

PROJECT 1

CINEMATIC TRENDS: EXPLORING MOVIE METADATA AND VIEWER PREFERENCES

Team Members:

Sashidhar Chary Viswanathula (11714360)

Dimpu Nithish Karanam (11823599)

Vidya Gangasani (11702121)

Problem Statement:

The movie industry produces a huge amount of data, from movie genres and release dates, ratings and popularity metrics. But it can be challenging to understand what truly drives a movie's success or failure. This project focuses on uncovering insights into viewer preferences, genre popularity, trends in movie ratings over time, and Profits earned by a movie.

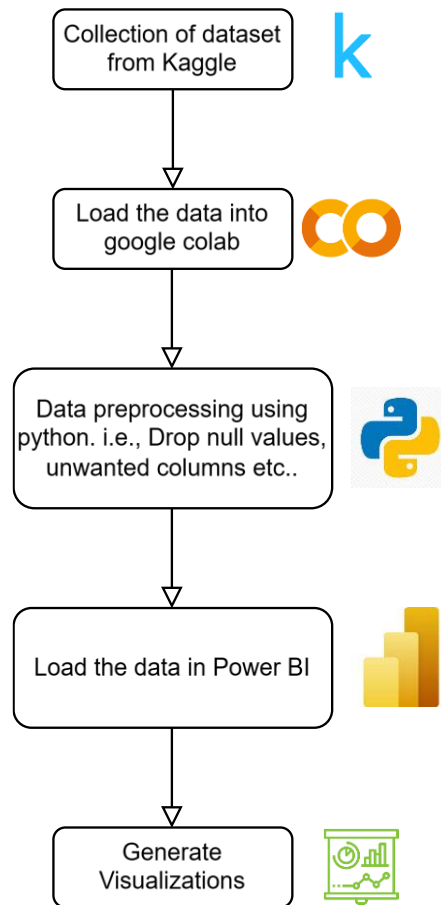
By analyzing the two datasets of movie metadata and ratings, we plan to answer questions like:

- Do certain genres consistently receive higher ratings?
- How does a movie's release year or runtime influence audience perception?
- Are there patterns in viewer behavior that correlate with popularity or average rating?
- What is the average budget of a genre, and average profits earned?

We are focusing on communicating these insights through interactive and engaging dashboards that reveal how people engage with films.

Workflow Diagram:

This is the Workflow diagram for the project 1, We started off by collecting the dataset from Kaggle and loading it to the Google collab, after that preprocessed the dataset using Python and loaded the preprocessed dataset into power BI to generate meaningful insights and visualizations.



Data Abstraction:

For this project, we are using datasets from Kaggle, which are publicly available and have rich information about the movies and their ratings. We've used a structured dataset, but it has complex columns which have a JSON type in it. We processed the JSON type columns to get the data we needed from those columns.

movies_metadata.csv (From Kaggle – TMDB dataset):

Contains information about movies like titles, genres, budget, revenue, runtime, popularity, release dates, production companies, and more. This will help us analyze movie characteristics and metadata.

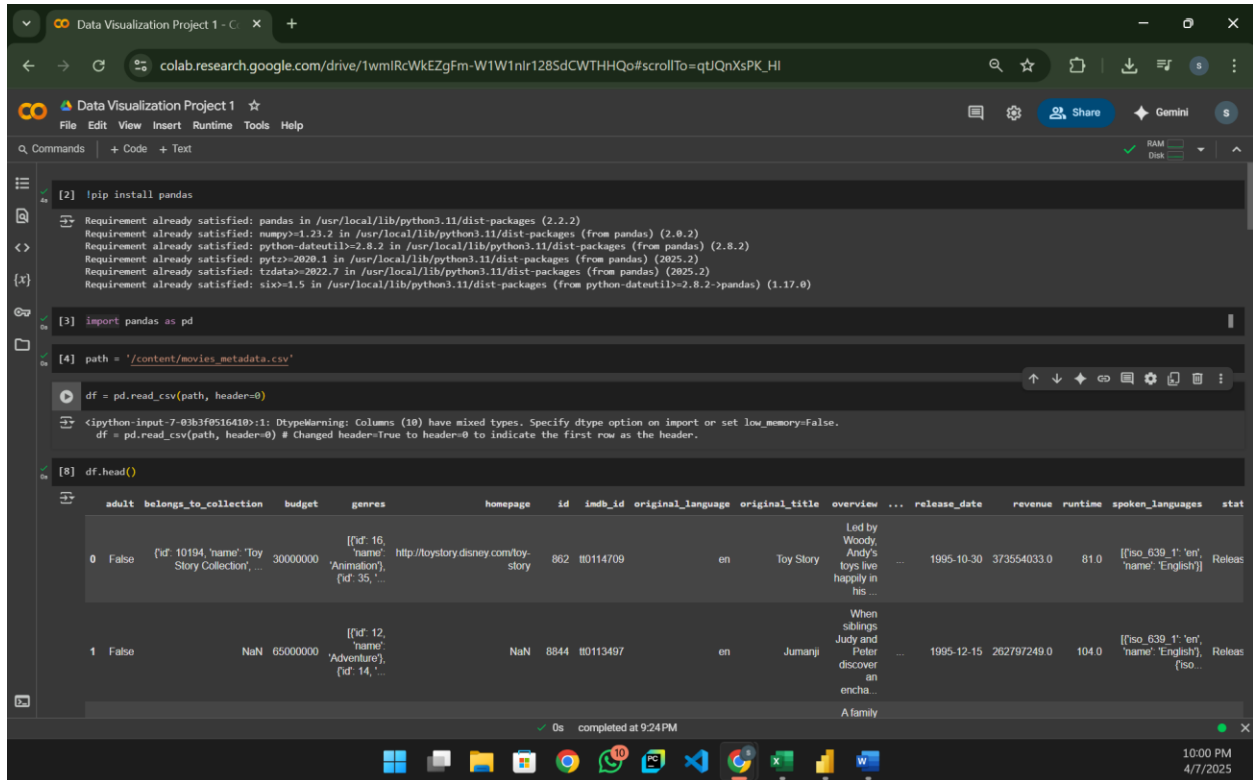
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
adult	belongs_to	budget	genres	homepage	id	imdb_id	original_language	original_title	overview	popularity	poster_path	production_companies	production_countries	release_date	revenue	runtime	spoken_languages	status	tagline
FALSE	{id: 10194	30000000	{id: 16, name: 'Toys		862	tt0114709	en	Toy Story	Led by Wor	21.94694	/rhlRbceof	{name: 'F	{iso_3166_1: 'US'		3.74E+08	81	{iso_639_1: 'en'	Released	
FALSE		65000000	{id: 12, name: 'Adv		8844	tt0113497	en	Jumanji	When sibli	17.01554	/vzmL6fP7	{name: 'T	{iso_3166_1: 'US'		2.63E+08	104	{iso_639_1: 'en'	Released	Ro
FALSE	{id: 11905	0	{id: 10749, name: 'F		15602	tt0113228	en	Grumpier (A family w		11.7129	/6ksm1sjK	{name: 'V	{iso_3166_1: 'US'		0	101	{iso_639_1: 'en'	Released	Sti
FALSE		16000000	{id: 35, name: 'Cor		31357	tt0114885	en	Waiting to Be Cheate	d	3.859495	/16XOMpE	{name: 'T	{iso_3166_1: 'US'		81452156	127	{iso_639_1: 'en'	Released	Fri
FALSE	{id: 96871	0	{id: 35, name: 'Cor		11862	tt0113041	en	Father of t	Just when	8.387519	/e64sOI48	{name: 'S	{iso_3166_1: 'US'		76578911	106	{iso_639_1: 'en'	Released	Ju
FALSE		60000000	{id: 28, name: 'Acti		949	tt0113277	en	Heat	Obsessive	17.92493	/zMyfPUelh	{name: 'F	{iso_3166_1: 'US'		1.87E+08	170	{iso_639_1: 'en'	Released	Al
FALSE		58000000	{id: 35, name: 'Cor		11860	tt0114319	en	Sabrina	An ugly dur	6.677277	/jQh15y5Y	{name: 'F	{iso_3166_1: 'US'		0	127	{iso_639_1: 'en'	Released	Yo
FALSE		0	{id: 28, name: 'Acti		45325	tt0112302	en	Tom and HA	mischiev	2.561161	/sGO5Qa5	{name: 'V	{iso_3166_1: 'US'		0	97	{iso_639_1: 'en'	Released	Th
FALSE		35000000	{id: 28, name: 'Acti		9091	tt0114576	en	Sudden De	Internatio	5.23158	/eoWvKD6	{name: 'L	{iso_3166_1: 'US'		64350171	106	{iso_639_1: 'en'	Released	Te

ratings.csv (From Kaggle – TMDB dataset):

This dataset includes over 100,000 user ratings for movies. Each rating includes a user ID, movie ID, rating score, and timestamp which allows us to evaluate user behavior, preferences, and seasonal trends based on the ratings.

userId	movieId	rating	timestamp
1	110	1	1.43E+09
1	147	4.5	1.43E+09
1	858	5	1.43E+09
1	1221	5	1.43E+09
1	1246	5	1.43E+09
1	1968	4	1.43E+09
1	2762	4.5	1.43E+09
1	2918	5	1.43E+09
1	2959	4	1.43E+09
1	4226	4	1.43E+09
1	4878	5	1.43E+09

Data Transformation:



```
[2] !pip install pandas
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy<=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

[3] import pandas as pd

[4] path = '/content/movies_metadata.csv'

df = pd.read_csv(path, header=0)
<ipython-input-7-03b3f0516410>:1: DtypeWarning: Columns (10) have mixed types. Specify dtype option on import or set low_memory=False.
df = pd.read_csv(path, header=0) # Changed header=True to header=0 to indicate the first row as the header.

[8] df.head()
```

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overview	release_date	revenue	runtime	spoken_languages	status
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...}	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, 'name': 'Comedy'}]	http://toystorydisney.com/toy-story	862	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in his room until Mr. Potter, the grumpy store owner, arrives, with the aim of replacing Andy's toys with new, more "educational" ones.	1995-10-30	373554033.0	81.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, 'name': 'Fantasy'}]	NaN	8844	tt0113497	en	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world of adventure, they discover an enchanted board game that opens the door to a magical world of adventure.	1995-12-15	262797249.0	104.0	[{'iso_639_1': 'en', 'name': 'English'}, {'iso_639_1': 'es', 'name': 'Spanish'}]	Released

Here we can see that the dataset is loaded using pandas framework and `df.head()` is called to expose the first five rows. The dataset is not clean and transformed. There are many unwanted columns, null values and nested data.

```

[9] new_df = df.drop(columns=['belongs_to_collection', 'homepage', 'overview', 'poster_path', 'spoken_languages', 'tagline', 'title', 'video'], errors='ignore')

import ast

def get_first_genre(genres_str):
    try:
        genres_list = ast.literal_eval(genres_str)
        if isinstance(genres_list, list) and len(genres_list) > 0 and 'name' in genres_list[0]:
            return genres_list[0]['name']
        else:
            return None
    except (ValueError, SyntaxError, IndexError, TypeError):
        return None

new_df['first_genre'] = new_df['genres'].apply(get_first_genre)
new_df.head()

```

	genres	id	imdb_id	original_language	original_title	popularity	production_companies	production_countries	release_date	revenue	runtime	status	vote_average	vote_count	first_genre
[{"id": 16, "name": "Animation"}, {"id": 33, "name": "Comedy"}]	862	#0114709	en	Toy Story	21.946943	[{"name": "Pixar Animation Studios", "id": 3}], [{"iso_3166_1": "US", "name": "United States of America"}]	1995-10-30	373554033.0	81.0	Released	7.7	5415.0	Animation		
[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Comedy"}]	8844	#0113497	en	Jumanji	17.015539	[{"name": "TriStar Pictures", "id": 559}, {"name": "United States of America"}]	1995-12-15	262797249.0	104.0	Released	6.9	2413.0	Adventure		
[{"id": 10749, "name": "Romance"}, {"id": 35, "name": "Comedy"}]	15602	#0113228	en	Grumpier Old Men	11.7129	[{"name": "Warner Bros.", "id": 6194}, {"name": "United States of America"}]	1995-12-22	0.0	101.0	Released	6.5	92.0	Romance		

Dropped unnecessary columns and retrieved the genre form the nested list.

```

def get_first_production_company(production_companies_str):
    try:
        production_companies_list = ast.literal_eval(production_companies_str)
        if isinstance(production_companies_list, list) and len(production_companies_list) > 0 and 'name' in production_companies_list[0]:
            return production_companies_list[0]['name']
        else:
            return None
    except (ValueError, SyntaxError, IndexError, TypeError):
        return None

new_df['first_production_company'] = new_df['production_companies'].apply(get_first_production_company)
new_df.head()

new_df = new_df.drop(columns=['production_companies'], errors='ignore')
new_df = new_df.rename(columns={'first_production_company': 'production_company'})
new_df.head()

```

	adult	budget	id	imdb_id	original_language	original_title	popularity	release_date	revenue	runtime	status	vote_average	vote_count	genre	production_company
0	False	30000000	862	#0114709	en	Toy Story	21.946943	1995-10-30	373554033.0	81.0	Released	7.7	5415.0	Animation	Pixar Animation Studios
1	False	65000000	8844	#0113497	en	Jumanji	17.015539	1995-12-15	262797249.0	104.0	Released	6.9	2413.0	Adventure	TriStar Pictures
2	False	0	15602	#0113228	en	Grumpier Old Men	11.7129	1995-12-22	0.0	101.0	Released	6.5	92.0	Romance	Warner Bros.
3	False	16000000	31357	#0114885	en	Waiting to Exhale	8.59495	1995-12-22	81452156.0	127.0	Released	6.1	34.0	Comedy	Twentieth Century Fox Film Corporation
4	False	0	11862	#0113041	en	Father of the Bride Part II	8.387519	1995-02-10	76578911.0	106.0	Released	5.7	173.0	Comedy	Sandollar Productions

Now we can see that the dataset is cleaned and transformed.

```
import ast
for col in ['budget', 'id', 'revenue', 'runtime', 'vote_count']:
    new_df[col] = pd.to_numeric(new_df[col], errors='coerce').astype('Int64')

new_df['adult'] = new_df['adult'].astype(bool)
for col in ['vote_average', 'popularity']:
    new_df[col] = pd.to_numeric(new_df[col], errors='coerce').astype('Float64')

print(new_df.info())
print("\nNull Values:")
print(new_df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   adult                 45466 non-null bool    
 1   budget                45463 non-null Int64  
 2   id                    45463 non-null Int64  
 3   imdb_id               45449 non-null object 
 4   original_language     45455 non-null object 
 5   original_title        45466 non-null object 
 6   popularity             45460 non-null Float64
 7   release_date          45379 non-null object 
 8   revenue               45460 non-null Int64  
 9   runtime               45283 non-null Int64  
10   status                45379 non-null object 
11   vote_average          45460 non-null Float64
12   vote_count            45460 non-null Int64  
13   genre                 43024 non-null object 
14   production_company    33585 non-null object 
dtypes: float64(2), int64(5), bool(1), object(7)
memory usage: 5.2+ MB
None

Null Values:
adult          0
budget         3
id             3
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   adult                 45466 non-null bool    
 1   budget                45463 non-null Int64  
 2   id                    45463 non-null Int64  
 3   imdb_id               45449 non-null object 
 4   original_language     45455 non-null object 
 5   original_title        45466 non-null object 
 6   popularity             45460 non-null Float64
 7   release_date          45379 non-null object 
 8   revenue               45460 non-null Int64  
 9   runtime               45283 non-null Int64  
10   status                45379 non-null object 
11   vote_average          45460 non-null Float64
12   vote_count            45460 non-null Int64  
13   genre                 43024 non-null object 
14   production_company    33585 non-null object 
dtypes: float64(2), int64(5), bool(1), object(7)
memory usage: 5.2+ MB
None

Null Values:
adult          0
budget         3
id             3
imdb_id       17
original_language  11
original_title  0
popularity     6
release_date   87
revenue        6
runtime        263
status         87
vote_average   6
vote_count     6
genre         2442
production_company 11881
dtype: int64
```

Here we have changed the datatypes of the columns to their appropriate datatypes like int and float. The dataset also have lots of null values, which are handled in the next step.

```
colab.research.google.com/drive/1wmlRcWkEZgFm-W1W1n1r1285dCWTHHqo#scrollTo=ICq_JbwPW3E
```

```
for col in ['genre', 'production_company']:
    new_df[col] = new_df[col].fillna('Unknown')

new_df.dropna(subset=['budget', 'id', 'revenue', 'runtime', 'vote_count', 'vote_average', 'popularity', 'adult', 'imdb_id', 'original_language', 'release_date', 'status'], inplace=True)

print(new_df.info())
print("\nNull Values:")
print(new_df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 45031 entries, 0 to 45465
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   adult                 45031 non-null  bool    
 1   budget                45031 non-null  Int64   
 2   id                    45031 non-null  Int64   
 3   imdb_id               45031 non-null  object  
 4   original_language     45031 non-null  object  
 5   original_title        45031 non-null  object  
 6   popularity            45031 non-null  Float64  
 7   release_date          45031 non-null  object  
 8   revenue               45031 non-null  Int64   
 9   runtime               45031 non-null  Int64   
10   status                45031 non-null  object  
11   vote_average          45031 non-null  Float64  
12   vote_count            45031 non-null  Int64   
13   genre                 45031 non-null  object  
14   production_company    45031 non-null  object  
dtypes: Float64(2), Int64(5), bool(1), object(7)
memory usage: 5.5+ MB
None

Null Values:
adult          0
budget         0
id             0
imdb_id        0
original_language  0
original_title  0
popularity     0
release_date   0
revenue        0
runtime        0
status         0
vote_average   0
vote_count     0
genre          0
production_company  0
dtype: object
```

completed at 10:13 PM

```
colab.research.google.com/drive/1wmlRcWkEZgFm-W1W1n1r1285dCWTHHqo#scrollTo=WOtdx6roOXCP
```

```
from google.colab import files
new_df.to_csv(modified_movies_metadata.csv', encoding = 'utf-8-sig')
files.download(modified_movies_metadata.csv')
```

```
8 revenue                45031 non-null  Int64  
9 runtime                45031 non-null  Int64  
10 status                45031 non-null  object  
11 vote_average          45031 non-null  Float64  
12 vote_count            45031 non-null  Int64  
13 genre                 45031 non-null  object  
14 production_company    45031 non-null  object  
dtypes: Float64(2), Int64(5), bool(1), object(7)
memory usage: 5.5+ MB
None

Null Values:
adult          0
budget         0
id             0
imdb_id        0
original_language  0
original_title  0
popularity     0
release_date   0
revenue        0
runtime        0
status         0
vote_average   0
vote_count     0
genre          0
production_company  0
dtype: object
```

completed at 10:13 PM

Here the dataset has no null values because they are dropped. Therefore, our dataset is clean and transformed ready for visualizations.

Target:

We are planning on analyzing:

1. Revenue VS Budget Analysis
2. Top genres by average rating
3. Trending popularity over years
4. Runtime distribution
5. Movies by language
6. Top production companies by revenue
7. Vote count vs vote average

Actions:

1. Filtered the data that doesn't make sense or bring any value to the project. Like null values, unwanted columns etc...
2. Aggregated the columns to get more granularity on the project and much more summarized analysis.
3. Found relation between budget, revenue, release date, popularity.
4. Utilized suitable visualizations to better understand the data. (E.g.: bar chart, scatterplot etc...)

Tool used for visualization:

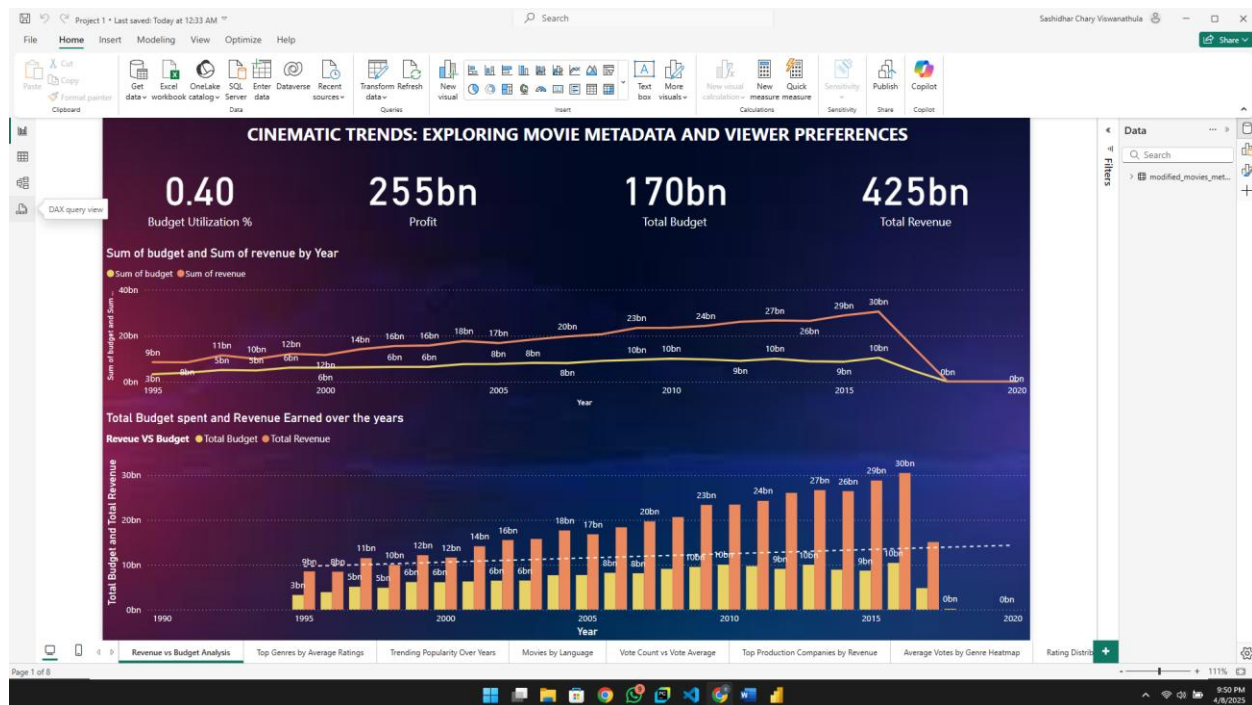
For this project, we have utilized Power BI to generate visualizations because it is easy to use interface and data loading is easier and flexible in this platform.

Data Loading:

The screenshot displays the Microsoft Power BI Desktop interface. The 'Get data' pane on the left shows the 'modified_movies_metadata.csv' file selected. The central area shows a preview of the data table with columns: adult, budget, id, imdb_id, original_language, original_title, popularity, release_date, revenue, and runtime. The 'Build' pane on the right shows various visualization suggestions. The bottom status bar indicates 'Page 1 of 1' and the system clock shows 10:34 PM on 4/7/2025.

	adult	budget	id	imdb_id	original_language	original_title	popularity	release_date	revenue	runtime
0	TRUE	30000000	862	tt0114709	en	Toy Story	21.946943	10/30/1995	373554033	1
1	TRUE	65000000	8844	tt0113497	en	Jumanji	17.015539	12/15/1995	262797249	1
2	TRUE	0	15602	tt0113228	en	Grumpier Old Men	11.7129	12/22/1995	0	1
3	TRUE	16000000	31357	tt0114885	en	Waiting to Exhale	3.859495	12/22/1995	81452156	1
4	TRUE	0	11862	tt0113041	en	Father of the Bride Part II	8.387519	2/10/1995	76578911	1
5	TRUE	60000000	949	tt0113277	en	Heat	17.924927	12/15/1995	187436818	1
6	TRUE	58000000	11860	tt0114319	en	Sabrina	6.677277	12/15/1995	0	1
7	TRUE	0	45325	tt0112302	en	Tom and Huck	2.561161	12/22/1995	0	1
8	TRUE	35000000	9091	tt0114576	en	Sudden Death	5.23158	12/22/1995	64350171	1
9	TRUE	58000000	710	tt0113189	en	GoldenEye	14.686036	11/16/1995	352194034	1
10	TRUE	62000000	9087	tt0112346	en	The American President	6.318445	11/17/1995	107879496	1
11	TRUE	0	12110	tt0112896	en	Dracula: Dead and Loving It	5.430331	12/22/1995	0	1
12	TRUE	0	21032	tt0112453	en	Balto	12.140733	12/22/1995	11348324	1
13	TRUE	44000000	10858	tt0113987	en	Nixon	5.092	12/22/1995	13681765	1
14	TRUE	98000000	1408	tt0112760	en	Cutthroat Island	7.284477	12/22/1995	10017322	1
15	TRUE	52000000	524	tt0112641	en	Casino	10.137389	11/22/1995	116112375	1
16	TRUE	16500000	4584	tt0114388	en	Sense and Sensibility	10.673167	12/13/1995	135000000	1
17	TRUE	4000000	5	tt0113101	en	Four Rooms	9.026586	12/9/1995	4300000	1
18	TRUE	30000000	9273	tt0112281	en	Ace Ventura: When Nature Calls	8.205448	11/10/1995	212385533	1
19	TRUE	60000000	11517	tt0113845	en	Money Train	7.337906	11/21/1995	35431113	1

Visualization 1: REVENUE VS BUDGET OVER TIME



This visualization compares total revenue vs total budget per year which is in the bar chart and trends of both overtime in the line chart. This page also has highlights summary KPIs for budget utilization percentage, total revenue, total budget, profit.

KPI's: By seeing the KPIs, we can say that the budget utilization is at 0.4%, which is extremely low, indicating very low budget utilization. The Profit is 255 billion, total budget is 170 billion, total revenue is 425 billion. Despite a low budget utilization percentage, the industry has generated a substantial profit of 255 billion, showing a strong return on investment over the observed period.

Line Chart:

The above chart shows the sum of budget and revenue by year from 1995 to 2020. As we can see, budget (Orange line) remains relatively studied between six billion and 10 billion from 2000 onward and revenue (Pink line) shows an upward trend, peaking sharply at 30 billion in 2018 then dropping to zero in 2020 (Possibly due to Covid 19 shutdowns).

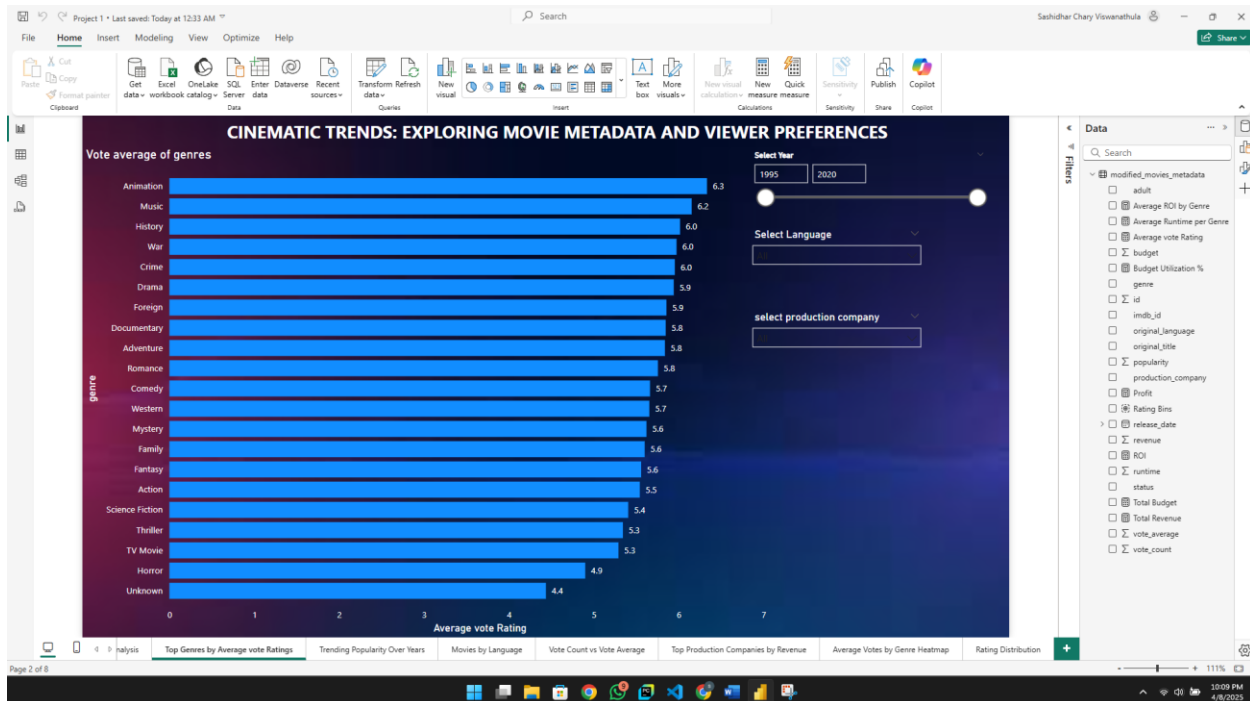
Stacked Column Chart:

The above stacked column chart shows revenue vs budget over the years yellow bars represent budget orange bars represent revenue. There is a visible gap between the two which is the two bars. Revenue bars Are always higher, signaling profit every year. The trend becomes more significant post 2005, showing increased revenue even though budgets grow only slightly.

Insights:

By observing this PBI report we can say that the movie industry has maintained a healthy financial trend over the years, with profits increasing significantly relative to budget. The 30 billion revenues in 2018 was the highest performing year the impact in 2020 is worth highlighting.

Visualization 2: VOTE AVERAGE OF GENRES



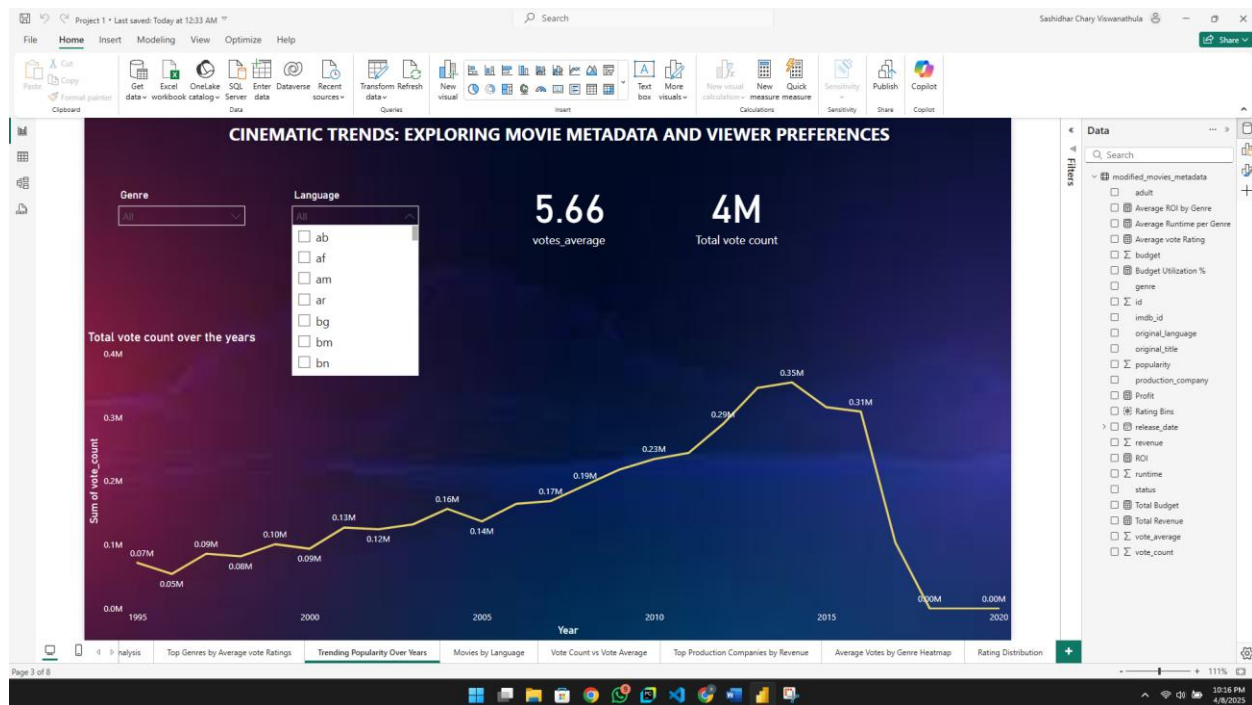
This visualization represents average voting votes of genres. This is a horizontal bar chart displaying the average vote rating for each movie genre we have used a dax measure called **average vote rating = AVERAGE (modified_movies_metadata[vote_average])**. The slicers (Year, language, production company) Remind the same, enabling interactive filtering for deep exploration.

Insights:

On the X axis we have average vote rating and on the Y axis we have genre. This chart displays the average audience rating for each movie genre based on voting data from the data set it allows us to compare how positively or negatively we have rated different genres on average.

By seeing the visualization, we can say that animation has a rating of 6.3, music 6.2, history or war or crime 6.0 are the highest rated genres. TV movie is 5.3 and thriller 5.3 are underperforming, possibly due to lower budgets, production quality or niche appeal.

Visualization 3: TRENDING POPULARITY OVER YEARS



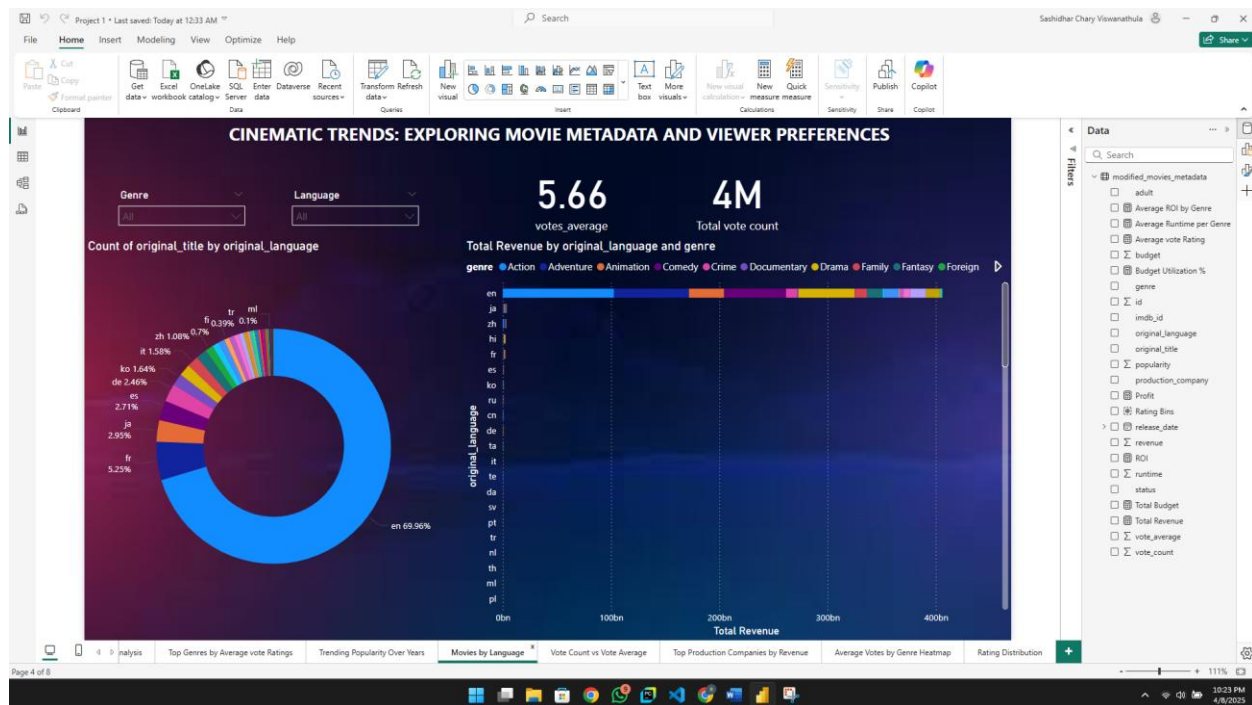
The above line chart shows the sum of vote count by year from 1995 – 2020. Here I've utilized two KPIs which are average board rating and total vote count and two filters which are genre and language.

By observing the chart, from 1995 to 2005 a steady slow growth in both counts from 0.04M to 0.12M which tells us that the digital movie platforms or metadata sources were limited in audience interaction. From 2006 to 2014 a significant spike from 0.16M to a peak of 0.35M votes in 2014 says that there is a sharp price in increased digital engagement, due to the rise of online movie databases, streaming platforms or mobile access to content from 2015 to 2018 vote counts remain relatively high indicating sustained viewer activity from 2019 to 2020 a dynamic drop to zero votes this is most likely due to disruptions like the COVID 19 pandemic which stalled movie releases and audience interaction.

Insights:

We can say that there is a clear correlation between year and digital audience engagement watch rise steadily with Internet and streaming growth. The sudden drop in 2019 to 2020 is because of the global impact of COVID 19 the inclusion of genre and language filters alert us for detailed investigation into how specific categories or language based films trended in popularity overtime.

Visualization 4: MOVIES BY LANGUAGE



In this Visualization we have used two charts, one is doughnut chart, and one is bar chart. Represents a distribution of movie count by the original language in which they were produced. There are also two KPIs here, one is average vote rating, other is total vote count.

Donut Chart: Percentage of language:

By observing the donut chart, we can say that English is the dominant language accounting for 69.96% of all movies. Other languages like Hindi are 5.2%, French 2.9%, Japanese 2.7%, Spanish 2.4%, Korean 1.6% and rest other languages are less than 1%.

This shows a heavy lean towards English language films, which is common in global movie datasets however non-English films do show up in notable volumes, especially from Bollywood.

Bar Chart: Revenue by language:

English not only dominates in volume but also in total revenue reaching over 400 billion non-English languages such as Hindi Japanese Spanish contributes smaller but still visible revenue figures. Each genre is color coded to show the difference.

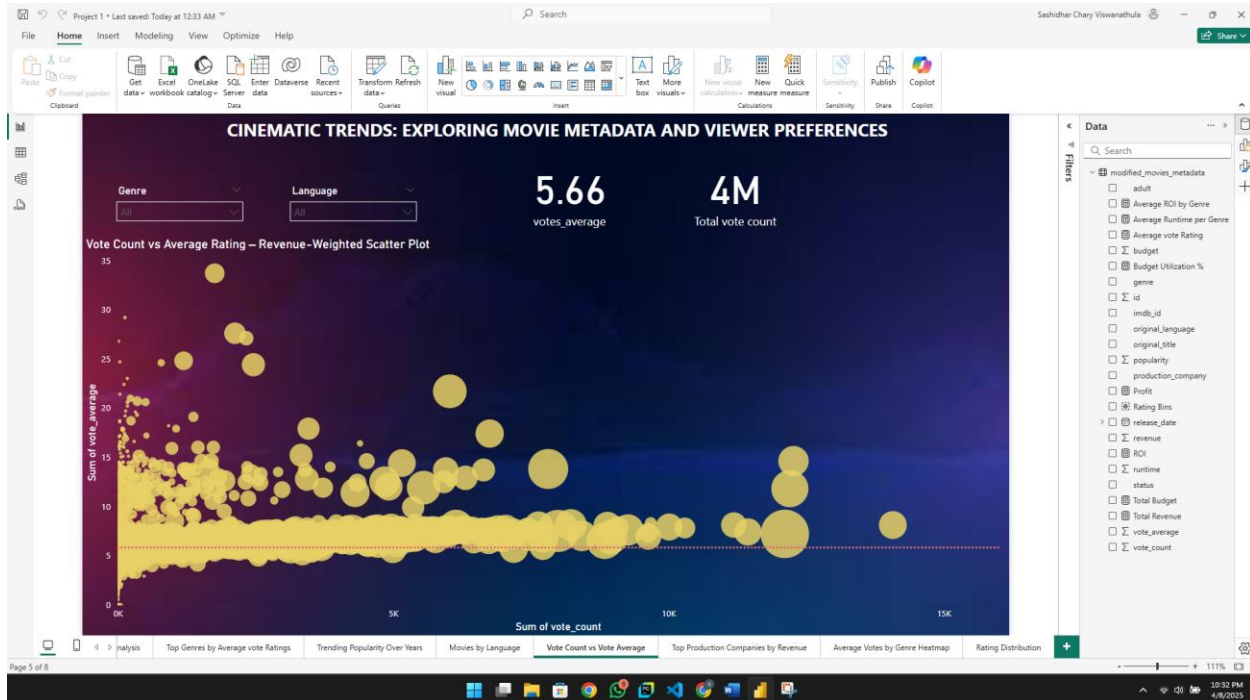
English language films dominate both quantity and profitability, but diverse journals and substantial revenue exists in global markets too, especially in highly specialized genres.

Insights:

This Visualization provides a clear global distribution view for content acquisition teams, showing where the majority of content originates from. Marketing or streaming platforms, helping tailor regional catalogs. production studios, Identifying untapped language - genre revenue pairs.

Visualization 5:

VOTE COUNT VS AVERAGE RATING – REVENUE-WEIGHTED SCATTER PLOT



The above chart is a scatter plot on X axis we have sum of word count on Y axis we have sum of vote average bubble size is proportional to revenue. Disk slatter plot allows us to analyze three performance metrics simultaneously.

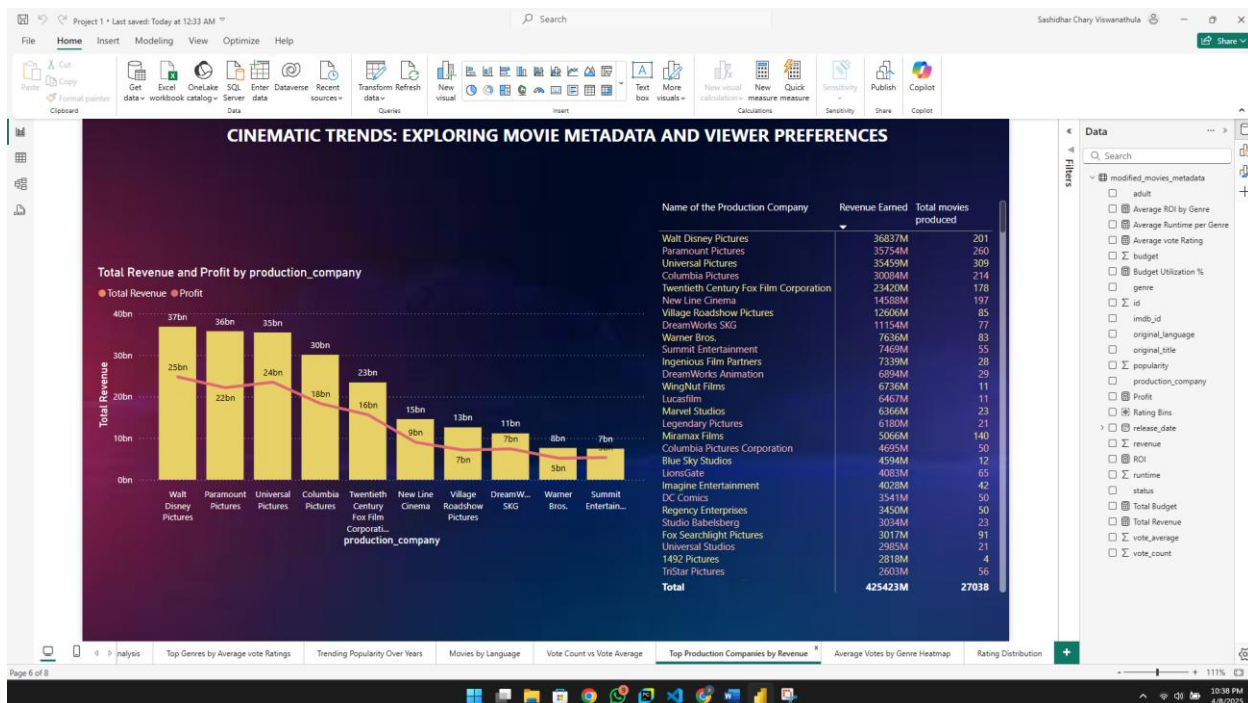
1. How many people voted for a movie?
2. How highly did those people rate the movie?
3. How much revenue does the movie make?

Visually we can observe that a majority of bubbles cluster around the mid lower Y axis indicating a concentration of average rated movies rating between five and 10. The horizontal spread of bubbles shows that some movies have very high vote counts exceeding 15,000 larger bubbles are generally on the right side which tells us the movies with more votes tend to generate modest revenue.

Insights:

By observing this scatterplot, we can say that high revenue is not equal to high ratings because many large bubbles are around the average rating 5 to 7. Success of a film lies in the mid rating but high vote count zone mainstream blockbusters that receive wide attention but average ratings high rating, low vote low revenue films suggest hidden gems or niche productions loved by a small but passionate audience.

Visualization 6: TOP PRODUCTION COMPANIES BY REVENUE



This visualization is a combination of bar and line chart. X axis we have production company on the Y axis we have total revenue, and the line indicates profit this chart compares revenue vs profit for top production companies. The table visualization has revenue earned in millions and total movies produced

The chart tells us Walt Disney Pictures has a revenue of 37 billion with a profit of 25 billion. By this we can say that Disney is the undisputed leader with strong margins and a large catalog of 201 movies. Paramount pictures and Universal pictures are two production companies whose revenue is close to Disney and profit is slight lower than Disney around 22 billion to 24 billion. which accounts for 260 plus movies from each production company. As a chart moves to the right side revenue and profit both decline but profit decreases more steeply than revenue for example Summit Entertainment and Warner Bros make 7 billion and 8 billion, with profits of one to 2 billion, suggesting higher costs or low ROI.

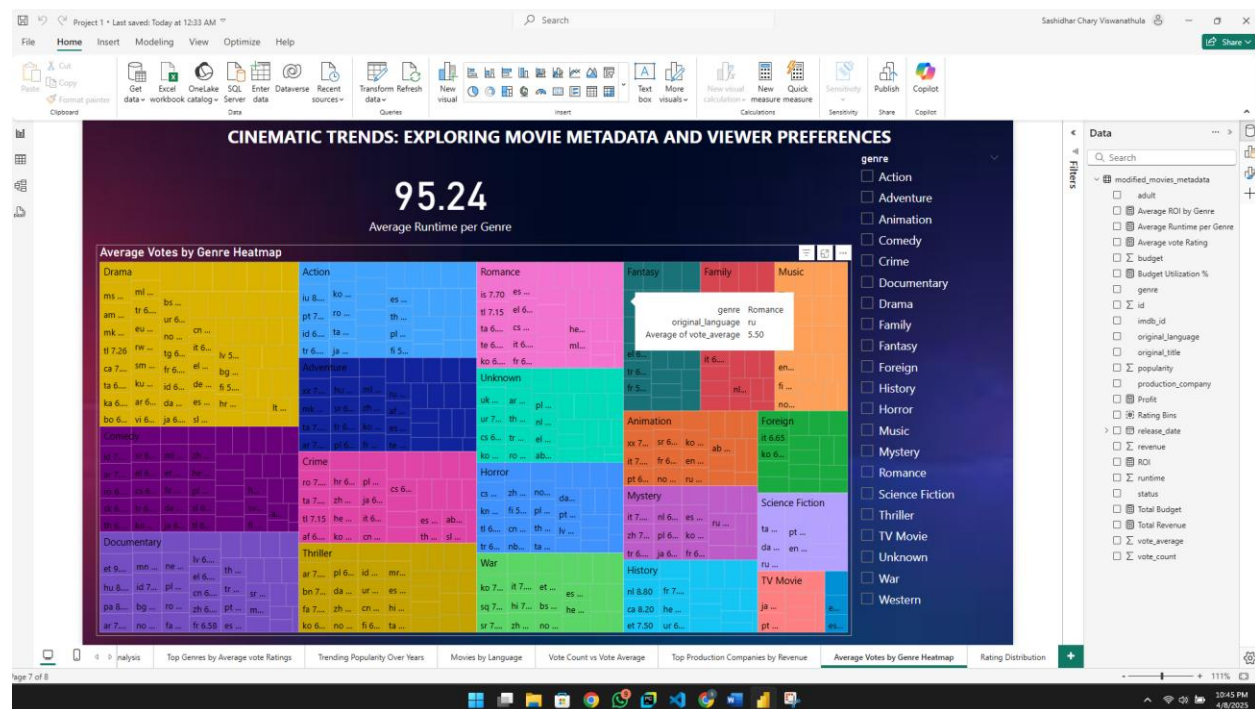
Matrix visualization adds important context by showing volume of production such as universal pictures leads in total movies of 39 dream works animation with fewer titles of 29 studios like

Marvel Studios and legendary pictures appear mid table with fewer movies but relatively high revenues, showing quality over quantity success.

Insights:

By observing the whole visualization, we can say that Walt Disney's dominance in both volume and profitability confirms its position as a market leader although there is a noticeable correlation between scale and success but not always fuse studios with small catalogs deliver strong returns companies with higher output but lower profits might need operational or content strategy adjustments.

Visualization 7: GENRE-LANGUAGE HEATMAP



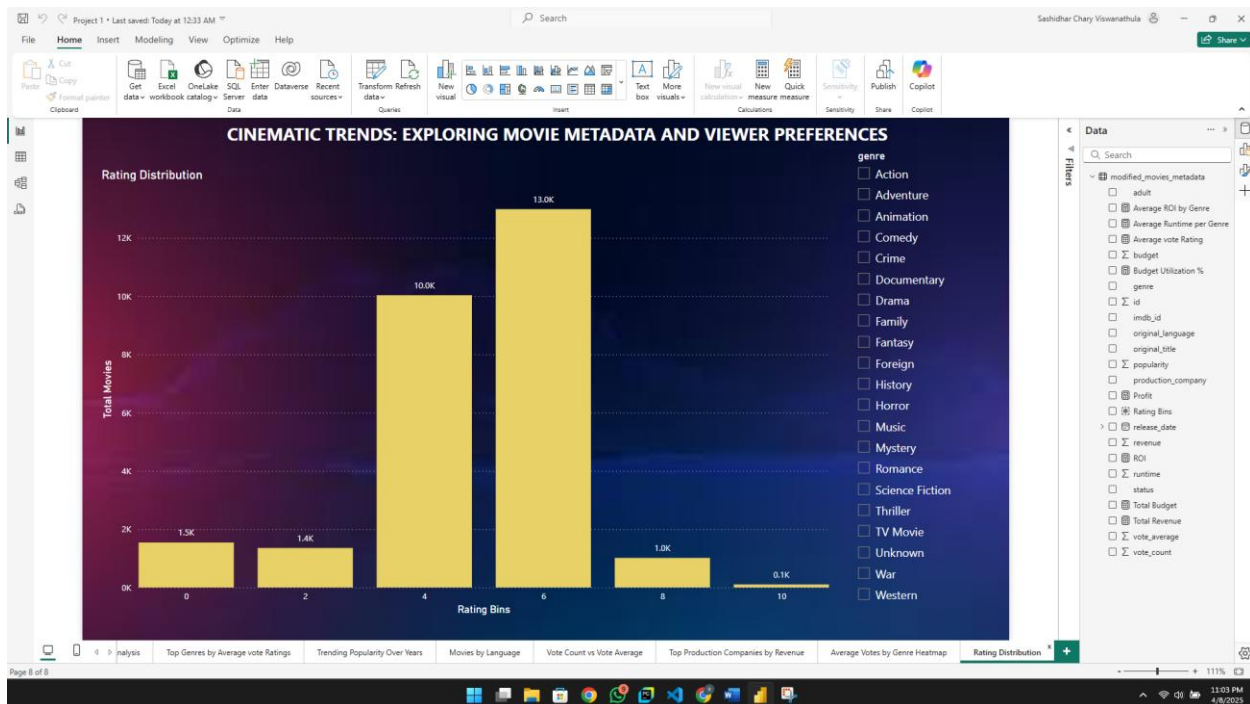
The above visualization is a heat map which shows average votes by genre. In category axis we have genre and original language, and the metric displayed here is voting average. This heat map is a multilayered analysis which shows how each genre performs in terms of audience rating and which original language versions of the genre received higher or lower ratings. Each small tile represents a unique combination of genre plus language and is proportional in size based on the number of titles and displays the average via rating for that combination.

Insights:

By observing the heat map, we can say that drama has the widest representation, indicating high volume and variety across languages like animation, foreign and documentary also show healthy ratings across multiple languages. For example, in the tool tip romance movies in Russian average a 5.5 rating which shows moderate performance foreign genre entries often hover above 6.5, showing strong audience appreciation for international cinema.

Some genres like TV movie horror tend to have lower ratings or fewer language variations. Crime thriller and mystery genres also display many language tiles with ratings close to the global average of 5.66.

Visualization 8: RATING DISTRIBUTION



Groups

Name *

Rating Bin

Field

vote_average

Group type

Bin

Bin type

Size of bins

Min value

0

Max value

10

Binning splits numeric or date/time data into equally sized groups. Enter bin size.

Bin size *

2

Reset to default

OK

Cancel

This visualization shows the rating distribution on the X axis we have ranges like 0-2, 2-4, 4-6, 6-8, 8-10 and exactly 10. On the Y axis we have the number of movies that fall into each rating bin. For this visualization I've binned the vote average field with the size of two, ranging from 0 to 10. The rating buckets are 0-2, 2-4, 4-6, 6-8, 8-10 and exactly 10.

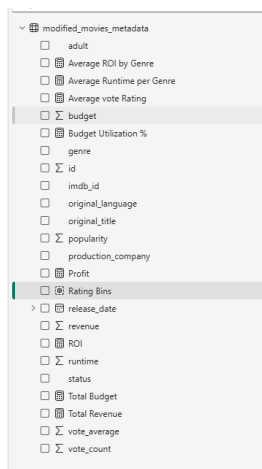
Insights:

By observing the chart, most movies are rated between 4 to 8, which captures 23,000 of the total movies out of 27,000. Specifically, 6 to 8 range is the peak, with 13,000 movies rated here. It is also strong with 1000 entries. Only 1000 movies scored between 8 to 10 and even fewer 100 hit a percent score of 10. Similarly, only 1500 scored between zero to two indicating few movies are universally disliked. Ratings show a bell-shaped distribution towards the center.

Storytelling Approach:

Stage 1: understanding and Exploring the Dataset:

What does raw data look like?



The dataset, as seen in the Fields pane, is from a table called `modified_movies_metadata`. It contains attributes such as:

Identifiers: `id`, `original_title`, `imdb_id`

Descriptive: `genre`, `original_language`, `production_company`

Performance metrics: `budget`, `revenue`, `vote_average`, `vote_count`, `runtime`, `ROI`, `Profit`, `Budget Utilization %`

Calculated fields: `Average ROI by Genre`, `Rating Bins`, etc.

What key variables and trends stand out?

Revenue & Profit Trends: Movies have consistently generated strong profits, with a total revenue of 425bn and profit of 255bn. Walt Disney, Paramount, and Universal are the top earners.

Viewer Sentiment: Average rating is 5.66 across 27,038 movies. Majority of movies fall in the 6-8 rating bin, indicating a large volume of moderately liked content. Animation, Music, and History genres received the highest ratings.

Language & Genre: English (69.9%) dominates, but foreign-language films also show strong niche performance. Performance of genres varies by language. e.g., Romance in Russian vs. Comedy in Korean.

Viewer Engagement: Vote count peaked around 2014-2018 and dropped off in 2020 (likely due to pandemic). High vote count doesn't always equal high ratings, popularity and quality are not strictly correlated.

Stage 2: Data Processing and Correlation Analysis

In this stage, we move from descriptive analysis to uncovering underlying patterns and relationships between variables. Key operations include:

- Cleaning missing or inconsistent values
- Merging rating and metadata tables
- Creating new fields (e.g., Rating Bins, ROI, Profit)
- Grouping and binning numeric fields for trend analysis

Correlation Analysis:

Vote Count vs Rating vs Revenue (Bubble Chart): High revenue movies often lie in the mid-rating zone (5–7), not necessarily the highest-rated. Some well-rated movies have low vote counts, revealing undiscovered gems.

Genre vs Average Rating: Top genres in ratings: Animation, Music, History. Bottom genres: TV Movie, Horror. Supports genre-based content curation or investment strategies.

Runtime vs Rating: Longer movies may have better storytelling and thus higher ratings. To be validated in later phases with scatter or trend lines.

Production Companies: Revenue vs Profit: Walt Disney leads with high revenue and profit efficiency. Marvel Studios shows high ROI with fewer productions. Smaller companies tend to have lower revenue-per-movie.

Work Management:

As mentioned in the proposal below, tasks have been completed for the project 1:

- Explored metadata features like genres, average ratings, popularity
- Detected basic correlations such as genre vs. average rating, release year vs. popularity
- Identified outliers and missing data
- Generated initial visual dashboards

Data Cleaning & Transformation:

- Handle missing or inconsistent values in the movies_metadata.csv file (e.g., missing runtimes, differently formatted release dates).
- Convert data types (e.g., release dates to datetime, budget/revenue to numeric).

Exploratory Data Analysis (EDA):

- Analyze the distribution of movie genres, release years, and popularity scores.
- Visualize the average user rating across different genres and decades.
- Create basic visualizations using Power BI

Initial Correlation Analysis:

- Identify relationships between variables such as:
- Movie runtime vs. average rating
- Budget/revenue vs. popularity
- Number of ratings vs. movie rating score

Name of the person	Responsibility taken	Percentage distribution
Sashidhar Chary Viswanathula	Worked on data cleaning, transformation, two visualizations and documentation	40%
Nithish Karanam	Worked on four visualizations, IEEE paper, PPT and documentation	30%
Vidya Reddy	Worked on two visualizations and documentation	30%