Final Project Submission

Please fill in:

- Student name: Alpha Omena Odiko
- Student pace: Part Time
- Scheduled project review date/time: 5/12/
- Instructor name: Rafael Ruiz
- Blog post URL:

Overview

Microsoft is looking to create a movie studio using the following movie datasets:

- Box Office Mojo
- IMDB
- Rotten Tomatoes
- Theatrical
- The Numbers

As a part of the Microsoft Film Studio's team, we are using our knowledge understand the great stats to find which types of film are currently doing the best at the box office.

Business Problem

What type of film Microsoft should invest in to create a profitable and successful movie?

Business Value

Many factors are considered the production of a film, including, directors, genre and stars are highly regarded to and movie goers. In this analysis, we will be looking at the following factors as well as what is in focus in the following questions:

- Where is the best time to release a movie?
- What is good production budget for a movie?
- Does production budget correlate to Profits/Losses?

Once we gather our information we will determine the recommendations Microsoft needs to release a film that will catch the eye of the audience.

What can we find from the data given to us?

We have 5 (three useful ones) ways in order to find the accurate recommendations for our Client:

- Lead and organize the data
- Find out what films are most on - time, becoming a Hit/Miss
- Find out what genre is most popular

When is the best time to release a movie based on Profit?

```
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

# Setting dataframe to gross.csv variable
gross = pd.read_csv('bom.movie_gross.csv')
# Checking the dataset
gross.info()
gross.shape
gross.describe()
gross.head(10)
gross.tail(10)
# Checking for duplicates
gross.duplicated().sum()
# Checking for null values
gross.isnull().sum()
# Dropping null values
gross.dropna(inplace=True)
# Checking for null values after dropping null values
gross.isnull().sum()
# Checking the dataset after dropping null values
gross.info()
gross.shape
gross.describe()
gross.head(10)
gross.tail(10)
```

```
<class pandas.core.frame.DataFrame>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 title            3387 non-null   object
 studio           3382 non-null   object
 domestic_gross   3359 non-null   float64
 foreign_gross    2037 non-null   object
 year             3387 non-null   int64
dtype: object, int64, float64(1)
memory usage: 132.4+ KB
(3387, 5)
       domestic_gross  year
count        3.359000  3387
mean         2.874585  2013
std          3.232808     2
min          0.000100  2010
25%          0.070000  2011
50%          1.700000  2013
75%          4.300000  2015
max         18.700000  2018
0    Toy Story 3
1     Inception
2    Shrek Forever After
3                NaN
4                NaN
5                NaN
6                NaN
7                NaN
8                NaN
9                NaN
Name: domestic_gross, dtype: float64
title            0
studio           5
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
title            0
studio           0
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
<class pandas.core.frame.DataFrame>
Int64Index: 3359 entries, 0 to 3386
Data columns (total 5 columns):
 title            3359 non-null   object
 studio           3359 non-null   object
 domestic_gross   3359 non-null   float64
 foreign_gross    2037 non-null   object
 year             3359 non-null   int64
dtype: object, int64, float64(1)
memory usage: 157.5+ KB
(3359, 5)
       domestic_gross  year
count        3.359000  3359
mean         2.874585  2013
std          3.232808     2
min          0.000100  2010
25%          0.070000  2011
50%          1.700000  2013
75%          4.300000  2015
max         18.700000  2018
0    Toy Story 3
1     Inception
2    Shrek Forever After
3                NaN
4                NaN
5                NaN
6                NaN
7                NaN
8                NaN
9                NaN
Name: domestic_gross, dtype: float64
title            0
studio           0
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
title            0
studio           0
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
<class pandas.core.frame.DataFrame>
Int64Index: 3359 entries, 0 to 3386
Data columns (total 5 columns):
 title            3359 non-null   object
 studio           3359 non-null   object
 domestic_gross   3359 non-null   float64
 foreign_gross    2037 non-null   object
 year             3359 non-null   int64
dtype: object, int64, float64(1)
memory usage: 157.5+ KB
(3359, 5)
       domestic_gross  year
count        3.359000  3359
mean         2.874585  2013
std          3.232808     2
min          0.000100  2010
25%          0.070000  2011
50%          1.700000  2013
75%          4.300000  2015
max         18.700000  2018
0    Toy Story 3
1     Inception
2    Shrek Forever After
3                NaN
4                NaN
5                NaN
6                NaN
7                NaN
8                NaN
9                NaN
Name: domestic_gross, dtype: float64
title            0
studio           0
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
title            0
studio           0
domestic_gross   0
foreign_gross    1350
year             0
dtype: int64
<class pandas.core.frame.DataFrame>
Int64Index: 3359 entries, 0 to 3386
Data columns (total 5 columns):
 title            3359 non-read   object
 studio           3359 non-read   object
 domestic_gross   3359 non-read   float64
 foreign_gross    2037 non-read   object
 year             3359 non-read
```