Final Project Submission

Please fill out:

- Student name: Shane Njenga Mwangi
- Student pace: Full Time Student
- Scheduled project review date/time: 5th August
- Instructor name: Faith Rotich and Asha Deen
- Blog post URL:

Phase 2 Project

Business Objective.

The company i work for now sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don't know anything about creating movies. I am charged with exploring what types of films are currently doing the best at the box office. I must then translate those findings into actionable insights that the head of the company's new movie studio can use to help decide what type of films to create.

Methodology

This project will be accomplished Using two datasets which are the Tmdb Movie and bom_movie gross dataset in which I will use Data cleaning process towards each individual dataset and then merge them together and use data cleaning processes, data visulization techniques, recommendations and conclusion in order to achieve the Objective.

First Step

The First thing i did was to import all relevant libraries into the jupyter notebook such as numpy,pandas,sqlite3,scipy,matplotlib,searborn.%matplotlib inline displays Matplotlib plots directly within the jupyter notebook.Itertools creates efficient iterators and complex operations on data. I used import warnings, warnings.filterwarnings('ignore') to suppress all warnings.Ast(Abstract Syntax Trees) which represent structure of a code it is used for code analysis. Pickle serializes (pickling) and deserializes (unpickling) Python objects.

```
# Importing necessary Libraries
import itertools
import numpy as np
import pandas as pd
from numbers import Number
import sqlite3
from scipy import stats
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
warnings.filterwarnings('ignore')
import ast


import pickle
```

The Second Step

I loaded the csv file to the dataframe in which i displayed the first 5 rows.I used the dropna method to drop nan values from the dataframe and lastly i used drop duplicates to remove any duplicates from the rows in the dataframe.

```
# Load Csv file to dataframe
df = pd.read_csv('tmdb.movies.csv')

# Display first 5 rows
print(df.head(5))

# Drop rows with missing values
clean = df.dropna()

# Drop duplicate values
clean = clean.drop_duplicates()
```

```
   Unnamed: 0            genre_ids     id original_language  \
0           0      [12, 14, 10751]  12444                en
1           1  [14, 12, 16, 10751]  10191                en
2           2        [12, 28, 878]  10138                en
3           3      [16, 35, 10751]    862                en
4           4        [28, 878, 12]  27205                en

                                original_title  popularity
release_date  \
0  Harry Potter and the Deathly Hallows: Part 1      33.533   2010-11-
19
1                     How to Train Your Dragon      28.734   2010-03-
26
2                                   Iron Man 2      28.515   2010-05-
07
3                                    Toy Story      28.005   1995-11-
22
4                                    Inception      27.920   2010-07-
16

                                        title  vote_average
vote_count
0  Harry Potter and the Deathly Hallows: Part 1           7.7
10788
```

```
1                       How to Train Your Dragon           7.7
7610
2                                    Iron Man 2           6.8
12368
3                                     Toy Story           7.9
10174
4                                     Inception           8.3
22186
```

Third Step

The code below is used in Pandas to filter out a column in the dataframe which name starts with the string "Unnamed".

```
#Remove Unnamed Column
df = df.loc[:, ~df.columns.str.startswith('Unnamed')]
print(df.head(5))
```

```
              genre_ids     id original_language  \
0      [12, 14, 10751]  12444                en
1  [14, 12, 16, 10751]  10191                en
2        [12, 28, 878]  10138                en
3      [16, 35, 10751]    862                en
4        [28, 878, 12]  27205                en

                                original_title  popularity
release_date  \
0  Harry Potter and the Deathly Hallows: Part 1      33.533   2010-11-
19
1                       How to Train Your Dragon      28.734   2010-03-
26
2                                    Iron Man 2      28.515   2010-05-
07
3                                     Toy Story      28.005   1995-11-
22
4                                     Inception      27.920   2010-07-
16

                                            title  vote_average
vote_count
0  Harry Potter and the Deathly Hallows: Part 1           7.7
10788
1                       How to Train Your Dragon           7.7
7610
2                                    Iron Man 2           6.8
12368
3                                     Toy Story           7.9
```

```
10174
4                                    Inception           8.3
22186
```

Fourth Step

I loaded the csv file to the dataframe in which i displayed the first 5 rows.I used the dropna method to drop nan values from the dataframe and lastly i used drop duplicates to remove any duplicates from the rows in the dataframe.This is a repeat of the second step process but with a done with a different dataset.

```
# Read the CSV file into a DataFrame
df = pd.read_csv('bom.movie_gross.csv')

# Display the first 5 rows
print(df.head(5))

# Drop rows with missing values
clean = df.dropna()

# Drop duplicate rows
clean = clean.drop_duplicates()
```

```
                                          title studio  domestic_gross
\
0                                Toy Story 3     BV     415000000.0

1                  Alice in Wonderland (2010)     BV     334200000.0

2  Harry Potter and the Deathly Hallows Part 1     WB     296000000.0

3                                   Inception     WB     292600000.0

4                           Shrek Forever After   P/DW     238700000.0


  foreign_gross  year
0     652000000  2010
1     691300000  2010
2     664300000  2010
3     535700000  2010
4     513900000  2010
```

The Fifth Step

The First thing i did is to create a dictionary which i named genre mapping numeric it contains genre IDs from the TMDB dataset in which i allocated readable genre names.

The second thing i did is to load two csv files Tmdb and bom movie datasets.

The third thing i did was to perform a merge operation between two Pandas DataFrames, tmdb and bom.

The fourth thing i did is to clean and convert the values in the 'domestic_gross' and 'foreign_gross' of the dataframe (merged) into numeric data by Iterating over list of column names, Converting values in the column to strings, Removing any dollar signs ($) or commas (,) from the string values in the columns using a regular expression and dropping null values from domestic gross and foreign gross.

The fifth thing is i added domestic gorss and foreign gross to get the total gross.

The sixth thing i did is i anlyzed movie data by selecting the genre_ids which will contain a string representation of a list i converted each string into a list of integers using ast.literal_eval.Movies without genre IDs are converted to empty lists then finally each list of genre has its own row.

The seventh thing i did is i replaced the genre id's with the names of movie genres and then Removed rows where the genre ID was not found in genre_mapping.

The eighth thing i did was to merge the data into a csv file named Combined data movies and i saved the file.

The last thing i did is to confirm if the Data is clean.

```
#Create a variable named Genre with Genre id data
genre= {
    28: 'Action', 12: 'Adventure', 16: 'Animation', 35: 'Comedy', 80:
'Crime',
    99: 'Documentary', 18: 'Drama', 10751: 'Family', 14: 'Fantasy',
36: 'History',
    27: 'Horror', 10402: 'Music', 9648: 'Mystery', 10749: 'Romance',
878: 'Science Fiction',
    10770: 'Tv Movie', 53: 'Thriller', 10752: 'War', 37: 'Western'
}
# Load datasets
tmdb = pd.read_csv('tmdb.movies.csv')
bom = pd.read_csv('bom.movie_gross.csv')

# Merge on movie title
merged = pd.merge(tmdb, bom, on='title', how='inner')

# Drop rows with missing gross values and clean numeric columns
for col in ['domestic_gross', 'foreign_gross']:
    merged[col] =
pd.to_numeric(merged_df[col].astype(str).str.replace(r'[$,]', '',
regex=True), errors='coerce')
clean = merged.dropna(subset=['domestic_gross', 'foreign_gross'])

# Calculate total gross Box Office
clean['total_gross']= df_clean['domestic_gross'] +
clean['foreign_gross']
```

```
# Analyze movie data
clean['genres'] = clean['genre_ids'].apply(lambda x:
ast.literal_eval(x) if pd.notnull(x) else [])
df_exploded = clean.explode('genres')

# Map genre names and drop rows with unknown genres
df_exploded['genre_name'] = df_exploded['genres'].map(genre)
df_exploded = df_exploded.dropna(subset=['genre_name'])

# Save the clean data and merged it to a new CSV file
df_exploded.to_csv('Combined_data_movies.csv', index=False)

print(" Data is clean.")
```

```
 Data is clean.
```

The Sixth step (Bar Chart)

The first thing i did is to calculate average gross by genre by Grouping the dataset by genre name, Calculating the average total gross for each genre using mean and Picking the top 10 genres with the highest average total gross.

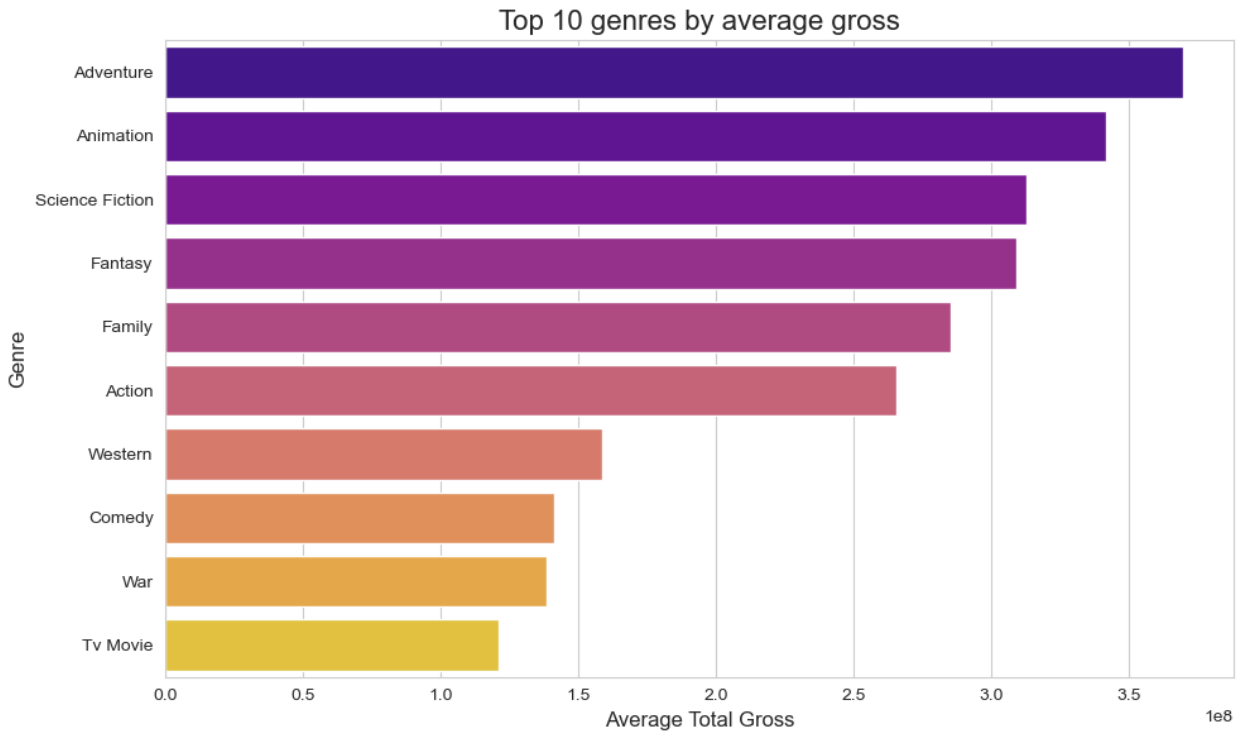
The second thing i did is set the plot figure to be 10 inches wide and 6 inches tall.

The third thing i did is to create the bar chart by Plotting the average gross amount on the x-axis and the genre name on the y axis and then I used a Sequential color map which represents information that has ordering .

The fourth thing i did was setting labels and titles with the appropriate font size

The last thing I did is i adjusted spacing to prevent labels from overlapping and Displayed the final plot.

```
# Visualization 1: Bar Chart of Average Total Gross by Genre
# Top 10 genres by average total gross
avg_gross = df_exploded.groupby('genre_name')
['total_gross'].mean().nlargest(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=avg_gross.values, y=avg_gross.index, palette='plasma')
plt.title('Top 10 genres by average gross', fontsize=16)
plt.xlabel('Average Total Gross', fontsize=12)
plt.ylabel('Genre', fontsize=12)
plt.tight_layout()
plt.show()
```



Seventh Step (Box Plot)

First thing i did is i Created the plot in which i Set the size of the plot to be 10 inches wide and 6 inches tall.

The second thing i did is to get the top 10 most common genres from the datasets.

The third thing i did is i Created the box plot in which Genres goes to the x axis, total box office gross goes to the y axis,filtered the dataframe to show rows from the top 10 genres and Applied a red-to-blue color.

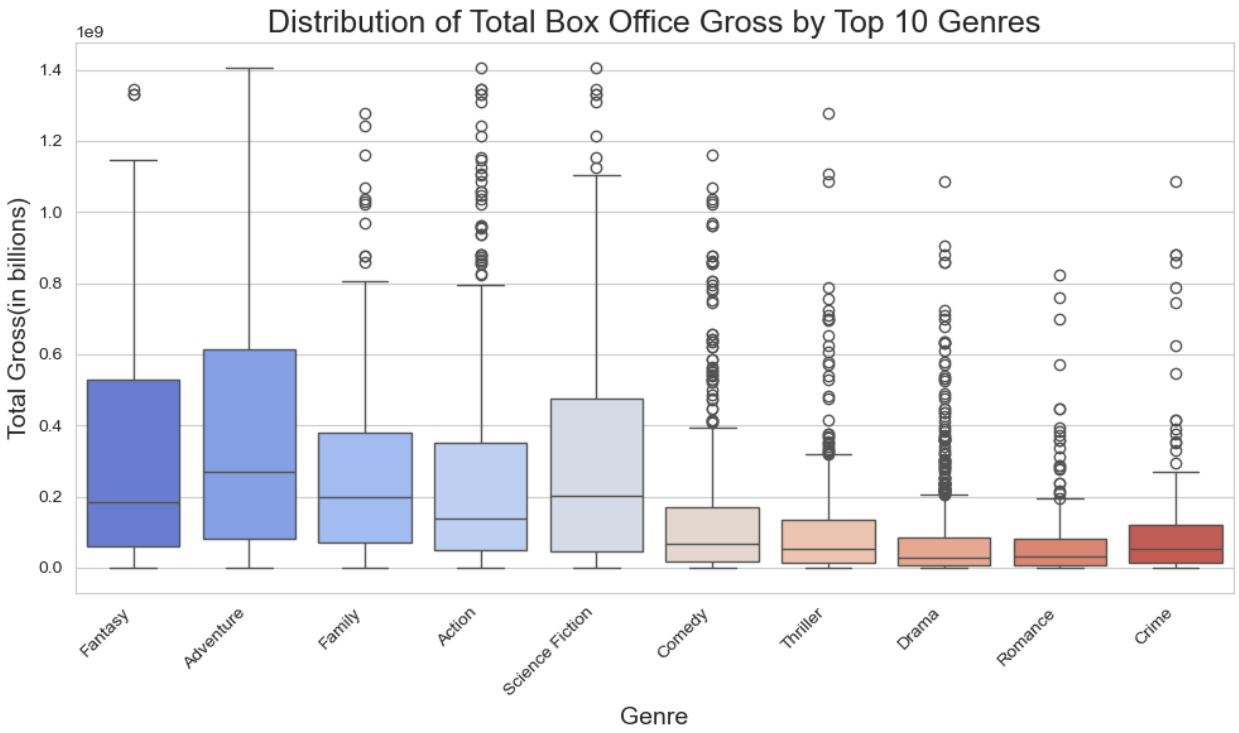
The fourth thing i did is i added titles and labels to make the chart simpler to understand with their appropriate font sizes.

The fifth thing i did is i Rotated the genre names on the x-axis by 45 degrees so they do not overlap and align them to the right.

Then the last thing is to make sure the elements fit well and display the chart.

```
# Visualization 2: Box Plot of Total Gross by Genre
plt.figure(figsize=(10, 6))
top_genres = df_exploded['genre_name'].value_counts().head(10).index
sns.boxplot(x='genre_name', y='total_gross',
data=df_exploded[df_exploded['genre_name'].isin(top_genres)],
palette='coolwarm')
plt.title('Distribution of Total Box Office Gross by Top 10 Genres',
fontsize=18)
plt.xlabel('Genre', fontsize=14)
```

```
plt.ylabel('Total Gross(in billions)', fontsize=14)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Eighth Step (Scatter Plot)

First thing i did is i Created the plot in which i Set the size of the plot to be 12 inches wide and 8 inches tall.

The second thing I did is create a scatter plot in which i Set the x-axis to the popularity score of the film (from TMDB),I set the y-axis to total box office gross (domestic + foreign).I Used a random sample of 1000 rows from the dataset df_exploded to make the plot easier to read and less cluttered and also ensured the sampling is reproducible. I also Colored the dots by film genre, allowing you to compare genres visually and finally Set the color scheme with 10 distinct colors.

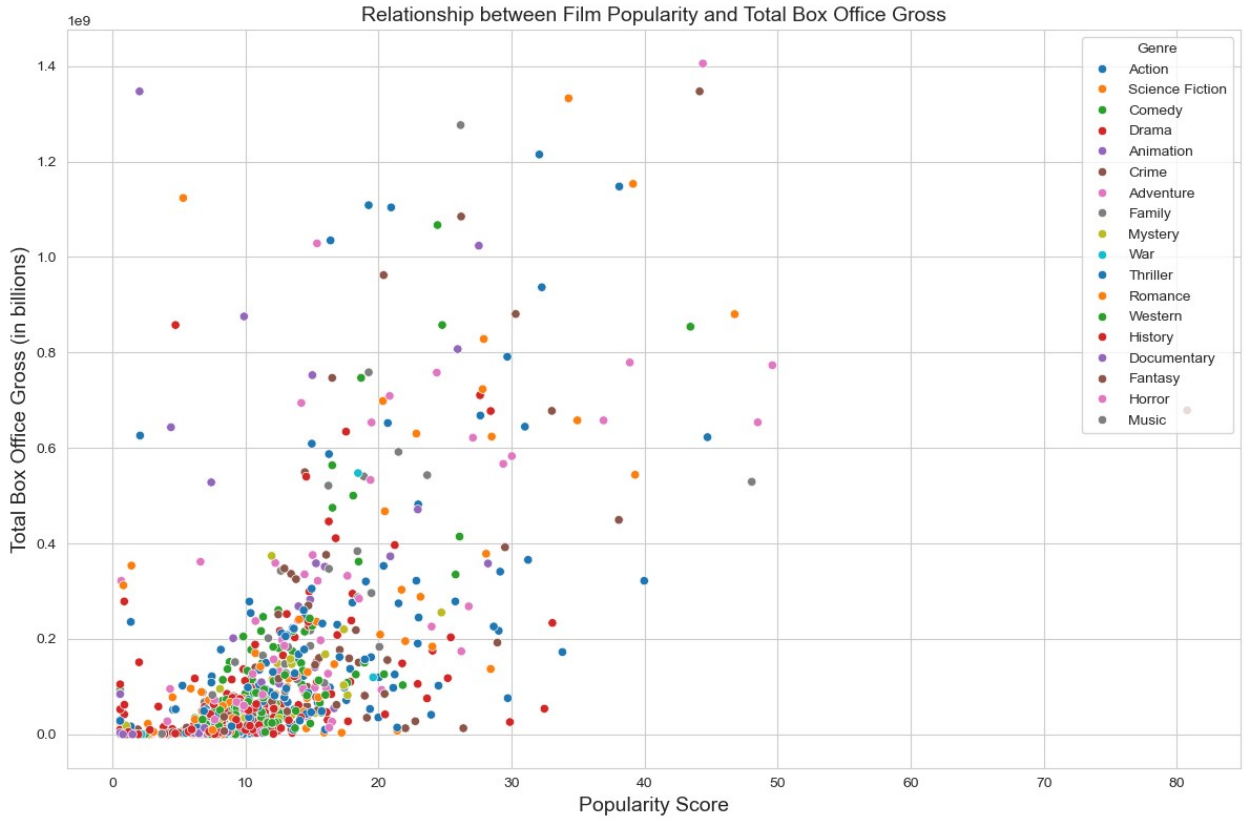
The thrid thing i did is i added Titles and labels with the appropriate font sizes.

The fourth thing i did is to Display a legend which explains what each color in the plot represents(the genres).

Then the last thing is to make sure the elements fit well and display the chart.

```
# Visualization 3: Scatter Plot of Popularity vs. Total Gross
plt.figure(figsize=(12, 8))
sns.scatterplot(x='popularity',
```

```
y='total_gross',data=df_exploded.sample(1000,
random_state=42),hue='genre_name', palette='tab10')
plt.title('Relationship between Film Popularity and Total Box Office
Gross', fontsize=14)
plt.xlabel('Popularity Score', fontsize=14)
plt.ylabel('Total Box Office Gross (in billions)', fontsize=14)
plt.legend(title='Genre')
plt.tight_layout()
plt.show()
```



The Last Step (Stacked Bar Chart)

The first thing i did is to Set up the figure size which was a size of 10 inches wide by 6 inches tall.

The second thing i did is to group and add all gross value by first filtering the dataframe and keeping only the top 10 most frequent genres in which i grouped the filtered data by genre and then adding up the domestic and foreign grosses for each genre.

The third thing i did is to create a stacked bar chart by plotting it, each bar represents a genre in which the bottom part is domestic gross and the top part is foreign gross.

The fourth thing i did i added Titles and labels with fontsizes.

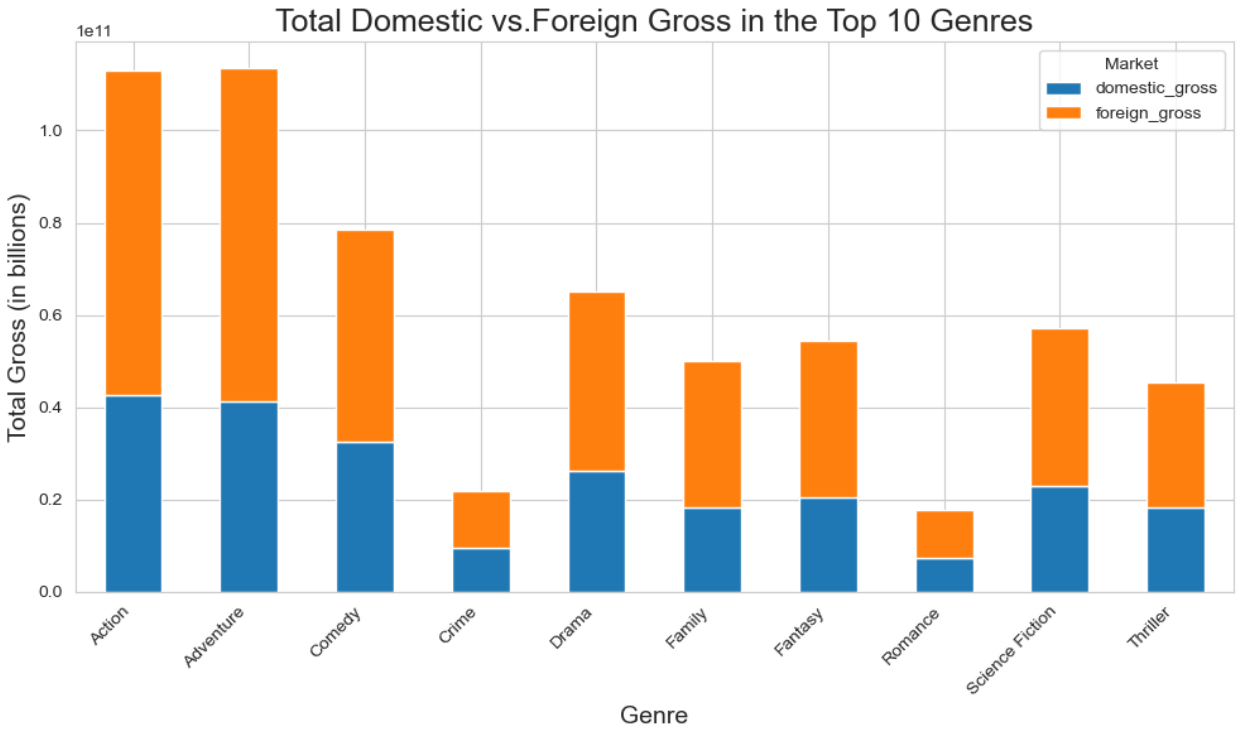
The fifth thing i did is i Rotated the genre names on the x-axis by 45 degrees so they do not overlap and align them to the right.

The sixth thing i did is to add a legend with the title "Market" to explain the two bar colors Domestic gross and foreign gross.

The last thing i did is to make sure the elements fit well and display the chart.

```
# Visualization 4: Stacked Bar Chart of Domestic vs. Foreign Gross by Top Genres
plt.figure(figsize=(10, 6))
genre_gross_split = df_exploded[df_exploded['genre_name'].isin(top_genres)].groupby('genre_name')[['domestic_gross', 'foreign_gross']].sum()
genre_gross_split.plot(kind='bar', stacked=True, figsize=(10,6), color=['#1f77b4', '#ff7f0e'])
plt.title('Total Domestic vs.Foreign Gross in the Top 10 Genres', fontsize=18)
plt.xlabel('Genre', fontsize=14)
plt.ylabel('Total Gross (in billions)', fontsize=14)
plt.xticks(rotation=45, ha='right')
plt.legend(title='Market')
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Recommendations

1: Focus on High grossing genres for example in this instance adventure has been having the highest average earnings which can create profit for the new movie studio.

2: Use Popularity as an indicator of which movies to choose to achieve the objective which in this case is financial success.The scatter plot shows the positive relationship between films popularity score and its total box office gross.

3: Develop Films with universal or international appeal, the stacked bar chart shows that foreign markets contribute a significant majority of the total box office revenue which can help in deciding which films the new movie studio can make.

4: Consider stable investments in terms of movies such as family and fantasy have a consistent high median gross and have a GE(Generally for Everyone) rating.

Conclusion

The best genre for the new movie studio is Adventure due to the high average earnings that it makes in domestic and foreign markets however family movies for the new movie studio is the more feasible option due to the lower costs of making Family movies than Adventure and also the consistent high median gross is good for stablity in terms of the new movie studio.