Udacity Machine Learning Nanodegree

Capstone Project

Movie Box-Office Predictive Analysis

Joel Vilanilam Zachariah

May 19th, 2019

I. Definition

Project Overview

In a world where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever. The multi-billion dollar industry involves a lot of risk in terms of how well the public will perceive the result that the team creates. Ideally, the revenue gained should be greater than the production budget to keep the work sustainable.

According to wikipedia[1], profit is a key force in the industry, due to the costly and risky nature of filmmaking. For example, *Avengers Endgame* by Marvel Studios garnered over $1.2 billion dollars[2] in the first weekend after release[3], while the budget for the movie was $356 million dollars. The production had to make risk assessments and decide the best conditions to arrive at this successful end result.

By drawing relevant conclusions the production team can optimize their resources to produce the movie with better success probability. I am personally interested to work on this domain because it will be interesting to not only read about the past movies but also to understand the factors that affect the audiences satisfaction in the content.

Problem Statement

A successful movie requires proper planning to minimize production cost and maximize revenue, that is, to maximize profit. Though it is true that experience plays a vital role in deciding the course of action to take for the creation of the movie, the audience acceptance rate can be enhanced if research is done. It can be done from past movies and studying the inter-relations in the system. This risk assessment[4] can improve the production teams success rate.

We need to predict the overall worldwide box office revenue by studying the correlations in the system and estimating which parameters affect the result the most (in terms of revenue). Following that, we need to train the model with the training data set and measure its accuracy with the testing data set.

The purpose is this project is to identify the main characteristics that affect the revenue of the movie based on a handful of machine learning algorithms that we shall test on the remaining features. The experimental results will be assessed by statistical metrics, As we get higher scores against these metrics, we will be able to better judge the success of a movie.

Metrics

For the training phase, we will utilize all the 23 parameters while for testing phase, the first 22 parameters are used to predict the revenue. Mean absolute error[7] will be the measure of quality for the model. Mean absolute error (MEA) is the measure of difference between two continuous variables.

Mean absolute error is a suitable metric for this problem as our task is to predict the revenue that a movie can earn based on the factors such as crew, genre etc. By using MEA as the loss function, we can estimate upto what extent did we deviate from the expected result.

In our case, we will be calculating the mean of the absolute error between predicted revenue values and actual revenue values of testing dataset movies. Over several epoch (In the case of neural networks), we shall try to minimize the error and then predict the revenue for a given movie characteristics.

II. Analysis

Data Exploration

The dataset is from *The Movie Database* (TMDB)[5], which is a database with metadata of over 3,000 movies from the past. I have used the dataset provided for a competition on Kaggle by TMDB[6].

The "tmdb-data.csv" data file has 3000 rows of data across 23 features.

Input Variables:

1. Id [integer] (Serial number in the database)
2. Belongs_to_collection [object file] (Collection it belongs to)
3. Budget [integer] (Cost of production)
4. Genres [object file] (Theme of the movie)
5. Homepage [object file] (Link to the internet page)
6. Imdb_id [object file] (reference id at IMDB)
7. Original_language [object file] (primary movie language)
8. Original_title [object file] (Title of the movie)
9. Overview [object file] (Short description of the movie)
10. Popularity [float] (relative measure of popularity)
11. Poster_path [object file] (Directory pathway to movie poster)
12. Production_companies [object file] (Name of production companies)
13. Production_countries [object file] (Country of production)
14. Release_date [object file] (Date when the movie was released)

15. Runtime [integer] (duration of the movie in minutes)
16. Spoken_languages [object file] (Languages used by actors)
17. Status [object file] (Released/ Not yet released/ etc)
18. Tagline [object file] (The one-line caption of the movie)
19. Title [object file] (Movie title)
20. Keywords [object file] (Related keywords of the movie)
21. Cast [object file] (List of actors)
22. Crew [object file] (List of crew members)

Let us examine a few attribute in more detail:

- Belongs_to_collection: [{'id', 'name', 'poster_path', 'backdrop_path'}]
- Genre: [{'id', 'name'}]
- Production_companies: [{'name', 'id'}]
- Production_countries: [{code, 'name'}]
- Keywords: [{'id', 'name'}]
- Cast: ['cast_id', 'character', credit_id', 'gender', 'id', 'name', 'order', 'profile_path']
- Crew: [{'credit_id', 'department', 'gender', 'id', 'job', 'name', 'profile_path'}]

Output Variable:

1. Revenue [integer] (Revenue of the movie)

Missing Attribute Values:

There are several missing values in the dataset. These can be treated as possible using imputation techniques.

A portion of the dataset can be visualized as follows:

```
df_movie.head(5)
```

	id	belongs_to_collection	budget	genres	homepage	imdb_id	original_language	original_title	overview	popularity	...	release_date
0	1	[{'id': 313576, 'name': 'Hot Tub Time Machine ...	14000000	[{'id': 35, 'name': 'Comedy'}]	NaN	tt2637294	en	Hot Tub Time Machine 2	When Lou, who has become the "father of the In...	6.575393	...	2/20/15
1	2	[{'id': 107674, 'name': 'The Princess Diaries ...	40000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	tt0368933	en	The Princess Diaries 2: Royal Engagement	Mia Thermopolis is now a college graduate and ...	8.248895	...	8/6/04
2	3	NaN	3300000	[{'id': 18, 'name': 'Drama'}]	http://sonyclassics.com/whiplash/	tt2582802	en	Whiplash	Under the direction of a ruthless instructor, ...	64.299990	...	10/10/14
3	4	NaN	1200000	[{'id': 53, 'name': 'Thriller'}, {'id': 18, 'n...	http://kahaanithefilm.com/	tt1821480	hi	Kahaani	Vidya Bagchi (Vidya Balan) arrives in Kolkata ...	3.174936	...	3/9/12
4	5	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 53, 'nam...	NaN	tt1380152	ko	마린보이	Marine Boy is the story of a former national s...	1.148070	...	2/5/09

5 rows × 23 columns

A portion of the statistics of the dataset:

```
df_movie.describe(include='all')
```

	id	belongs_to_collection	budget	genres	homepage	imdb_id	original_language	original_title	overview	p
count	3000.000000	604	3.000000e+03	2993	946	3000	3000	3000	2992	300
unique	NaN	422	NaN	872	941	3000	36	2975	2992	
top	NaN	[{'id': 645, 'name': 'James Bond Collection', ...	NaN	[{'id': 18, 'name': 'Drama'}]	http://www.transformersmovie.com/	tt0084788	en	The Gift	A seasoned team of bank robbers, including Gor...	
freq	NaN	16	NaN	266	4	1	2575	2	1	
mean	1500.500000	NaN	2.253133e+07	NaN	NaN	NaN	NaN	NaN	NaN	
std	866.169729	NaN	3.702609e+07	NaN	NaN	NaN	NaN	NaN	NaN	1
min	1.000000	NaN	0.000000e+00	NaN	NaN	NaN	NaN	NaN	NaN	
25%	750.750000	NaN	0.000000e+00	NaN	NaN	NaN	NaN	NaN	NaN	
50%	1500.500000	NaN	8.000000e+06	NaN	NaN	NaN	NaN	NaN	NaN	
75%	2250.250000	NaN	2.900000e+07	NaN	NaN	NaN	NaN	NaN	NaN	1
max	3000.000000	NaN	3.800000e+08	NaN	NaN	NaN	NaN	NaN	NaN	29

11 rows × 23 columns

There are missing values that are causing the system to not generate useful statistics of the table for most of the columns.

The columns with number of missing values in each is as given below:

```
df_movie.isna().sum().sort_values(ascending=False)
```

```
belongs_to_collection    2396
homepage                 2054
tagline                   597
Keywords                  276
production_companies      156
production_countries       55
spoken_languages           20
crew                       16
cast                       13
overview                    8
genres                      7
runtime                     2
poster_path                 1
original_language           0
budget                      0
imdb_id                     0
revenue                     0
original_title              0
popularity                  0
release_date                0
status                      0
title                       0
id                          0
dtype: int64
```

In the later stages, we shall perform the necessary preprocessing steps (such as imputation techniques) to deal with missing values.

Exploratory Visualization

Let us examine the features in more detail.

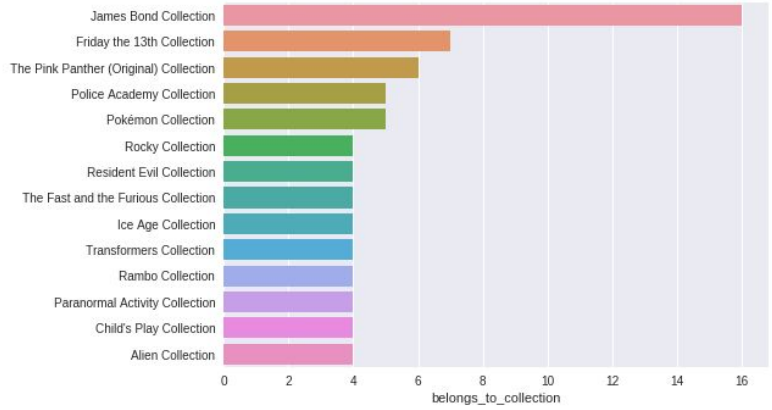
The **‘belongs_to_collection’** column indicates that a movie belongs to a particular franchise (eg: *The spy who loved me* is a movie in the *James Bond* Series[8]). Let us see how many collections are there in the dataset:

```
df_movie['belongs_to_collection'].apply(lambda x:len(x) if x!= {} else 0).value_counts()
```

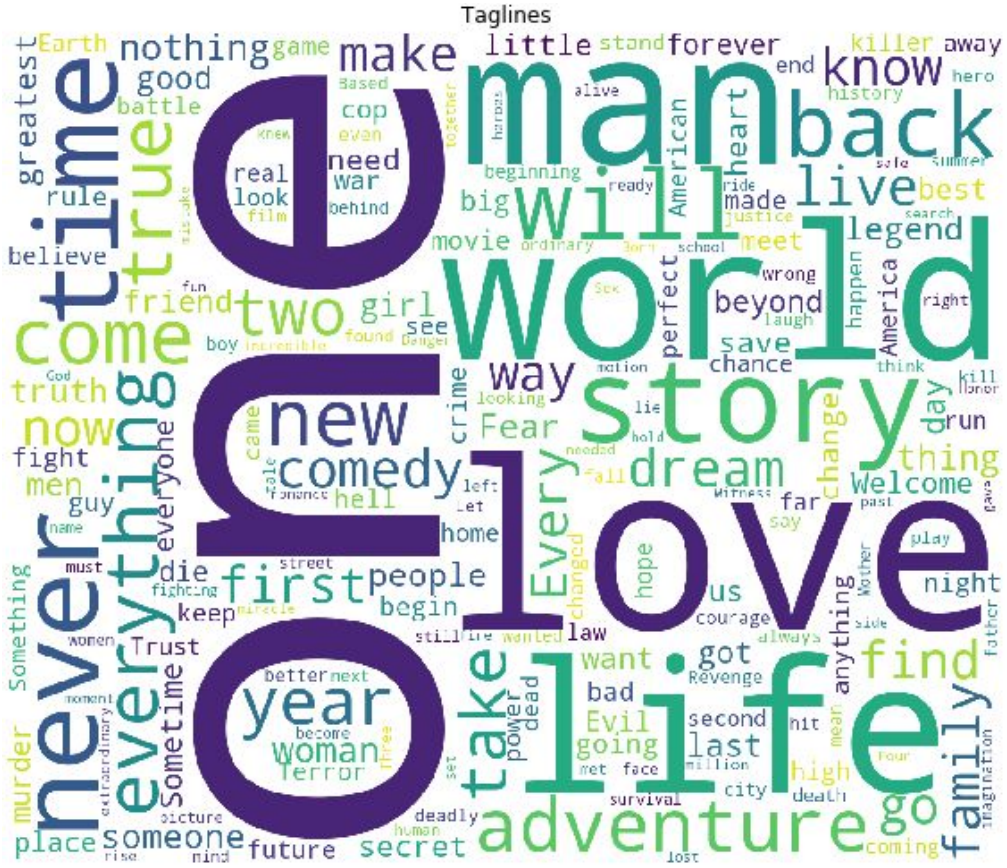
```
0    2396
1     604
Name: belongs_to_collection, dtype: int64
```

Of the 3000 movies in the dataset, Only 604 belong to some collection while the rest are independent stand alone movies.

```
collections=df_movie['belongs_to_collection'].apply(lambda x : x[0]['name'] if x!= {} else '?').value_counts()[1:15]
sns.barplot(collections,collections.index)
plt.show()
```



The ‘**Tagline**’ column has an input for 2403 movies while 597 movies do not have one. A word cloud generated based on this data is shown below:

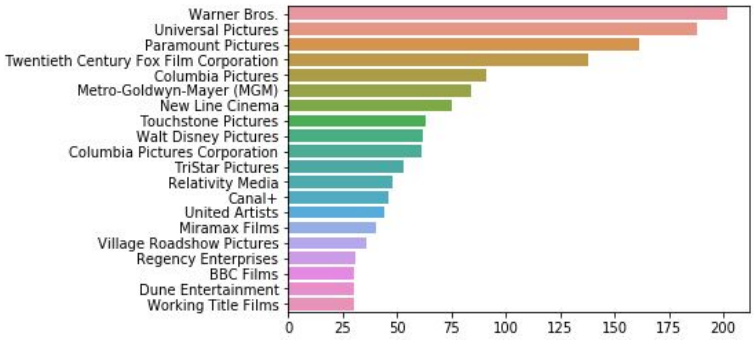


Similarly, the word cloud for '**Keyword**' column is as shown below:

The top 20 production companies are shown below:

```
x=df_movie['production_companies'].apply(lambda x : [x[i]['name'] for i in range(len(x))] if x != {} else []).values
count = Counter([i for j in x for i in j]).most_common(20)
sns.barplot([val[1] for val in count], [val[0] for val in count])
```

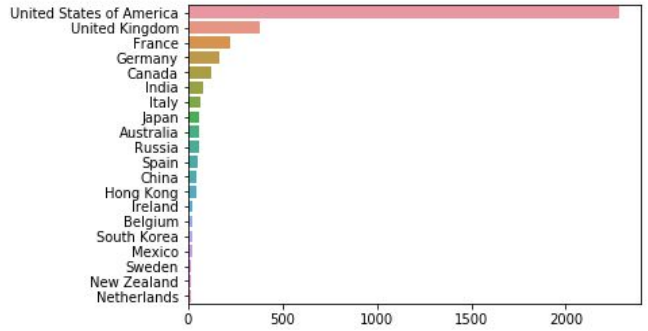
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a1b5f7d30>
```



The top 20 production countries are shown below:

```
countries=df_movie['production_countries'].apply(lambda x: [i['name'] for i in x] if x!={} else []).values
count=Counter([j for i in countries for j in i]).most_common(20)
sns.barplot([val[1] for val in count],[val[0] for val in count])
```

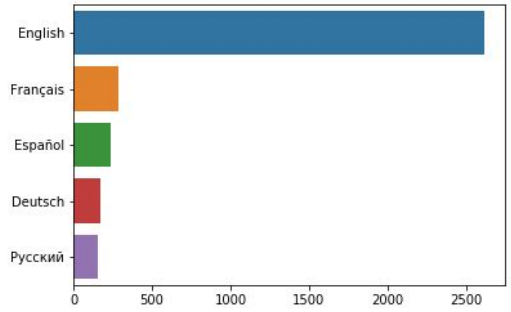
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a1b4d20b8>
```



English is the most common language used in movies:

```
lang=df_movie['spoken_languages'].apply(lambda x: [i['name'] for i in x] if x != {} else [])
count=Counter([i for j in lang for i in j]).most_common(5)
sns.barplot([val[1] for val in count],[val[0] for val in count])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a1b5eefd0>
```

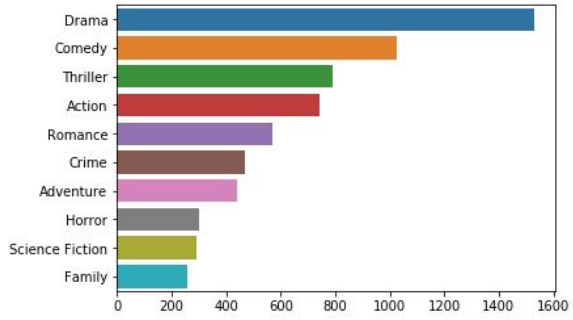


From bar graph, we see that most movies are in English, then French.

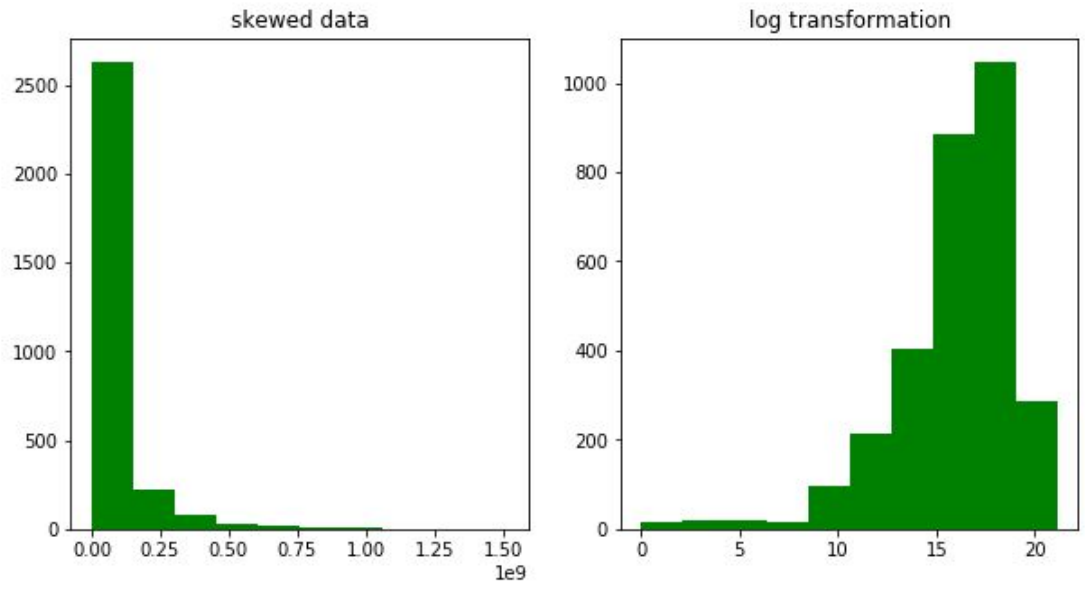
More than 1500 movies fall under the drama genre:

```
genre=df_movie['genres'].apply(lambda x: [i['name'] for i in x] if x != {} else [])
count=Counter([i for j in genre for i in j]).most_common(10)
sns.barplot([val[1] for val in count],[val[0] for val in count])
```

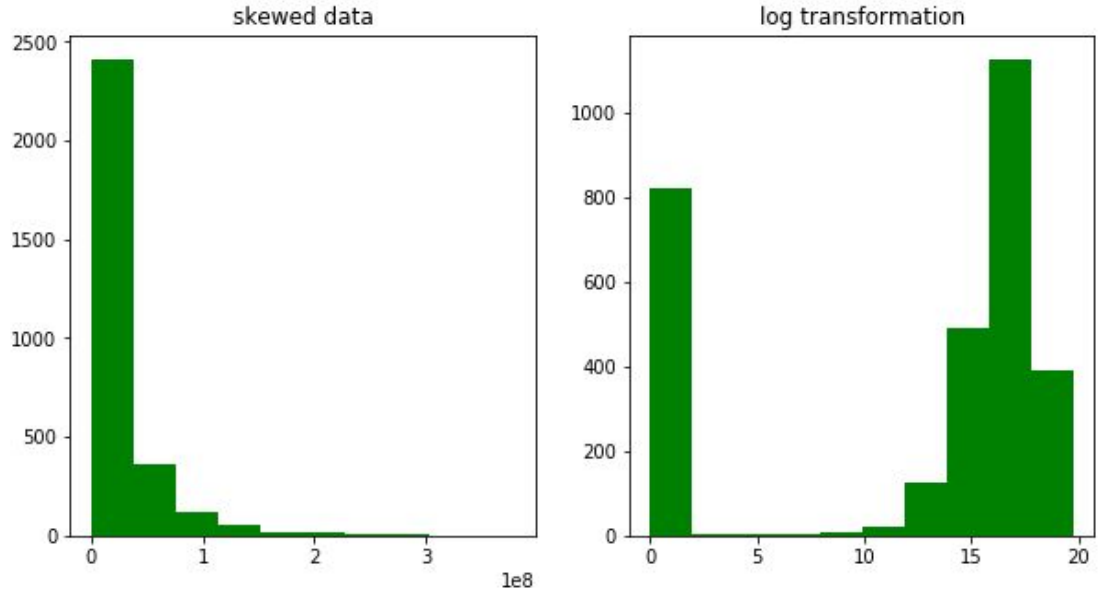
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a1b652438>
```



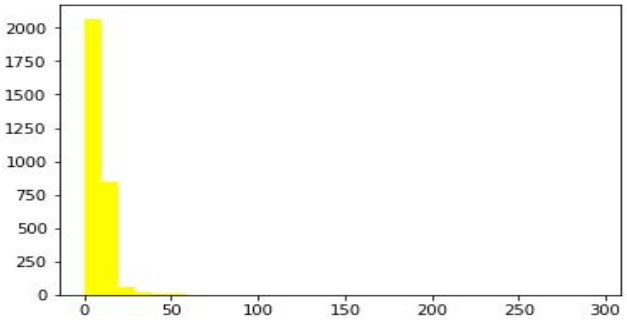
Revenue graph appears to be skewed and so we perform the logarithmic transformation:



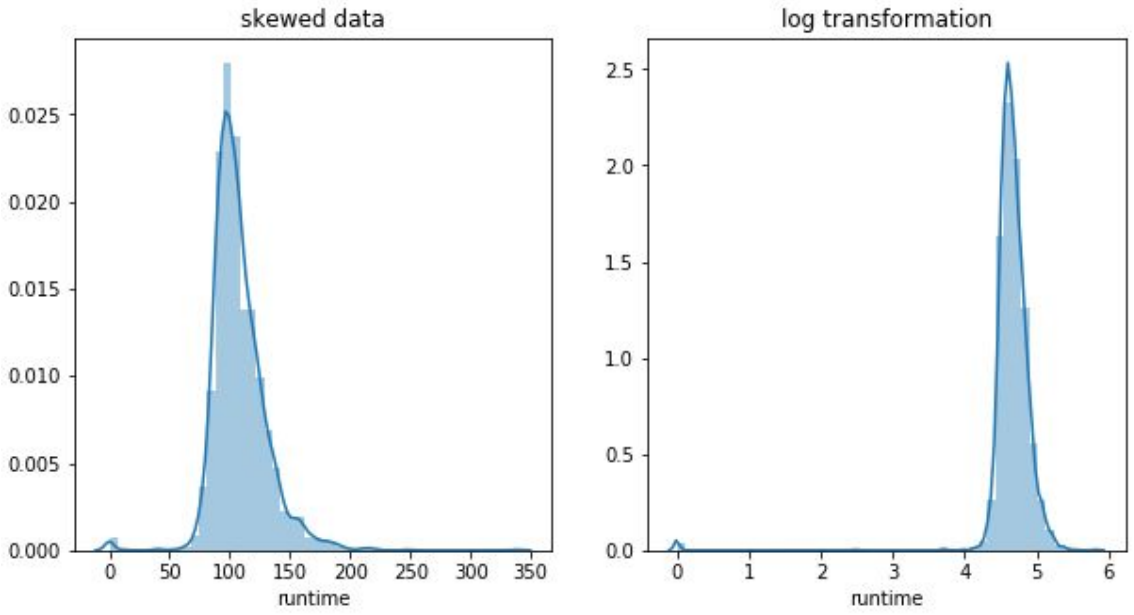
Budget graph also appears to be skewed:



Most of the movies tend to have a low popularity measure, but the range of the popularity value is 0-50:

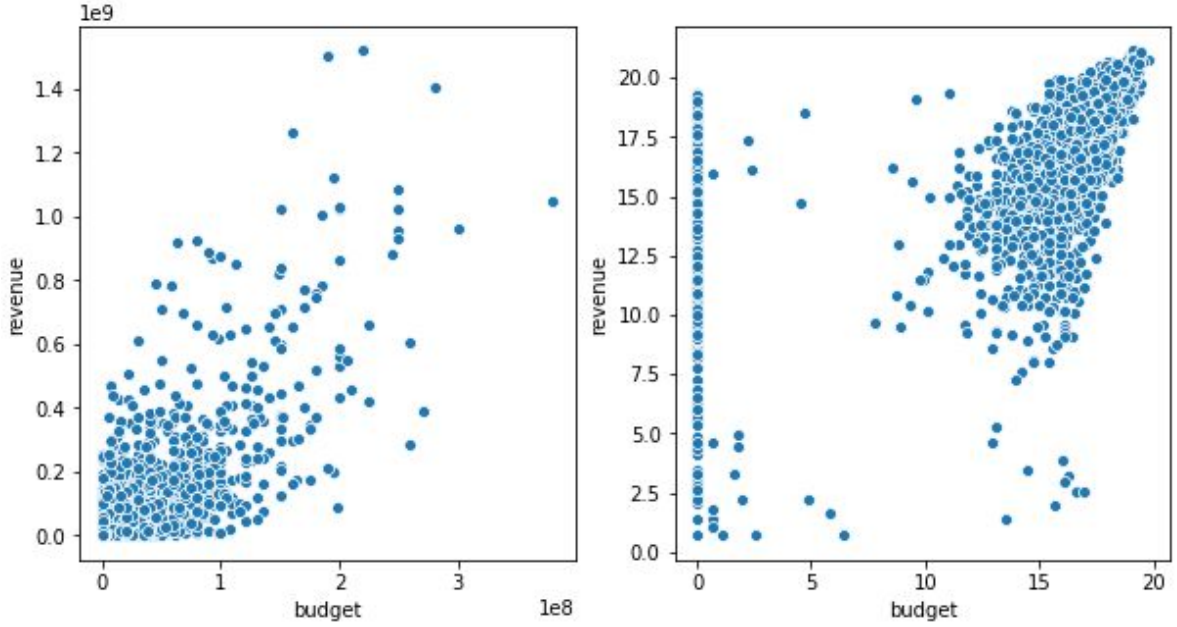


Most of the movies have a runtime around 100 minutes:

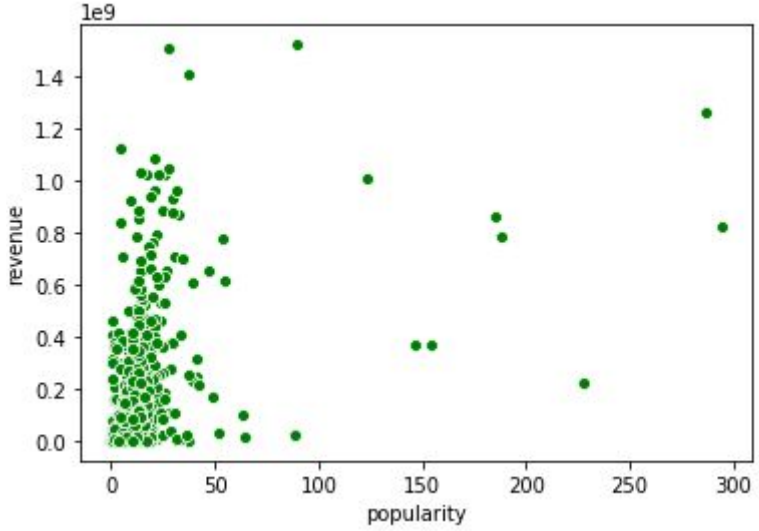


Now let us examine the correlations in the dataset:

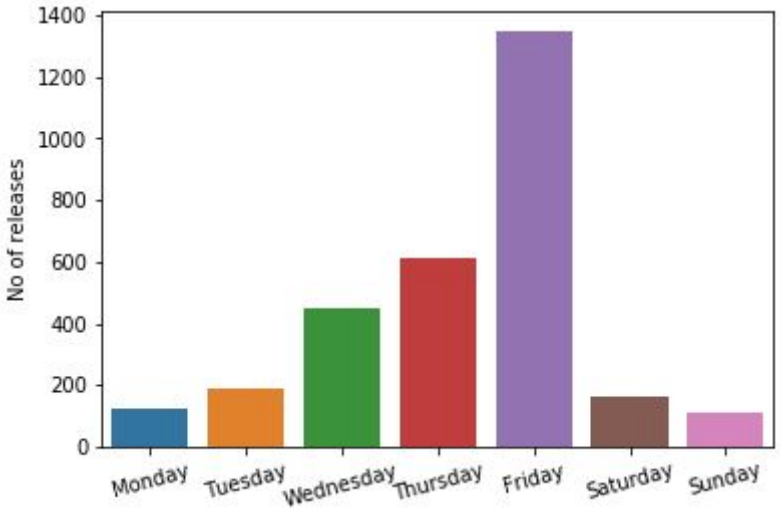
There is a correlation between budget and revenue, but it is not linearly dependent:



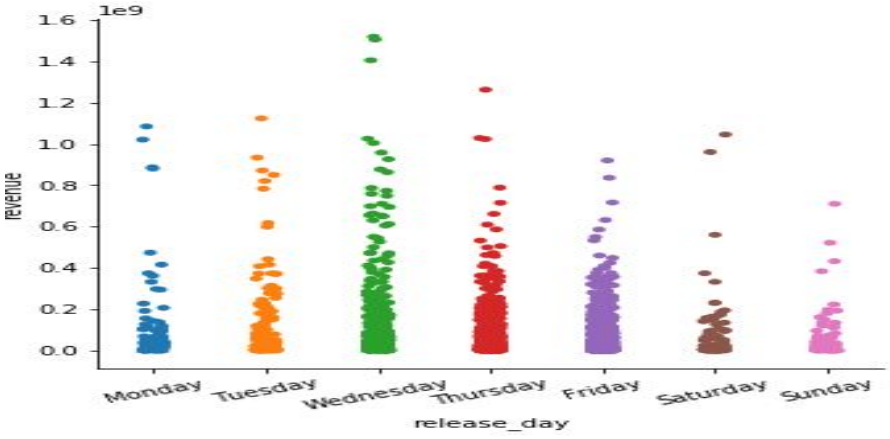
No significant relation between the revenue and popularity:



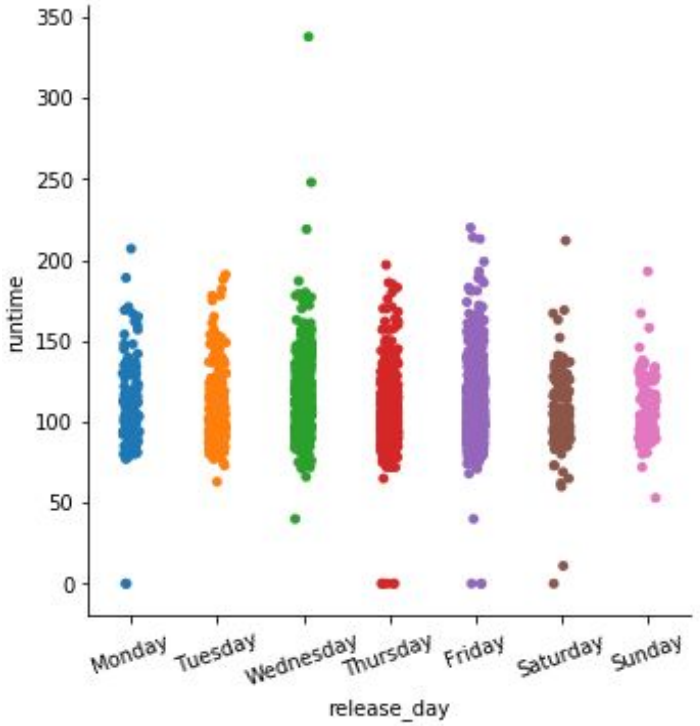
It appears to be that most movies are released on a Friday. This might be to gain the weekend audience:



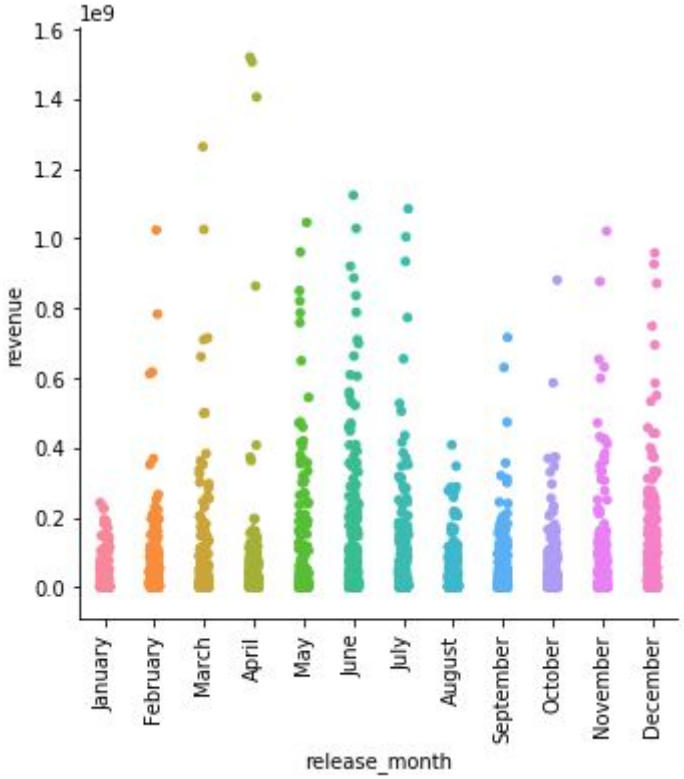
Revenue gained according to the day of release is shown below:



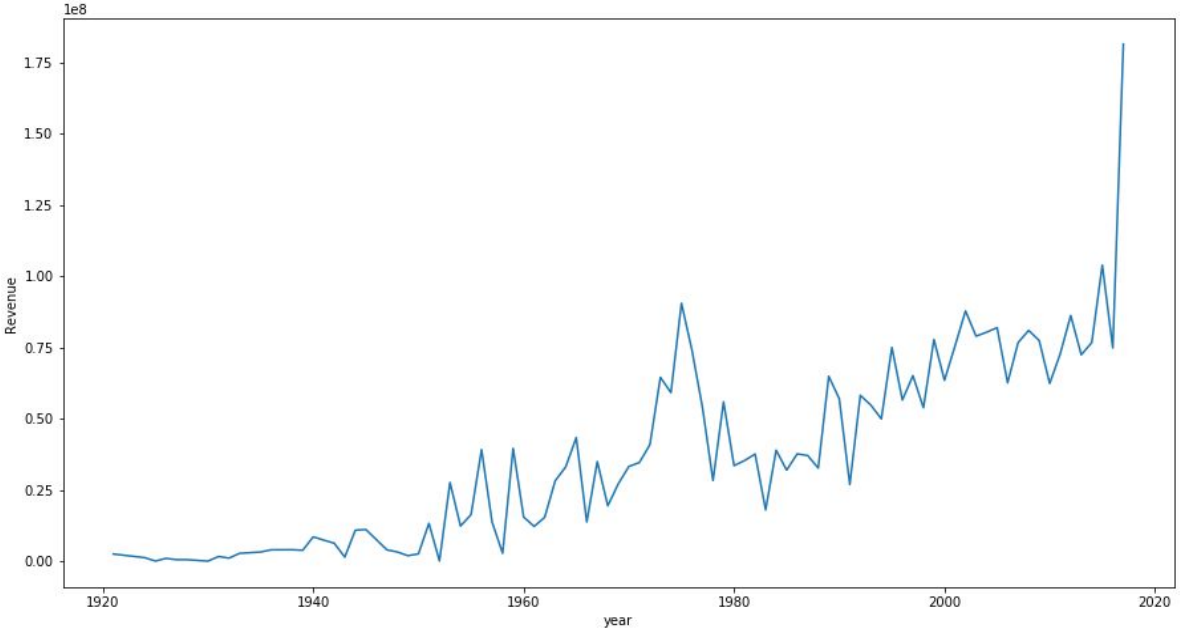
Runtime of the movie with respect to the release day:



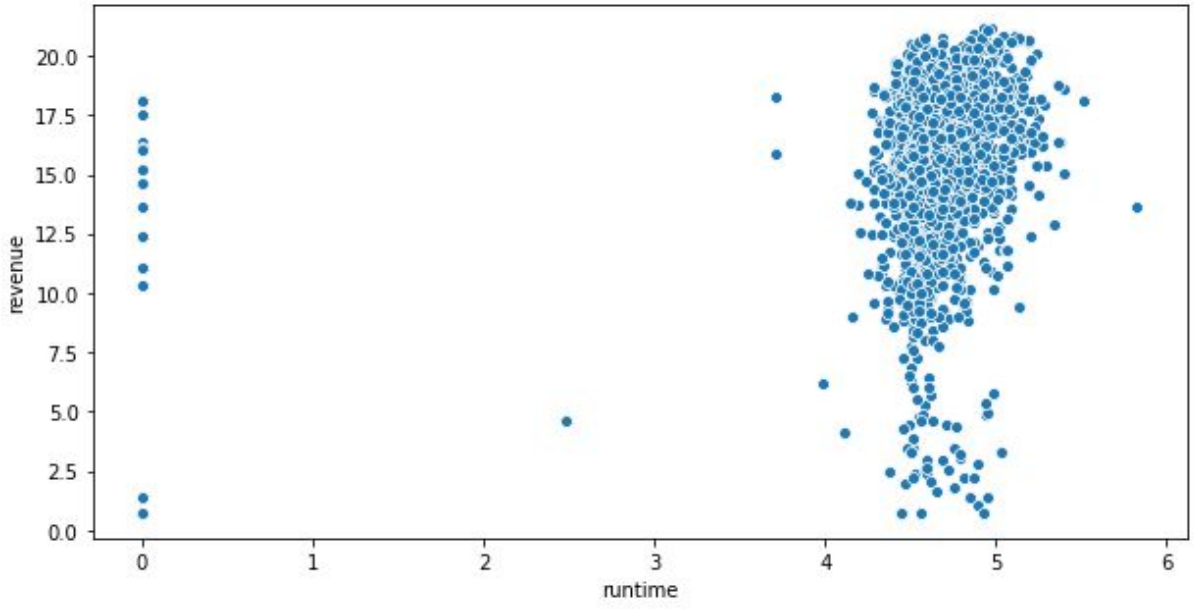
Revenue gained with respect to the month of release of the movie:



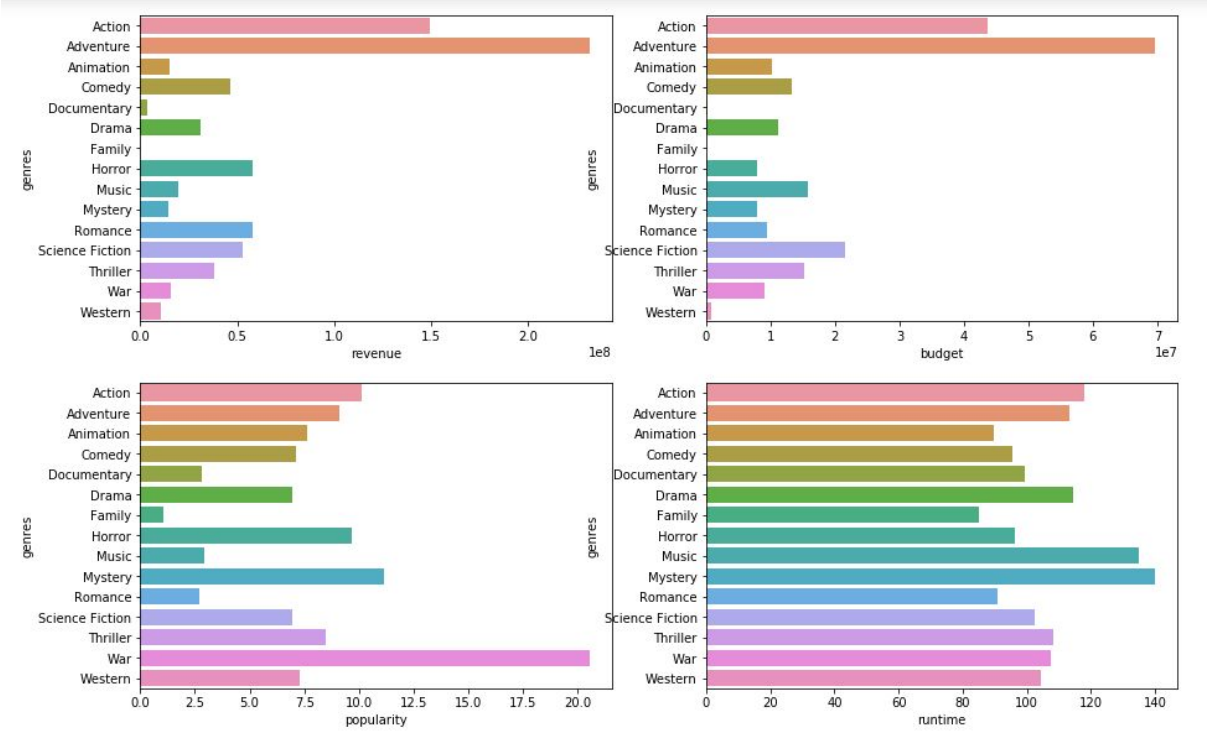
Revenue gained over the years in the movie industry:



Revenue gained with respect to the runtime of the movie:



It appears to be that the genre vs budget and genre vs revenue graphs are quite similar. War themed movies are most in demand and Music, mystery movies have the most runtime.



Algorithms and Techniques

1. *Dummy Regressor:* This is the baseline model that takes the mean of the training data to predict the target variable.

2. *Decision Tree*: Starts with a single mode and then branches out with a decision being made at every branch point. It can be used to predict whether a particular variable would matter in the a revenue of the movie. I have used a decision tree regressor for the problem.

3. *Linear Regression*: A linear regression is performed based on the training dataset and then it is used to predict the target variable values.

4. *Keras Model*: A custom made Keras model with suitable layers and filtering mechanisms to perform the action.

Benchmark Model

As a benchmark, I plan to compare the result of my model against the *Dummy Regressor* model that uses only the mean of the training dataset revenue. This is a naive solution but works well to set the baseline for the system.

III. Methodology

Data Preprocessing

During the Data exploration phase, we discovered a lot of parameters tend to show inter-relations so their ratio has significant importance for our model to predict the revenue. Accordingly, the following ratios were computed:

1. Budget-Runtime Ratio
2. Budget-Popularity Ratio
3. Budget-Release Year Ratio
4. Release Year-Popularity Ratio
5. Popularity - Release Year Ratio

Each of these have a dedicated newly formed column in the dataset.

Additionally, the following columns were created for the stated reasons:

1. has_homepage: set to 1 if a homepage exists, 0 otherwise
2. _num_Keywords: set to number of keywords present if available, 0 otherwise
3. _num_cast : set to number of cast actors present if available, 0 otherwise
4. isbelongto_coll: set to 0 unless the movie belongs to a collection (then 1)
5. isTaglineNA: set to 0 unless a tagline exists for the movie, in which case 1
6. isOriginalLangiageEng: set to 0 unless the movie is in english (then 1)
7. ismovie_released: set to 1 unless movie has not been released (then 1)
8. no_spoken_languages: set to the number of spoken languages
9. original_title_letter_count: set to the letter count of original title
10. original_title_word_count: set to word count of original title
11. title_word_count: set to the word count of the title
12. overview_word_count: set to the word count of the overview
13. tagline_word_count: set to the word count of the tagline
14. collection_id: set to the collection ID of the movie if applicable
15. production_countries_count: set to the number of production countries
16. production_companies_count: set to the number of production companies
17. cast_count: set to the number of cast members
18. crew_count: set to the number of crew members

The newly formed columns were normalized and the previously used columns which are now repeated in information are dropped. Thus we have cleaned our data to better represent the same information in a easy-to-interpret format by the model.

Implementation

After the preprocessing stage, we define a get_json function to create a dictionary of information encoded in the JSON files. Following this, the dataset is passed through the function and stored in a dictionary. Some of these entries are removed if found to contain less than the threshold value.

At this point we can observe that the data frame contains 173 columns due to the preprocessing stage. ‘y’ holds the target variable (revenue) values while ‘x’ holds the training dataset (all columns except revenue).

Refinement

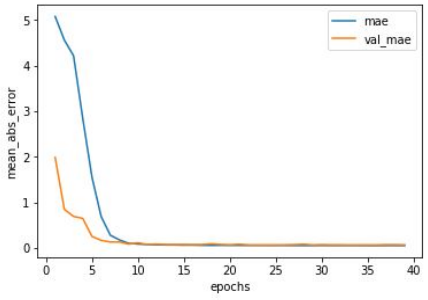
- We test against the baseline model *‘DummyRegressor’* with the mean as the parameter. The mean absolute error was found to be 2.14799
- We test using the *Decision Tree Regressor* and find the mean absolute error to be 2.02538
- We test using *Linear Regression* and find the mean absolute error to be 1.43821
- Finally we test using a custom made Keras model (neural network) with following parameters:
    - First layer has 356 nodes, relu activation function and l1 regularization.
    - Second layer has 256 nodes, relu activation function and l1 regularization.
    - Final layer has a single node
    - K Fold clustering is used with 3 splits
    - Model is run for 40 epochs using validation data.

IV. Results

Model Evaluation and Validation

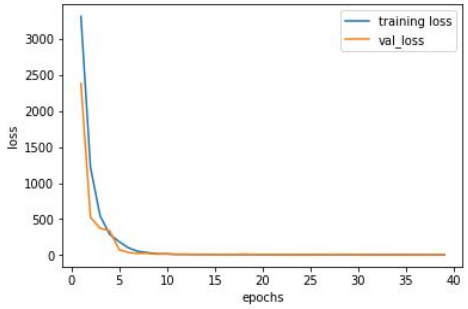
```
mae=hist.history['mean_squared_logarithmic_error']
plt.plot(range(1,40),mae[1:],label='mae')
plt.xlabel('epochs')
plt.ylabel('mean_abs_error')
mae=hist.history['val_mean_squared_logarithmic_error']
plt.plot(range(1,40),mae[1:],label='val_mae')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f0e50882668>
```




```
mae=hist.history['loss']
plt.plot(range(1,epochs),mae[1:],label='training loss')
plt.xlabel('epochs')
plt.ylabel('loss')
mae=hist.history['val_loss']
plt.plot(range(1,epochs),mae[1:],label='val_loss')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f0e508fb4e0>
```



Justification

Though there is scope for improvement in the final model, the error has significantly decreased as indicated visually and so I resort to using this model for this report.

V. Conclusion

Reflection

The most time consuming phase proved to be the data exploration phase as determining what correlations mattered was important for understanding how to preprocess and restructure the dataset.

Improvement

Improvement would be to use models such as Xgboost or LightGBM as such powerful algorithms have proven to be quite resourceful in such applications.

References

[1] https://en.wikipedia.org/wiki/Film
[2] https://www.vox.com/2019/4/29/18521581/avengers-endgame-box-office-1-2-billion
[3] https://www.the-numbers.com/movie/Avengers-Endgame-(2019)
[4] https://www.filmsourcing.com/film-production-risk-assessment/
[5] https://www.themoviedb.org/en
[6] https://www.kaggle.com/c/tmdb-box-office-prediction
[7] https://en.wikipedia.org/wiki/Mean_absolute_error
[8] https://en.wikipedia.org/wiki/The_Spy_Who_Loved_Me_(film)