Final Project Submission

Please fill out:

- Student name: Nancy Maina
- Student pace: self paced / part time / full time: DSF-FT 09 Hybrid
- Scheduled project review date/time: 26/7/24
- Instructor name: Antonny Muiko
- Blog post URL:https://github.com/Nmwangu2/dsc-phase-2-project-v3.git

Project Overview

Most of the big companies are generating original video content. In this context, this organization had decided to join in the fun by creating new movie studio, but it lacks adequate knowledge regarding this venture. In this context, this project explores the various kinds of films and their performance at the box office. The findings are then translated into actionable insights that the head of the new movie studio can apply when making critical decisions regarding the types of films to produce.

Business Understanding

Objective: Establish a new movie studio and determine which types of films to produce based on current box office trends and audience preferences.

Business Goals: Maximize box office revenues, establish brand recognition, and cater to diverse audience preferences.

Key Questions:

a) Which genres are currently performing well at the box office?

b) How do factors like budget and directors impact box office success?

c) Which studios will we be competing with?

Data Understanding

Data Sources:

IMDB SQLite Database (im.db): Contains movie metadata including titles, genres, ratings. Box Office Mojo (bom.movie_gross.csv.gz): Provides box office gross revenue data.

Explore data to understand distributions, correlations, and completeness.

Identify relevant variables such as genre, budget, ratings, release dates.

Data Analysis

1. Data Exploration

1.1 Importing relevant libraries and exploring data structure and type

2. Genre Performance Analysis:

2.1 Data cleaning

2.2 Calculate total box office revenues by genre.

2.3 Evaluate the relationship between production budget, marketing spends, and box office returns

2.4 Top studios in relation to box office returns

1. Data Exploration

1.1 Importing relevant libraries and exploring data structure and type

```
# import the relevant libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sqlite3 as sql
```

```
#accessing movie_basics from this database
q = """
SELECT *
FROM movie_basics
"""
conn = sqlite3.connect("im.db")
mb = pd.read_sql(q, conn)
mb.head()
```

	movie_id	primary_title	original_title	start_year	runtime_minutes	
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	

```
#Accessing the columns needed from this database
conn = sqlite3.connect("im.db")
q = """
SELECT
    mb.primary_title,
    mb.genres,
    p.primary_name
FROM movie_basics AS mb
JOIN directors AS d
    ON mb.movie_id = d.movie_id
JOIN persons AS p
    ON d.person_id = p.person_id
WHERE p.primary_profession LIKE '%director%'
GROUP BY mb.primary_title, mb.genres, p.primary_name
"""
imdb = pd.read_sql(q, conn)
imdb.head()
```

	primary_title	genres	primary_name
0	!Women Art Revolution	Documentary	Lynn Hershman-Leeson
1	#1 Serial Killer	Horror	Stanley Yung
2	#5	Biography,Comedy,Fantasy	Ricky Bardy
3	#5	Documentary	Trisha Fuentes Allen
4	#50Fathers	Comedy	Joddy Eric Matthews

```
#Affirming that only the columns have been selected
imdb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150469 entries, 0 to 150468
Data columns (total 3 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   primary_title  150469 non-null  object
 1   genres         146953 non-null  object
 2   primary_name   150469 non-null  object
dtypes: object(3)
memory usage: 3.4+ MB
```

```
#upload additional data from box office mojo
bom = pd.read_csv('bom.movie_gross.csv')
bom.head()
```

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010

```
bom.info() #exploring the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   title           3387 non-null   object
 1   studio          3382 non-null   object
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2032 non-null   float64
 4   year            3387 non-null   int64
 5   total_revenue   2004 non-null   float64
dtypes: float64(3), int64(1), object(2)
memory usage: 158.9+ KB
```

```
#upload more datasets
tmdb_movies = pd.read_csv('tmdb.movies.csv')
tmdb_movies.head()
```

	Unnamed: 0	genre_ids	id	original_language	original_title	popularity	release_d
0	0	[12, 14, 10751]	12444	en	Harry Potter and the Deathly Hallows: Part 1	33.533	2010-1
1	1	[14, 12, 16, 10751]	10191	en	How to Train Your Dragon	28.734	2010-0

```
#exploring tmdb_movies
tmdb_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26517 entries, 0 to 26516
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         26517 non-null  int64
 1   genre_ids          26517 non-null  object
 2   id                 26517 non-null  int64
 3   original_language  26517 non-null  object
 4   original_title     26517 non-null  object
 5   popularity         26517 non-null  float64
 6   release_date       26517 non-null  object
 7   title              26517 non-null  object
 8   vote_average       26517 non-null  float64
 9   vote_count         26517 non-null  int64
dtypes: float64(2), int64(3), object(5)
memory usage: 2.0+ MB
```

```
#upload movies_budgets
movies_budgets = pd.read_csv('tn.movie_budgets.csv')
movies_budgets.head()
```

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
			Avengers:			

```
movies_budgets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5782 non-null   int64
 1   release_date       5782 non-null   object
 2   movie              5782 non-null   object
 3   production_budget  5782 non-null   object
 4   domestic_gross     5782 non-null   object
 5   worldwide_gross    5782 non-null   object
dtypes: int64(1), object(5)
memory usage: 271.2+ KB
```

This project will use IMDB database as it has large datasets for analysis(more than 150,000 movies) using three primary columns selected including genres, primary_name(directors) and primary title. The project will also use the bom.movie_gross.csv data sets to calculate the box office revenue per genre.

2. Data Analysis

Genre Performance Analysis:

2.1 Data cleaning

2.2 Calculate total box office revenues by genre.

2.3 Evaluate the relationship between production budget, marketing spends, and box office returns.

```
imdb.head(3)
```

	primary_title	genres	primary_name
0	!Women Art Revolution	Documentary	Lynn Hershman-Leeson
1	#1 Serial Killer	Horror	Stanley Yung
2	#5	Biography,Comedy,Fantasy	Ricky Bardy

```
# comparing top most produced genres
Top5_Genres = imdb ['genres'].value_counts().head(5)
Top5_Genres
```

```
genres
Documentary     34687
Drama           22125
Comedy           9186
Horror           4991
Comedy,Drama     3616
Name: count, dtype: int64
```

```
# Identifying top  directors on genres

Directors = imdb['primary_name'].value_counts().head(5)
Directors
```

```
primary_name
Omer Pasha       62
Larry Rosen      53
Rajiv Chilaka    49
Stephan Düfel    48
Graeme Duane     45
Name: count, dtype: int64
```

```
# Exploring total box office revenues
bom.head(3)
```

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010

```
bom.info() #exploring the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   title           3387 non-null   object
 1   studio          3382 non-null   object
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2032 non-null   float64
 4   year            3387 non-null   int64
 5   total_revenue   2004 non-null   float64
dtypes: float64(3), int64(1), object(2)
memory usage: 158.9+ KB
```

```
#Data cleaning to address missing values
bom.isna().sum()
```

```
title               0
studio              5
domestic_gross     28
foreign_gross    1355
year                0
total_revenue    1383
dtype: int64
```

```
# address the missing values by inserting mean
bom['domestic_gross'].fillna(bom['domestic_gross'].mean(), inplace=True)
bom['foreign_gross'].fillna(bom['foreign_gross'].mean(), inplace=True)
```

```
# confirming no missing values
bom.isna().sum()
```

```
title            0
studio           5
domestic_gross   0
foreign_gross    0
year             0
total_revenue    0
dtype: int64
```

```
# calculate total revenues
# Check data types of columns
print(bom['domestic_gross'].dtype)
print(bom['foreign_gross'].dtype)

# Convert columns to numeric type, handling errors
bom['domestic_gross'] = pd.to_numeric(bom['domestic_gross'], errors='coerce')
bom['foreign_gross'] = pd.to_numeric(bom['foreign_gross'], errors='coerce')

# Calculate total revenue
bom['total_revenue'] = bom['domestic_gross'] + bom['foreign_gross']

bom.head(5)
```

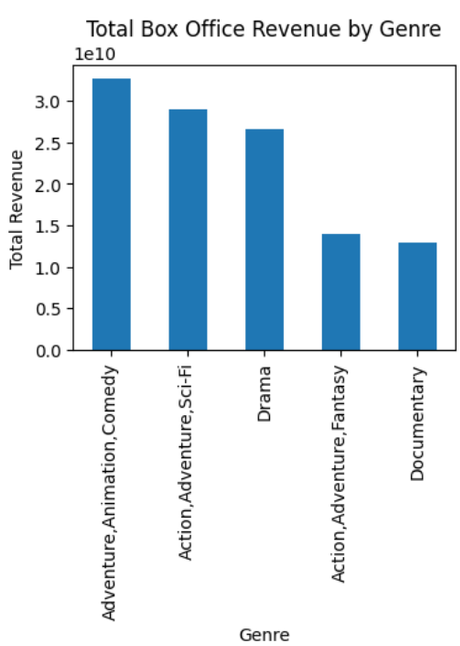
```
float64
float64
```

	title	studio	domestic_gross	foreign_gross	year	total_revenue
0	Toy Story 3	BV	415000000.0	652000000.0	2010	1.067000e+09
1	Alice in Wonderland (2010)	BV	334200000.0	691300000.0	2010	1.025500e+09
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000.0	2010	9.603000e+08
3	Inception	WB	292600000.0	535700000.0	2010	8.283000e+08

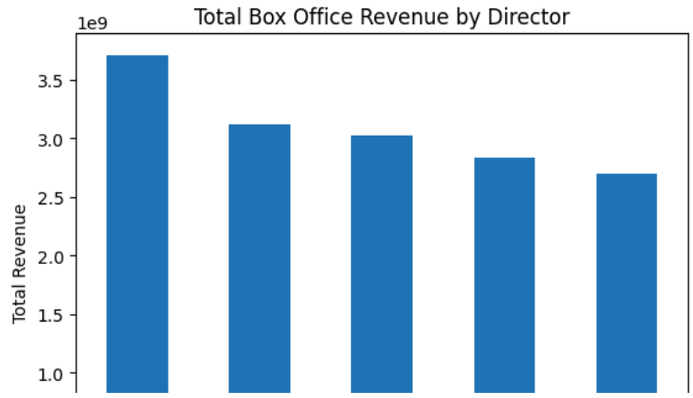
```
# merge BOM datasets with IMDB database for further analysis

# Rename the 'primary_title' column in 'imdb' to 'title' to match 'bom'
imdb.rename(columns={'primary_title': 'title'}, inplace=True)


# Merge 'bom' and 'imdb' DataFrames first
bom_imdb = pd.merge(bom, imdb, on='title', how='inner')


bom_imdb.head()
```

	title	studio	domestic_gross	foreign_gross	year	total_revenue	
0	Inception	WB	292600000.0	535700000.0	2010	828300000.0	Action,Adve
1	Shrek Forever After	P/DW	238700000.0	513900000.0	2010	752600000.0	Adventure,Animat
	The Twilight						

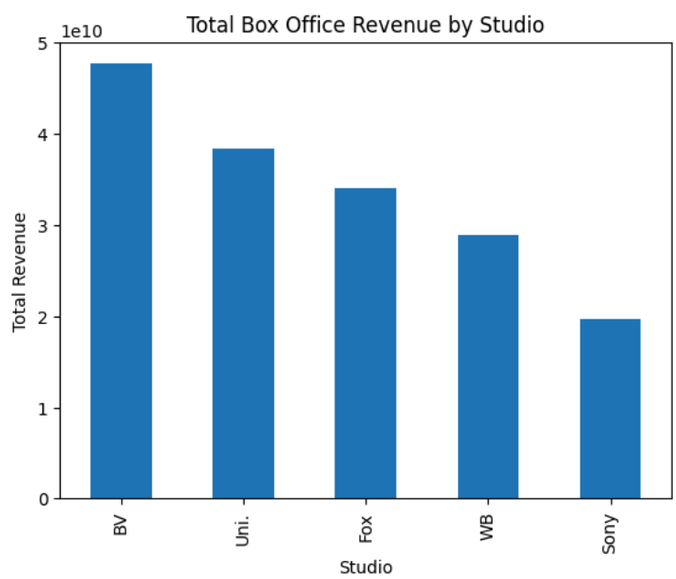
```
#analyse best genres against total_revenue and plot top 5
bom_imdb.groupby('genres')['total_revenue'].sum().sort_values(ascending=False)
fig = plt.figure(figsize=(4, 3))
plot = bom_imdb.groupby('genres')['total_revenue'].sum().sort_values(ascending=False).head(5).plot(kind='bar')
plot.set_title('Total Box Office Revenue by Genre')
plot.set_xlabel('Genre')
plot.set_ylabel('Total Revenue')
plt.show()
```




```
#analyse best directors to work with using  total_revenue and plot top 5
bom_imdb.groupby('primary_name')['total_revenue'].sum().sort_values(ascending=False)
plot = bom_imdb.groupby('primary_name')['total_revenue'].sum().sort_values(ascending=False).head(5).plot(kind='bar')
fig = plt.figure(figsize=(4, 3))
plot.set_title('Total Box Office Revenue by Director')
plot.set_xlabel('Director')
plot.set_ylabel('Total Revenue')
plt.show()
```




```
# generate revenue by studio
bom_imdb.groupby('studio')['total_revenue'].sum().sort_values(ascending=False)
plot = bom_imdb.groupby('studio')['total_revenue'].sum().sort_values(ascending=False).head(5).plot(kind='bar')
fig = plt.figure(figsize=(4, 3))
plot.set_title('Total Box Office Revenue by Studio')
plot.set_xlabel('Studio')
plot.set_ylabel('Total Revenue')
plt.show()
```




```
<Figure size 400x300 with 0 Axes>
```

Three recommendations

1. Focus on producing Action and Adventure movies: These genres consistently generate the highest box office revenues.
2. Collaborate with successful directors: Partnering with experienced directors like Christopher Nolan,Pierre Coffin,Peter Jackson,Michael Bay, and Francis Lawrence can significantly increase the chances of office success.
3. Consider benchmarking top 5 studios: The following studios tend to perform exceptionally well including BV, Uni., Fox, WB, and Sony, to learn ways they strategically can maximize revenue potential.