

A man and a woman are dancing against a night sky with mountains in the background. The woman is wearing a yellow dress and the man is wearing a white shirt and a dark tie. They are both pointing their arms upwards.

EXPLORATORY DATA ANALYSIS OF TMBD DATASET

AND

REVENUE PREDICTOR
USING LINEAR REGRESSION METHOD

BY :

AHMAD FADLAN AMIN


A close-up, profile view of a man with a light beard and brown hair, looking out towards the right. He is wearing a white shirt. The background is a blurred outdoor festival scene with other people and structures.

TABLE OF CONTENT

PROJECT OVERVIEW

DATA SET

DATA CLEANING

EXPLORATION AND
ANALYSIS

REVENUE PREDICTOR

ABOUT AUTHOR



project OVERVIEW

This project showcase data exploration, analysis and visualization of TMDB Movie Data Set from Kaggle using python librabries' such as pandas, seaborn and matplotlib etc.

Area of analysis consists of timeseries analysis, linear regression and correlation.

Using multiple regression model and FastAPI, this project features a revenue predictor API that predict film revenues based off its budget and popularity.

matplotlib

pandas



FastAPI



kaggle

Github
Repository



DATA SET

TMBD(The Movie Database) contains movies from from the year 1961 – 2015.

This dataset consists of 24 columns and 1287 rows of data for each films.

This data set contains 1 null value and no duplicated data

Author : Success Ikuku

kaggle

**link to kaggle database*

- Genres: Category
- Production companies: Companies that produced the movie
- Release date: Date movie was released
- Vote count:
- Vote average
- Release year: year movie was released
- Budget adj: Budget In terms of 2010 dollars
- Revenue adj: Revenue in terms of 2010 dollars

- ID: Movie ID
- Popularity: Popularity score
- Budget: Amount spent to make the movie
- Revenue: Amount realized from the movie
- original title: Movie title
- Cast: Actors & Actresses in the movie
- Homepage: Movie website
- Director: Director(s) of the movie
- Tagline: Catchphrase(s)/slogan of the movie
- keywords: Words associated with a movie
- Overview: Movie Summary
- Runtime: length of a movie

```
#Check if there's null value  
df.isna().sum()
```

```
imdb_id      0  
popularity    0  
original_title  0  
cast         0  
director     0  
runtime      0  
genres       0  
production_companies  0  
release_date  0  
vote_count   0  
vote_average  0  
release_year  0  
budget_adj   0  
revenue_adj   0  
profit       0  
popularity_level  1  
dtype: int64
```

```
#Check for duplicated value  
df.loc[df.duplicated()]  
  
#There's no duplicate value
```

```
imdb_id popularity original_title cast director runtime genres production_companies release_date vote_count vote_average releas
```


DATA CLEANING

**click title above to the full pyhton code github link*

- Check for Null and Duplicated Values
 - Removing Unnecesary Columns
 - Setting appropriate data type
 - Adding numerical category for cast, genres, category, popularity and production company
-
- Converting columns with multiple values, and set it to 1 value for genres, production_companies and casts
 - The first value of the columns will be considered as the 'main' values that will be used for further analysis

company_code	cast_code	genres_code	director_code	popularity_code
299	111	0	128	4
302	593	0	236	4
275	553	1	613	4

genres	production_companies
Action Adventure Science Fiction Thriller	Universal Studios Amblin Entertainment Legenda...
Action Adventure Science Fiction Thriller	Village Roadshow Pictures Kennedy Miller Produ...
Adventure Science Fiction Thriller	Summit Entertainment Mandeville Films Red Wago...

to

genres	production_companies
Action	Universal Studios
Action	Village Roadshow Pictures
Adventure	Summit Entertainment

cast
Chris Pratt Bryce Dallas Howard Irrfan Khan Vi...
Tom Hardy Charlize Theron Hugh Keays-Byrne Nic...
Shailene Woodley Theo James Kate Winslet Ansel...

to

cast
Chris Pratt
Tom Hardy
Shailene Woodley

CLEANED DATA SET

This is the cleaned Dataset that will be used for analysis.

- This data set contains data for 1286 films, from the year of 1961 to 2015
- For budget, revenue and profit, adjusted 2010 data will be used
- Average budget for each film is 54M, with revenue of 199M and 124M in profit
- Average ratings of film is 6.0/10 with 110 minutes run time.
- There are over 300 production company and

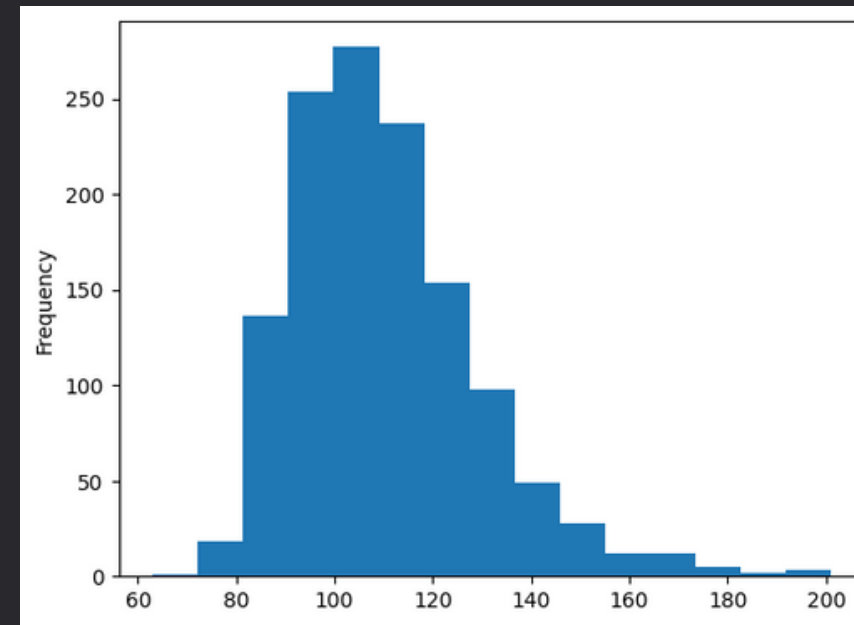
	popularity	runtime	release_date	vote_count	vote_average	release_year	budget_adj	revenue_adj	profit	company_code	cast_code	genres_code	director_code
count	1286.0	1286.0	1286	1286.0	1286.0	1286.0	1286.0	1.286000e+03	1.286000e+03	1286.0	1286.0	1286.0	1286.0
mean	2.0	110.0	2008-06-27 15:10:21.461897472	948.0	6.0	2007.0	54650580.0	1.992828e+08	1.243079e+08	162.0	318.0	5.0	392.0
min	0.0	63.0	1972-03-15 00:00:00	10.0	2.0	1961.0	1.0	4.300000e+01	-4.139124e+08	-1.0	-1.0	0.0	0.0
25%	1.0	97.0	2006-03-09 06:00:00	179.0	6.0	2005.0	15191800.0	2.754364e+07	3.133386e+06	64.0	158.0	1.0	198.0
50%	1.0	107.0	2009-10-10 00:00:00	440.0	6.0	2009.0	35571641.0	8.689619e+07	4.532795e+07	174.0	319.0	3.0	390.0
75%	2.0	121.0	2012-01-19 00:00:00	1174.0	7.0	2011.0	76336859.0	2.351241e+08	1.471200e+08	278.0	471.0	6.0	586.0
max	33.0	201.0	2071-12-22 00:00:00	9767.0	8.0	2015.0	425000000.0	2.827124e+09	2.544506e+09	321.0	628.0	17.0	787.0
std	2.0	19.0	NaN	1256.0	1.0	8.0	55271158.0	2.969429e+08	2.184179e+08	112.0	181.0	4.0	224.0

EXPLORATORY DATA ANALYSIS

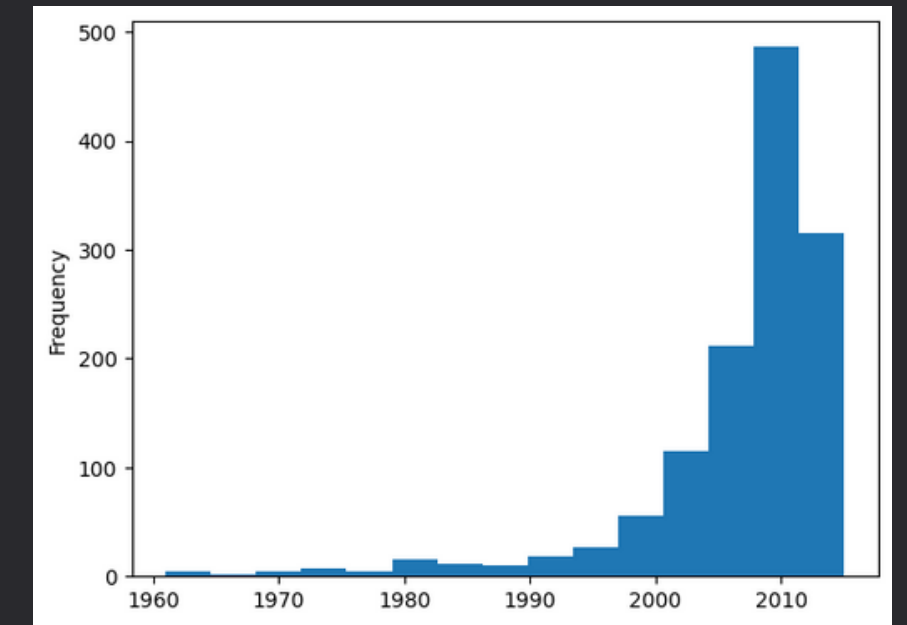
**click title above to the full pyhton code github link*

DATA DISTRIBUTION

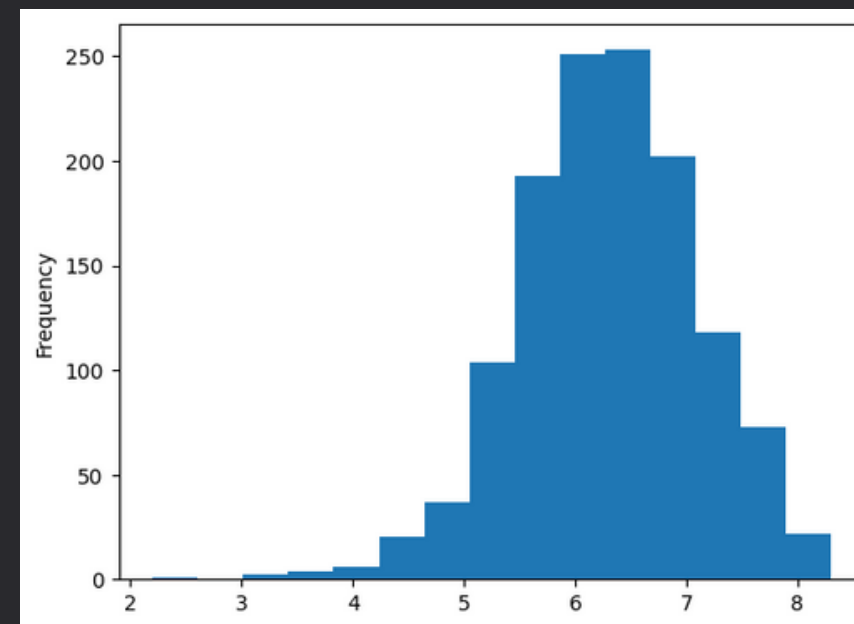
- The graph shows that most film have an average ratings of 6 - 7.8
- Common runtime for films is range between 90 - 130 minutes
- Most of the films in this data set release after the year 2000
- and have less than 3000 vote



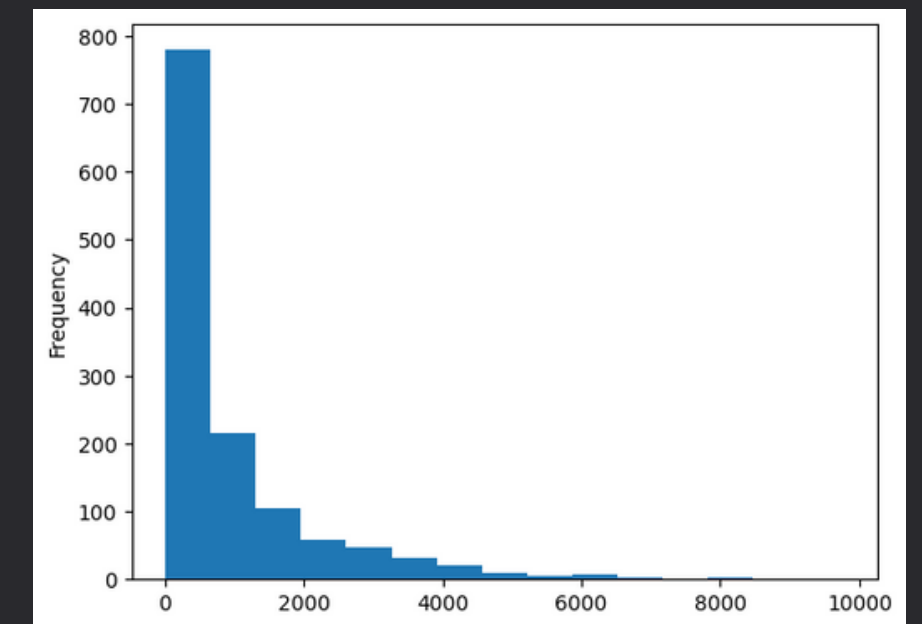
Runtime



Release Year

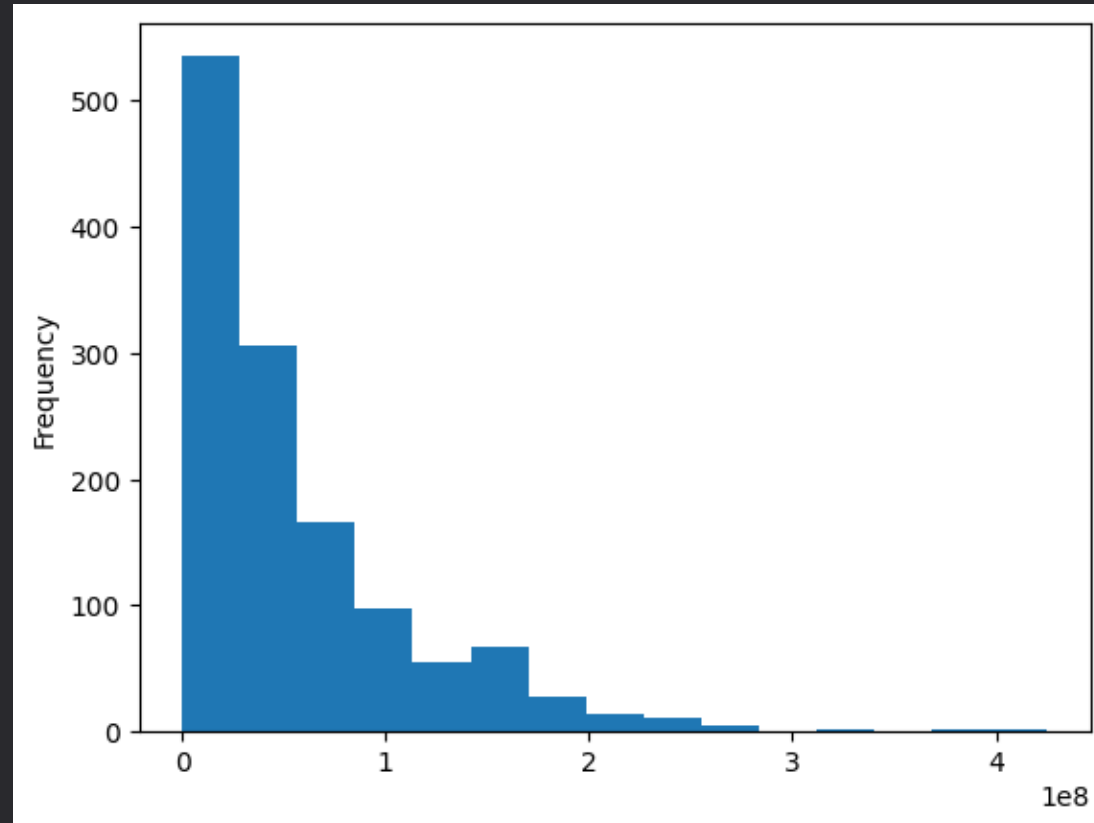


Vote Average
(ratings)

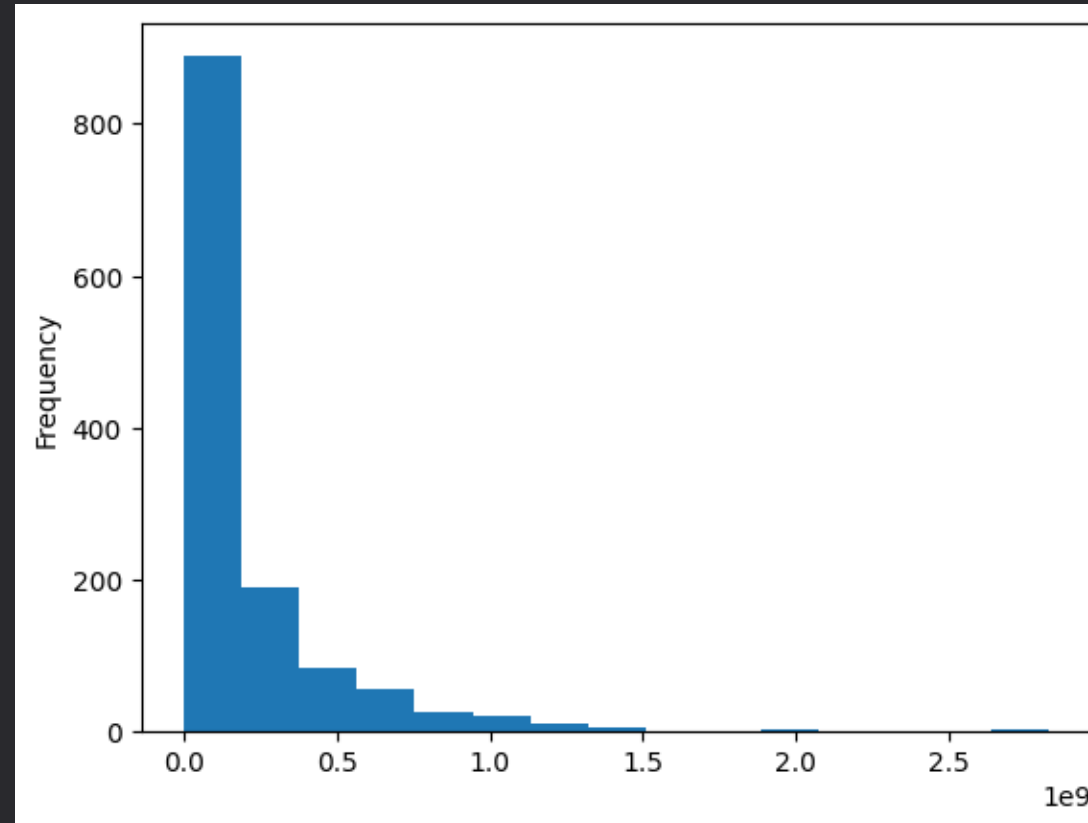


Vote Count

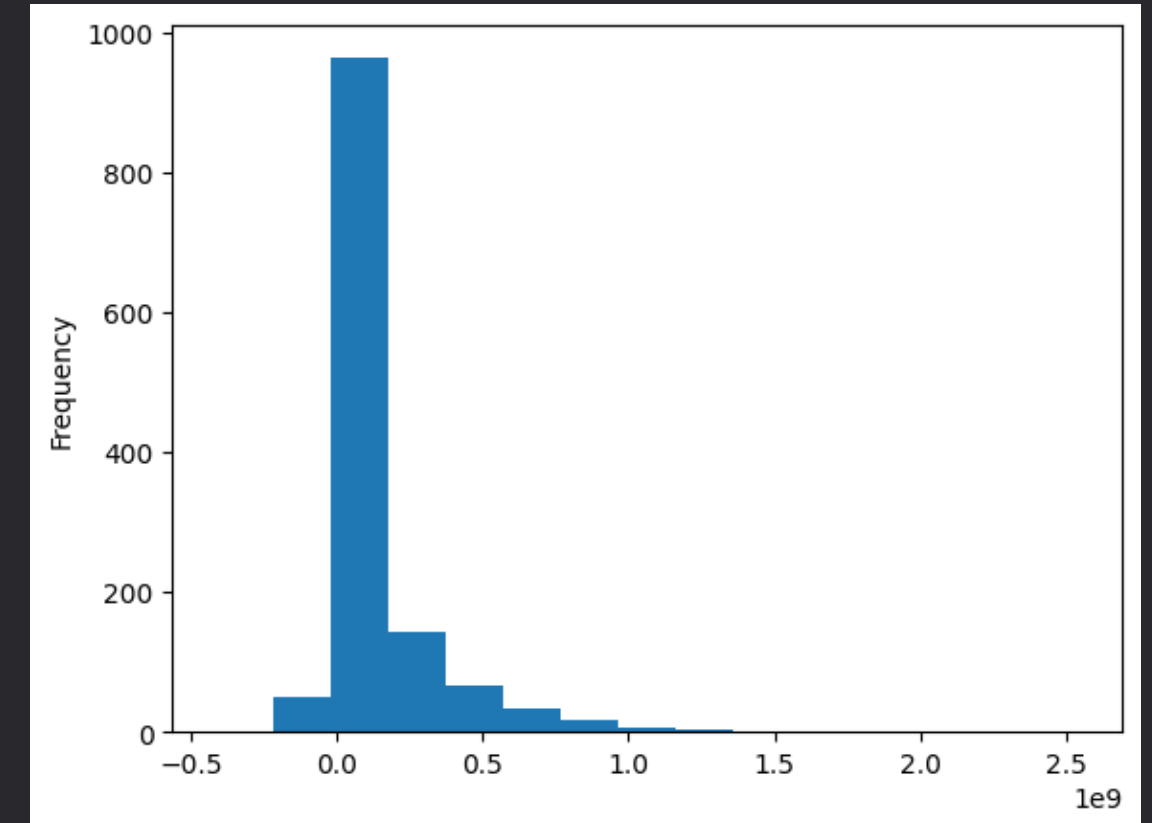
DATA DISTRIBUTION



Budget



Revenue

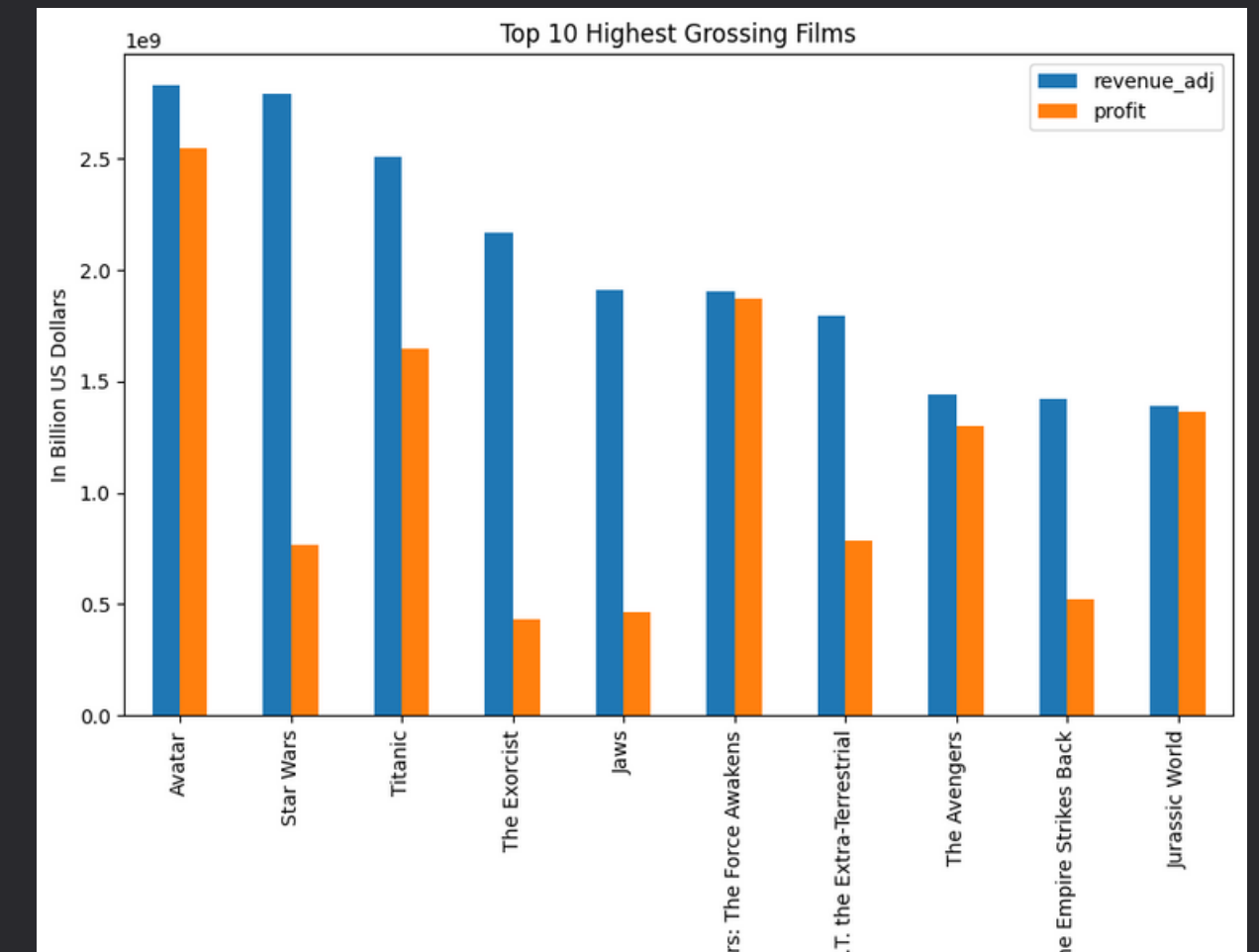
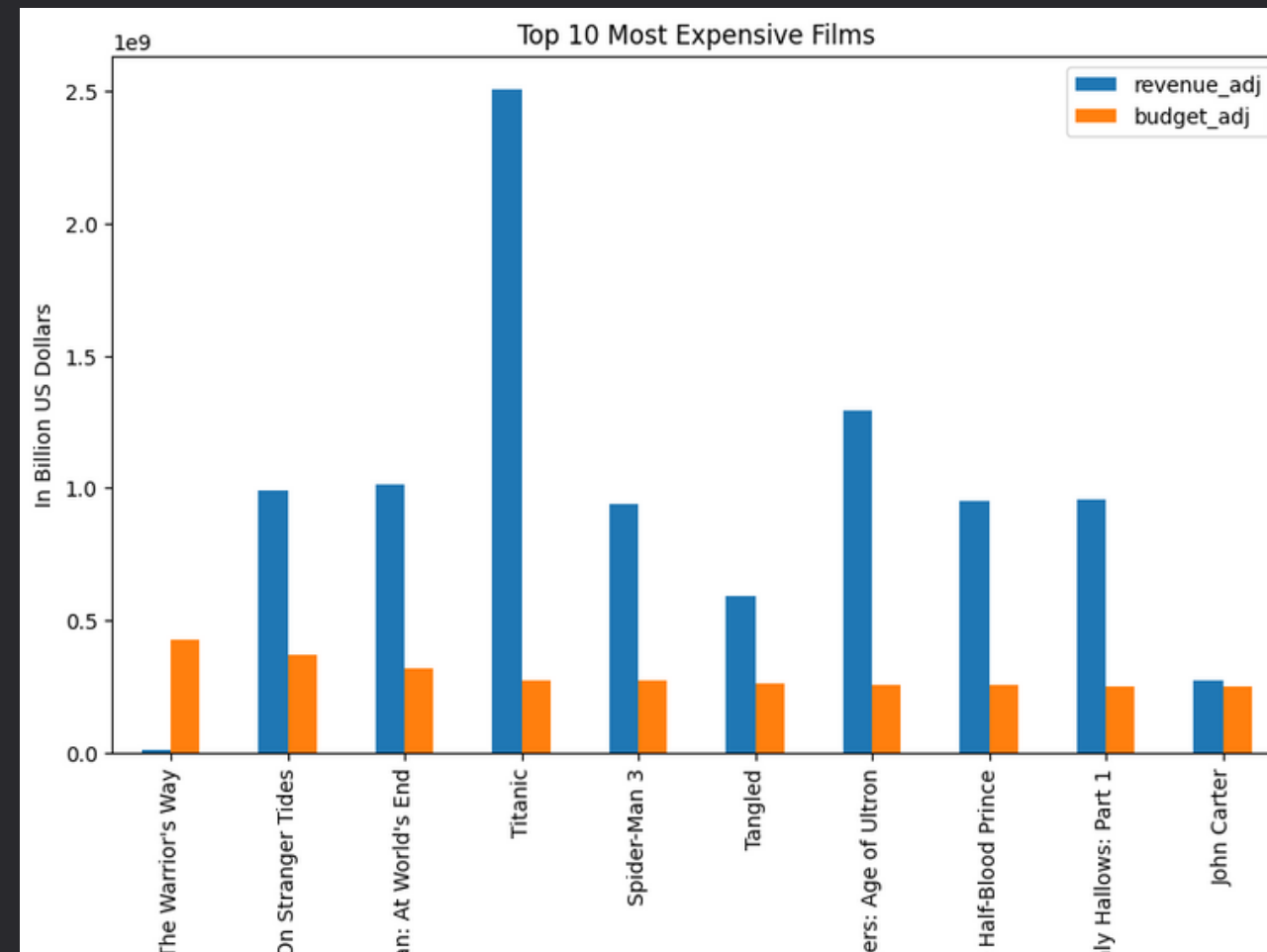


Profit

- Most of the films have a budget of 100M dollar, with around 500M dollar in revenue.
- Most common profit is around 200M dollars, with some films have a loss around 100M
- The Warrior's Way is an outlier with budget of 400M, revenue in 10M and loss in 300M

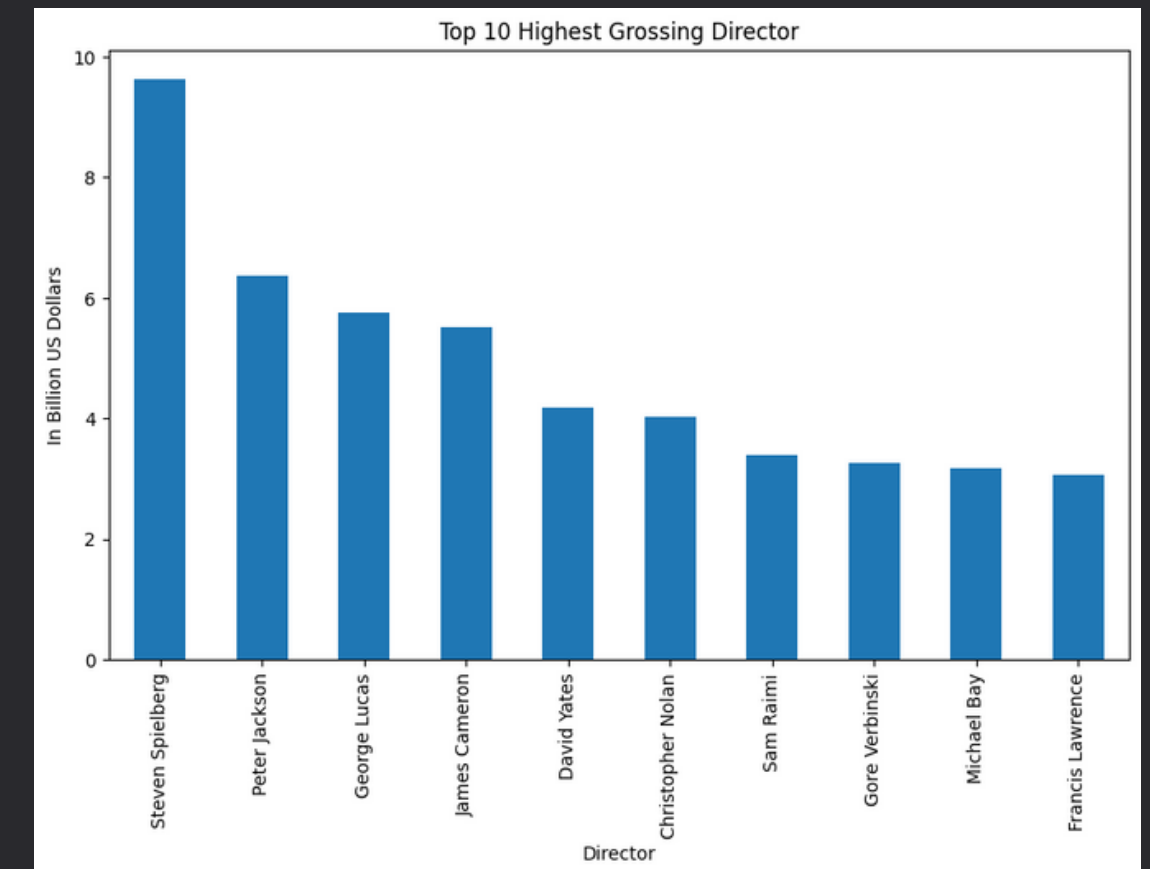
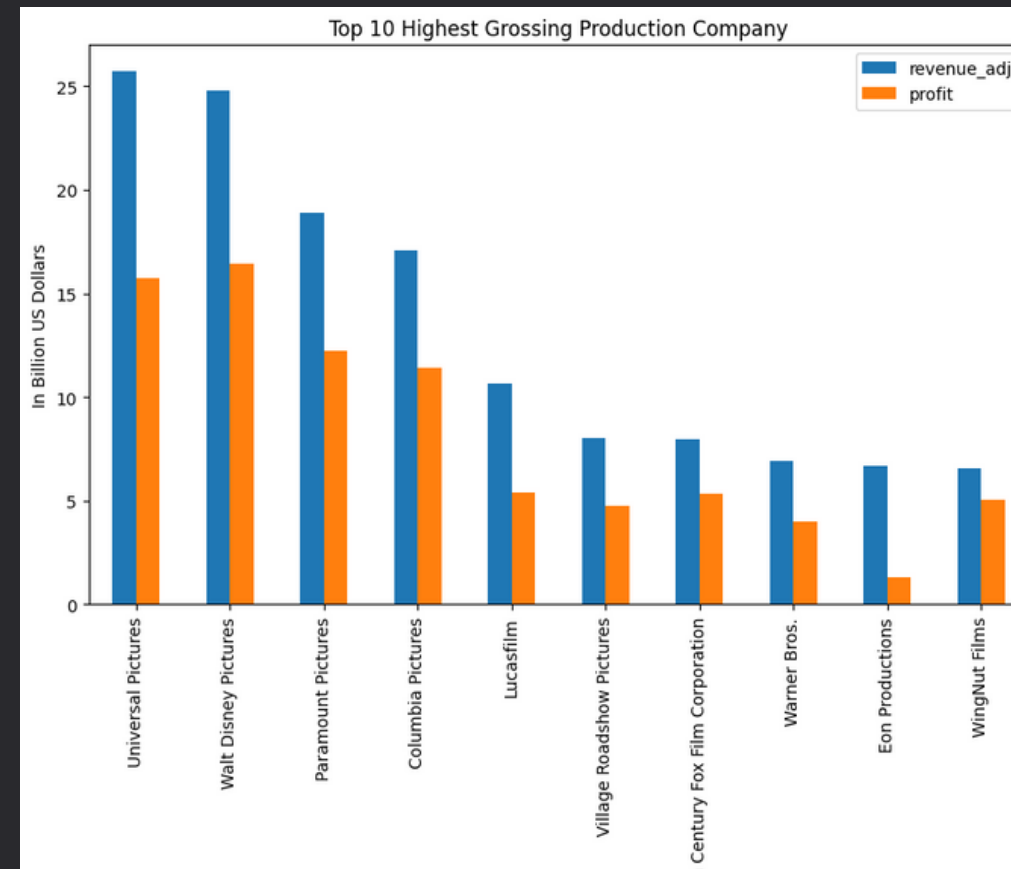
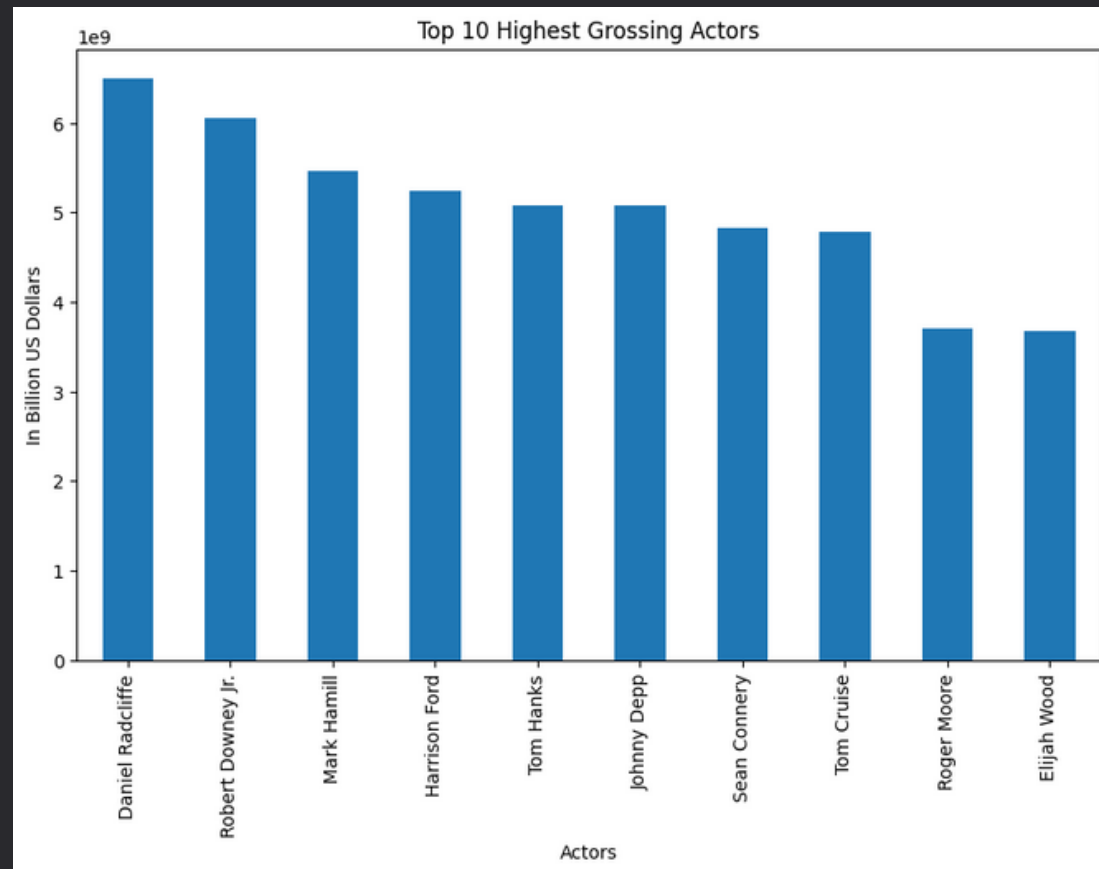
DATA EXPLORATION

Q : WHAT ARE THE TOP GROSSING AND THE MOST EXPENSIVE FILMS ?



- Avatar is the highest grossing film with revenue at almost 3 billion dollar, it's also the most profitable film with profit around 2.5 billion. Star Wars and Titanic followed Avatar as the 2nd and 3rd highest grossing film, respectively.
- The Warrior's Way has the highest budget with around 400 million dollars, and has the biggest loss of almost 390M. Pirates of The Caribbean : On Stranger Tides and Man of Steel followed as films with highest budget.

Q : WHAT ARE THE TOP GROSSING ACTORS, DIRECTOR AND PRODUCTION COMPANY ?

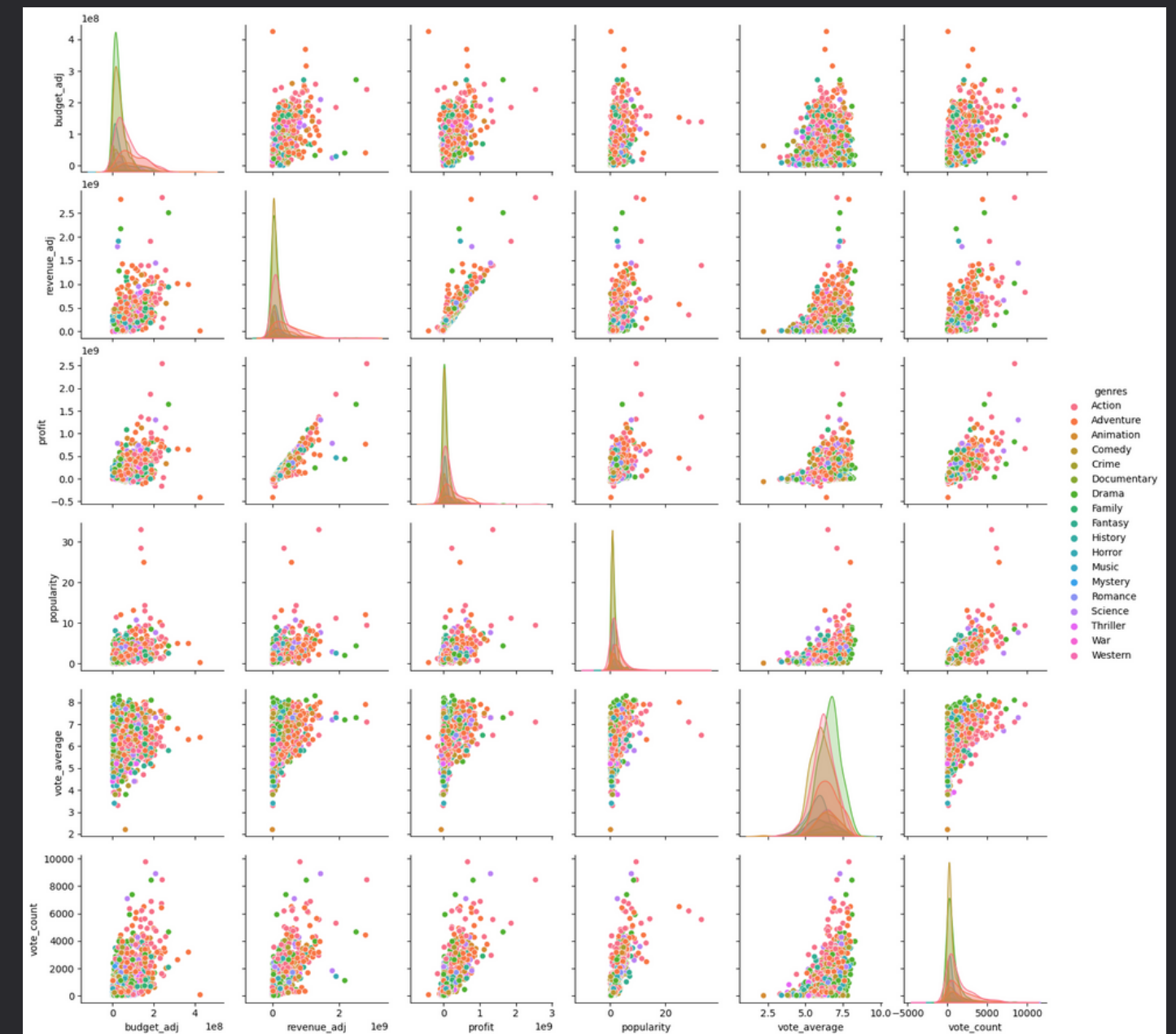


- Daniel Radcliffe is the top grossing actor, that has been on films with total revenue over 6 billion dollars, followed by Robert Downey Jr and Mark Hamill
- Steven Spielberg is the highest grossing director. He has directed films with total revenue at almost 9 billion dollars, followed by Peter Jackson and George Lukas
- Universal Pictures is a production company that produce the highest revenues, with over 25 billion dollars in total revenue, and 15 billion in profit. Walt Disney and Paramount followed as the 2nd and 3rd top grossing production companies, respectively.

Q : WHAT ARE VARIABLES THAT HAVE SIGNIFICANT / INSIGNIFICANT CORRELATION ?

CORRELATION PAIR-PLOT

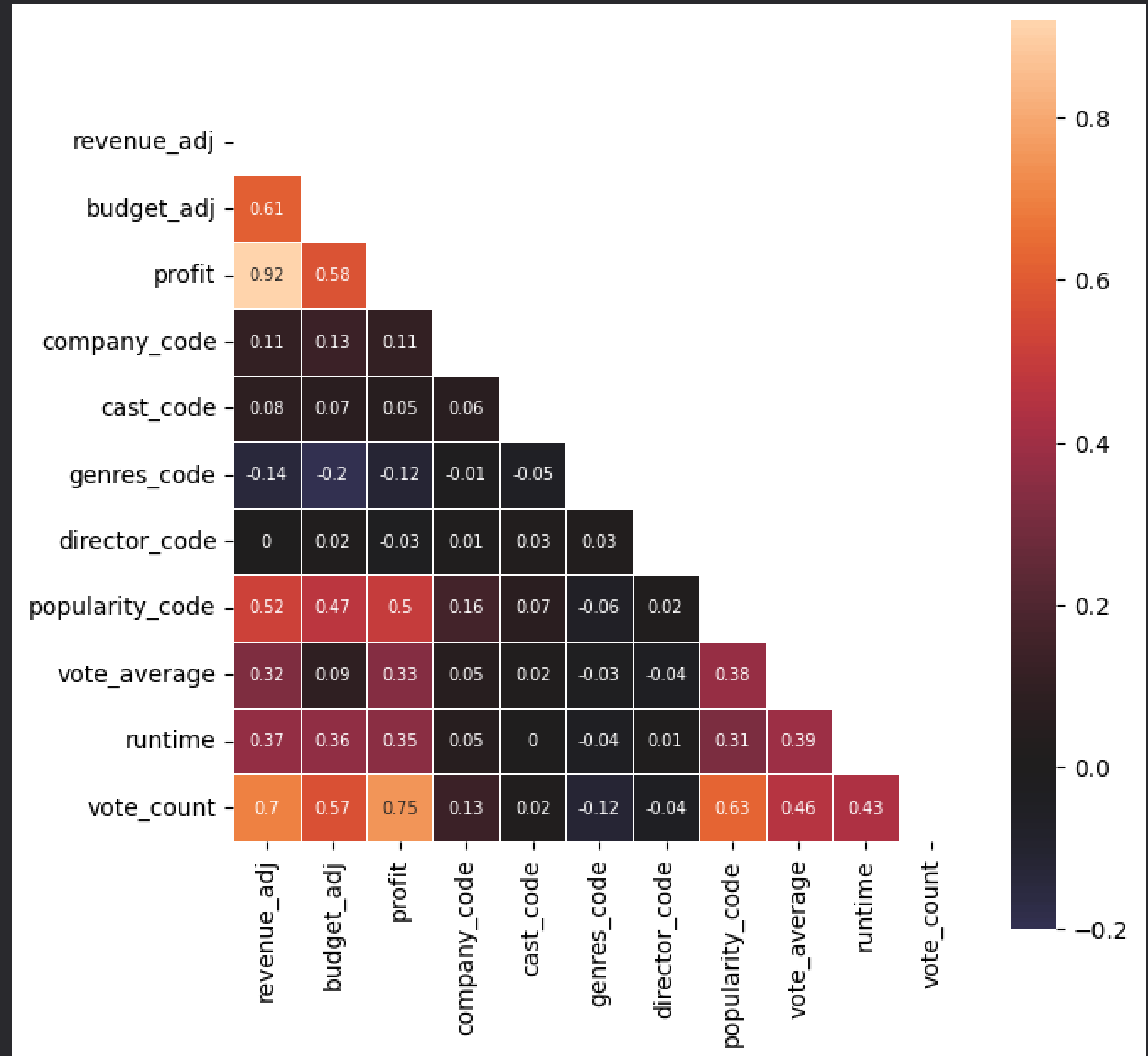
The Pair-Plot shows the data distribution and relationship between each other of :
'*budget_adj*', '*revenue_adj*',
'*profit*', '*popularity*',
'*vote_average*', '*vote_count*',
and shows in a scatter plot in a matrix form



HEATMAP

CORRELATION

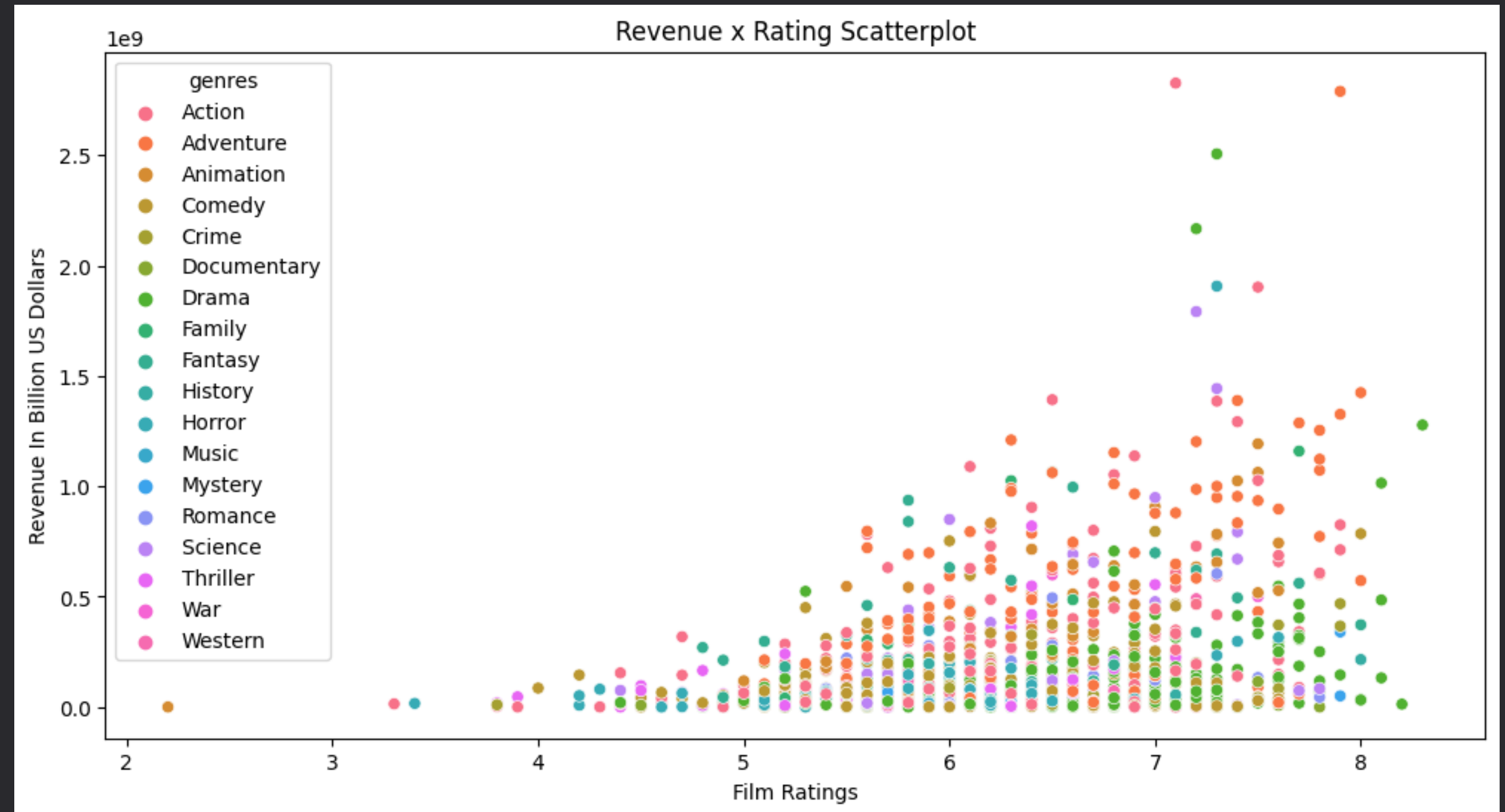
The Correlation heatmap shows how significant (or insignificant) each feature correlate to each other, the features consists of : *'revenue_adj'*, *'budget_adj'*, *'profit'*, *'company_code'*, *'cast_code'*, *'genres_code'*, *'director_code'*, *'popularity_code'*, *'vote_average'*, *'runtime'*, *'vote_count'*.



INSIGHTS

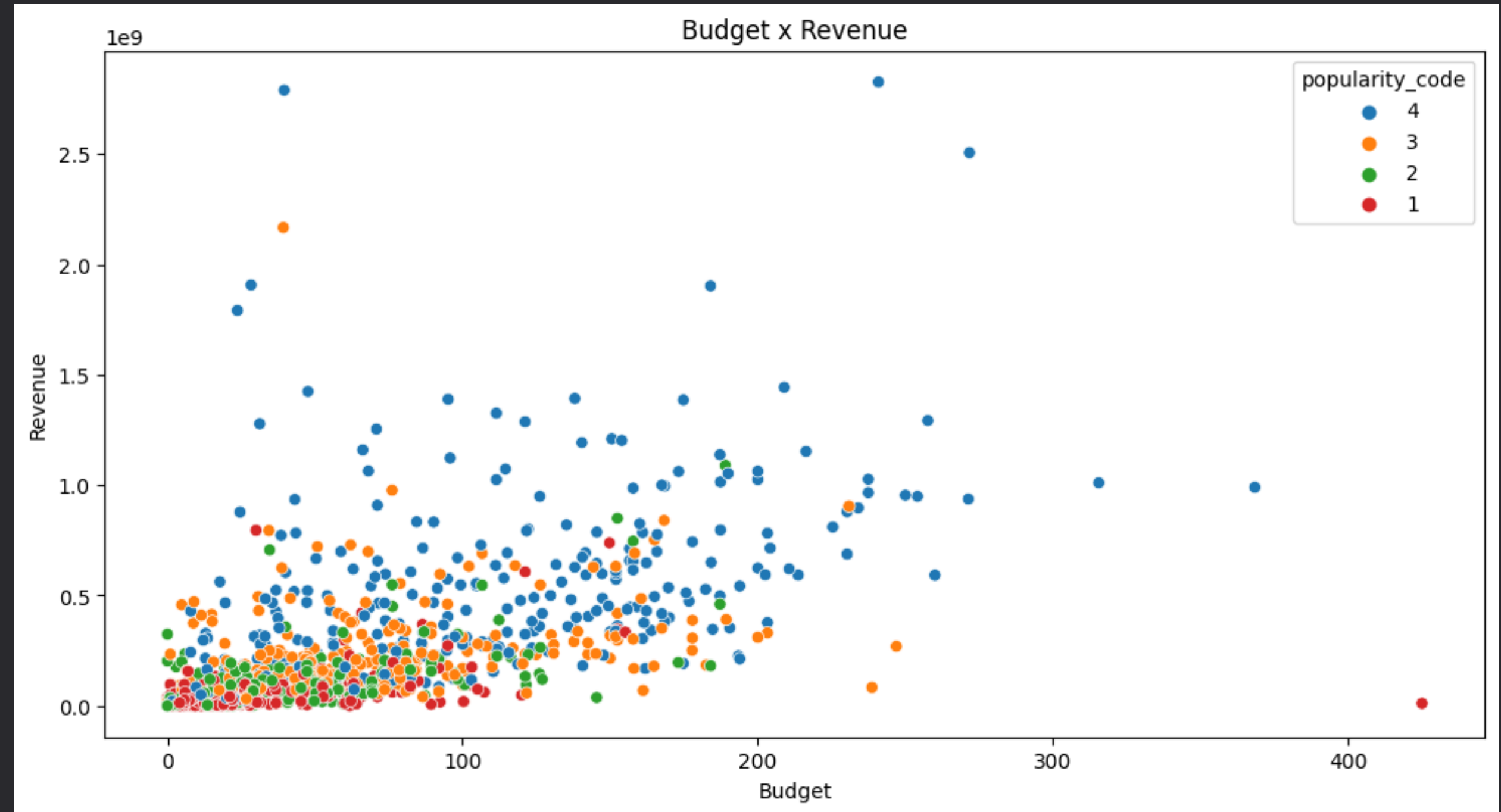
- There's no significant (weak) correlation between casts, director and production company over films revenue, profit and popularity.
- Vote Count and Popularity has adequate-strong correlation over revenue and profit
- Genres, production company, director and casts has no significant correlation with any other values
- Runtime and vote average (film ratings) have weak correlation over popularity
- Runtime has weak correlation over revenue and film ratings (vote Average)

RELATIONSHIP OF REVENUE AND RATING



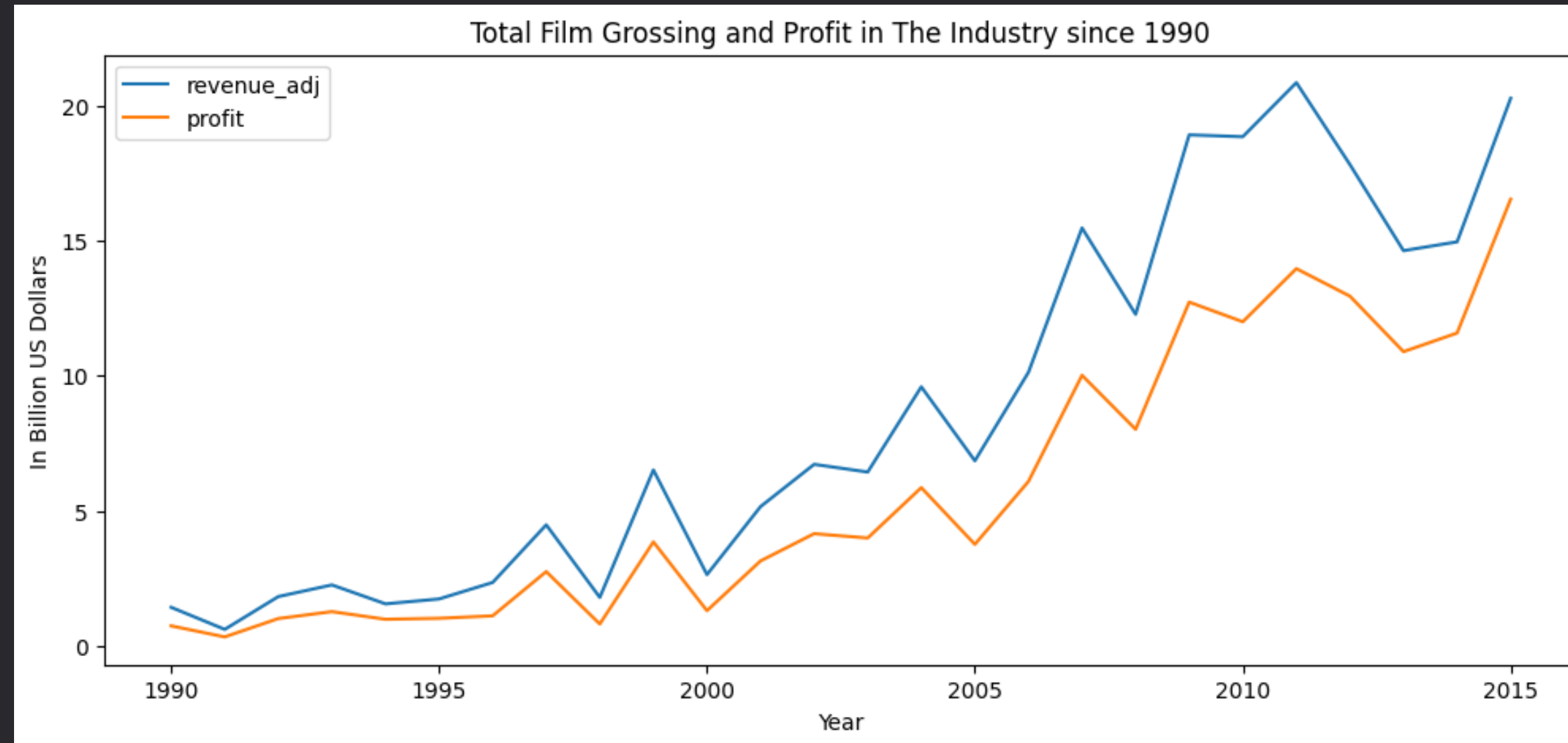
The scatterplot visualize relationship between film ratings and revenue. The graph shows that films with higher ratings, generate more revenue, although it's not a significant increase.

RELATIONSHIP OF REVENUE AND POPULARITY SCORE



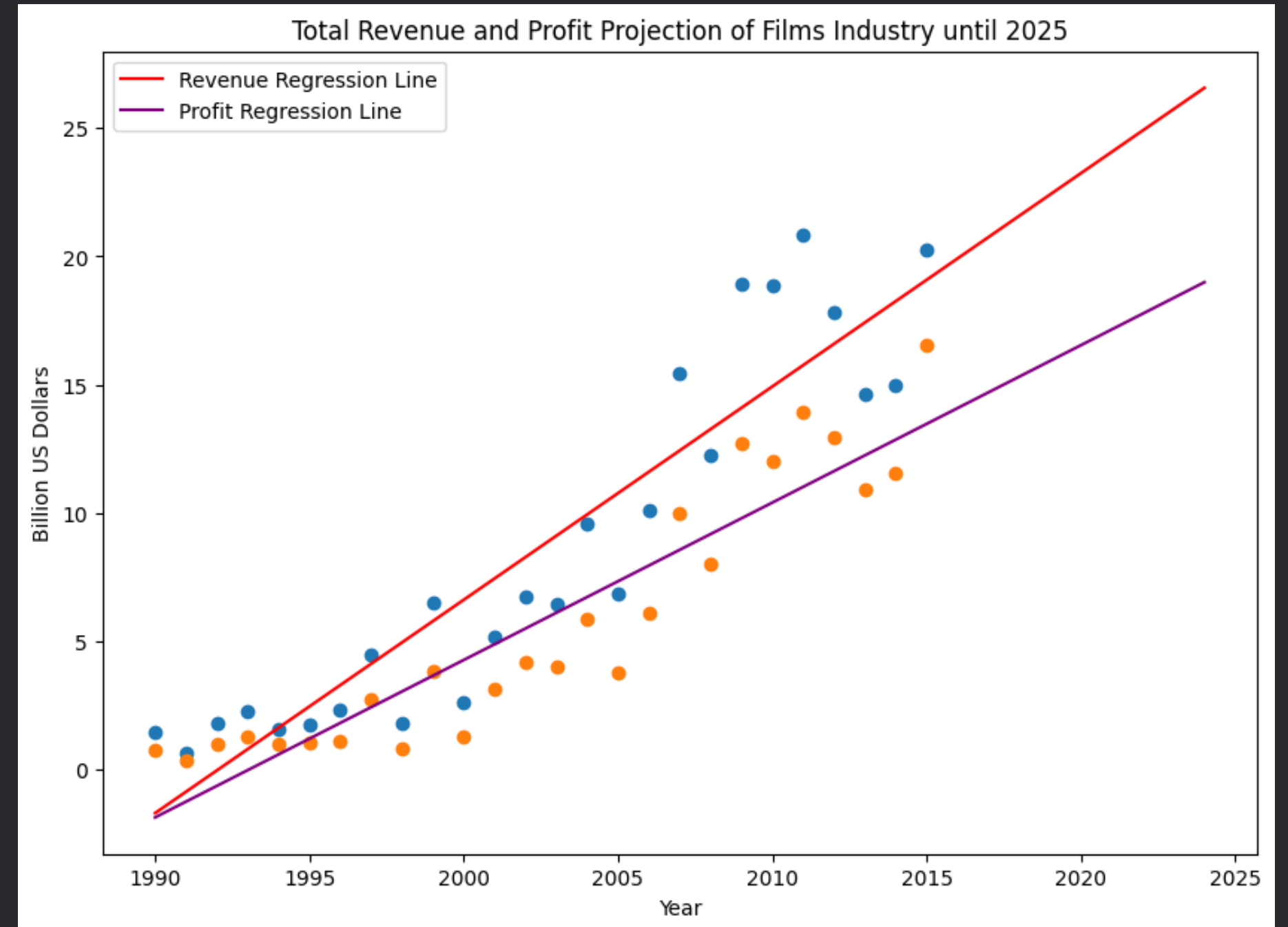
The scatterplot showcase relationship between budget and revenue, grouped by its popularity. It's shown that, films with higher budget and that are popular tends generates more in revenue.

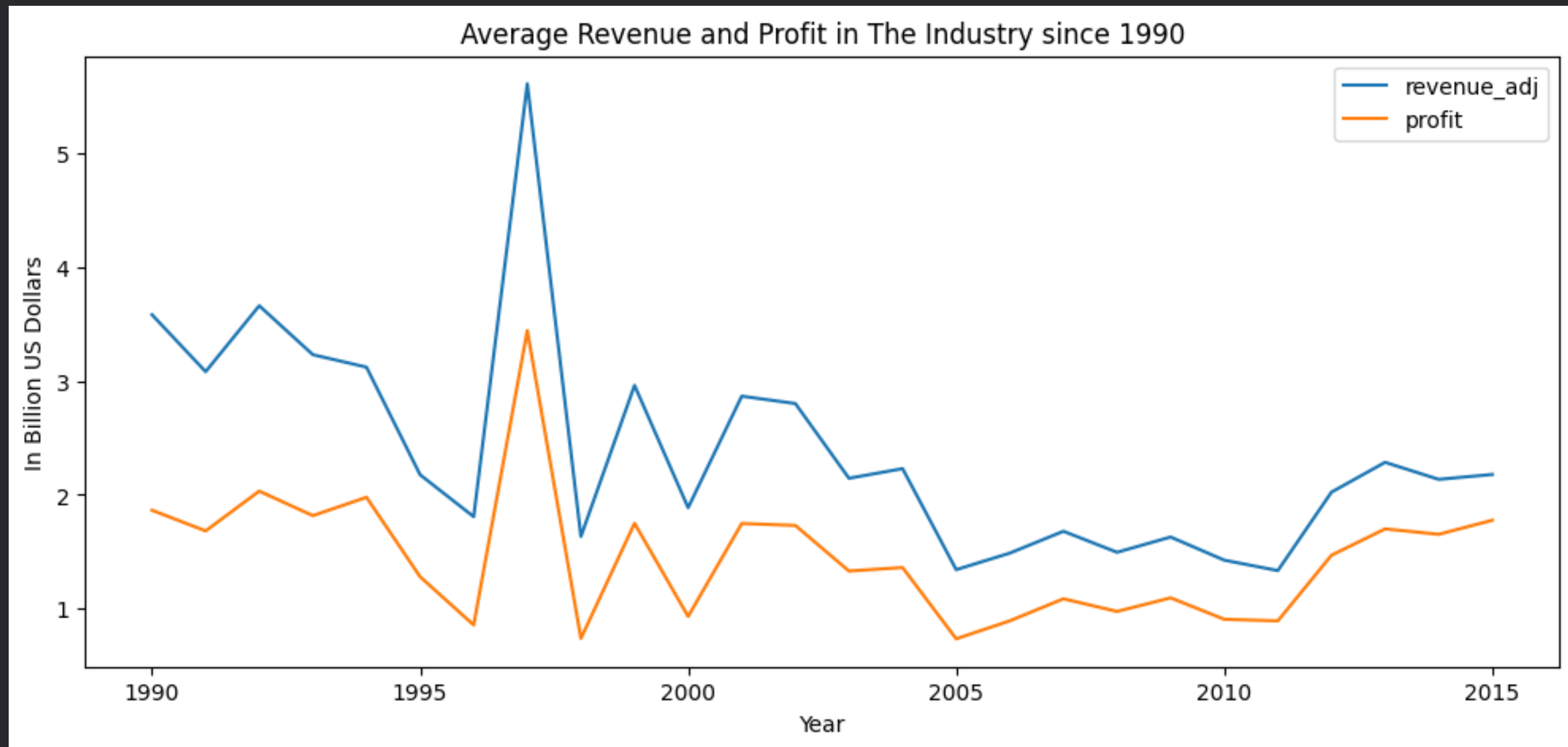
Q : DESCRIBE THE PROFITABILITY OF FILM INDUSTRY FROM 1990 - 2015, AND ITS PROJECTION UNTIL THE YEAR 2025



The figure above visualizes the timeseries of **total** film revenue and profit, throughout the year 1990 until 2015. From 1990 to 2000, there's a constant, but small increase, peaking in around 7 billion dollars. And through the year 2000 until 2015 there's a big increase in total revenue and profit, peaking in around 22 billion dollars.

Using *Linear Regression*, there's an increase in trend of total revenue and profit in the film industry. In 2025, It's projected to reach a total of around 27 billion dollar in revenue and 17 billion dollar in profit.

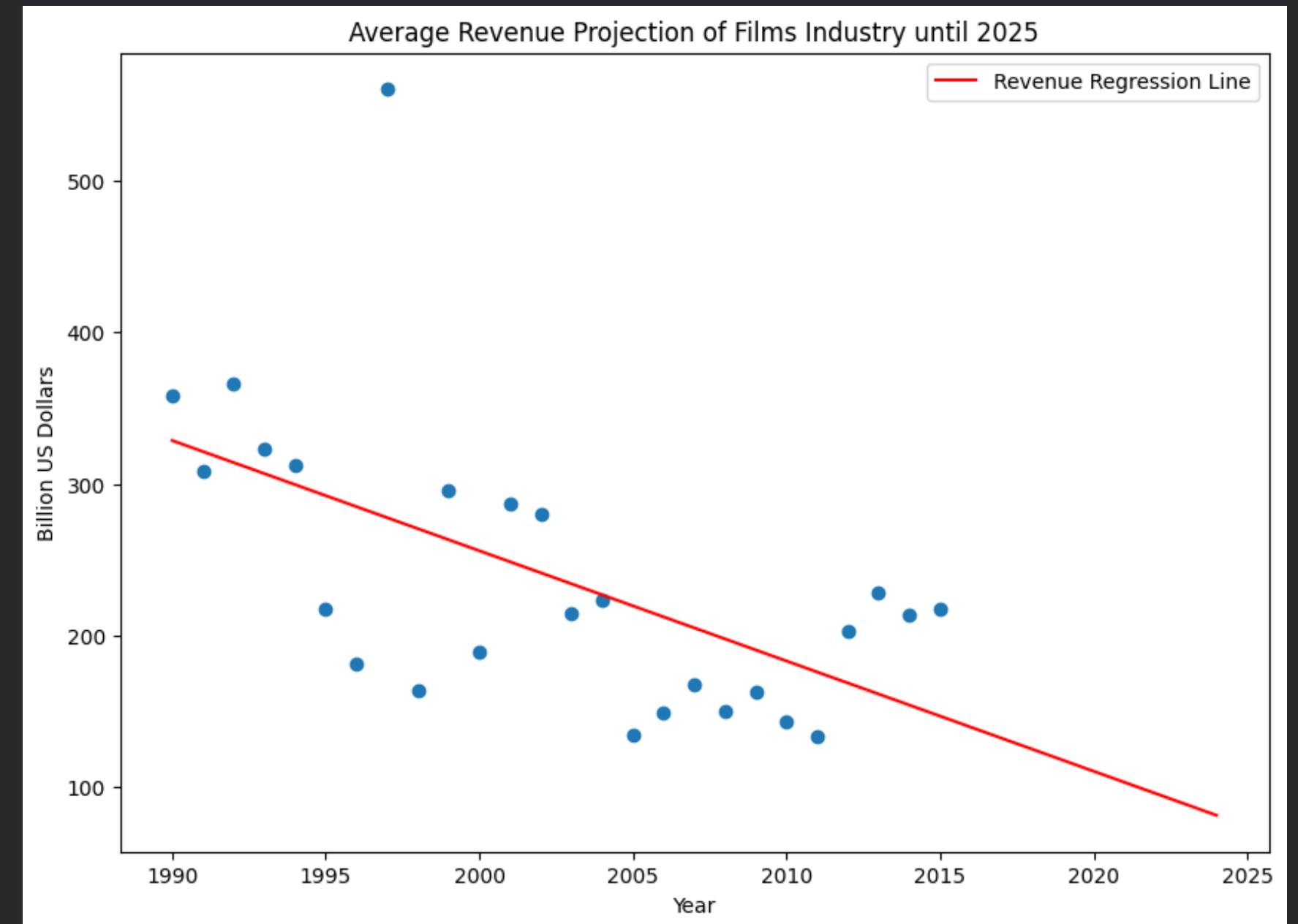


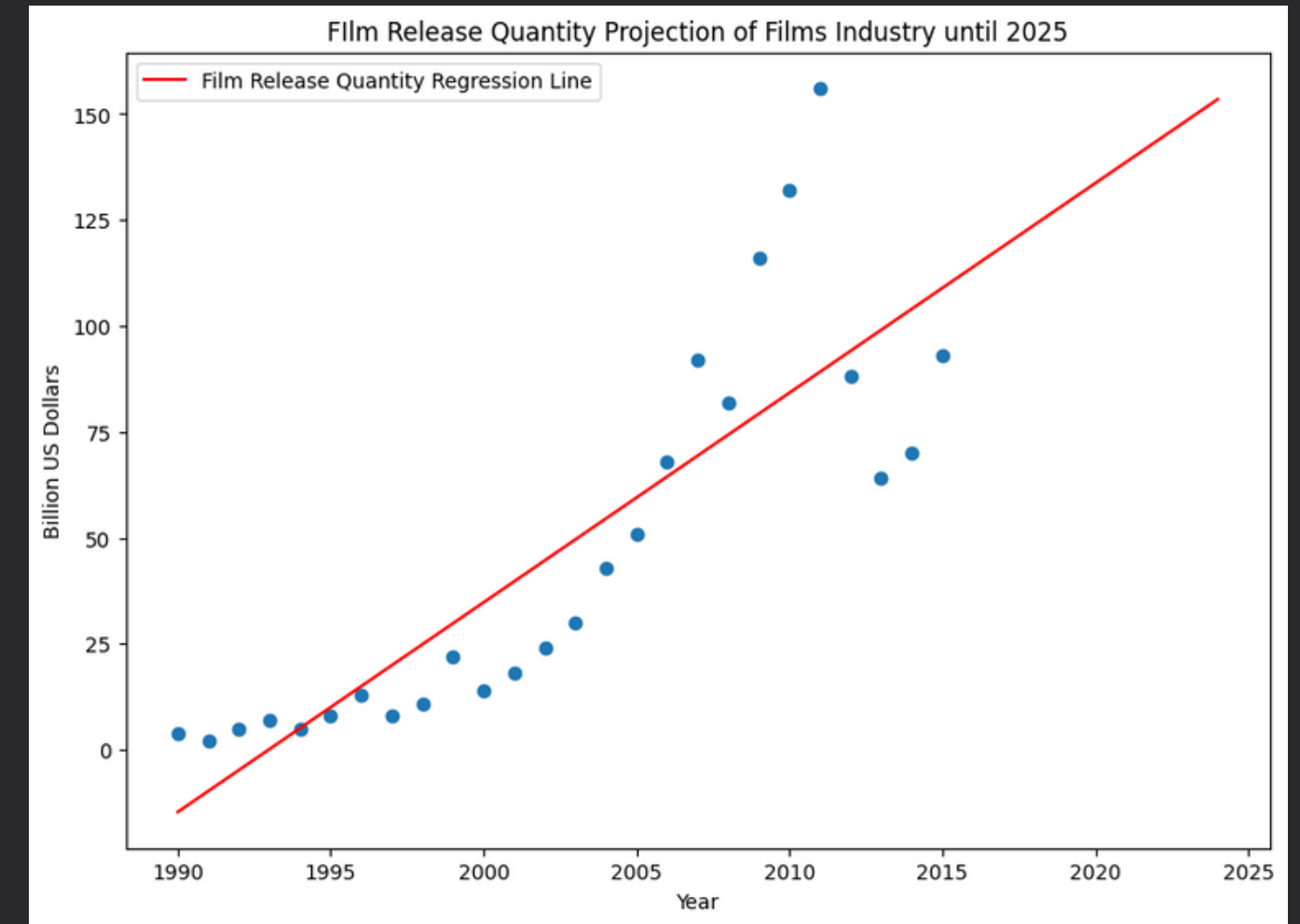
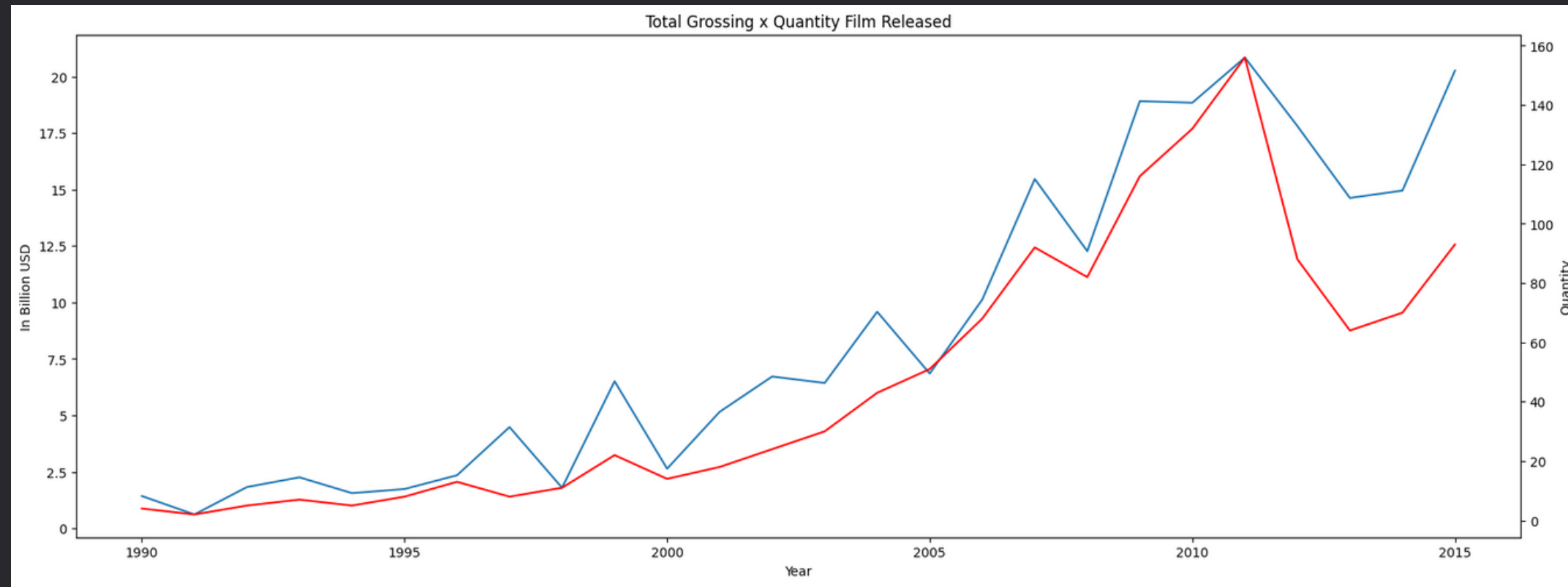


But, if we look at the **average** revenue and profit from each films, shown in the figure above, there's a decreasing trend. Aside from a massive increase that happened in 1997, peaking at average 500M dollar, there's a steady decrease throughout 1999 through 2010. Lowest average revenue occurs in 2005 with around 130M dollars, and an increase the following year.

Using *Linear Regression*, it is projected that on average revenue and profit will have a meaningful decrease trend. In 2025, average revenue is projected to be around 100M.

If we take a step back, there's an anomaly where **total** and **average** revenue graph **contradict** each other..





The anomaly that happened between total and average profitability can be explained through the total film release graph. Throughout the year of 2000 until 2012, there's a significant increase in quantity film release, and it is projected that it will keep increasing until 2025, with a total film released of 150.

The significant increase in quantity of quantity film released creates an increasing trend in total film revenue, and the average revenue decreased.

REVENUE PREDICTOR API

USING LINEAR REGRESSION MODEL

**click title above to the full pyhton code github link*

DATA AND MODEL SET UP

```
#with train test split
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

1] ✓ 1.5s

#Data that will be used for linear regression model is from 1990
df_linreg = df[['release_year', 'budget_adj', 'revenue_adj', 'popularity_code']].sort_values('release_year')
df_linreg = df_linreg[df_linreg['release_year'] > 1989]

4] ✓ 0.0s

#Setting up and Model Training
x_reg = df_linreg[['budget_adj', 'popularity_code']]
y_reg = df_linreg[['revenue_adj']]
xtrain, xtest, ytrain, ytest = train_test_split(x_reg, y_reg, test_size=0.15, random_state=90)

model = LinearRegression()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
#score shows coefficient of determinitaion atau r2 (rsquared)

3] ✓ 0.0s

0.661496394929237
```

- SKlearn, LinearRegression model & train_test_split used to set up the model
- Data used is from the year 1990 onwards
- Budget and Popularity used as independent variable, and revenue used as dependent variable
- Using train_test_split method, 85% of the data used to train the linear regression model, while 15% used as the test
- Test results shows the accuracy of this model is around 66%

MODEL TESTING AND PYTHON FUNCTION SET UP

- Once the linear regression model trained, model function is created then tested to make sure it works properly
- Predictor file contained the linear regression model function is created so that be load into an API

PREDICTOR FUNCTION FILE

```
#Testing Prediction
features = np.array([[1000000,4]])
print(model.predict(features))
```

9] ✓ 0.0s

```
[[1.21989172e+08]]
```

```
#Importing Library Datframe
import import_ipynb
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import import_ipynb
from Data Exploration import df_linreg

#Setting up data and model
x_reg2 = df_linreg[['budget_adj', 'popularity_code']]
y_reg2 = df_linreg[['revenue_adj']]
xtrain, xtest, ytrain, ytest = train_test_split(x_reg2, y_reg2, test_size=0.15, random_state=89)

model = LinearRegression()
model.fit(xtrain, ytrain)

#Function for the prediction
def rev_predictor(budget,pop) :
    features = np.array([[budget,pop]])
    rev_pred = model.predict(features)
    return rev_pred
```


SETTING UP API

- fastAPI is used to create an API with path parameters and user input of 'budget' and 'popularity scale' using get method,

API FILE

```
from fastapi import FastAPI, Path, Query
from fastapi.responses import JSONResponse
import import_ipynb
from Predictor_Function import rev_predictor

app = FastAPI()

@app.get("/")
async def root():
    return {"message": 'Hello! This is the main page for Revenue Predictor for a Film',
            "instruction": 'type /docs at the end of the address bar to proceed',
            "Author": "Ahmad Fadlan Amin"}

@app.get("/predict/{budget},{popularity}")
async def get_data(
    #Setting up the user input
    budget:float = Path(description='Budget of the film')
    ,popularity :int = Path(description='High: 4, Moderately High: 3, Medium: 2, Low: 1', gt=0, lt=5)
):

    results = rev_predictor(budget,popularity)

    #The results is in ndarray format, needs to convert to list so that it can convert to json file
    prediction = {'prediction':results.tolist()}

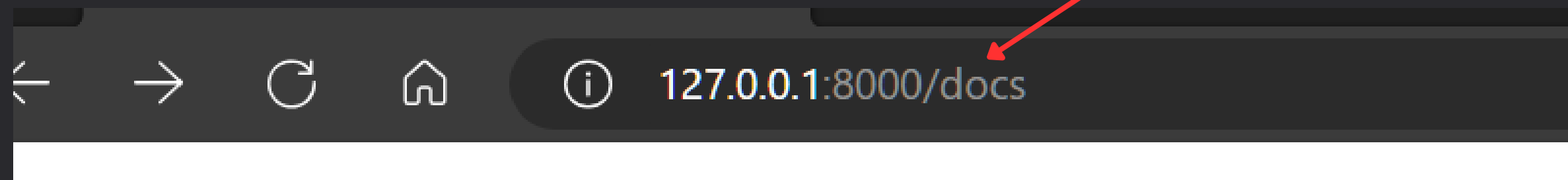

    return JSONResponse(prediction) # Results needs to be jsonified
```

OPERATE THE API

- Open the main.py file
- On terminal, type '**uvicorn API.main:app**' to run the fastAPI
- Click the link while pressing ctrl

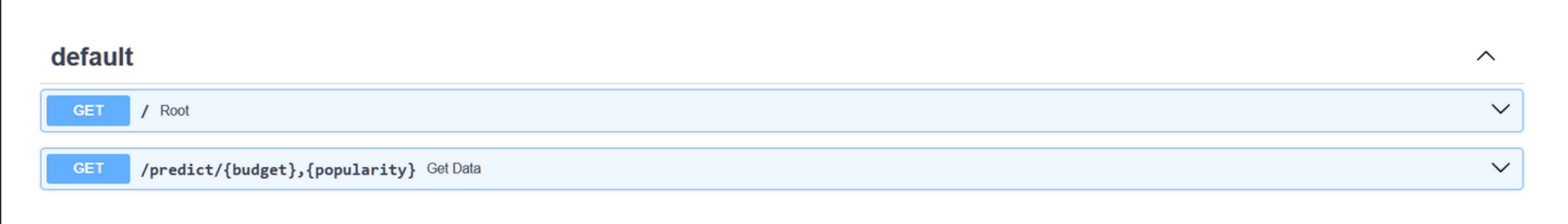
```
Victor> uvicorn API.main:app
```

```
INFO: Application startup complete.  
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)  
INFO: 127.0.0.1:58741 - "GET / HTTP/1.1" 200 OK
```

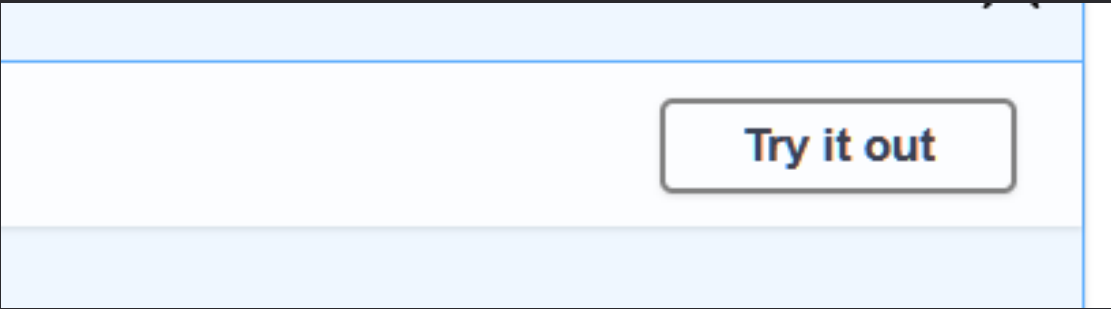


- On the address bar of the browser type '/docs'

OPERATE THE API



- On the Revenue Predictor page, click the tab above and on the top right corner click ‘Try It Out’



Name	Description
budget * required number (path)	Budget of the film <input type="text" value="budget"/>
popularity * required integer (path)	High: 4, Moderately High: 3, Medium: 2, Low: 1 <input type="text" value="popularity"/>

- Input the budget and the popularity scale
- Both of the field is required, and for ppularity has a scale of 1-4 (anything than that will shows and error)
- Once filled, click execute
- The predicted revenue will be shown below

EXAMPLE USES OF THE API

Name	Description
budget * required number (path)	Budget of the film <input type="text" value="100000000"/>
popularity * required integer (path)	High: 4, Moderately High: 3, Medium: 2, Low: 1 <input type="text" value="3"/>

Execute

Response body

```
{
  "prediction": [
    [
      334992426.3004036
    ]
  ]
}
```

- Example use of the API : with the total budget of 100M and popularity level at 3 (moderately high), the revenue of the film predicted to be around 330M

introduction **ABOUT ME**

Graduated cumlaude from Bakrie University with Business Management Major and experienced working in a dynamic e-commerce environment.

With Data Analysts certification from Binar Academy and personal projects, i aspire to pursue a career in Data Analytics field.



**AHMAD
FADLAN
AMIN**

The background of the slide is a composite of two movie scenes. The top half shows a couple standing on a balcony at night, looking out over a city with lights reflecting on the water. The bottom half shows a couple in formal wear standing in front of a large, ornate building at night. The text is overlaid on the dark blue background.

THANK YOU!

LET'S DISCUSS YOUR FAVOURITE FILMS!

- +62 857 0930 9304
- fathelan@gmail.com
- github.com/fadlanamin
- linkedin.com/in/fadlanamin