M.Sc. Data Analytics and Technologies

DAT7302- Big Data Analytics

Assessment 1

Submitted By:

Student ID: 2310413

Submitted To:

Dr. Anchal Garg

Module Tutor

Submission Date: 4th January 2025

Time of the workshop session: 9:00 am

Declaration

This is my original work as part of the Big Data Analytics assignment requirement.

No generative AI has been used in this assignment.

List of Figures

List of Tables

Table of Contents

1.0 Introduction

The film industry is a multifaceted sector of the economy that has social and commercial functions for society. A significant economic sector, the film industry draws investments and influences the social, ideological, and financial facets of the economy (Manakbayeva, 2022). It encompasses many aspects, starting from production to distribution, and has an endeavour on the local economy as well as the sector that is related to it. Applying analytical tools in the film sector has improved financial planning, production techniques, and, therefore, customer relations. Through the incorporation of data insights, producers and shareholders make strategic decisions about what to produce and what direction to take the cinema industry, which directs its development.

1.1. Business Problem

In this line of business, competition in the entertainment industry makes it difficult to determine the tastes and preferences the market has in store as well as come up with sound strategies for production that will effectively serve the market. The use of IMDb data (given dataset) is a great chance to analyse the trends, engagement, and success indicators for content to become popular and highly rated. Many difficulties concern the entertainment industry in the conditions of the contemporary dynamic media world. The business problem is centered on how to extract useful information from the large IMDb dataset to make decisive and actionable decisions in content production and marketing of content, talent acquisition, and other areas of business. Specifically, it aims to address:

What steps can entertainment businesses take towards unlocking the values of IMDb data to formulate good decisions on draughting their content, choosing talents to cast, and identifying the right markets to target with the ultimate aim of catering to audiences’ needs and making good sales?

These industries require information on the specific media content that is most popular amongst the general public. This includes analysing which genres are popular at the moment and how rating and popularity are influenced by the choice of genre. It also examines the effects that runtime has on the audiences as well as factors that influence the scoring system to hit high ratings on IMDb.

Selecting the right talent defines the success of a project to a very large extent.

Including types of story and style in drama and the cast/crew members playing in a title’s performance.

Knowing the audience preferences in each region and by age is also important. This involves looking at the regional preferences for movies, comparing the effectiveness of television series rather than films, and studying current tendencies in adult industries and their demand.

In this way, entertainment companies can make more informed decisions when it comes to content production, eliciting better engagement from an audience and therefore having better chances of greater revenue within the highly saturated and rival industry.

1.2 Business Questions

1. What affects the ratings and votes of users?
2. Which regions have the highest concentration of popular movies based on ratings and votes?
3. What are the roles (actors, directors, or crew members) that starred in high-rated movies?
4. Which runtime range exhibits the highest average ratings?
5. Which movies released across all years have the highest audience ratings?
6. What are the most common genres for high-rated movies?
7. How does each genre's level of popularity evolve throughout time?
8. What is the primarytitle with the maximum runtime minutes?

After such analysis, a business can ensure that its content is as close to these factors as possible, improving the viewers' satisfaction and engagement. High ratings and votes in a title's visibility thus increase the traffic to movie halls or the number of advertisements shown.

As informed by ratings and votes for films, it is possible to define the areas where popular movies are most often released, and therefore, it will be rational to concentrate on the business positioning in these regions. Since people are diverse in the geographical locations they frequent, marketers can separate the regions where the clients are most receptive to certain forms of content to better forecast the revenue and viewership rates.

Picking the most popular genres according to the ratings and the number of votes these movies can help businesses focus on the genres that are going to be liked by the viewers. Purchasing well-liked genres helps to match content production with demand, thus also decreasing the likelihood of low successful works and increasing profitability.

The average length of the movies across and within genres shows what duration of movies is most preferred by the audience in certain genres. For instance, audiences may wish to watch short movies if they fall under the comedy genre, while the prolonged long runtime might be preferable to the movie falling under the drama or fantasy category. Knowledge of these preferences enables the producers and the sites hosting the content to align the duration of the content to meet the audience's expectations and, as a result, increase satisfaction levels.

It means that by analysing who the actors, directors, or the rest of the crew are involved with in movies with high ratings, one will find that sometimes it is possible to pinpoint talent that proves cohesion to the success factor within the business industry. It can inform casting and partnership choices to help producers put together teams that could raise the prospect of hitting cultural milestones as well as financial targets.

Studying the tendencies in the consumption of genres enables businesses to anticipate what kind of content will be liked by audiences in the future. For instance, if the current trend suggests a growing interest in science fiction, such as that it is an emerging trend, a firm can ensure it invests in the particular content type before others do, thus gaining a cutting edge on rivals for emerging markets.

The division of program runtime into different ranges to identify which range has the highest average rating enables excellent audience targeting. Since people have their preferences on movie duration, the producers may direct their productions to those timeframes, hence guaranteeing higher audience satisfaction.

Answering these questions helps to improve typical business decisions based on data, content strategies, audience targeting, and revenues. The people’s choice and the market demands can be identified, and thus, to promote and sell content accordingly to sustain both the business structures and the entertainment industry in the long run.

2.0 Review of Literature

Genre plays a crucial role in defining the performance of a film, influencing everything from production and marketing strategies to reception by the audience. As the field progresses, the evaluation of genre and its effects on movie processing continues to be a key focus of interest to filmmakers, producers, and marketers. An analysis of genre has been the venue of focus in several studies. Matthews and Glitre (2024) adopted topic modelling via plot summaries within the space of movie genres; recent research specified how movie genres change (Matthews and Glitre, 2024). It has also been established that certain genres are preferred by the audience, and other genres, especially drama, will get higher ratings than others (Juan, 2019).

Many works attempted to examine the connection between different characteristics and IMDb ratings. Pavan and Manjunath (2024) established that cultural relevance is a critical factor responsible for the construction of audience knowledge and selection. Namely, budget, genre, vote count, and popularity do concern the case of revenues (Pavan and Manjunath, 2024).

The film production industry is inherently unpredictable, and therefore, direction decision-making requires reliable data. The consumer's choice is one critical factor in decision-making when it comes to film production based on online ratings. Gavilan et al.(2019) report that one way in which aggregated numerical ratings ease decision-making for viewers is that they lower the perceived risk of choosing films (Gavilán et al., 2019). This discovery is rather important to producers since the results of the improvement of an online rating of a specific film can lead to increased numbers of viewers and, therefore, a grossing factor. Kumar (2024) also underlines that ratings play a critical role in consumers' choices, which are determined by marketing approaches, such as social network presence, regarding moviegoing (Kumar and Sharma, 2024). This goes a long way to justify the need for the promotion of a movie on a social media platform to achieve better results.

The use of big data in the film industry has gained prominence as directors and companies turn to the analytical tools to make their decisions on which films they should shoot next. The integration of data analytics, sentiment analysis, and consumer behaviour data from sites such as IMDb has revolutionised decision-making on the production, marketing, and distribution of films.

Star power and casting decisions also have the ability to minimise the fame risk associated with film production. McMahon (2023) explores how the star system in Hollywood lessens the risk for repetitive control in casting, stating that studios can refer to IMDB for star trend analysis to inform the casting process. In this way it can help add to the marketability of a film and raise the prospect of its success (McMahon 2023).

Therefore, it critically analysed how the industry relies on data in their decision-making mechanisms to produce films. Thus, it is possible to make particular conclusions as to how filmmakers can use the opportunities of the online ratings, interpret the consumer, and apply analytics to make the right decisions in a high-risk, highly competitive environment.

3.0 Methodology

This analysis follows the use of CRISP-DM (Cross-Industry Standard Process for Data Mining), which lays down a guideline for data-driven projects. The six phases of CRISP-DM are as follows:

3.1 Business Understanding

In order to ensure alignment with organisational goals, this first phase focuses on identifying project objectives and requirements from a business perspective (Tunca 2024). The first step of the process has been described in the introduction part of this proposal when the business issue and goals were outlined. These will be reviewed and adjusted, as the study proceeds, to reflect various objectives of data mining. The key business goals are:

To find out more about the factors behind influencing the high ratings and the votes on the movie dataset. To make company-specific adjustments with relation to regions and genres of movies. To optimise the runtime for achieving better audience engagement.

3.2 Data Understanding

This phase involves gathering initial data, analysing it, and finding quality problems and insights—all of which are essential for making well-informed decisions.

First, a comprehensive analysis of the dataset will be conducted to investigate the structure and contents of the five files offered. The datasets (title.basics, title.ratings, title.akas, title.principals, and name.basics) will be explored to understand their structure, attributes, and quality.

Table 1: Datasets description

File name	Attributes descriptions
Title.akas.csv (contains information for titles)	• titleId (string) - a tconst, an alphanumeric unique identifier of the title. • ordering (integer) – a number to uniquely identify rows for a given titleId. • title (string) – the localised title. • region (string) - the region for this version of the title. • types (array) - Enumerated set of attributes for this alternative title. • isOriginalTitle (boolean) – 0: not original title; 1: original title.
title.basics.csv (contains information for titles)	•tconst (string) - alphanumeric unique identifier of the title. • titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc). • primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release. • startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year • runtimeMinutes – primary runtime of the title, in minutes. • genres (string array) – includes genres associated with the title.
title.principals.csv (contains the principal cast/crew for titles)	• tconst (string) - alphanumeric unique identifier of the title. • ordering (integer) – a number to uniquely identify rows for a given titleId. • nconst (string) - alphanumeric unique identifier of the name/person. • category (string) - the category of job that person was in.
title.ratings.csv (contains the IMDb rating and votes information for titles)	• tconst (string) - alphanumeric unique identifier of the title. • averageRating – weighted average of all the individual user ratings. • numVotes - number of votes the title has received.

name.basics.csv (contains the following information for names)	• nconst (string) - alphanumeric unique identifier of the name/person. • primaryName (string)– name by which the person is most often credited. • birthYear – in YYYY format. • deathYear – in YYYY format if applicable, else . • primaryProfession (array of strings)– the top-3 professions of the person. • knownForTitles (array of tconsts) – titles the person is known for.

This process will include being able to determine the nature of the data, observe if there are any problems with the data, and draw first insights that may shape subsequent processes. In this particular phase of data wrangling, the schema will be reflected upon to ensure primary and foreign relationships, including tconst, and nconst, as well as any missing or invalid values such as \N located in fields such as runtime. The data will then be looked at for a common range including averageRating ranges between 1 and 10 and the startYear varying from early 1900s to the current decades. Some observations to be made include the large cardinality of unique titles and names; data can be of various types: numeric (runtime, averageRating), categorical/ string (genres, region, etc.). There were also some matches, which can feature in title.akas, that there might be different localised names for the movie with the same tconst, therefore, it is crucial to perform data cleaning and normalisation.

3.3 Data Preparation

This phase includes cleaning and transforming data to get the final dataset ready for modelling. The data preparation phase will consist of analysing the files given and merging them using SQL to obtain an extensive dataset. It will include cleaning the data, handling cases with missing values or inconsistent data, and performing CRUD operations. Using SQL to merge datasets in an adequate format to form a complete dataset. For rating information, the merging process will require the joining of title.basics and title.ratings on tconst, joining title.principals and name.basics for the cast and crew and an inner or left join on title.akas to incorporate both region and language. The data pipeline where the extracted data will be in CSV format and will be loaded to a staging area (SQL or Data Frames) and then the raw data will be cleansed, standardised, and joined before loading the cleaned data to an analytical data pipeline for further exploratory analysis and modeling in Python, Spark. CSV file will be converted to JSON to perform queries and explore specific insights dynamically in MongoDB.

3.4 Modelling

Although CRISP-DM typically includes advanced predictive or clustering models, this project focuses on exploratory data analysis or simple statistical analysis.

Aggregations: Group by genre, region, year.

Correlations: Looking at the difference between runtime and the rating number, between votes and the rating number.

Distribution: histograms for rating and box plots for runtime.

Bar charts: Top 10 first-tier genres by mean or sum.

Heatmaps: Interdependencies between numeric characteristics that form the basis for the searches—rating, votes, runtime, year.

Line plots: In the past ten years, the trend of specific genres.

3.5 Evaluation

This phase evaluates the model's effectiveness and makes sure it meets the business's objectives, emphasising the significance of validation (Bemthuis, 2024). The evaluation phase will check whether the models deliver measurable business value based on the flawlessly achieved overall goal. Such cross-validations involve repeated testing of the models with the data or evaluating the sensitivity of the results and making generalisations in the broad perspective of the entertainment industry.

3.6 Deployment

The deployment phase will involve putting the model into practice in a real-world setting, offering the results of this study in formats easy to digest and applicable to decision-makers in entertainment industries. These will include the formulation of strategies in light of the analysis, as well as the initiation of strategies on how the findings can be integrated into business processes.

Applying CRISP-DM, an iterative approach of IMDb dataset examination starting from business and data understanding stages up to the deployment states. This is especially important during the analysis phase of a project, where it is easier to detect new facts or more business questions as they emerge. This method maximises the value of insights for the entertainment sector by coordinating technical work with business goals.

4.0 Implementation and Result

Business question 1: What affects the ratings and votes of users?

1. Using SQL DB browser to join/merge datasets. To find out what affects user ratings (averageRating) and votes (numVotes), the following columns are particularly useful:

title.basics.csv

tconst (this variable is a unique constant of the table and is used for merging)
titleType (movie, tvSeries and so on)
primaryTitle
startYear (potentially relevant: older vs. newer releases)
runtimeMinutes (could covary with ratings)]
genres (genre(s) could affect popularity)

title.ratings.csv

tconst (for joining)
averageRating (the variable concerned with)
numVotes (variable of concern)

```sql
CREATE TABLE ratings_analysis AS
SELECT
    b.tconst,
    b.titleType,
    b.primaryTitle,
    b.startYear,
    b.runtimeMinutes,
    b.genres,
    r.averageRating,
    r.numVotes
FROM "title.basics" AS b
JOIN "title.ratings" AS r
    ON b.tconst = r.tconst;
```

Figure 1: SQL query for joining the datasets

The SQL script joins title.basics with title.ratings by the key tconst; the result is a table that consists of two parts: concise attributes of each title (genres, year, runtime, etc.) and the IMDb rating indicators (averageRating, numVotes). This query uses inner joins, meaning only rows that are present in both tables are included. Regarding the business question of what impacts the user ratings and the votes, this script provides all the necessary attributes necessary for creating an answer, including genre, runtime, release year, rating, and, most importantly, votes, in one consolidated table.

	tconst	titleType	primaryTitle	startYear	runtimeMinutes	genres	averageRating	numVotes
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	tt0000001	short	Carmencita	1894	1	Documentary,Short	5.7	2100
2	tt0000002	short	Le clown et ses chiens	1892	5	Animation,Short	5.6	282
3	tt0000003	short	Poor Pierrot	1892	5	Animation,Comedy,Romance	6.5	2119
4	tt0000004	short	Un bon bock	1892	12	Animation,Short	5.4	182
5	tt0000005	short	Blacksmith Scene	1893	1	Comedy,Short	6.2	2851
6	tt0000006	short	Chinese Opium Den	1894	1	Short	5.0	200
7	tt0000007	short	Corbett and Courtney Before the …	1894	1	Short,Sport	5.4	891
8	tt0000008	short	Edison Kinetoscopic Record of a …	1894	1	Documentary,Short	5.4	2248
9	tt0000009	movie	Miss Jerry	1894	45	Romance	5.4	215
10	tt0000010	short	Leaving the Factory	1895	1	Documentary,Short	6.8	7753
11	tt0000011	short	Akrobatisches Potpourri	1895	1	Documentary,Short	5.2	403
12	tt0000012	short	The Arrival of a Train	1896	1	Documentary,Short	7.4	13178
13	tt0000013	short	The Photographical Congress Arrives …	1895	1	Documentary,Short	5.7	2020
14	tt0000014	short	The Waterer Watered	1895	1	Comedy,Short	7.1	5995
15	tt0000015	short	Around a Cabin	1894	2	Animation,Short	6.1	1231
16	tt0000016	short	Boat Leaving the Port	1895	1	Documentary,Short	5.9	1622
17	tt0000017	short	Italienischer Bauerntanz	1895	1	Documentary,Short	4.6	363
18	tt0000018	short	Das boxende Känguruh	1895	1	Short	5.2	644
19	tt0000019	short	The Clown Barber	1898	\N	Comedy,Short	5.1	32
20	tt0000020	short	The Derby 1895	1895	1	Documentary,Short,Sport	4.7	402
21	tt0000022	short	Blacksmith Scene	1895	1	Documentary,Short	5.1	1176
22	tt0000023	short	The Sea	1895	1	Documentary,Short	5.7	1550

1 - 23 of 472,747 Go to: 1

Figure 2: New table rating_analysis

The outcome in the new table, ratings_analysis, has one row per title for titles found in both title.basics and title.ratings with the genre, runtime, and year of the work and average rating and total voting numbers.

2. Using Python to perform exploratory data analysis (EDA) and visualisation

After creating the above table in SQL, export and read the tables to Python (preferably using pandas).

```
[58] import pandas as pd

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

```
[59] df.head(10)
```

Show hidden output

```
[60] df.tail(10)
```

Show hidden output

```
[61] df.shape

(472747, 8)
```

Figure 3: Loading dataset into pandas

Reading the csv file into a dataframe. Then checked the first and last 10 rows of the dataset. To display the number of rows and columns in the dataset.

```
[74] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 472747 entries, 0 to 472746
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   tconst          472747 non-null  object 
 1   titleType       472747 non-null  object 
 2   primaryTitle    472747 non-null  object 
 3   startYear       472722 non-null  float64
 4   runtimeMinutes  472747 non-null  object 
 5   genres          472747 non-null  object 
 6   averageRating   472747 non-null  float64
 7   numVotes        472747 non-null  int64  
dtypes: float64(2), int64(1), object(5)
memory usage: 28.9+ MB
```

Figure 4: To check the datatypes

```
[76] df.isna().sum()
```

	0
tconst	0
titleType	0
primaryTitle	0
startYear	25
runtimeMinutes	0
genres	0
averageRating	0
numVotes	0

dtype: int64

Figure 5: To check missing values

There are 25 missing values in the rating_analysis dataset.

```
[ ] df1 = df.copy()

[55] df1['startYear']=df['startYear'].fillna(0)

[56] df1['startYear'].isna().sum()

0

[57] df.shape

(472747, 8)

[44]
print(df.isna().sum())

tconst            0
titleType         0
primaryTitle      0
startYear         0
runtimeMinutes    0
genres            0
averageRating     0
numVotes          0
dtype: int64
```

Figure 6: Handling missing values

Here, handling missing values was done by replacing them with 0.

```python
import matplotlib.pyplot as plt
import seaborn as sns
num_cols = ['runtimeMinutes', 'averageRating', 'numVotes']
df1[num_cols].boxplot()
plt.title("Box Plots of Numeric Columns")
plt.show()
```

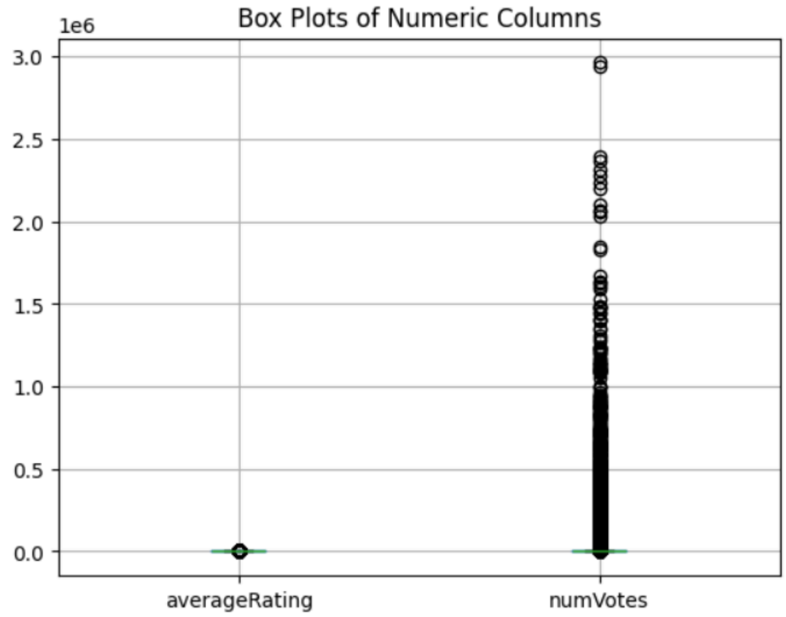


Figure 7: checking for outliers using a boxplot.

```python
df1['startYear'] = pd.to_numeric(df1['startYear'], errors='coerce')
df1['runtimeMinutes'] = pd.to_numeric(df['runtimeMinutes'], errors='coerce')
df1.dropna(subset=['averageRating', 'numVotes'], inplace=True)
```

```python
[97] print(df1[['averageRating','numVotes','runtimeMinutes','startYear']].describe())
```

```
       averageRating      numVotes  runtimeMinutes      startYear
count  472747.000000  4.727470e+05   382412.000000  472722.000000
mean        6.703200  1.634125e+03       63.001425    1983.966587
std         1.302287  2.531435e+04       47.650899      22.563132
min         1.000000  5.000000e+00        0.000000    1888.000000
25%         5.900000  1.700000e+01       29.000000    1971.000000
50%         6.900000  4.300000e+01       60.000000    1991.000000
75%         7.600000  1.550000e+02       90.000000    2002.000000
max        10.000000  2.963270e+06     5220.000000    2024.000000
```

```python
[98] import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(df1['averageRating'], bins=20, kde=True)
plt.title("Distribution of IMDb Ratings")
plt.show()
```

Figure 8: Converting datatype

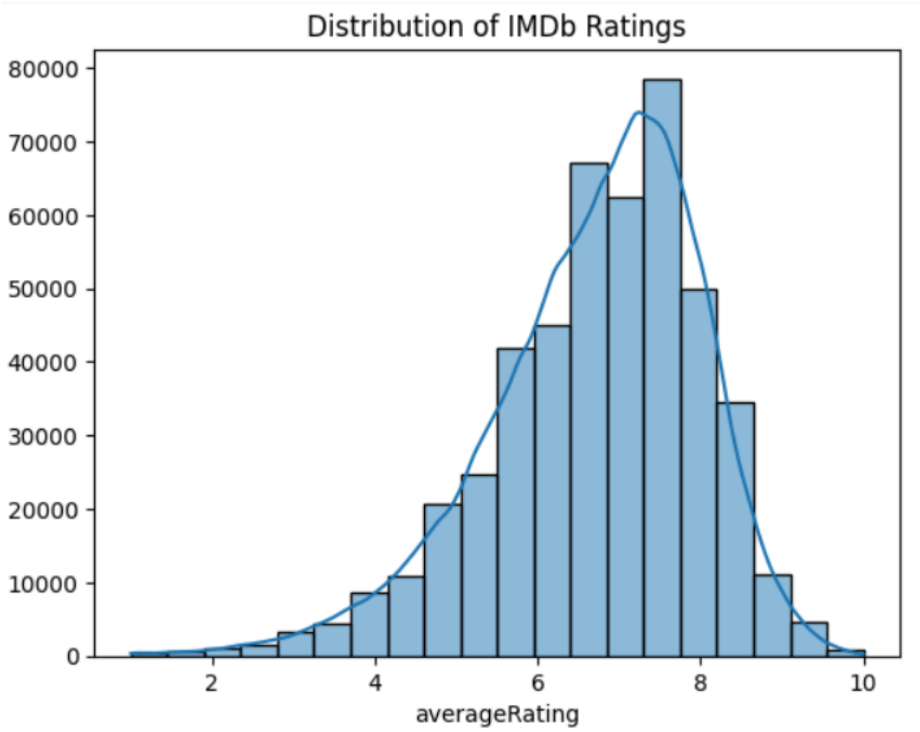


Figure 9: Distribution of ratings

```
[19] corr_matrix = df1[['averageRating','numVotes','runtimeMinutes','startYear']].corr()
     print(corr_matrix)
     sns.heatmap(corr_matrix, annot=True, cmap='Blues')
     plt.show()
```

```
                averageRating  numVotes  runtimeMinutes  startYear
averageRating        1.000000  0.027377       -0.167250   0.151348
numVotes             0.027377  1.000000        0.065668   0.029338
runtimeMinutes      -0.167250  0.065668        1.000000   0.019097
startYear            0.151348  0.029338        0.019097   1.000000
```

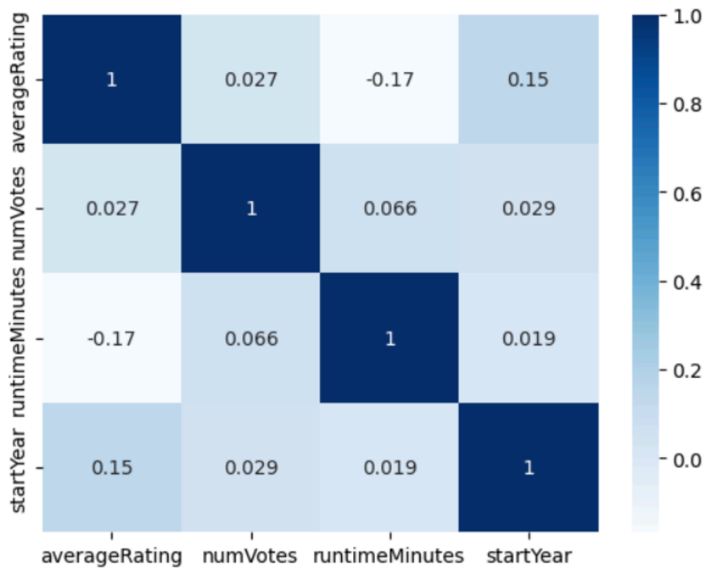


Figure 10: Correlation with numeric feature

```python
df1['primaryGenre'] = df1['genres'].apply(lambda x: x.split(',')[0] if pd.notnull(x) else x)
genre_stats = df1.groupby('primaryGenre').agg({
    'averageRating': 'mean',
    'numVotes': 'mean',
    'tconst': 'count'
}).rename(columns={'tconst': 'titleCount'}).reset_index()

genre_stats.sort_values('averageRating', ascending=False, inplace=True)
print(genre_stats.head(10))
```

```
   primaryGenre  averageRating     numVotes  titleCount
27      Western       7.176482   137.740480        9244
15        Music       7.101094    69.377863        3930
7   Documentary       6.991876   169.294444       33137
6         Crime       6.971794  2185.476352       36219
23        Sport       6.928597    98.887067        1098
13      History       6.901778   109.153333         450
4     Biography       6.878353  4860.516730        6814
9        Family       6.801551    99.202742        6126
2     Adventure       6.788089  3294.444614       26144
5        Comedy       6.760726  1284.471209      122818
```

Figure 11: Exploring how genres affect ratings and votes

```python
plt.figure(figsize=(10, 5))
sns.barplot(x='averageRating', y='primaryGenre', data=genre_stats, order=genre_stats['primaryGenre'])
plt.title("Average IMDb Rating by Genre")
plt.xlabel("Average Rating")
plt.ylabel("Genre")
plt.show()
```

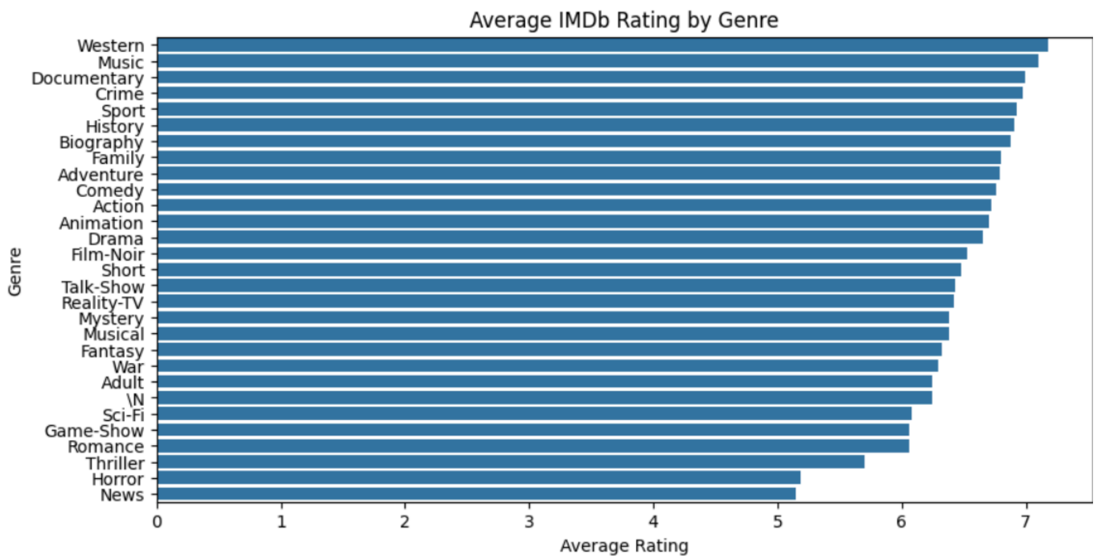


Figure 12: Genre vs. average ratings

Interpreting the Visuals:

Correlation Heatmap: A positive coefficient of runtimeMinutes and averageRating indicates the fact that long movies will make more sense (though there exists a weak positive correlation) to the viewers as they give more time to understand a movie.

Genre Bar Plot: To check, for instance, if music or western has greater mean ratings or more people voted, it reveals the preferences of users.

Business question 2: Which regions have the highest concentration of popular movies based on ratings and votes?

1. In this task, using the given dataset (mainly title.akas for the region and title.ratings for popularity). Identify which areas of the world are most closely associated with popular films, where ‘popularity’ could be measured by: High averageRating, High numVotes

```

CREATE TABLE movie_ratings AS
SELECT
    b.tconst,
    b.primaryTitle,
    b.startYear,
    r.averageRating,
    r.numVotes
FROM "title.basics" AS b
JOIN "title.ratings" AS r
    ON b.tconst = r.tconst
WHERE b.titleType = 'movie';

```

```sql
CREATE TABLE movie_ratings_region AS
SELECT
    mr.*,
    a.region
FROM movie_ratings AS mr
JOIN "title.akas" AS a
    ON mr.tconst = a.titleId;
```

Figure 13: Combined dataset with movie info

movie_ratings: Retrieves movies only by having titleType = ‘movie’ and returns the movie’s rating, votes, and basic details like title, year, etc.

movie_ratings_region: Copies the region from title.akas. This is because title.akas can be many localised versions of the title for a single movie (title by the region, by language, etc.), hence a single movie may have many different titles if it is distributed or known in different regions.

The next step is identifying the highest concentration of popular movies.

```sql
CREATE TABLE popular_movies AS
SELECT
    tconst,
    primaryTitle,
    startYear,
    averageRating,
    numVotes,
    region
FROM movie_ratings_region
WHERE averageRating >= 7.5
  AND numVotes >= 50000;
```

```sql
SELECT
    region,
    COUNT(*) AS popular_movie_count
FROM popular_movies
GROUP BY region
ORDER BY popular_movie_count DESC;
```

Figure 14: Summarising popularity by region

This query starts with the creation of a filtered table named popular_movies that contains only those movie titles that fit or surpass the entered “popular” parameters. Post that, it proceeds to count the number of popular movies by each region. That is sorted in descending order and shows which region code corresponds to the most number of popular titles.

	region	popular_movie_count
1	US	1050
2	FI	904
3	IN	902
4	CA	862
5	GR	754
6	JP	751
7	ES	736
8	GB	498
9	MX	497
10	FR	497
11	BR	478
12	IT	477
13	PL	464
14	PT	455
15	HU	454
16	SE	453
17	DE	451
18	UA	449
19	\N	446
20	AR	437
21	RO	435

Figure 15: Outcome of movie popularity by region

2. Further analysis and visualisation in Python

Once this has been created or exported, other processes or graphical representations can be performed (for example, bar graphs to determine how many of the popular movies each region has).

```
total_by_region = (
    df_region
    .groupby('region')['tconst']
    .nunique()
    .reset_index(name='total_titles')
)

print(total_by_region.head(10))
```

```
  region  total_titles
0     AE           830
1     AF             6
2     AL           105
3     AM            33
4     AO             4
5     AR         11977
6     AT          7876
7     AU         15765
8     AZ           326
9     BA            82
```

Figure 16: Total titles by region

df_region = df_region.groupby('region') ['tconst'] Takes the DataFrame df_region, applies logic to the rows of this DataFrame to group them by their 'region,' and finally selects the 'tconst' within each group.

Basically, it forms a grouping structure in which each of the regions is affiliated with a subset of rows.

The .nunique() function is used to determine tconst count by region. To make the function run as required. This makes sure that the titles that come more than once with the same region are not counted many times, which will provide a unique title count to the region. Grouped data is being reset into a DataFrame and renames the aggregated column to 'total_titles.'. The resulting DataFrame has two columns: region and total_titles. print(total_by_region.nlargest(10, total)). Displays the first 10 rows of the DataFrame to see which regions appear and how many total or unique titles each region possesses.

```
[ ] pop_by_region = (
        df_movies
        .groupby('region')['tconst']
        .nunique()
        .reset_index(name='popular_titles')
    )

    print(pop_by_region.head(10))
```

```
  region  popular_titles
0     AE             231
1     AL              17
2     AM               2
3     AR             403
4     AT             222
5     AU             412
6     AZ             160
7     BA               4
8     BD               7
9     BE             163
```

Figure 17: Pop by region

With this data, the first step of the analysis was to divide the df_movies data frame by the region column and then return the tconst series for each region. Essentially the same logic of grouping; however, here the DataFrame used is df_movies. The DataFrame is constructed only of popular titles. Partitioned based on the 'region' dimension and the 'tconst' attribute.

.nunique() The result of counting the tconst (title IDs) frequency per region is the estimated occurrence of unique titles present in this "popular movies" data set.

.reset_index(name='popular_titles') Sets the grouping result back to standard DataFrame form and renames the new column as popular_titles. So the DataFrame that get here has region and the count of the number of distinct popular titles within a region.

print(pop_by_region.head(10)) displays the DataFrame to show only the first 10 rows; this will help to recognise which regions exist and how many of them have "popular" titles.

```
region_stats = total_by_region.merge(pop_by_region, on='region', how='left')
region_stats['popular_titles'] = region_stats['popular_titles'].fillna(0)

region_stats['popularity_ratio'] = (
    region_stats['popular_titles'] / region_stats['total_titles']
)

region_stats = region_stats.sort_values('popularity_ratio', ascending=False)
region_stats.head(10)
```

	region	total_titles	popular_titles	popularity_ratio
70	KG	5	4.0	0.800000
112	QA	6	4.0	0.666667
101	NP	3	2.0	0.666667
136	UZ	359	182.0	0.506964
126	TJ	6	3.0	0.500000
129	TO	2	1.0	0.500000
74	KZ	332	163.0	0.490964
8	AZ	326	160.0	0.490798
53	GT	14	6.0	0.428571
21	CG	5	2.0	0.400000

Figure 18: Merging and calculating popularity ratio

To observe a more detailed picture of distribution, these two aggregations are joined side by side to examine not only the number of popular movies for each region but also the relative share.

```python
import matplotlib.pyplot as plt
top10 = region_stats.head(10)
plt.figure(figsize=(8,4))
plt.bar(top10['region'], top10['popularity_ratio'], color='skyblue')
plt.title("Top 10 Regions by Popularity Ratio")
plt.xlabel("Region")
plt.ylabel("Popularity Ratio")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

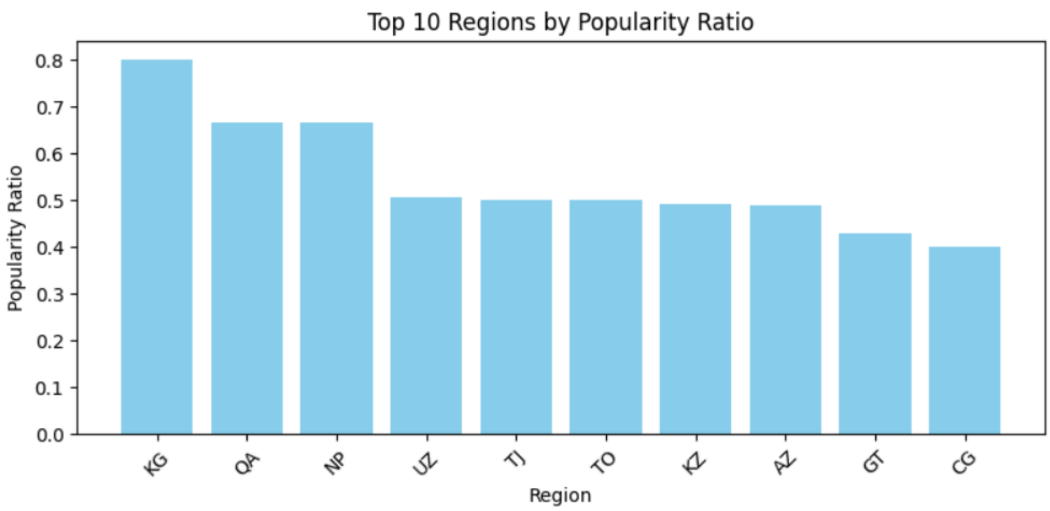


Figure 19: Barplot of the 10 regions by popularity ratio

Business question 3: What are the roles (actors, directors, or crew members) starred in high-rated movies?

```sql
CREATE TABLE high_rated_movies_cast_test AS
SELECT
  b.tconst,
  b.primaryTitle,
  r.averageRating,
  p.nconst,
  p.category,
  n.primaryName
FROM "title.basics" AS b
JOIN "title.ratings" AS r ON b.tconst = r.tconst
JOIN "title.principals" AS p ON b.tconst = p.tconst
JOIN "name.basics" AS n ON p.nconst = n.nconst;
```

Figure 20: Joining tables in SQL

This query names a new table as high_rated_movies_cast that filters all movies with their averageRating of 8.0 and above. It contains the individual name (tconst, primaryName), type of the person (actor, director, etc.), and the movie (primaryTitle, averageRating).

```python
import pandas as pd

df = pd.read_csv("high_rated_movies_cast_test.csv")

person_counts = df.groupby(['nconst', 'primaryName', 'category'])['tconst'].nunique()
person_counts = person_counts.reset_index(name='num_high_rated_movies')

top_people = person_counts.sort_values('num_high_rated_movies', ascending=False)
print(top_people.head(20))
```

```
       nconst            primaryName  category  num_high_rated_movies
0   nm0000356          Sybil Danning   actress                     19
1   nm0000434            Mark Hamill     actor                      7
24  nm0958249            Karol Beffa  composer                      1
23  nm0953125        Harrison Zanuck     actor                      1
22  nm0923928         Nicola Wheeler   actress                      1
21  nm0901552      Ekaterina Volkova   actress                      1
20  nm0780435          Ryan Seacrest      self                      1
19  nm0747328           Rosita Royce      self                      1
18  nm0735348         Ivan Rodriguez     actor                      1
17  nm0607510          Joshua Morrow     actor                      1
16  nm0601168         Elizabeth Moore   actress                      1
15  nm0554788            Óscar Martín     actor                      1
14  nm0549460  Publius Vergilius Maro    writer                      1
13  nm0522550       Jacqueline Lovell   actress                      1
12  nm0440788            Katyana Kass   actress                      1
11  nm0404895            Simone Hyams   actress                      1
10  nm0374416          Amelia Heinle   actress                      1
9   nm0346533             Cary Guffey     actor                      1
8   nm0339998          Simon Gregson     actor                      1
7   nm0295568          Annette Frier   actress                      1
```

Figure 21: Exploring data in Python

groupby('nconst, 'primaryName, 'category') divides data by the certain person and the job title. The .nunique() function on 'tconst' counts how many high-rated movies each person contributed to. The sort_values call brings towards the top the people who feature in most of the movies that have a rating of 8.0 or higher.

Business question 4: Which runtime range exhibits the highest average ratings?

This approach includes defining runtime ranges. Finding the average of the values and then averaging those averages by each range. Then, finding out which range has the highest average rating.

```
CREATE TABLE basics_ratings AS
SELECT
    b.tconst,
    b.runtimeMinutes,
    r.averageRating
FROM "title.basics" AS b
JOIN "title.ratings" AS r
    ON b.tconst = r.tconst;
```

Figure 22: Creating new table name "basic_ratings"

```
SELECT
    CASE
        WHEN runtimeMinutes < 30  THEN '0-30'
        WHEN runtimeMinutes < 61  THEN '31-60'
        WHEN runtimeMinutes < 121 THEN '61-120'
        WHEN runtimeMinutes < 181 THEN '121-180'
        ELSE '180+'
    END AS runtimeRange,
    AVG(averageRating) AS avg_rating,
    COUNT(*) AS titles_count
FROM basics_ratings
WHERE runtimeMinutes IS NOT NULL
GROUP BY runtimeRange
ORDER BY avg_rating DESC;
```

	runtimeRange	avg_rating	titles_count
1	31-60	7.20710279554816	114110
2	0-30	6.92550562969141	95920
3	180+	6.87619483630556	93925
4	121-180	6.69874254186181	15587
5	61-120	6.08309585196306	153205

Figure 23: Creating ranges with case.

CASE: classifies based on runtimeMinutes into buckets: 0-30 min, 31-60 min, 61-120 min, 121-180 min, 180+ min.

AVG(averageRating): Calculates the average rating for each range of runtime measurement. COUNT(*): depicts the number of titles that belong to each range.

ORDER BY avg_rating DESC: The first one is the range that has the highest time and the highest average rating.

Performing exploratory analysis in Python (Pandas) and answering this business question:

```
[45] import pandas as pd

     df11 = "/content/basics_ratings.csv"
     data1 = pd.read_csv(df11)
```

```
[37] data1.head(10)
```

	tconst	runtimeMinutes	averageRating
0	tt0000001	1	5.7
1	tt0000002	5	5.6
2	tt0000003	5	6.5
3	tt0000004	12	5.4
4	tt0000005	1	6.2
5	tt0000006	1	5.0
6	tt0000007	1	5.4
7	tt0000008	1	5.4
8	tt0000009	45	5.4
9	tt0000010	1	6.8

```
data1.tail(10)
```

Show hidden output

Figure 24: Performing EDA

```
[47] df['runtimeMinutes'] = pd.to_numeric(df['runtimeMinutes'], errors='coerce')
```

```
[48] print(df['runtimeMinutes'].dtype)
     df['runtimeMinutes'].isna().sum()  # count how many became NaN
```

```
float64
90335
```

```
df.dropna(subset=['runtimeMinutes'], inplace=True)
```

Figure 25: Converting String to Numeric

Used pd.to_numeric to convert strings to numeric; upon doing that, invalid entries became NaN. errors='coerce' replaced any non-convertible values ('\\N') with NaN instead of giving an error.

```
[50] import pandas as pd

     bins = [0, 30, 60, 120, 180, 9999]
     labels = ["0-30", "31-60", "61-120", "121-180", "180+"]

     df['runtimeRange'] = pd.cut(
         df['runtimeMinutes'],
         bins=bins,
         labels=labels,
         include_lowest=True
     )
```

```
range_stats = df.groupby('runtimeRange').agg({
    'averageRating': 'mean',
    'runtimeMinutes': 'count'  # optional, to see how many titles
}).rename(columns={'runtimeMinutes': 'titles_count'})

range_stats.sort_values('averageRating', ascending=False, inplace=True)
print(range_stats)
```

```
              averageRating  titles_count
runtimeRange
31-60              7.197245         80989
180+               7.053120          3590
0-30               7.003970        129041
121-180            6.698743         15587
61-120             6.083096        153205
```

Figure 26: Grouped and aggregated

Using pd.cut which is a utility that is used to bins of numeric data. This forms a new column in df called runtimeRange having categories such as ‘0-30’, ‘31-60’.

groupby('runtimeRange'): After that, according to the new bucket, groups rows.

mean of 'averageRating': Gives the average rating by range. count of 'runtimeMinutes' (renamed to 'titles_count'`): Others demonstrate how many titles belong to each category. It is also for the same reason that it is sorted by 'averageRating' in descending order so as to be able to see which of the range is highest. The highest avg_rating row represents the amount of time in minutes the average user rating is highest.

```
[52] import matplotlib.pyplot as plt

     plt.figure(figsize=(6,4))
     plt.bar(range_stats.index, range_stats['averageRating'], color='skyblue')
     plt.title("Average Ratings by Runtime Range")
     plt.xlabel("Runtime Range (minutes)")
     plt.ylabel("Average Rating")
     plt.show()
```

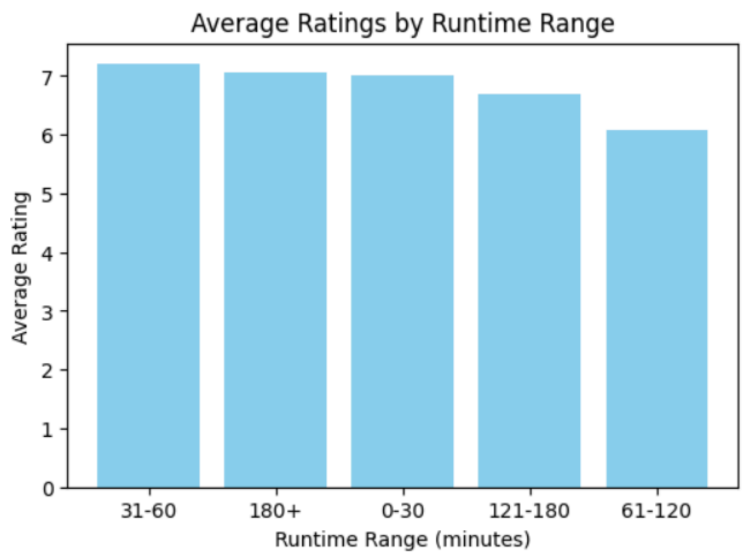


Figure 27: Runtime range vs. average ratings

This provides a good visual on how the average rating of each range stands.

Business question 5: Which movies released across all years have the highest audience ratings?

To solve this question, first joining title.ratings table which contains ratings information, with title.basics table using tconst as the common key.

```sql

CREATE TABLE movie_ratings AS
SELECT
    b.tconst,
    b.primaryTitle,
    b.startYear,
    r.averageRating,
    r.numVotes
FROM "title.basics" AS b
JOIN "title.ratings" AS r
    ON b.tconst = r.tconst
WHERE b.titleType = 'movie';

```

Figure 28: New table called movie ratings.

+ Code + Text

```python
[2] import pandas as pd
    import json

    df9 = pd.read_csv("/content/movie_ratings.csv")

    json_movies= "/content/json_movies.json"
    df9.to_json(json_movies, orient="records", indent=1)
```

Figure 29: CSV to JSON format

After creating the new table named "movie_ratings" converted to JSON format in Python to perform queries in mongoDB.

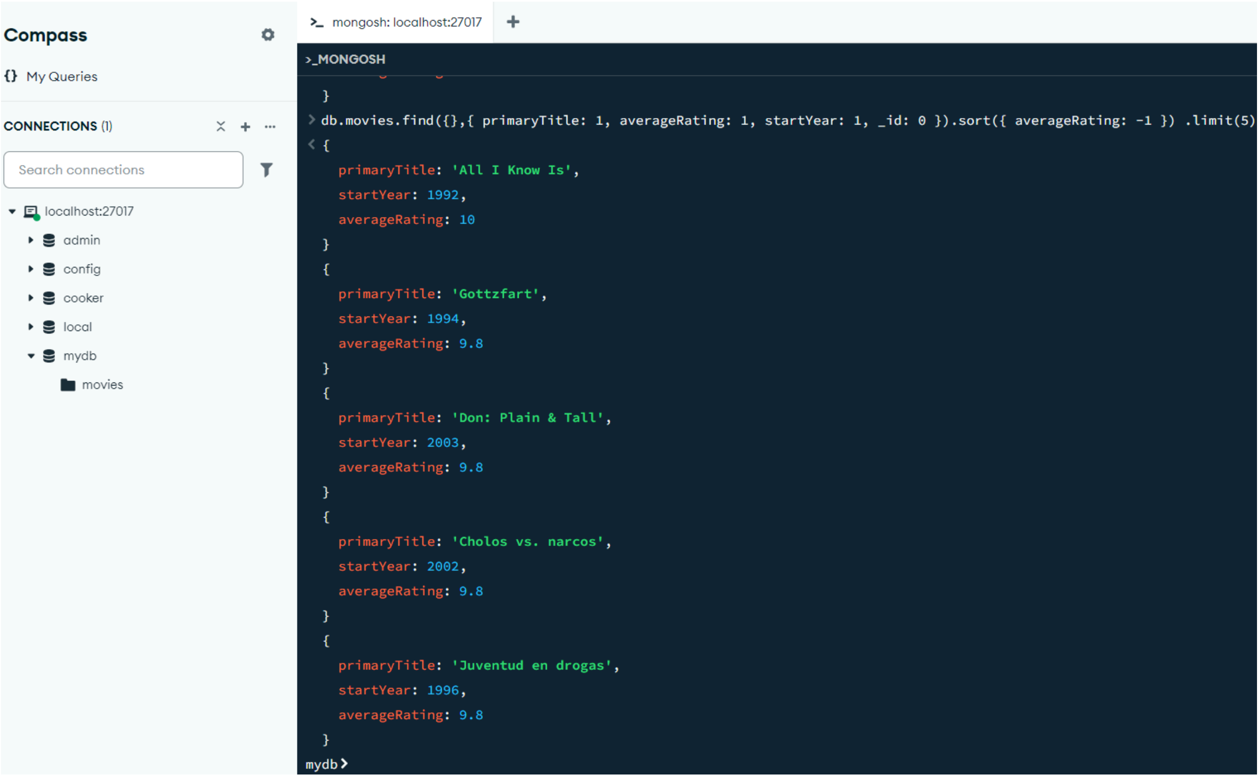


Figure 30: Top 5 movies across all years.

The query db.movies.find({}, { primaryTitle: 1, averageRating: 1, startYear: 1, _id: 0 – this query means that all documents are selected from the movies collection without any condition _id field, only the fields such as primary-Title which is equal to the movie title, averageRating which refers to the average rating of a given movie, and startYear that refers to the year in which the movie was produced, should be included in the result. The .sort({ averageRating: It has added an order clause limiting the results by averageRating and sorting it [-1] in the manner of creating a descending order list so if a movie has a higher rating than another it will be displayed first. Lastly, .limit(5) when added at the end of a query, brings only the five topmost results when a query has been sorted.

Business question 6: What are the most common genres for high-rated movies?

```
[1] from pyspark.sql import SparkSession
    from pyspark.sql.functions import col, explode, split, desc

    spark = SparkSession.builder.appName("HighRatedGenres").getOrCreate()

    title_basics = spark.read.csv("title.basics.csv", header=True, inferSchema=True)
    title_ratings = spark.read.csv("title.ratings.csv", header=True, inferSchema=True)

    joined_df = title_basics.join(title_ratings, on="tconst", how="inner")

    high_rated_df = joined_df.filter(col("averageRating") >= 8.0)


    genres_exploded = high_rated_df.withColumn(
        "genre",
        explode(split(col("genres"), ","))
    )

    genre_counts = (
        genres_exploded.groupBy("genre")
        .count()
        .orderBy(desc("count"))
    )

    genre_counts.show(10, truncate=False)
```

Figure 31: High-rated movie genre

Created SparkSession

Function to generate the primary context to use PySpark.

Loading Data: title.basics is one of them, and it generally contains columns such as tconst, titleType, primaryTitle, runtimeMinutes, and genres.

Both DataFrames are merged based on matching the keys tconst.

If rows do not match in either, the default how=”inner” retains only these matching in both DataFrames. Afterward, using averageRating >= 8.0 helped pick only the titles with the highest rating. Adjust this cutoff as needed.

Split and Explode Many IMDb titles of the ‘Genre’ have two or more genres but are specified in one row, for instance, Action or thriller. split(“,”) converts the string such as “Action, Thriller” into an array, containing “Action” and “Thriller.”.

explode(...) creates a new row for every item in that array. Hence, when the number of genres is more, the data frame with columns with movie names and counts for each genre becomes the data frame with the same name but with many rows, each having one genre only.

Group & Count Listed by genres, summed up how many titles belong to the mentioned genre. A movie can belong to more than one genre; in this case, every genre is counted. Used. orderBy(desc("count")) it will arrange words from most frequently to least found. Showed the 10 genres based on the frequency for all high-ratedmovies (8.0 and above in this case).

```
+-----------+-----+
|genre      |count|
+-----------+-----+
|Drama      |28763|
|Comedy     |24159|
|Crime      |9905 |
|Documentary|9223 |
|Family     |8666 |
|Action     |8400 |
|Adventure  |8389 |
|Animation  |5941 |
|Short      |5871 |
|Romance    |5077 |
+-----------+-----+
only showing top 10 rows
```

Business question 7: How does each genre's level of popularity evolve throughout time?

```sql
CREATE TABLE genre_ratings AS
SELECT
    b.tconst,
    b.genres,
    b.startYear,
    r.averageRating,
    r.numVotes
FROM "title.basics" AS b
JOIN "title.ratings" AS r ON b.tconst = r.tconst
```

Figure 32: Joining in SQL

SELECT: Selected Picks columns required for identifying trends in the genre over time. b.genres (which is usually a string of the known genres separated by a comma and arrow, for instance, (Action, Thriller). JOIN: Having an inner join on title.basics (b) with the title.ratings (r) with tconst.

Table: genre_ratings

	tconst	genres	startYear	averageRating	numVotes
1	tt0000001	Documentary,Short	1894	5.7	2100
2	tt0000002	Animation,Short	1892	5.6	282
3	tt0000003	Animation,Comedy,Romance	1892	6.5	2119
4	tt0000004	Animation,Short	1892	5.4	182
5	tt0000005	Comedy,Short	1893	6.2	2851
6	tt0000006	Short	1894	5.0	200
7	tt0000007	Short,Sport	1894	5.4	891
8	tt0000008	Documentary,Short	1894	5.4	2248
9	tt0000009	Romance	1894	5.4	215
10	tt0000010	Documentary,Short	1895	6.8	7753
11	tt0000011	Documentary,Short	1895	5.2	403
12	tt0000012	Documentary,Short	1896	7.4	13178
13	tt0000013	Documentary,Short	1895	5.7	2020
14	tt0000014	Comedy,Short	1895	7.1	5995
15	tt0000015	Animation,Short	1894	6.1	1231
16	tt0000016	Documentary,Short	1895	5.9	1622
17	tt0000017	Documentary,Short	1895	4.6	363
18	tt0000018	Short	1895	5.2	644
19	tt0000019	Comedy,Short	1898	5.1	32
20	tt0000020	Documentary,Short,Sport	1895	4.7	402
21	tt0000022	Documentary,Short	1895	5.1	1176
22	tt0000023	Documentary,Short	1895	5.7	1550
23	tt0000024	News,Short	1895	3.8	149
24	tt0000025	News,Short,Sport	1896	3.8	47

1 - 24 of 472,747

Figure 33: New table genre_ratings

```
[2] import pandas as pd

    df = pd.read_csv("/content/genre_ratings.csv")
```

```
[3] df['startYear'] = pd.to_numeric(df['startYear'], errors='coerce')
    df['averageRating'] = pd.to_numeric(df['averageRating'], errors='coerce')
    df['numVotes'] = pd.to_numeric(df['numVotes'], errors='coerce')
```

```
[4] df['genres_split'] = df['genres'].str.split(',')
    df_exploded = df.explode('genres_split').rename(columns={'genres_split': 'genre'})
```

```
genre_year_stats = (
    df_exploded
    .groupby(['startYear', 'genre'])
    .agg(
        avg_rating=('averageRating', 'mean'),
        avg_votes=('numVotes', 'mean'),
        count_titles=('tconst', 'nunique')
    )
    .reset_index()
)
```

Figure 34: Analysing in Python

Loaded the dataset into pandas and converted columns to numeric. Genres can be multiple in one row; for example, in IMDb, it is written as Action, Thriller. To be able to track the yearly popularity of each genre separately, it is necessary to split and explode. After splitting genres, grouped by year and genre to compute the average rating (mean of numVotes) per (startYear, genre).

```
[6]
top_genre_per_year = genre_year_stats.loc[
    genre_year_stats.groupby('startYear')['avg_rating'].idxmax()
]
```

```
import matplotlib.pyplot as plt

some_genres = ['Drama', 'Comedy', 'Action']
df_subset = genre_year_stats[genre_year_stats['genre'].isin(some_genres)]

for genre in some_genres:
    sub = df_subset[df_subset['genre'] == genre]
    plt.plot(sub['startYear'], sub['avg_rating'], label=genre)

plt.xlabel("Year")
plt.ylabel("Average Rating")
plt.title("Genre Popularity Over Time")
plt.legend()
plt.show()
```

Figure 35: Visualising in Python

Analysed trends over time to see which genre has the highest average rating each year.

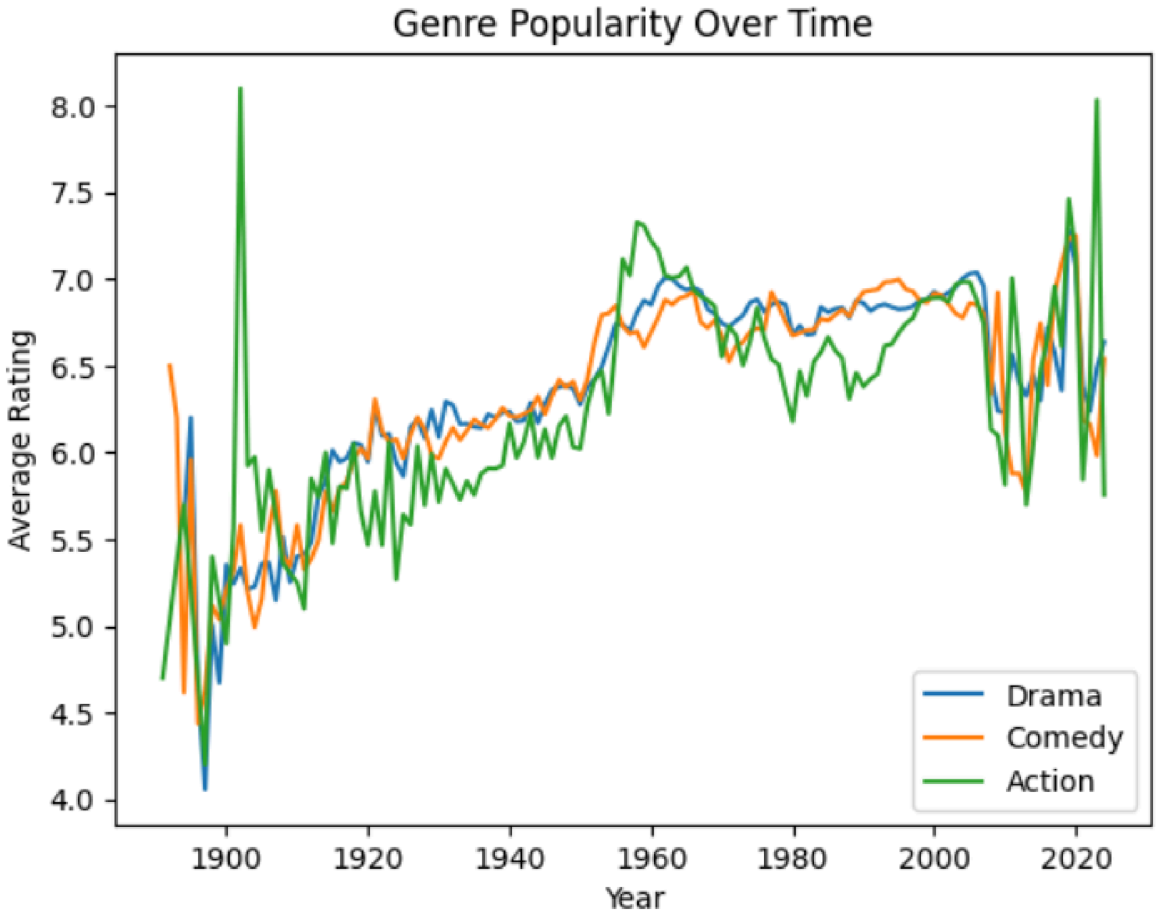


Figure 36: Genre popularity in average rating vs. year

Business question 8: What is the primarytitle with the maximum runtime minutes?

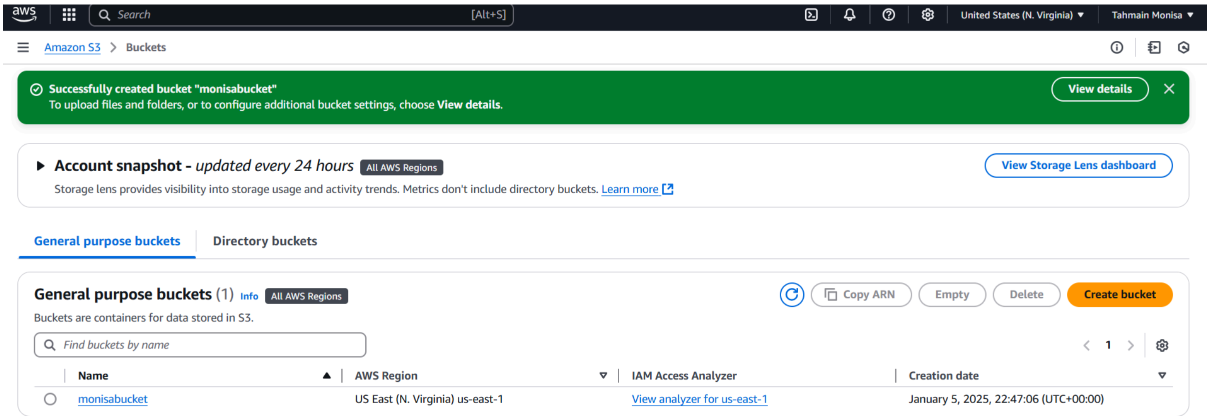


Figure 37: created S3 bucket

According to the proposed pipeline, Amazon S3 forms the general data storage tier of the system. Source data comes from several sources and is stored in S3 in its raw form, including log data, CSV, JSON, or Parquet. S3 offering can be considered as a reliable, cost-efficient, and highly available option to meet data storage and enable easy access for downstream processing. It can be divided into a well-defined structure where it can be filed or named with subgroups or prefixes for easy presentation.

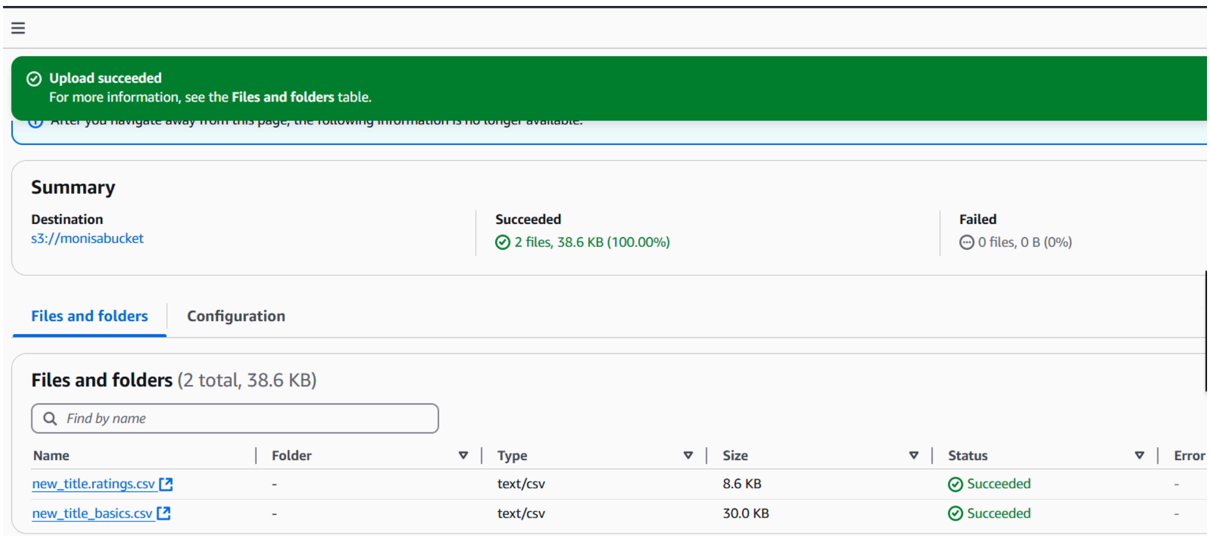


Figure 38: Uploaded datasets

To answer this question, two separate datasets containing 500 rows have been uploaded to perform an AWS Athena query.

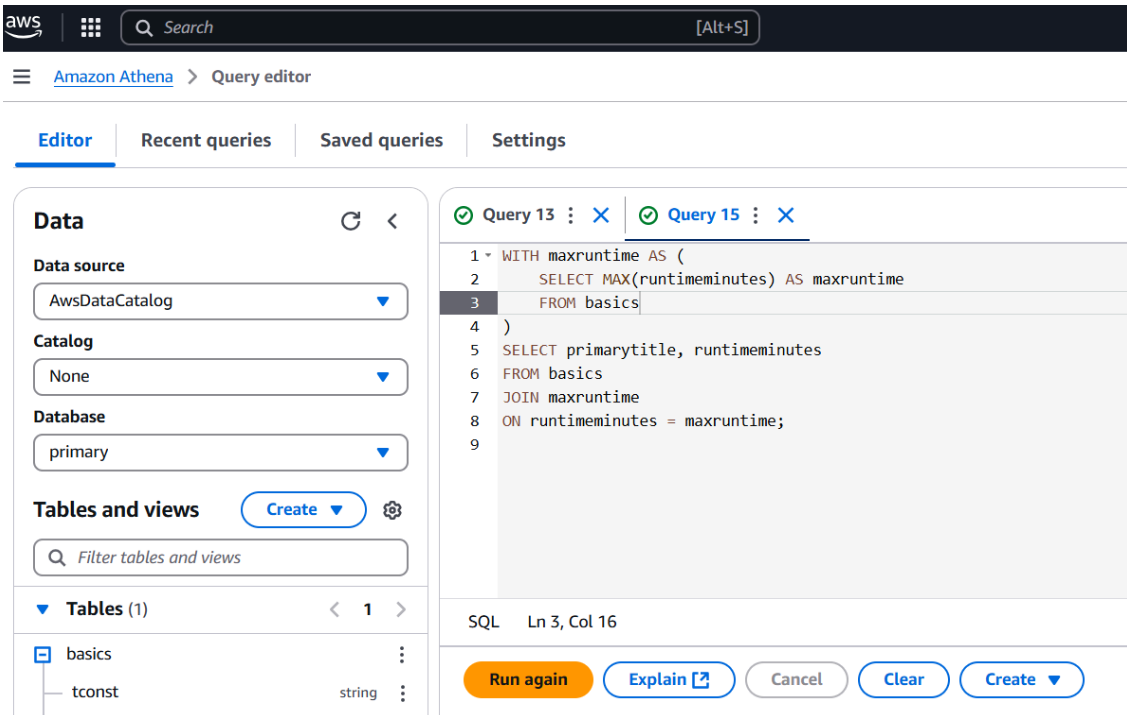


Figure 39: Athena query performed in table named basics

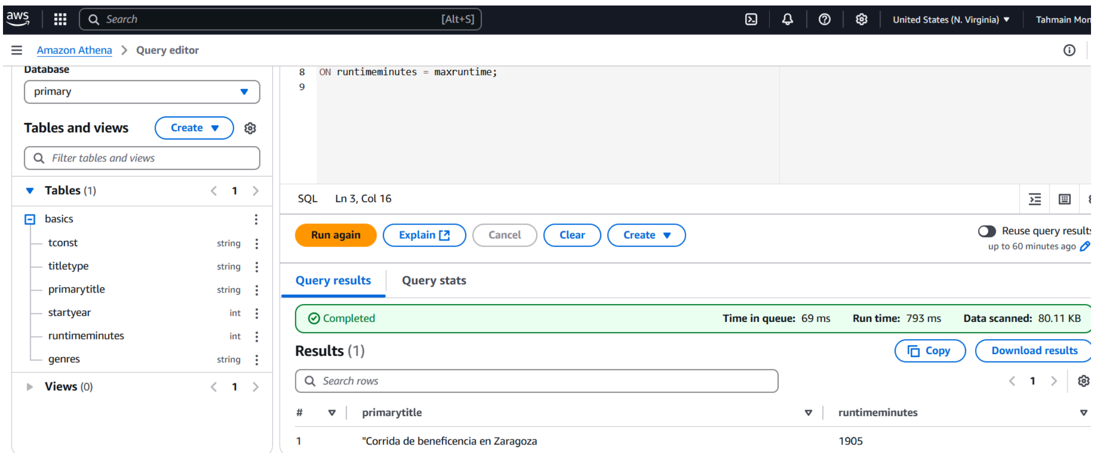


Figure 40: Outcome primarytitle with runtimeMinutes.

5.0 Discussion

5.1 Derived Insights

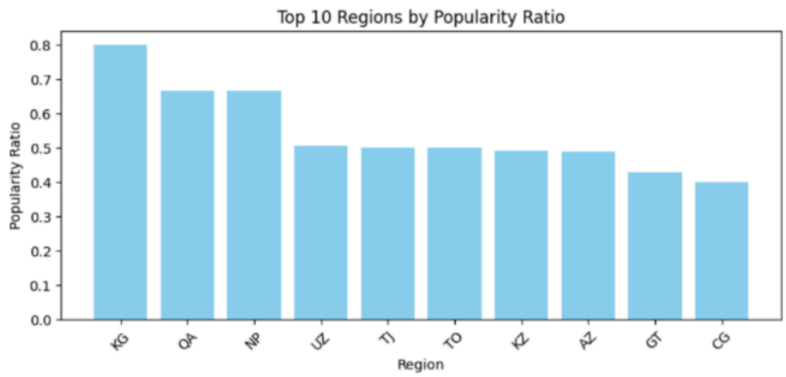
1.

```
df1['primaryGenre'] = df1['genres'].apply(lambda x: x.split(',')[0] if pd.notnull(x) else x)
genre_stats = df1.groupby('primaryGenre').agg({
    'averageRating': 'mean',
    'numVotes': 'mean',
    'tconst': 'count'
}).rename(columns={'tconst': 'titleCount'}).reset_index()

genre_stats.sort_values('averageRating', ascending=False, inplace=True)
print(genre_stats.head(10))
```

```
   primaryGenre  averageRating     numVotes  titleCount
27      Western       7.176482   137.740480        9244
15        Music       7.101094    69.377863        3930
7   Documentary       6.991876   169.294444       33137
6         Crime       6.971794  2185.476352       36219
23        Sport       6.928597    98.887067        1098
13      History       6.901778   109.153333         450
4     Biography       6.878353  4860.516730        6814
9        Family       6.801551    99.202742        6126
2     Adventure       6.788089  3294.444614       26144
5        Comedy       6.760726  1284.471209      122818
```

Looking at the business question 1 insights, one can see that some genres receive always higher average ratings while other genres attract the most viewers and a total number of votes. Western and music films seem to come out as popular with average ratings just above seven; it may therefore attract a loyal, however, smaller audience that enjoys these speciality genres. Neutro, which comprises movies such as action, comedy, and drama movies, has a higher average rating compared to the industry average; however, they receive more total votes and more titles, indicating that they are more popular movies that may not necessarily be rated high but enjoy a large followership.



2. Figure 19: Barplot of the 10 regions by popularity ratio

An analysis of the bar chart shows that some of the areas like “KG,” “QA,” and “NP” maintain a high “popularity ratio,” which exceeds 0.6. That is, these regions provide a big share of what can be termed as ‘popular’ movies based on their ratings and votes concerning the number of ‘titles’ produced. Such outcomes could be explained by local emphasis on the quality of a product, cultural narratives familiar to the theatre audience of the world, or by the inclusion of targeted advertising and community engagement boosting positive feedback. Conversely, IDs such as “AZ,” “GT,” and “CG” are presented lower on the chart, which means that a lower percentage of their entire output receives the top-rated, or highly-voted, tag. Some of the reasons can be (a) the program has a greater number of productions (at least some of them may be narrow or low-profile worldwide) and (b) limited access to audiences or advertising that does not allow wide recognition. These two regions could therefore increase the ratio of their high-performing titles by engaging in deeper cross-regional cooperation, investing in culturally sensitive content, or optimising marketing strategies to wider audiences.

```python
import pandas as pd

df = pd.read_csv("high_rated_movies_cast_test.csv")

person_counts = df.groupby(['nconst', 'primaryName', 'category'])['tconst'].nunique()
person_counts = person_counts.reset_index(name='num_high_rated_movies')

top_people = person_counts.sort_values('num_high_rated_movies', ascending=False)
print(top_people.head(20))
```

```
       nconst          primaryName  category  num_high_rated_movies
0   nm0000356         Sybil Danning   actress                     19
1   nm0000434          Mark Hamill     actor                      7
24  nm0958249          Karol Beffa  composer                      1
23  nm0953125      Harrison Zanuck     actor                      1
22  nm0923928       Nicola Wheeler   actress                      1
21  nm0901552     Ekaterina Volkova   actress                      1
20  nm0780435        Ryan Seacrest      self                      1
19  nm0747328         Rosita Royce      self                      1
18  nm0735348       Ivan Rodriguez     actor                      1
17  nm0607510        Joshua Morrow     actor                      1
16  nm0601168      Elizabeth Moore   actress                      1
15  nm0554788         Óscar Martín     actor                      1
14  nm0549460  Publius Vergilius Maro    writer                      1
13  nm0522550     Jacqueline Lovell   actress                      1
12  nm0440788         Katyana Kass   actress                      1
11  nm0404895         Simone Hyams   actress                      1
10  nm0374416        Amelia Heinle   actress                      1
9   nm0346533          Cary Guffey     actor                      1
8   nm0339998        Simon Gregson     actor                      1
7   nm0295568        Annette Frier   actress                      1
```

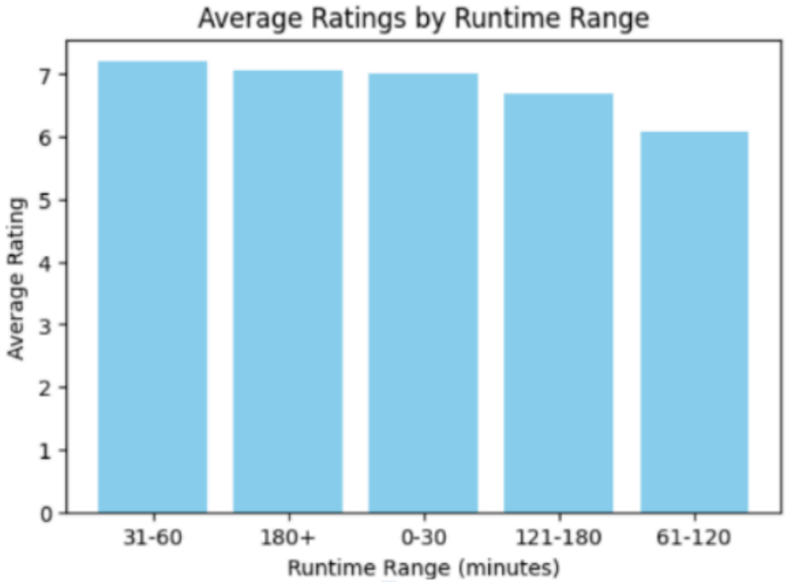
3.

A preliminary analysis of the data suggests that people belonging to some categories—especially actors and actresses—appear most often in the highest-rated movies. Sometimes there are other roles identified in a play that are not so frequently used as the main four roles; they include the composer and the 'self.' The idea of decoding the chart is based on the assumption that there are performers whose names can be linked with several highly appraised films, which in turn may be related to films' critical and/or box-office success.

```
[52] import matplotlib.pyplot as plt

     plt.figure(figsize=(6,4))
     plt.bar(range_stats.index, range_stats['averageRating'], color='skyblue
     plt.title("Average Ratings by Runtime Range")
     plt.xlabel("Runtime Range (minutes)")
     plt.ylabel("Average Rating")
     plt.show()
```

4.

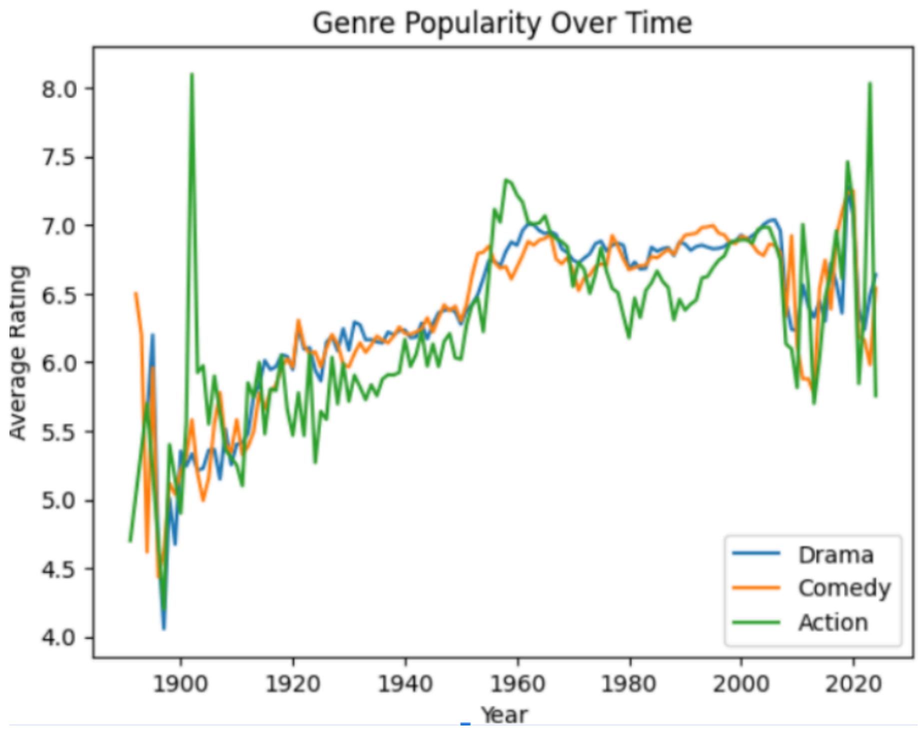


The bar chart shows that titles with a runtime of 31-60 and those with a runtime greater than 180 minutes have higher average ratings that frequently are north of 7.0. The movies that are within the 61-120 minute range get slightly lower mean values, that is usually below 6. It also pointed towards an increased preference towards more precise stories in the case of programs in the duration range of 31-60 minutes, probably because of the potential of passing significant information under favorable conditions free from excessive fluff, as well as to the advantage of narrow and deep narratives within extended product releases that last over 3 hours, to cater to the committed viewer base.

```
+-----------+-----+
|genre      |count|
+-----------+-----+
|Drama      |28763|
|Comedy     |24159|
|Crime      |9905 |
|Documentary|9223 |
|Family     |8666 |
|Action     |8400 |
|Adventure  |8389 |
|Animation  |5941 |
|Short      |5871 |
|Romance    |5077 |
+-----------+-----+
only showing top 10 rows
```

5.

For business question 6, insight reveals that Drama has the highest number of titles, over 28,500, followed closely by Comedy, with almost 24,500. These genres occupy most of the positions; this indicates that these genres belong to the most popular and successfully produced ones. Other categories, such as Crime and Documentary, have far fewer titles, yet they are among the top ten, suggesting that while their quantity may be low, they do well in terms of rating. Both older and younger audiences can be identified, as well as entertainment genres and, more broadly, family and animation.

6. 

In business question 7, the line graph represents The time series of genres shows a different pattern for Drama, Comedy, and Action. In the case of the average ratings, drama has always had a stable level of popularity throughout decades, which proves its unchangeable demand. Comedy is less stable than tragedy, and its changes through different periods fluctuate more strongly. Its popularity peaked in the middle of the twentieth century, probably because of the boosting of classic comedies and legends. However, this genre received lower ratings in the later years of the twentieth century, indicating that the audiences may have moved away from this format or comedies may have evolved. Of course, comedy does not lose its position in modern world entertainment even though it is rather unpredictable sometimes. Action, as a genre, revealed that it was the most popular type of genre in the later part of the timeline. It started small but emerged as modern with the increase in the second half of the twentieth century, along with phenomenal growth in special effects and the beginning of the concept of filmmaking. It carries on its upward progression into modernity, where action becomes one of the leading international film genres characterised by high-budget films with general appeal.

5.2 Recommendations

1. Business question 1: Based on these lessons learned, the following advice can be made when a film's priority is to enhance its critical acclaim or prestige: focus on high-rating sub-genres in niches. Western or music productions, though not as regular as drama, may add value to an organisation's image in terms of quality and story depth. However, it is also important to keep creating ordinary action, comedy, and drama categories for the appeal to the large audience and profitability purposes. Another may have to do with a concept of hybrids: constructing projects that would guarantee the appeal to specially-driven communities while having appeal for the majority at the same time.
2. From a business level, collaboration with the best performers, for instance, in the areas of "KG" or "QA," might provide access to focused catalogues of the aforementioned well-received content, which in turn might boost both reputation and satisfaction for the audience. On the other hand, regions with lower ratios could use production investments based on data and inform and publicising the notable local titles to the global audience. Targeting assumptions regarding user preference, distribution channels, and cultural appeal helps media stakeholders to promote the most engaging content within the target areas effectively.

3. From the perspective of gaining credibility and visibility with the viewers, it may be beneficial for the studios or streamers that offer interesting works to partner with or cast the people who often deliver series or movies with high ranks. Even though the mere list does not mean that a given actor or actress is solely and directly responsible for hit films, such an actor or actress consistently appearing in successful films does indicate recognisable talent with viewers, audience loyalty, or simply a focused career on quality projects.
4. Studios or streaming services may consider opportunities to establish pieces with these two runtimes as one of achieving the goal of creating higher ratings.
5. One recommendation for this business question is to invest in diverse content production: As for the highest-ranking genres, the key is to diversify content and add crime, documentary, and family genres into the program. Used sparingly in some markets, these genres may therefore represent promising areas for expansion and product differentiation.
6. Based on these trends, drama will remain popular, and comedy can be presented in new styles and combinations with other dramatic genres while the action is on a steady rise path, creating valuable opportunities for businesses. Moreover, the most recent trend across all shows the need to come up with great multi-genre projects for the ever-changing market. Such trends, if monitored frequently, will enable the content developer or creator to prepare and position himself well for future changes in the trend to tap into the trends to the maximum, thereby increasing audience response and success.

6.0 Conclusion

By addressing these business questions, the optimistic fundamentals indicate that improved decision-making based on data can have a strong positive impact on content production and marketing in the entertainment industry. This analysis has illustrated how effective big data analytics can be when it comes to addressing concerns together with identifying prospects within the movie business. Through the use of movie datasets and the use of methodologies built around CRISP-DM, entertainment businesses can define and capitalise on patterns and key variables that drive success in the movies. Thus, it became possible to define which genres, runtimes, regional preferences, and stars bring high ratings and popularity, which helps to clearly outline the busts of content production and promotion. By positioning entertainment production to meet audience consumption needs and cultural preferences, entertainment firms can boost interaction, revenue, and industry leadership in a field characterised by high levels of volatility (Stimpert et al., 2008).

With an emphasis on targeting, content optimisation, and personalisation tactics—all of which are essential for understanding consumer behaviour and enhancing monetisation tactics in the entertainment industry—data-driven insights increase market share in the media sector (Mehra, 2023).

Furthermore, the present research also highlights the need to take data as one of the key foundations for the management of the challenges of the entertainment market. It is recommended that future efforts in the industry focus on adopting advanced analytical techniques into business contexts to predict the needs of the audience, create meaningful innovations, and ensure future stability.

7.0 Personal Reflection

Completing this assignment provided me with a wealth of knowledge that exposed the relevance of big data analytics in solving real business problems. Analysing a complex IMDb dataset under the direction of the CRISP-DM approach allowed me to recognise the major steps in data analysis. Not only did I learn more about the technical work performed using tools such as SQL, Python, MongoDB, aws, but I also learned the intended goals of this technical work and what it is to deliver.

A fundamental insight was also delivered on the use of data to analyse features and characteristics of the entertainment business. For example, examining the relationships between genres, runtime, and ratings showed how minor changes have a big effect on the preferences of the viewers. This highlighted how crucial data-driven approaches are when making decisions in a variety of businesses, not just entertainment.

However, this journey had not been very smooth and easy. The process of data comprehension and data preprocessing that takes place in the data understanding step was the most challenging. The scale and variety also posed some challenges when it came to the 'data wrangling' data preparation by handling the inconsistencies/missing values/dependencies between the tables. One of the main challenges was managing to achieve both the technical accuracy and the originality when performing data preprocessing and analysis, respectively. Moreover, the process of joining raw data with conclusions that are substantial for the business world was not only inspiring from an academic point of view but included the presence of owners’ visions as well.

In the future, I am now eager to learn more about higher levels of analytics. Also, I plan to enhance my knowledge of visualisation to be able to better present my results to diverse audiences.

To sum up, this assignment was enjoyable and demanding. It has given me useful skills, reinforced the importance of data in achieving business goals, and inspired me to continue my education in this field. I'm determined to use what I've learned going forward in practical situations, emphasising creativity, effectiveness, and making significant decisions.

Word count: 7015

References

Bemthuis. (2024). [2404.01114] A CRISP-DM-based Methodology for Assessing Agent-based Simulation Models using Process Mining. [online]. Available from: https://arxiv.org/abs/2404.01114 [Accessed January 4, 2025].

Gavilán, D., Lores, S.F. and Martínez-Navarro, G. (2019). The influence of Online Ratings on Film Choice: Decision Making and Perceived Risk. *Communication &Amp; Society*, 32(2). [online]. Available from: https://doi.org/10.15581/003.32.2.45-59 [Accessed January 3, 2025].

Juan. (2019). Refining IMDb Scores: A Better Ranking | Toptal®. *Toptal Engineering Blog*. [online]. Available from: https://www.toptal.com/data-science/improving-imdb-rating-system [Accessed January 3, 2025].

Kumar and Sharma. (2024). Impact of Marketing Strategies on Consumer Buying Behaviour with Specific Reference to Movies as a Medium. *International Research Journal on Advanced Engineering and Management (IRJAEM)*, 2(03). [online]. Available from: https://doi.org/10.47392/irjaem.2024.0038 [Accessed January 3, 2025].

Manakbayeva, A.B. (2022). The Film Industry as a Sector of the Economy: Current Problems and Trends. *Economy: strategy and practice*, 17(1), pp.226–237.

Matthews, P. and Glitre, K. (2024). (PDF) Genre analysis of movies using a topic model of plot summaries. *ResearchGate*. [online]. Available from: https://www.researchgate.net/publication/351939382_Genre_analysis_of_movies_us ing_a_topic_model_of_plot_summaries [Accessed January 3, 2025].

McMahon, J. (2023). Star Power and Risk: A Political Economic Study of Casting Trends in Hollywood. *Quarterly Review of Film and Video*, 41(8). [online]. Available from: https://doi.org/10.1080/10509208.2023.2215355 [Accessed January 3, 2025].

Mehra, A. (2023). Leveraging Data-Driven Insights to Enhance Market Share in the Media Industry. *Journal for Research in Applied Sciences and Biotechnology*, 2(3), pp.291–304.

Pavan and Manjunath. (2024). A Study on Movie Genre Analysis Using IMDb Data. *International Journal of Advanced Research in Science, Communication and Technology*, pp.101–104.

Stimpert, J.L. et al. (2008). Factors Influencing Motion Picture Success: Empirical Review And Update. *Journal of Business & Economics Research (JBER)*, 6(11). [online]. Available from: https://clutejournals.com/ [Accessed January 9, 2025].

Tunca, S. (2024). Forecasting the Enrolment of Bank Term Deposits: A case study approach with Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. [online]. Available from: https://www.researchsquare.com/article/rs-3921578/v1 [Accessed January 3, 2025].