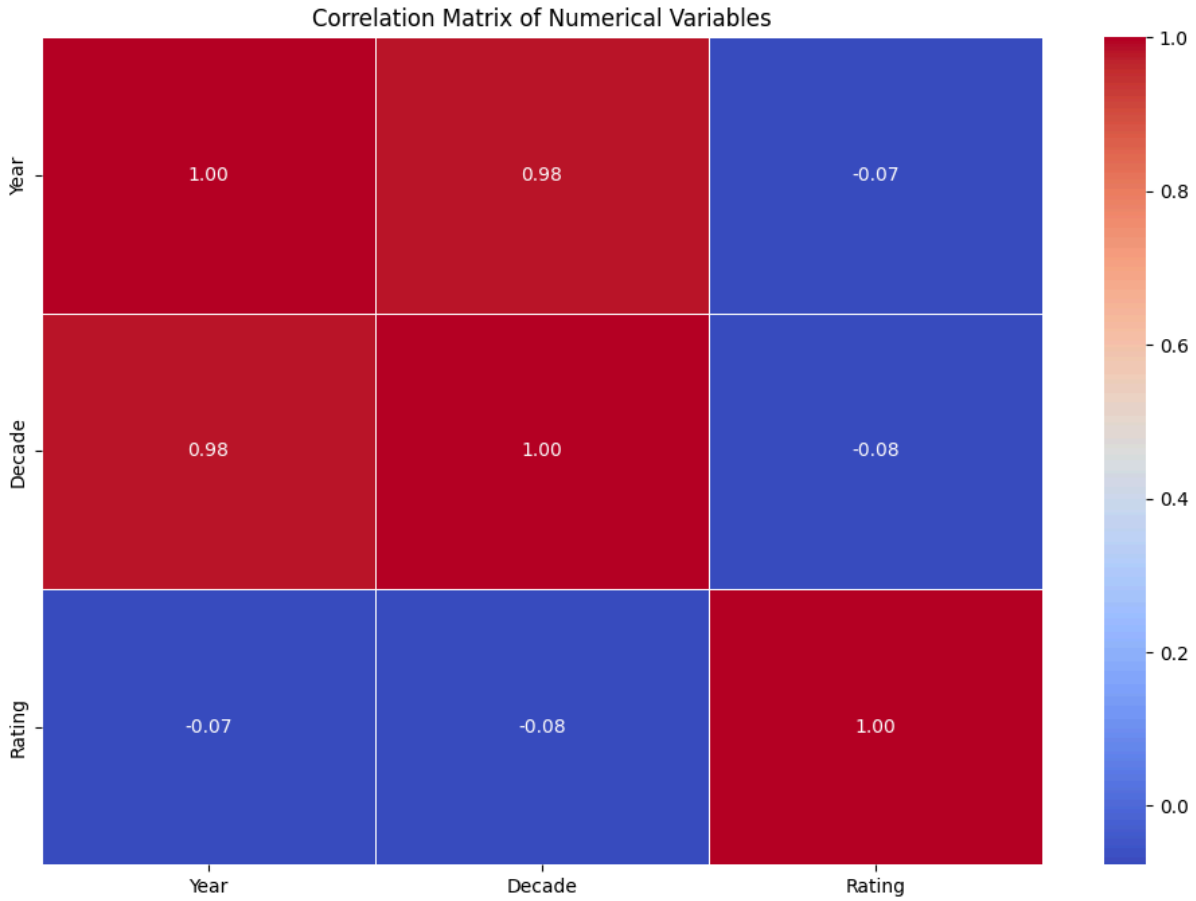
- Tương quan giữa các biến định lượng.

- Để trực quan tương quan giữa các biến định lượng, trước hết, chúng em loại bỏ các biến liên quan đến “ID".
- Sau đó, chúng em minh hoạ ma trận tương quan bằng thư viện seaborn.
- Cài đặt:

```
num_list = [col for col in df_merged.dtypes[df_merged.dtypes != 'object'].index if 'ID' not in col]
correlation_matrix = df_merged[num_list].corr()
plt.figure(figsize=(12, 8))
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title('Correlation Matrix of Numerical Variables')
plt.show()
```

- Kết quả:



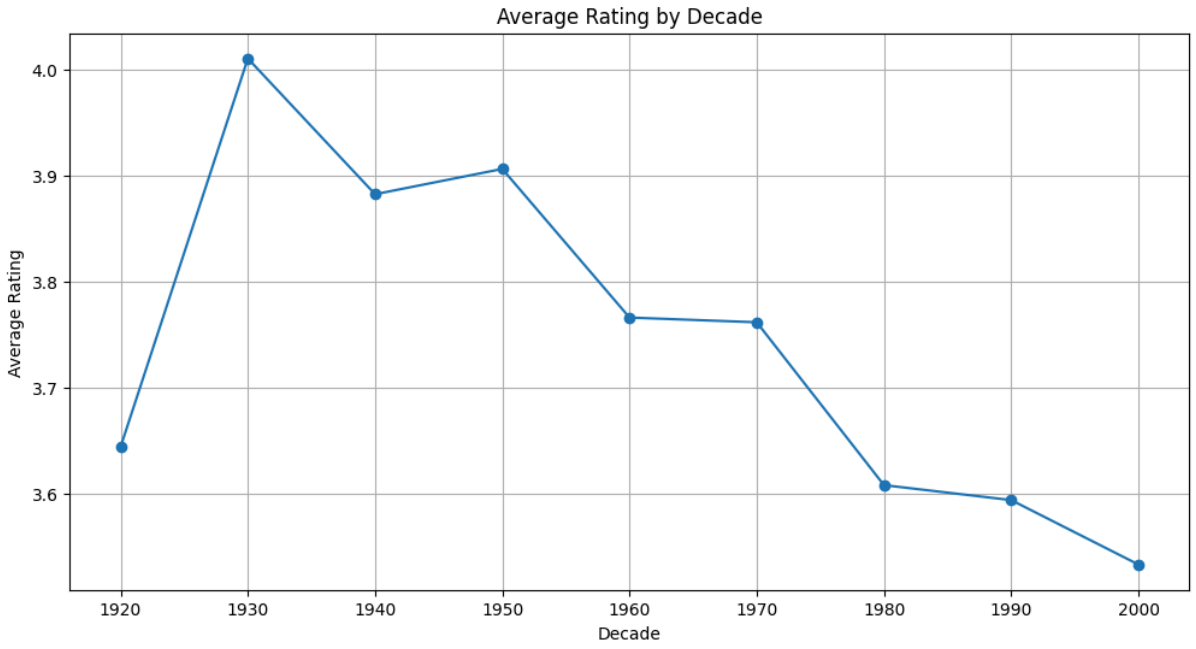
- Nhận xét:
  - Biến “year" và “decade” có hệ số tương quan là 0.98, biểu thị mối quan hệ “đồng biến” rất mạnh. Điều này hoàn toàn hợp lý bởi thuộc tính “decade” thực chất được sinh ra từ thuộc tính “year”.
  - Ngược lại, cả hai biến “year” và “decade” đều có hệ số tương quan âm so với biến “rating” (lần lượt là -0.07 và -0.08, biểu thị mối quan hệ “nghịch biến”.

- Giá trị trung bình của các đánh giá theo từng thập kỉ.
    - Chúng em thực hiện tính giá trị trung bình của tất cả các đánh giá trong một thập kỉ. Như vậy, mỗi thập kỉ sẽ có một giá trị đánh giá trung bình.
    - Tiếp theo, chúng em sử dụng biểu đồ đường để trực quan sự thay đổi của giá trị trung bình theo thời gian.
    - Cài đặt:

```
# Compute the average rating for each decade
decade_avg_rating =
df_merged.groupby('Decade')['Rating'].mean().sort_index()

# Create the line graph
plt.figure(figsize=(12, 6))
decade_avg_rating.plot(kind='line', marker='o')
plt.title('Average Rating by Decade')
plt.xlabel('Decade')
plt.ylabel('Average Rating')
plt.xticks(rotation=0)
plt.grid(True)
plt.show()
```

    - Kết quả:



    - Nhận xét:

- Giá trị trung bình của các đánh giá có xu hướng giảm qua từng thập kỉ. Điều này có thể phản ảnh chất lượng bộ phim giảm hoặc thị hiếu của độc giả trở nên khắt khe hơn.
- Thập kỉ có giá trị đánh giá trung bình tốt nhất là 1930, với trung bình đánh giá trên 4 sao.

5. Kết luận

- Thông qua việc phân tích và trả lời câu hỏi, chúng em rút ra được các góc nhìn thú vị sau:
    - Số lượng bộ phim tăng vọt trong khoảng thời gian từ năm 1915 đến 2005.
        - Cụ thể, trong những năm thuộc thập kỉ 1920, chỉ có chưa tới 100 bộ phim được sản xuất.
        - Tuy nhiên, chỉ trong nửa đầu thập kỉ 2000 đã có tới gần 7,000 bộ phim được sản xuất.
    - Hơn một nửa các đánh giá là từ 4 đến 5 sao.
        - Cụ thể, các đánh giá 4 và 5 sao chiếm trên 55% tổng số lượng đánh giá.
    - Tuy nhiên, giá trị trung bình của các đánh giá có xu hướng giảm theo thời gian.