Project: IMDB MOVIE DATA INVESTIGATION

Table of Contents

- Introduction
- Data Wrangling
- Exploratory Data Analysis
- Conclusions

+ Code + Markdown

[120]:
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[121]:
```python
df=pd.read_csv('../input/imdb-data/imdb-movies.csv')
df.head()
```

Out[121]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overview	runtime	genres	production_companies	release_date	vote_count	vote_av
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.	...	Twenty-two years after the events of Jurassic ...	124	Action|Adventure|Science Fiction|Thriller	Universal Studios|Amblin Entertainment|Legenda...	6/9/15	5562	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.	...	An apocalyptic story set in the furthest reach...	120	Action|Adventure|Science Fiction|Thriller	Village Roadshow Pictures|Kennedy Miller Produ...	5/13/15	6185	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You	...	Beatrice Prior must confront her inner demons ...	119	Adventure|Science Fiction|Thriller	Summit Entertainment|Mandeville Films|Red Wago...	3/18/15	2480	
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...	J.J. Abrams	Every generation has a story.	...	Thirty years after defeating the Galactic Empi...	136	Action|Adventure|Science Fiction|Fantasy	Lucasfilm|Truenorth Productions|Bad Robot	12/15/15	5292	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/	James Wan	Vengeance Hits Home	...	Deckard Shaw seeks revenge against Dominic Tor...	137	Action|Crime|Thriller	Universal Pictures|Original Film|Media Rights ...	4/1/15	2947	

5 rows × 21 columns

Introduction

Imdb dataset included such as movies, directors, genres of moveies, release date and year, budget, production companies etc. I will try to answer 2 questions.

1. Which genres are most popular from year to year?
2. Does bugdet effect popularity of movie? How?

At this section, I checked dataset information and missing value. I defined my questions accordingly missing value. Budget, release_date, popularity do not have any missing value but genres column has 23 missing value. So I will only drop 23 rows to complete my study.

[122]:
```python
# Missing value check
df.isnull().sum()
```

Out[122]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7930
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

[123]:

```
# Data info check
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10866 non-null  int64
 1   imdb_id               10856 non-null  object
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64
 4   revenue               10866 non-null  int64
 5   original_title        10866 non-null  object
 6   cast                  10790 non-null  object
 7   homepage              2936 non-null   object
 8   director              10822 non-null  object
 9   tagline               8042 non-null   object
 10  keywords              9373 non-null   object
 11  overview              10862 non-null  object
 12  runtime               10866 non-null  int64
 13  genres                10843 non-null  object
 14  production_companies  9836 non-null   object
 15  release_date          10866 non-null  object
 16  vote_count            10866 non-null  int64
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

+ Code   + Markdown

Data Wrangling

Duplicated Items

[124]:

```
#duplicated lines check
sum(df.duplicated())
```

Out[124

```
1
```

[125]:

```
df.drop_duplicates(inplace=True)
```

[126]:

```
#Check is drop_duplicates function worked - result should be 0
sum(df.duplicated())
```

Out[126

```
0
```

Missing Value

+ Code + Markdown

[127]:
```
# genres column has 23 missing data. Genres is object so we can not fill these empty items with mean. I will drop 23 rows with missing data.
df = df[df['genres'].notna()]
```

```
#recheck missing value of genres
#I only dropped missing genres rows but kept other missing values.
#If I would used dropna function, all missing rows will be deleted. In this case, dataset lose lots of data.
#genres has 0 missing value now.
df.isnull().sum()
```

Out[128
```
id                         0
imdb_id                    8
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      75
homepage                7911
director                  42
tagline                 2806
keywords                1475
overview                   3
runtime                    0
genres                     0
production_companies    1016
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

Exploratory Data Analysis

Research Question 1 : Which genres are most popular from year to year?

[129]:
```
#df filtered for question 1
df_1=df.filter(items=['genres','popularity','release_year'])
df_1.head()
```

Out[129]:

	genres	popularity	release_year
0	Action|Adventure|Science Fiction|Thriller	32.985763	2015
1	Action|Adventure|Science Fiction|Thriller	28.419936	2015
2	Adventure|Science Fiction|Thriller	13.112507	2015
3	Action|Adventure|Science Fiction|Fantasy	11.173104	2015
4	Action|Crime|Thriller	9.335014	2015

[130]:
```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10842 entries, 0 to 10865
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   genres        10842 non-null  object
 1   popularity    10842 non-null  float64
 2   release_year  10842 non-null  int64
dtypes: float64(1), int64(1), object(1)
memory usage: 338.8+ KB
```

[131]:

```
#Checked genres types and counts. There are lots of category for genres but some of them contain only 1 sample.
df_1['genres'].value_counts()
```

Out[131]

```
Drama                                        712
Comedy                                       712
Documentary                                  312
Drama|Romance                                289
Comedy|Drama                                 280
                                            ...
Adventure|Horror|Thriller|Mystery              1
Action|Fantasy|Comedy|Horror|Mystery           1
Drama|Adventure|Science Fiction|Romance        1
Thriller|Science Fiction|Horror                1
Action|Animation|Thriller                      1
Name: genres, Length: 2039, dtype: int64
```

[132]:

```
#Checked if a category has over than 200 sample
df_1['genres'].value_counts()[df_1['genres'].value_counts()>200]
```

Out[132]

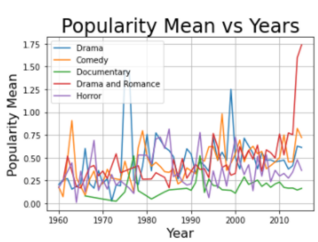
```
Drama                   712
Comedy                  712
Documentary             312
Drama|Romance           289
Comedy|Drama            280
Comedy|Romance          268
Horror|Thriller         259
Horror                  253
Comedy|Drama|Romance    222
Name: genres, dtype: int64
```

[33]:

```
# Categories are grouped by released year and calculated mean of popularity per year
# Drawed graphic to view popularity mean change per year for categories
drama=df_1.query('genres=="Drama"').groupby(['release_year'])['popularity'].mean()
comedy=df_1.query('genres=="Comedy"').groupby(['release_year'])['popularity'].mean()
documentary=df_1.query('genres=="Documentary"').groupby(['release_year'])['popularity'].mean()
drama_romance=df_1.query('genres=="Drama|Romance"').groupby(['release_year'])['popularity'].mean()
horror=df_1.query('genres=="Horror"').groupby(['release_year'])['popularity'].mean()
plt.plot(drama, label="Drama")
plt.plot(comedy, label="Comedy")
plt.plot(documentary,label="Documentary")
plt.plot(drama_romance, label="Drama and Romance")
plt.plot(horror, label="Horror")
plt.title('Popularity Mean vs Years', fontsize=24)
plt.xlabel('Year', fontsize=16)
plt.ylabel('Popularity Mean', fontsize=16)
plt.grid(True)
plt.legend()
```

Out[33]

```
<matplotlib.legend.Legend at 0x7f14f409fa10>
```



Popularity mean per year for 5 categories is visible on graphic. Graphic shows popularity mean change years by years. For example Drama and Romance category improved popularity later 2010 a lot. My detailed observations are under Conclusions

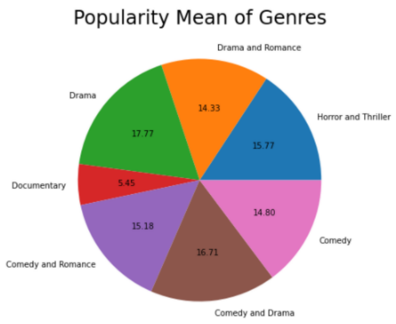
[34]:

```
df_1_filtered=df_1.query('genres=="Drama" or genres=="Comedy" or genres=="Documentary" or genres=="Drama|Romance" or genres=="Comedy|Drama" or genres=="Comedy|Romance" or genres=="Horror|Thriller"')
labels=['Horror and Thriller','Drama and Romance', 'Drama', 'Documentary', 'Comedy and Romance', 'Comedy and Drama','Comedy']

fig = plt.figure(figsize =(10, 7))
plt.pie(df_1_filtered.groupby(['genres']).mean()['popularity'], labels=labels, autopct='%.2f')
df_1_filtered.groupby(['genres']).mean()['popularity']
plt.title('Popularity Mean of Genres',fontsize=24 )
```

Out[34]:

```
Text(0.5, 1.0, 'Popularity Mean of Genres')
```



Pie chart shows that mean of popularity accordingly genres. Comedy and Drama has highest popularity than others. But documentary genre has lowest popularity than others.

Research Question 2 : Does bugdet effect popularity of movie? How?

[135]:

```
#Created new dataframe to answer question 2
# genres, popularity and budget are included
df_2=df.filter(items=['genres','popularity','budget'])
df_2.head()
```

Out[135]:

	genres	popularity	budget
0	Action|Adventure|Science Fiction|Thriller	32.985763	150000000
1	Action|Adventure|Science Fiction|Thriller	28.419936	150000000
2	Adventure|Science Fiction|Thriller	13.112507	110000000
3	Action|Adventure|Science Fiction|Fantasy	11.173104	200000000
4	Action|Crime|Thriller	9.335014	190000000

[136]:

```
#Grouped by genres and took mean of budget per genres and sorted
grouped_budget=df_2.groupby(['genres'])['budget'].mean().sort_values(ascending=False).reset_index()
#Filtered budget which are greater than 190000000
filtered_budget=grouped_budget[grouped_budget['budget']>=190000000.0]
filtered_budget
```

Out[136]:

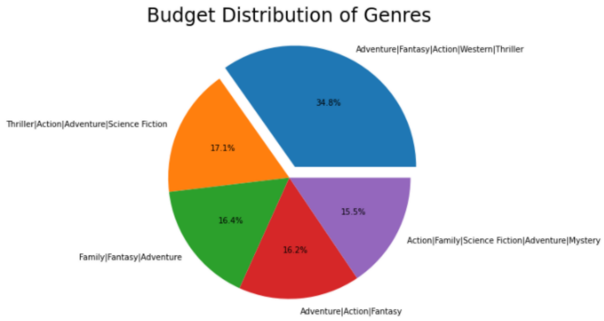
	genres	budget
0	Adventure|Fantasy|Action|Western|Thriller	425000000.0
1	Thriller|Action|Adventure|Science Fiction	209000000.0
2	Family|Fantasy|Adventure	200000000.0
3	Adventure|Action|Fantasy	198000000.0
4	Action|Family|Science Fiction|Adventure|Mystery	190000000.0

[46]:

```
#created pie chart for 5 category according to budget
labels=['Adventure|Fantasy|Action|Western|Thriller','Thriller|Action|Adventure|Science Fiction','Family|Fantasy|Adventure',
        'Adventure|Action|Fantasy','Action|Family|Science Fiction|Adventure|Mystery']
fig = plt.figure(figsize =(10, 7))
plt.pie(filtered_budget['budget'],labels=labels, autopct='%1.1f%%', explode=[0.1, 0.0, 0.0, 0.0, 0.0])
plt.title('Budget Distribution of Genres', fontsize=24)
#Adventure|Fantasy|Action|Western|Thriller category has greater budget than other categories
```

Out[46]

```
Text(0.5, 1.0, 'Budget Distribution of Genres')
```



Pie chart shows budget mean distribution per genres. Adventure,Fantasy,Action,Western,Thriller category has highest budget. And Action,Family,Science, Advanture, Mystery category has lowest budget.

[138]:

```
#comedy category have 712 sample
df_2['genres'].value_counts()
```

Out[138]

```
Drama                                           712
Comedy                                          712
Documentary                                     312
Drama|Romance                                   289
Comedy|Drama                                    280
                                               ...
Adventure|Horror|Thriller|Mystery                 1
Action|Fantasy|Comedy|Horror|Mystery              1
Drama|Adventure|Science Fiction|Romance           1
Thriller|Science Fiction|Horror                   1
Action|Animation|Thriller                         1
Name: genres, Length: 2039, dtype: int64
```

[139]:

```
#Only comedy category is filtered
comedy=df_2[df_2['genres']=='Comedy']
comedy['popularity'].sort_values(ascending=False)
```

Out[139]

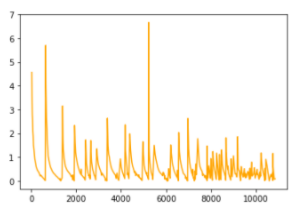
```
5230     6.668990
646      5.701683
26       4.564549
653      4.105685
1397     3.153060
           ...
6074     0.002838
10592    0.001567
3370     0.001317
6961     0.001115
6080     0.000620
Name: popularity, Length: 712, dtype: float64
```

[40]:

```
#popularity change plotted
popularity=comedy['popularity']
budget=comedy['budget']
plt.plot(popularity, color='orange')
```

Out[40]

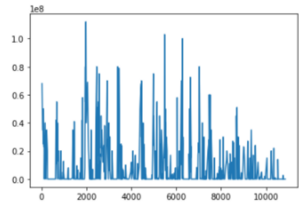
```
[<matplotlib.lines.Line2D at 0x7f14dfefc710>]
```



Graphic shows popularity change for comedy category

[41]:
```
#budget change plotted
plt.plot(budget)
```

Out[41]: [<matplotlib.lines.Line2D at 0x7f14dfe67ed0>]



Graphic shows change of budget for comedy category.

When I checked both graphic, I could not find a similarity to make some comments. Graphics are slightly different than each other.

So I changed my method. I decided to categorize popularity 1,2,3,4,5,6 and find mean of budget for each popularity range.

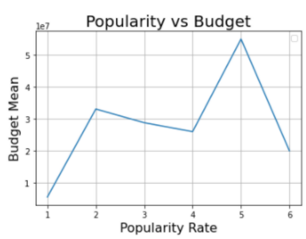
Please see below research.

[142]:
```
#popularity highest rate is 6.6
comedy['popularity'].sort_values(ascending=False)
```

Out[142]:
```
5230     6.668990
646      5.701683
26       4.564549
653      4.105685
1397     3.153060
           ...
6074     0.002838
10592    0.001567
3370     0.001317
6961     0.001115
6080     0.000620
Name: popularity, Length: 712, dtype: float64
```

[143]:
```
#samples are grouped by 1,2,3,4,5,6 popularity and taken mean of budget per range
comedy_1=comedy[comedy['popularity']<1]
budget_1=comedy_1['budget'].mean()
comedy_2=comedy[(comedy['popularity']>1) & (comedy['popularity']<2) ]
budget_2=comedy_2['budget'].mean()
comedy_3=comedy[(comedy['popularity']>2) & (comedy['popularity']<3) ]
budget_3=comedy_3['budget'].mean()
comedy_4=comedy[(comedy['popularity']>3) & (comedy['popularity']<4) ]
budget_4=comedy_4['budget'].mean()
comedy_5=comedy[(comedy['popularity']>4) & (comedy['popularity']<5) ]
budget_5=comedy_5['budget'].mean()
comedy_6=comedy[comedy['popularity']>5 ]
budget_6=comedy_6['budget'].mean()
```

[44]:
```
#plotted budget vs popularity rate
budget_list=[budget_1,budget_2,budget_3,budget_4,budget_5,budget_6]
popularity_list=[1,2,3,4,5,6]
plt.plot(popularity_list,budget_list)
plt.title('Popularity vs Budget', fontsize=20)
plt.xlabel('Popularity Rate', fontsize=16)
plt.ylabel('Budget Mean', fontsize=16)
plt.legend()
plt.grid(True)
```



Graphic shows change of budget mean accordingly popularity range (1-6). There is not any linear/non linear relationship between popularity and budget mean for comedy category. You can find my detailed observations and comment under Conclusions

Conclusions

Question 1:

My observations:

- Drama and romance movies improved popularity a lot later 2010.
- Drama movies improved popularity between 1970-1980 six times more.
- Documentary category has lowest popularity
- Comedy movies was most popular category at beginning of 60s.
- Horror movies was the most popular category at beginning of 90s.

Notes:

- CSV data is enough to answer this question for some categories.
- Movies mostly labeled with many genres. I first decided to separate categories. For example, If a movie labeled as Drama and Romance, I thought I can add 2 line for this movie, first one is Drama, second one is Romance. But that seemed to me complicated and I gave up.
- Later I decided to use genres which has more samples than others.
- Some categories has 1 sample. 1 sample is not enough to answer this question.
- I faced some difficulties when I tried to plot graphics. It was usually lack of practice. Later I overwhelmed.

Question 2:

My observations:

- I tried to observe if budget effects popularity of a movie.
- In this case, I wanted to analysis comedy category because this category had greater sample than others.
- When I plotted budget vs popularity, I could not observe any logical result.
- So I decided to divide popularity to 1,2,3,4,5,6 category and I took mean of budget for each range.
- Latest graphic showed that there is not linear/bounded relationship between this 2 category.
- For example when popularity improved to 2 from 1, mean of budget also increased.
- But when popularity improved to 3 from 2, mean of budget decreased.
- So we can say that mean of budget directly effect to popularity for comedy category

Notes:

- I selected comedy category to assess budget effects to popularity.
- When I plotted popularity vs budget with all data, I could not observe a relationship between graphics.
- Later I decided to categorize popularity to 1,2,3,4,5,6. I calculated mean of budget for each popularity category.
- Finally I drawed line chart graphic for mean of budget vs popularity range. In this case I could not observe direct relationship between these subjects. We can not say budget increasement improves popularity or opposite.
- Accordingly my opinion, csv data is enough to evaluate this question but accordingly my observation there is not linear/nonlinear relationship between budget and popularity.
- When I drawed popularity vs budget graphics I could not observe a relationship. In this case, I felt stuck because I could not find a solution how I can answer this question. Later I thought maybe I can categorize popularity in a range and try to create more clear graphic than first one. And it worked for my observation.