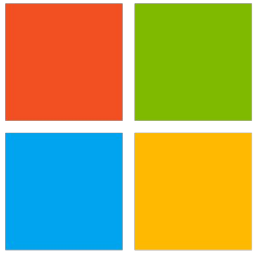
Microsoft Movie Analysis

Justin Balwan
February 7th, 2022



Microsoft

Summary

This project focuses on providing Microsoft Movie Studio with recommendations on how they can produce successful films.

Given a dataset containing files from popular film related websites, descriptive analysis was conducted to discover the top grossing genres, top grossing years, and top grossing studios. The results were used for recommendations, and Microsoft Movie Studio can effectively utilize these recommendations to create attractive, successful films.

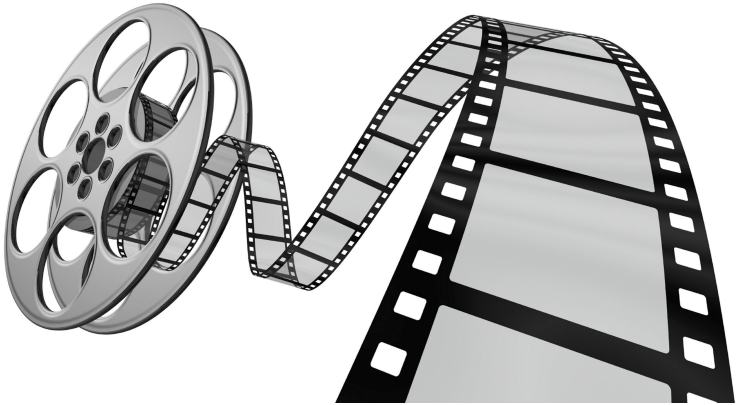
- Create movies that are either fantasies, musicals, sci-fi's, adventures, or animations
- Study top movies from the years 2010, 2012, 2016, 2017, and 2018
- Build partnerships with studios such as MUBI, Spanglish, Super, Blue Fox, or RLJ

Outline

- Business Problem
- Data & Methods
- Results
- Conclusions

Business Problem

- New movie studio
- Find the current best films in the Box Office
- Provide effective recommendations



Data & Methods

- The data represented in this project were taken from the websites Box Office Mojo, IMDB, and The Numbers. The files used in this project represented movie-related factors, such as genre, year, and studio. Given the target variable worldwide success, I used each of these factors and filtered them to show the top worldwide grossing element of each.

- Descriptive analysis was conducted to show the top worldwide grossing genres, years, and studios. This provides Microsoft's Movie Studio with the necessary information to create successful films.

Data Cleaning

Box Office Mojo Gross Data

```
In [11]: # Dropping Columns
         bom_movie_gross_df.drop(columns = ['domestic_gross', 'foreign_gross'], inplace=True)

In [12]: bom_movie_gross_df.head()
```

Out[12]:

title	studio	year
Toy Story 3	BV	2010
Alice in Wonderland (2010)	BV	2010
Harry Potter and the Deathly Hallows Part 1	WB	2010
Inception	WB	2010
Shrek Forever After	P/DW	2010

IMDB Title Basics Data

```
In [13]: # Simplifying Column Names & Dropping Columns
         imdb_title_basics_df.columns = imdb_title_basics_df.columns.str.lower().str.replace(' ', '_')
         imdb_title_basics_df.drop(columns = ['original_title'], inplace=True)

In [14]: imdb_title_basics_df.head()
```

Out[14]:

tconst	primary_title	start_year	runtime_minutes	genres
tt0063540	Sunghursh	2013	175.0	Action,Crime,Drama
tt0066787	One Day Before the Rainy Season	2019	114.0	Biography,Drama
tt0069049	The Other Side of the Wind	2018	122.0	Drama

Data Preparation

Finding The Top Grossing Genres

I created two new variables called genres_worldwide_gross and genres_worldwide_gross_df to ultimately compute the top grossing genres. I plan to use the results as one of my recommendations to Microsoft.

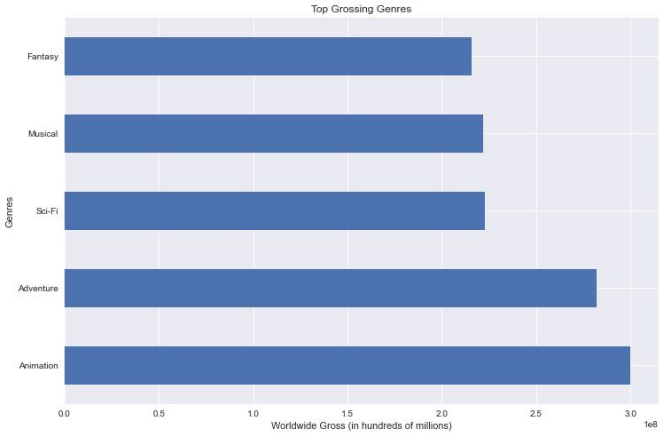
```
In [23]: # Creates a new variable called genres_worldwide_gross to store the merge between title_information_df and tn_movie_
         genres_worldwide_gross = pd.merge(title_information_df,
                 tn_movie_budgets_df,
                 left_on=['primary_title'],
                 right_on=['movie'],
                 how='inner')
         genres_worldwide_gross.head()
```

Out[23]:

	primary_title	start_year	runtime_minutes	genres	averagerating	movie	production_budget	worldwide_gross
0	Foodfight!	2012	91.0	Action,Animation,Comedy	1.9	Foodfight!	$45,000,000	$73,706
1	Mortal Kombat	2021	NaN	Action,Adventure,Fantasy	NaN	Mortal Kombat	$20,000,000	$122,133,227
2	The Overnight	2010	88.0	NaN	7.5	The Overnight	$200,000	$1,165,996
3	The Overnight	2015	79.0	Comedy,Mystery	6.1	The Overnight	$200,000	$1,165,996
4	On the Road	2012	124.0	Adventure,Drama,Romance	6.1	On the Road	$25,000,000	$9,313,302

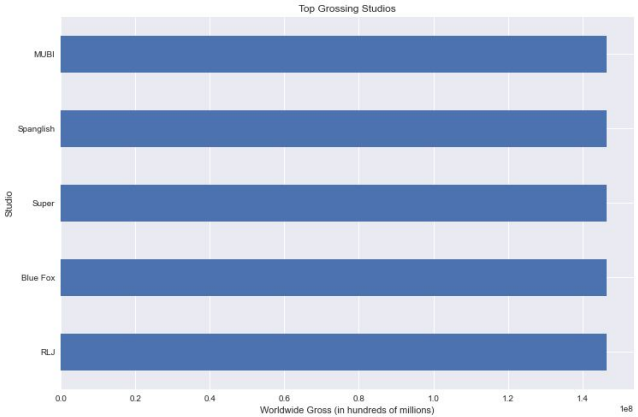
```
In [24]: # Gets rid of the '$' and ',' characters
         genres_worldwide_gross['worldwide_gross'] = genres_worldwide_gross['worldwide_gross'].apply(lambda x: x.replace('$',

In [25]: # Converts worldwide_gross to an int so I can calculate it's mean
         genres_worldwide_gross['worldwide_gross'] = genres_worldwide_gross['worldwide_gross'].astype(int)

In [26]: rs while converting production_budget to an int
         get'] = genres_worldwide_gross['production_budget'].apply(lambda x: x.replace('$', '').replace(',', '')).astype(int)

In [27]: genres_worldwide_gross
```

Results



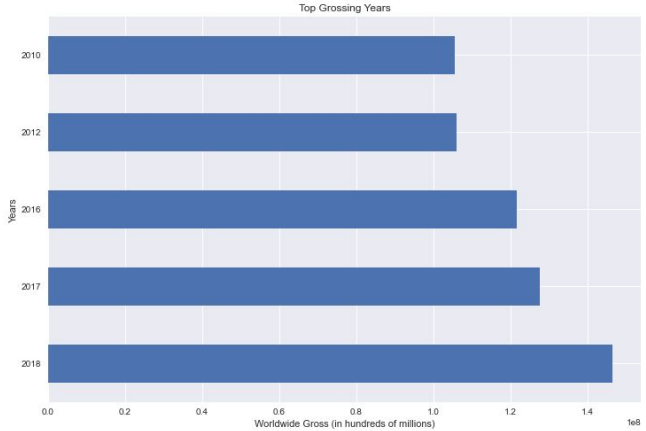
Top Grossing Years:

- 2010
- 2012
- 2016
- 2017
- 2018



Top Grossing Genres:

- Fantasy
- Musical
- Sci-Fi
- Adventure
- Animation



Top Grossing Studios:

- MUBI
- Spanglish
- Super
- Blue Fox
- RLJ

Conclusion

- Create movies that are either fantasies, musicals, sci-fi's, adventures, or animations
- Study top movies from the years 2010, 2012, 2016, 2017, and 2018
- Build partnerships with studios such as MUBI, Spanglish, Super, Blue Fox, or RLJ

Next Steps

- Make models more engaging (more color; different types of graphs)
- Creating models for other variables (i.e. directors, actors/actresses)
- Predicting likelihood of Microsoft's Movie Studio's success

THANK YOU!

Email: justinbalwan20@gmail.com

Github: @justinbalwan

LinkedIn: www.linkedin.com/in/justinbalwan