Investigate_a_Dataset

February 3, 2019

1 Project: Investigating a Movie Dataset

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysi with conclusions
## Introduction
We are going to explore a Movie Dataset to do our analysis.In this dataset set we have various variables such as Movie Genres,Budget for making ,revenue collected to state a few.we will try to get insight of various factors which will answer our stated questions below.from this dataset will try to give insight on following questions: >1.How a movie will perform in terms of revenue based upon the popularity?

2.What type of genre is popular?

3.Does budget really decides the popularity of a given movie or is its just the content of the Movie which matters ?

4.Finding out trends in the number of movie release by each year?

1.1.1 Importing all the necessary LibrarIes

```
In [86]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline
```

# Data Wrangling
Always before analysing any dataset we must first understand the following

Dataset structure

Variables that defines particular qualities of the dataset given

Dimension/Shape of the dataset

Each variables type.

Following through the above process can give us more understanding about the given dataset which makes our further anaylsis step easier.

load the Movie dataset

```
In [88]: df=pd.read_csv('tmdb-movies.csv')
```

1.Dataset structure

```
In [89]: df.head()
```

```
Out[89]:        id    imdb_id  popularity     budget     revenue  \
         0  135397  tt0369610   32.985763  150000000  1513528810   
         1   76341  tt1392190   28.419936  150000000   378436354   
         2  262500  tt2908446   13.112507  110000000   295238201   
         3  140607  tt2488496   11.173104  200000000  2068178225   
         4  168259  tt2820852    9.335014  190000000  1506249360   

                          original_title  \
         0                Jurassic World   
         1            Mad Max: Fury Road   
         2                     Insurgent   
         3  Star Wars: The Force Awakens   
         4                     Furious 7   

                                                         cast  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   
         1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...   
         2  Shailene Woodley|Theo James|Kate Winslet|Ansel...   
         3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...   
         4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...   

                                                     homepage          director  \
         0                      http://www.jurassicworld.com/   Colin Trevorrow   
         1                        http://www.madmaxmovie.com/     George Miller   
         2     http://www.thedivergentseries.movie/#insurgent  Robert Schwentke   
         3  http://www.starwars.com/films/star-wars-episod...       J.J. Abrams   
         4                           http://www.furious7.com/         James Wan   

                                  tagline      ...       \
         0              The park is open.      ...        
         1             What a Lovely Day.      ...        
         2     One Choice Can Destroy You      ...        
         3  Every generation has a story.      ...        
         4            Vengeance Hits Home      ...        

                                                     overview runtime  \
         0  Twenty-two years after the events of Jurassic ...     124   
         1  An apocalyptic story set in the furthest reach...     120   
         2  Beatrice Prior must confront her inner demons ...     119   
         3  Thirty years after defeating the Galactic Empi...     136   
         4  Deckard Shaw seeks revenge against Dominic Tor...     137   
```

```
                                      genres  \
0  Action|Adventure|Science Fiction|Thriller   
1  Action|Adventure|Science Fiction|Thriller   
2         Adventure|Science Fiction|Thriller   
3   Action|Adventure|Science Fiction|Fantasy   
4                      Action|Crime|Thriller   

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562   
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185   
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480   
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292   
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947   

   vote_average  release_year    budget_adj   revenue_adj  
0           6.5          2015  1.379999e+08  1.392446e+09  
1           7.1          2015  1.379999e+08  3.481613e+08  
2           6.3          2015  1.012000e+08  2.716190e+08  
3           7.5          2015  1.839999e+08  1.902723e+09  
4           7.3          2015  1.747999e+08  1.385749e+09  

[5 rows x 21 columns]
```

By looking at the structure of movie dataset,we can see there are lot of variables/qualites which are unnecessary or useless for our Analysis and creates more confusion . once we are done with Data wrangling step we can (DROP or Select only those variables needed represented by new dataframe) those column labels in Data cleaning step.

2.Variables that defines particular qualities of the dataset given From the above structure we can say the variables like popularity,budget,original_title,runtime,genres,release_year,vote_count will be very helpful in our quest to search answer's for the above proposed questions.

3.Dimension/Shape of the dataset

```
In [90]: df.shape
```

```
Out[90]: (10866, 21)
```

4.Each variables type.

```
In [91]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
```

```
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Datatypes of each essential qualities are all in preffered format so need to change the datatype

1.2 Cleaning the data

In order to further analyse the movie dataset we must clean and refine the dataset that is detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data

step1.Create a dataframe with only relevant variables to do our analysis Here as discussed earlier we are selecting particular variables/Labels and representing those with a new dtaframe named df_movies, moving forward we will refer the new data frame created instead of df for our Analysis.

```
In [92]: col_labels=['id','popularity','budget','revenue','original_title','runtime','genres','r
         df1=df[col_labels]
         df1.head()
```

```
Out[92]:        id  popularity     budget     revenue                original_title  \
         0  135397   32.985763  150000000  1513528810                Jurassic World
         1   76341   28.419936  150000000   378436354            Mad Max: Fury Road
         2  262500   13.112507  110000000   295238201                     Insurgent
         3  140607   11.173104  200000000  2068178225  Star Wars: The Force Awakens
         4  168259    9.335014  190000000  1506249360                     Furious 7
```

```
   runtime                                    genres  release_year  \
0      124  Action|Adventure|Science Fiction|Thriller          2015
1      120  Action|Adventure|Science Fiction|Thriller          2015
2      119         Adventure|Science Fiction|Thriller          2015
3      136   Action|Adventure|Science Fiction|Fantasy          2015
4      137                      Action|Crime|Thriller          2015

   vote_count
0        5562
1        6185
2        2480
3        5292
4        2947
```

step2:check for Non null/missing values and remove if any Now since we have a relevant dataframe we can check for missing values and this can be performed by following code where it return boolean 'True' if there is missing value in given column and 'False' if there is no

1.Removing Nan values

```
In [93]: df1.isnull().any(axis=0)  #axis=0 for columns and 1 for rows
```

```
Out[93]: id                False
         popularity        False
         budget            False
         revenue           False
         original_title    False
         runtime           False
         genres             True
         release_year      False
         vote_count        False
         dtype: bool
```

```
In [94]: #we will remove the entries with Nan values

         df1.dropna(inplace=True)

         df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10843 entries, 0 to 10865
Data columns (total 9 columns):
id                10843 non-null int64
popularity        10843 non-null float64
budget            10843 non-null int64
revenue           10843 non-null int64
original_title    10843 non-null object
runtime           10843 non-null int64
```

```
genres             10843 non-null object
release_year       10843 non-null int64
vote_count         10843 non-null int64
dtypes: float64(1), int64(6), object(2)
memory usage: 847.1+ KB


/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  This is separate from the ipykernel package so we can avoid doing imports until
```

step2:Remove duplicates we need to remove redundant rows from the dataframe to do that firstly we need to check wether we have any duplicates or not

```
In [95]: #check for duplicates ie check for duplicates count
         sum(df1.duplicated())

Out[95]: 1

In [96]: #removing duplicates
         df1.drop_duplicates(inplace=True)
         df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10842 entries, 0 to 10865
Data columns (total 9 columns):
id                 10842 non-null int64
popularity         10842 non-null float64
budget             10842 non-null int64
revenue            10842 non-null int64
original_title     10842 non-null object
runtime            10842 non-null int64
genres             10842 non-null object
release_year       10842 non-null int64
vote_count         10842 non-null int64
dtypes: float64(1), int64(6), object(2)
memory usage: 847.0+ KB


/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
```

In this way we performed Data wrangling and Data cleaning in order to get a relevant dataset now we can apply codes in order to answers to the above proposed questions along with visuals in order to get more insights into the dataset.

1.2.1 3.Split each Genre into separate rows

As per our observation ,there are multiple values in a generes which needs to be seperated using split and the seperator seperating those values ie(|) and create a new column named genre and will remove the older genres.

```
In [97]: df_genre=df1.join(df1.genres.str.strip('|').str.split('|',expand=True).stack().reset_in
```

```
In [98]: #drop the older genres colums since its irrelevant
         df1=df_genre.drop(['genres'],axis=1)
         df1.head()
```

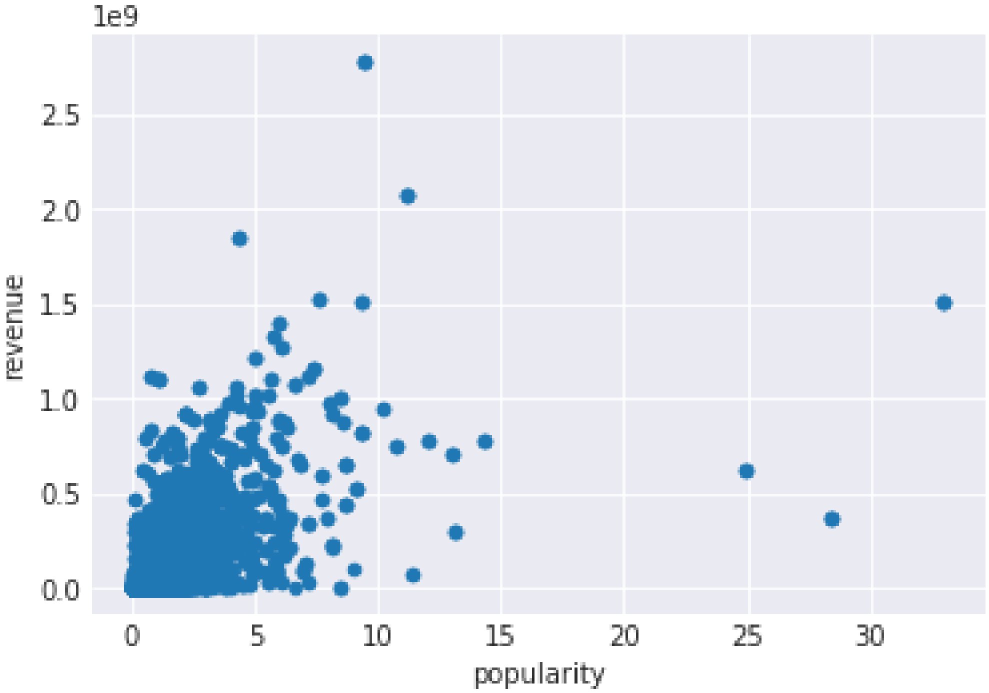
```
Out[98]:        id  popularity     budget     revenue      original_title  runtime  \
         0  135397   32.985763  150000000  1513528810      Jurassic World      124
         1  135397   32.985763  150000000  1513528810      Jurassic World      124
         2  135397   32.985763  150000000  1513528810      Jurassic World      124
         3  135397   32.985763  150000000  1513528810      Jurassic World      124
         4   76341   28.419936  150000000   378436354  Mad Max: Fury Road      120

            release_year  vote_count            genre
         0          2015        5562           Action
         1          2015        5562        Adventure
         2          2015        5562  Science Fiction
         3          2015        5562         Thriller
         4          2015        6185           Action
```

### Exploratory Data Analysis with conclusions

1.2.2 1.How a movie will perform in terms of revenue based upon the popularity?

```
In [99]: # Use this, and more code cells, to explore your data. Don't forget to add
         #   Markdown cells to document your observations and findings.
         sns.set_style('darkgrid')
         df1.plot(x='popularity',y='revenue',kind='scatter');
```



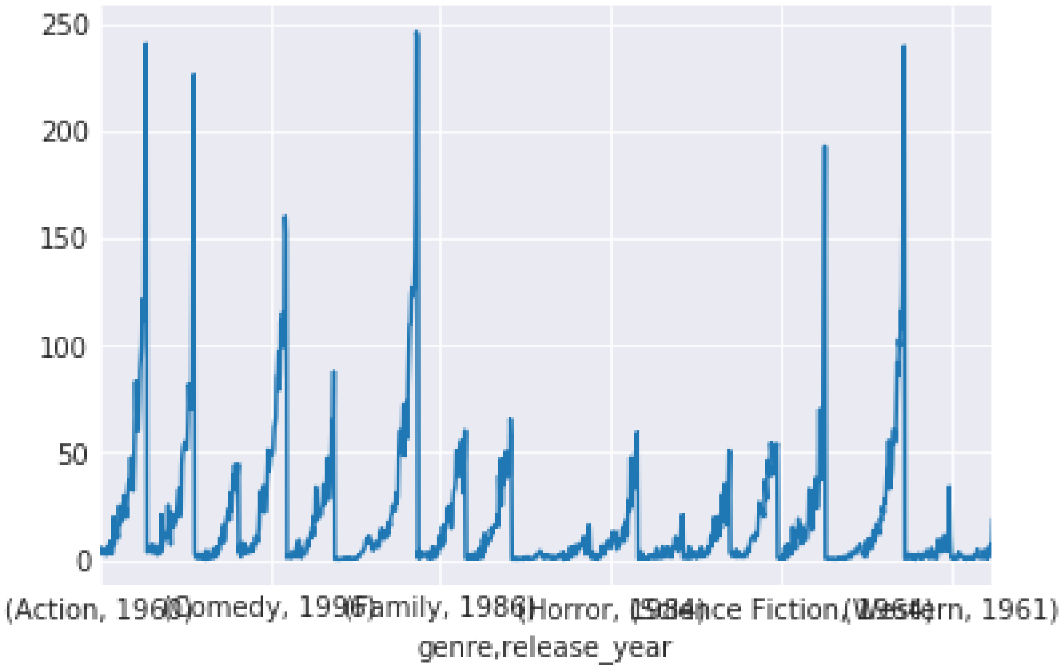
observation : From the above scattered plot we can say, >Most movies with less popularity tends to generate less revenue in Boxoffice,

As the movie garner good popularity, it tends to generate more in revenue,

There are Exceptional cases in which Movies with good popularity collected less revenue and Movies with average popularity tends to generate more revenue.

2.What type of genre is popular?

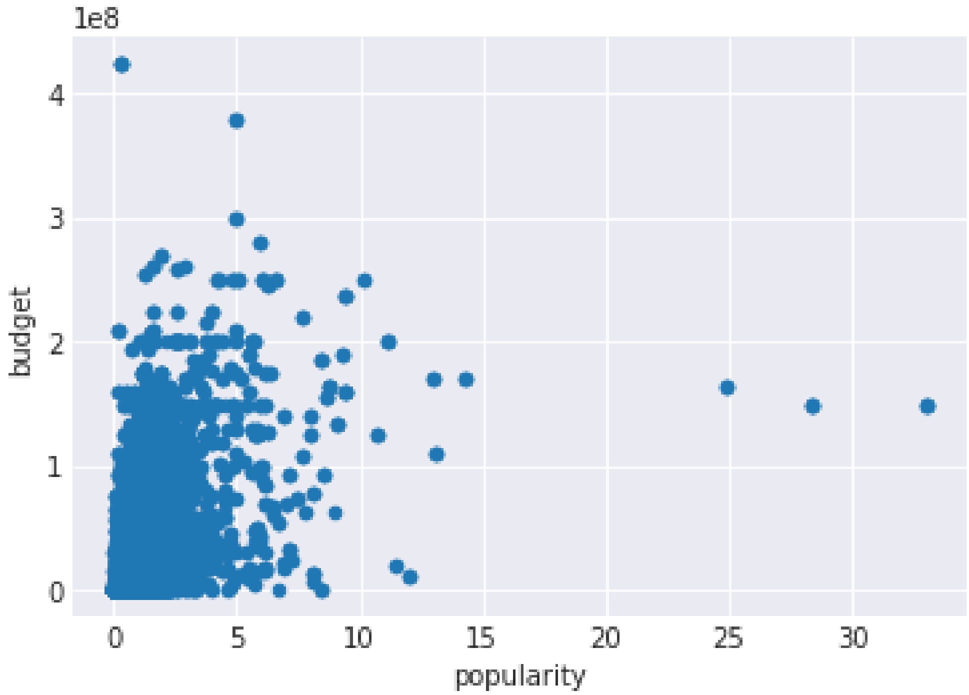
```
In [100]: #group genre with release year and find popularity for each year along with genre
          df2=df1.groupby(['genre','release_year']).sum()['popularity']
          df2.plot(x=("genre","release_year"),y="popularity",kind="line");
```



observation : From the above plot we can say,
movies of comedy,action and science fiction are most popular among the audience
Also we by seeing the plot it is evident that horror ,western and family genre are less popular among the audience

3.Does budget really decides the popularity of a given movie or is its just the content of the Movie which matters ?

```
In [102]: df1.plot(x="popularity",y="budget",kind="scatter");
```



observation : From the above scattered plot we can say that Its evident that movies with higher budgets sometimes doesnt gain that much of popularity as many of the times contents do matter only few movies are popular having higher budget. At the endwe can say most of the movies with higher budgets are not so popular

1.3 Submitting your Project

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

```
In [103]: from subprocess import call
          call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

```
Out[103]: 0
```

```
In [ ]:
```