Faculty of Engineering
Cairo University

Big Data Final Document

TMDB Box Office Prediction

Submitted to:
Eng Hussein

Submitted by:

- Ehsan Sayed — Email: ehsansayed2019@gmail.com
- Reham Abdellatif — Email: reham.abdallatef47@gmail.com
- Sara Mohamed — Email: sarahasanien932@gmail.com
- Omar Hashim — Email: omarhashim94@gmail.com

List of Contents

1. Brief problem description

In a world...where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever. But what movies make the most money at the box office? How much does a director matter? Or the budget? For production companies that pay much money to produce one movie, they are always fear about the revenue. Will they get the money they paid or not? How much money they will gain? (Problem Statement)

So, It is obvious that we need something to tell us whether we should produce this movie and start working on it or not? Will this movie make the money we expect or not? That is exactly what we seek to predict in this project, given some information such as the language, budget, crew or production company, we try to predict how much revenue can the movie make at the box office.

2. Project Pipeline

a. Discovery Phase

In this phase we start to explore our problem domain and available resources and ask ourselves questions like “what may be different about a movie that make high revenue and other not” is the crew, budget,cast, production companies,production countries or release date make change. Then we asked ourselves another questions “Do we have sufficient data?” and we found a dataset (17 MB exactly 7398 Movies) which has been collected from TMDB and can be found HERE.

b. Data Preparation

In this phase, we start to explore the available data through some visualizations to get insights and see what is most important, what to ignore, what affect most and what does not affect at all.
The following points are some of the insights we conclude from the data visualization phase:

- There is a very clear linear relation between budget and the revenue.
- The movies which have home page makes higher revenue than movies with no home page.
- The movies which in English language makes higher revenue than movies is not in English.
- The count of the crew and cast makes difference.

The following points are some of the insights that make us ignore some features:

- The poster image and poster id were ignored.
- The status of the movie whether was released or not was ignored because there are only 4 rows out of 3000 which were not released so these small number of rows did not help us to make use of this feature.
- The overview and the title of the movie were ignored.

In the end of this phase, our new dataset that consists of more than 100 columns which are extracted from the original dataset such as (cast_count - crew_count - hasHomePage - Production_Countries_Count) and some ratios such as (budget_per_year_ratio or popularity_mean_year), all these features can be found in our code files.

c. Model Planning and Building

a. Model Planning

In this part, we start to plan how our model will look like, we have two ways. The first way is to deal with the problem as a regression problem and try to predict the revenue as a continuous number, we can use many regression algorithms if we decide this way such as linear regression or XGBoost.

The second way is to deal with the problem as a classification problem and try to predict the revenue in categories. For example, very high , high , medium , low, very low. we can use many classification algorithms if we decide this way such as logistic regression or K-NN, train a neural network, build an SVM or decision tree.

b. Model Building

In this part, we select more than one model that we think they will achieve high accuracy which are:

- SVM
- Linear Regression
- Decision Tree
- XGBoost
- Lbgm

(More details in model building section)

d. Results

In this phase, we start to test our models accuracy to select the best model. Our final results are :

- SVM — Accuracy: 56.5% - 72.6%
- Decision Tree — Accuracy: 43.7% - 60.3%
- Linear Regression — RSME: 1.52
- XGBoost — RSME: 1.59
- Lgbm — RSME: 1.74

(More details in results section)

e. Enhancements and Optimizations

In this phase, we found that our model accuracy is not too high. So, we think of two enhancements which can be done in the future and may increase the model accuracy significantly, which are data accuracy and data size.
(More details in enhancements section)

3. Analysis and solution of the problem

a. Data Visualization

The first part of the project concerns with the Visualization, we try to plot the relation between the revenue and some of available features which are:

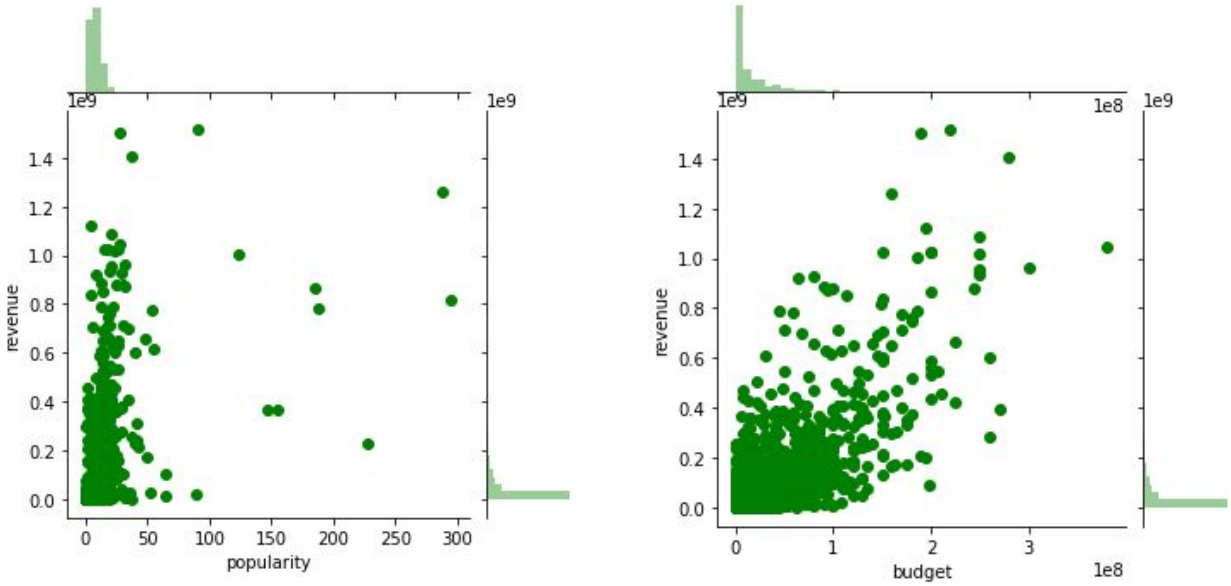
- Budget
- Homepage
- Original_language
- Popularity
- Release_date
- Runtime
- Tagline
- Crew
- Genres
- Spoken_languages
- Production_companies
- Production_countries

There are some data columns that we found they are not important so we ignore them in this phase, which are:

- Imdb_id
- Title
- Keywords
- Original_title
- Overview
- Poster_path
- Status

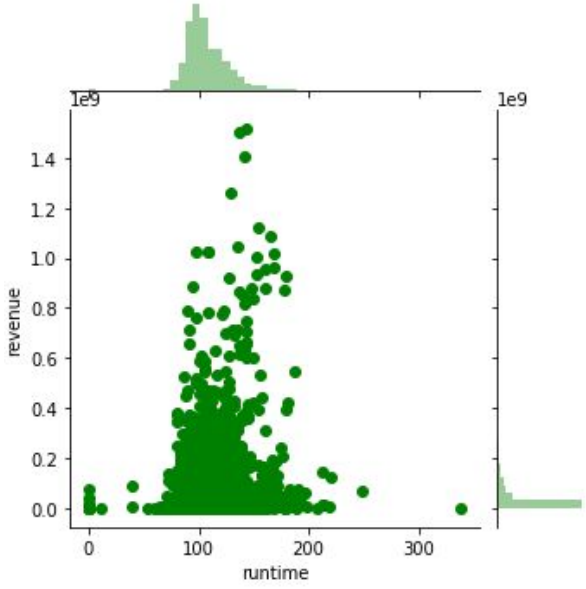
Note: In the following pages we will show the most important plots that help us in the upcoming phases. However, we plot each feature separately with the revenue in the code files.

Figure 1



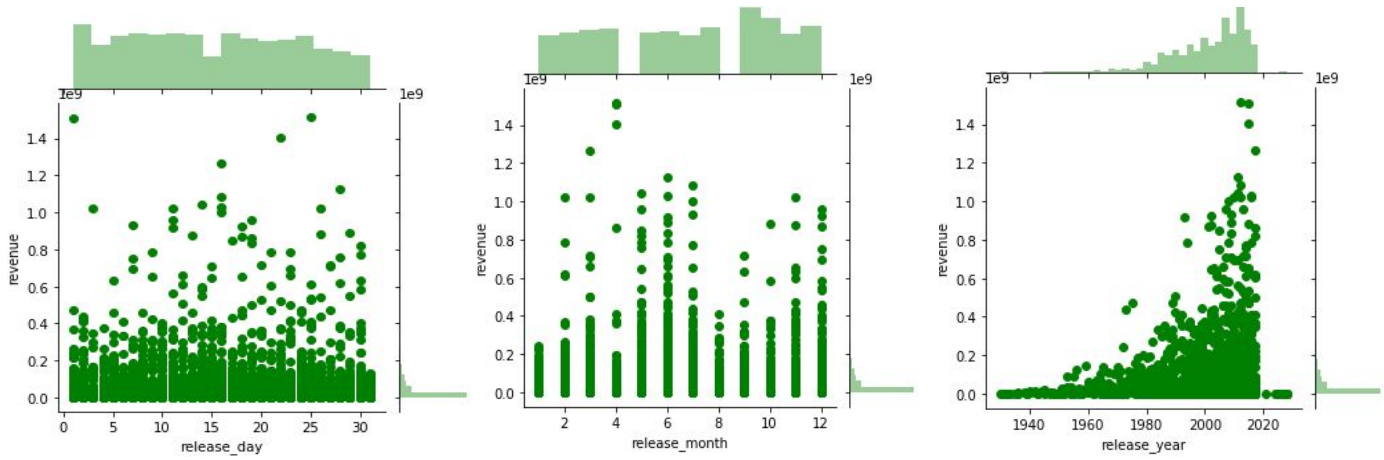
Observations: The relation between the revenue and budget is clearly linear. In contrast, popularity does not change with the increase of the revenue.

Figure 2



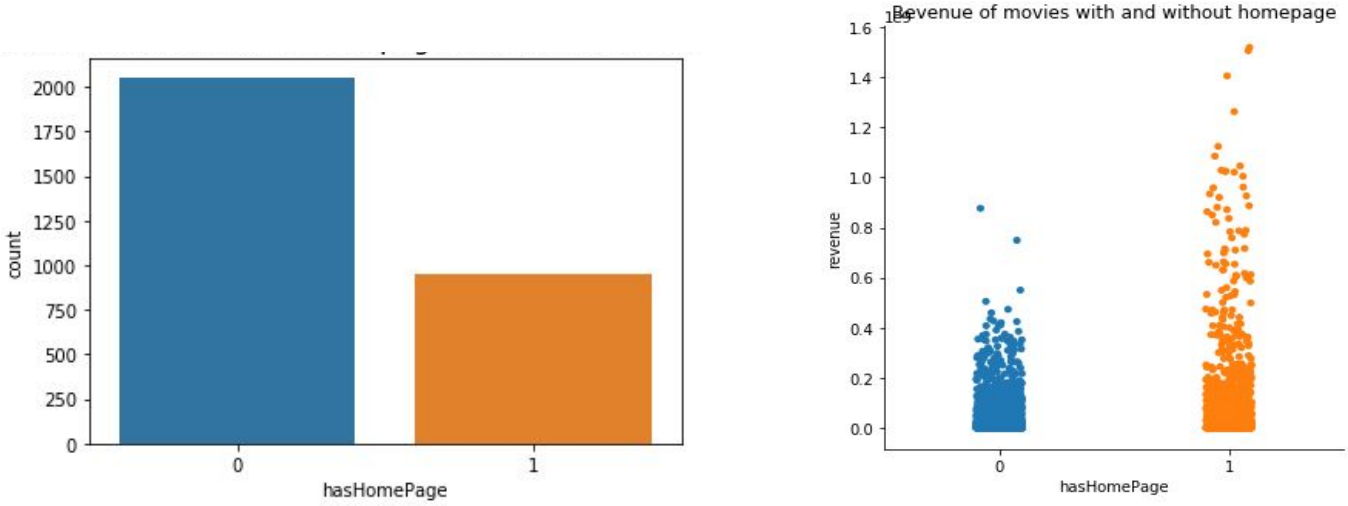
Observations: The runtime is normally distributed with the value of the revenue.

Figure 3



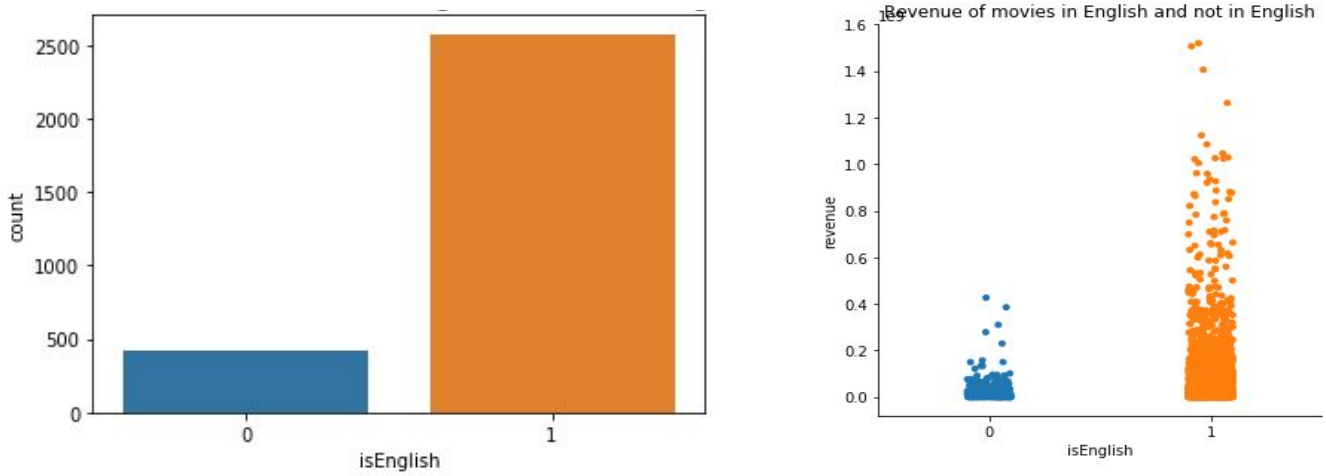
Observations: The relation between the revenue and day of release and month of release do not help us because there is not specific months or days that make high revenue than others. In contrast, the revenue increase the the release date.

Figure 4



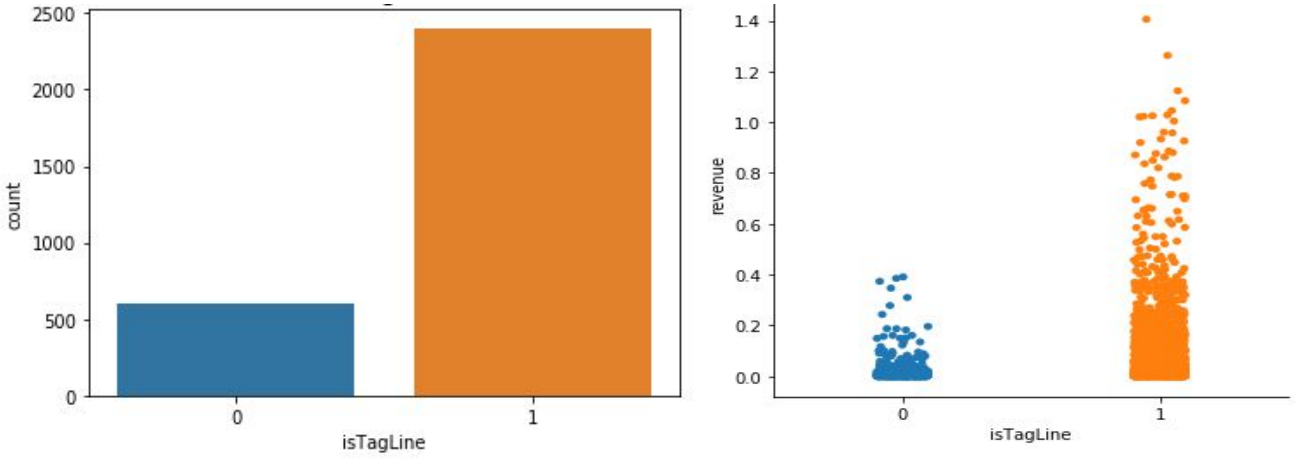
Observations: Most of the movies in our data do not have home page but the movies that have home page make higher revenue than movies that do not have home page.

Figure 5



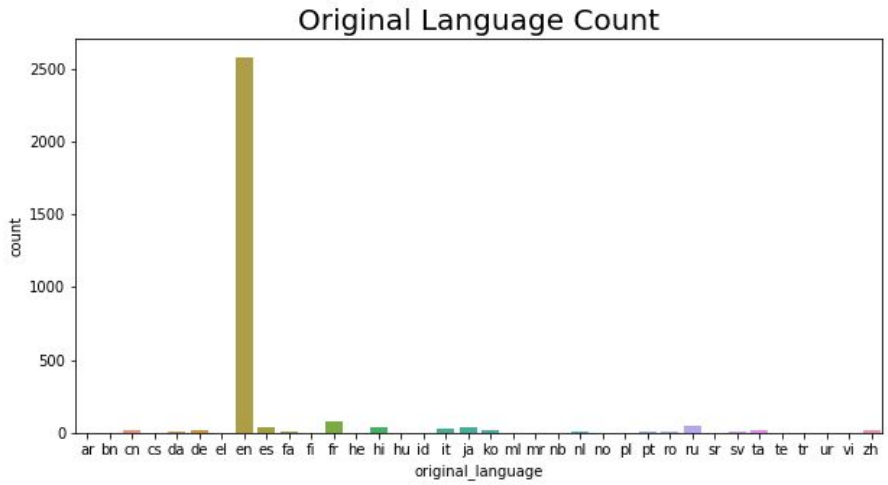
Observations: Most of the movies in our data is in English language and the movies that is english make higher revenue than movies that not in english.

Figure 6



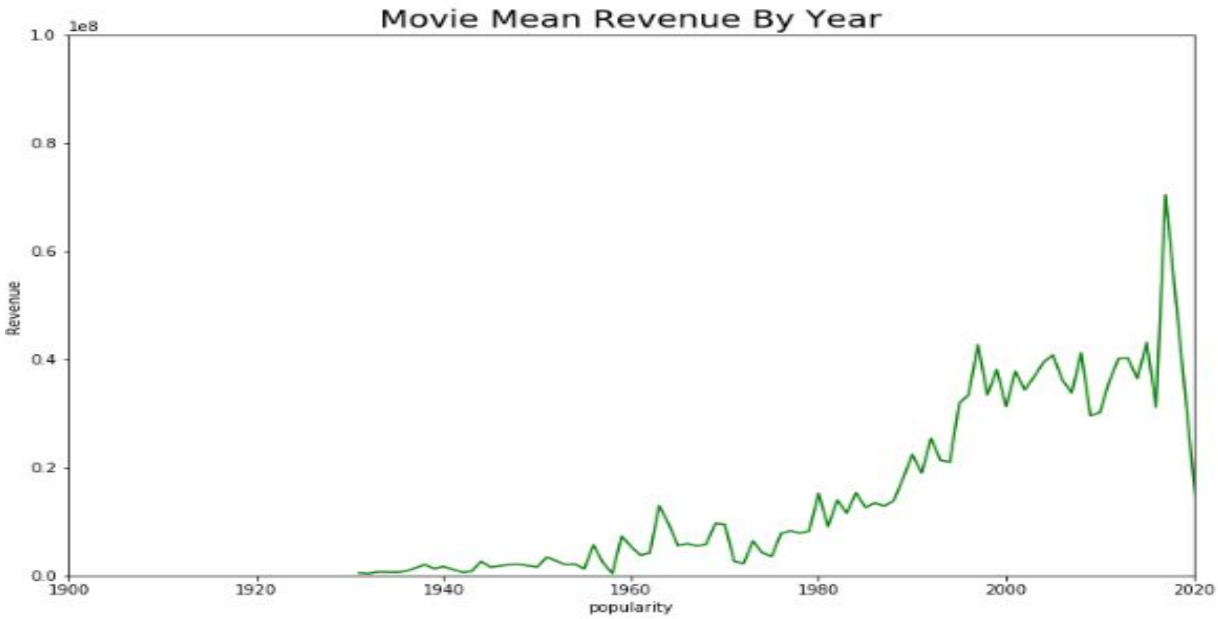
Observations: Most of the movies in our data is have tagline and the movies that have tagline make higher revenue than movies that not have tagline.

Figure 7



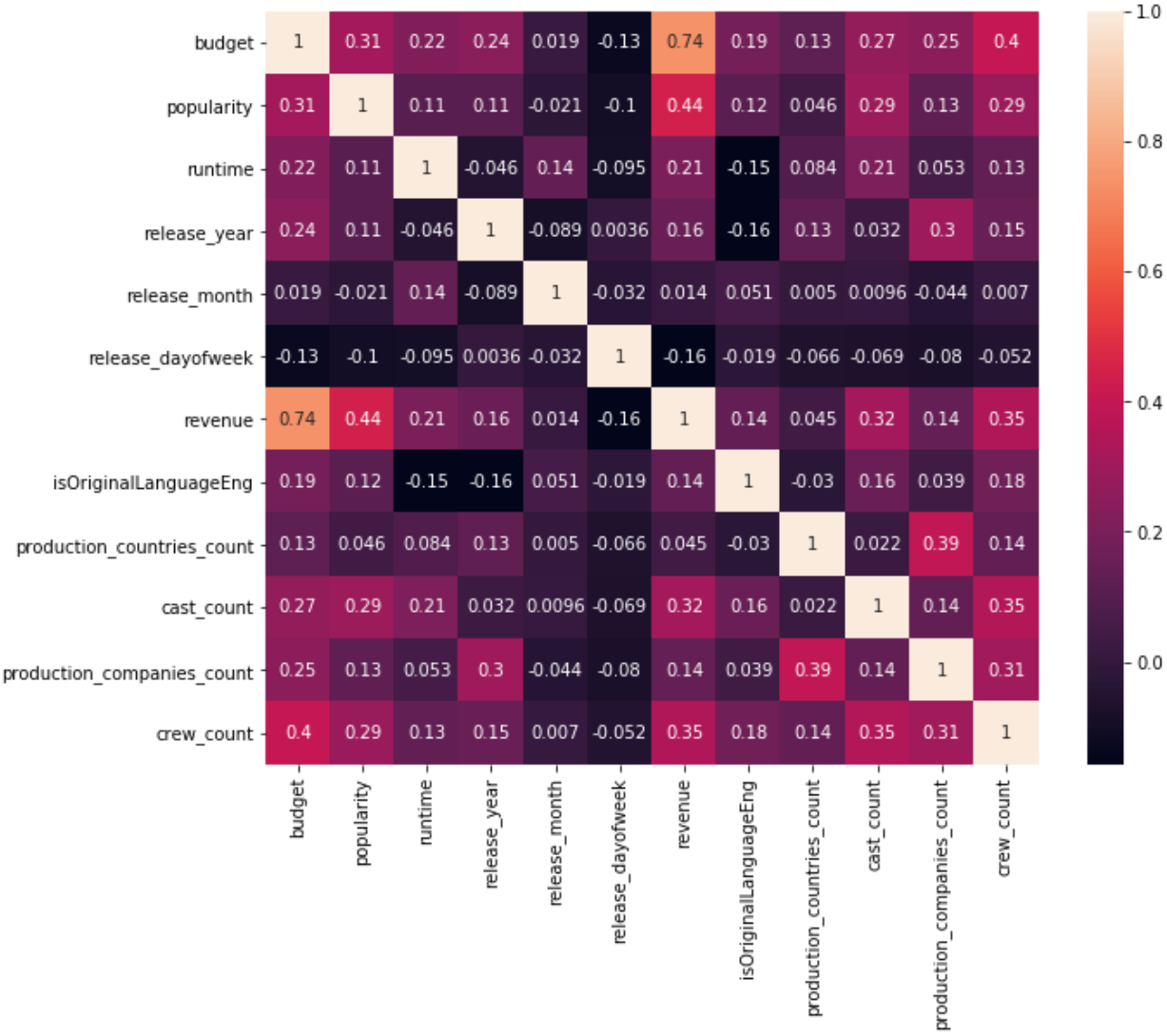
Observations: Most of the movies in our data, the english is its original spoken language, we did not use this feature because other languages are too small to train our model with them.

Figure 8



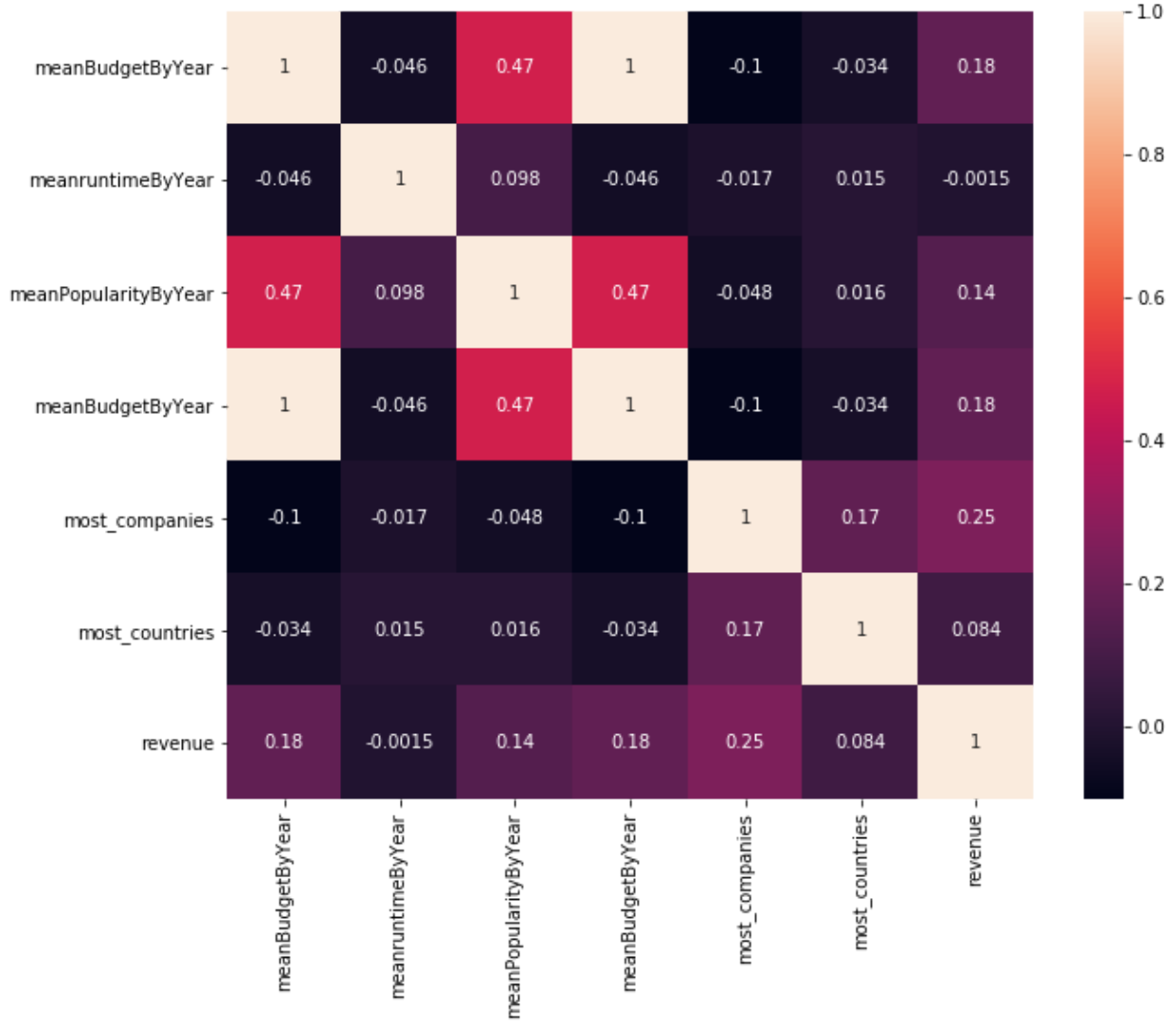
Observations: we can see that the revenue increases with the year which make sense as the value of money decrease year by year.

Figure 9



Observations: The matrix show the correlation between some variables, we can see that the correlation between (crew_count - cast_count - popularity - budget) and the revenue are the highest values so we use them in our models.

Figure 10



Observations: The matrix show the correlation between some variables, we can see that the correlation between (most_companies - meanBudgetByYear- meanPopularityByYear) and the revenue are the highest values so we use them in our models.

Figure 11

Observations: The matrix show the correlation between some variables, we can see that the correlation between (budget_runtime_ratio - budget_year_ratio) and the revenue are the highest values so we use them in our models.

b. Data Preprocessing

In this phase we concern more on cleaning, transforming json objects and try to extract new features from the original features, the following points are what we do exactly at this phase:

- Clearing nan elements from features columns after calculations.
- Clearing rows that have zero values in columns (budget , runtime).
- Reading json objects in column(production_companies) then we create the following features:
  - production_companies_count : number of production_companies participated in this movie
  - most_companies : we calculated how many times every company exist in rows then we take most occurance of companies then put in array so if film has any of most common companies will be 1 otherwise will be 0.
- Reading json objects in column(cast) then we create the following features:
  - Cast_count: we count number of people in cast in every film
- Reading json objects in column(production_countries) then we create features:
  - production_countries_count : every row contain number of production_countries for this film
  - most_companies : we calculate how many times every company occur in rows then we take most (5) occurance of countries then put in array so if film has any of most countries will be 1 other will be 0
- Reading json objects in column(Keywords_count) then we create features:
  - Keywords_count : every row contain number of Keywords_count for this film
- Split Release_date to columns(Release_day - Release_month - Release_year)
- For Release_year if (year is two digits we make it four digits (like 09 will be 2009)

c. Model Building

(a) Regression Approaches

Used Features
Popularity_mean_year
Budget_runtime_ratio
Budget_popularity_ratio
Budget_year_ratio
ReleaseYear_popularity_ratio
Cast_Count
Crew_Count
Revenue (Target)
Budget
Production_companies_count
has_homepage
_releaseYear_popularity_ratio2

(b) Classification Approaches

Used Features
Popularity_mean_year
Budget (Categorized to five groups)
Release_Year
Production_companies_count
Cast_Count
Crew_Count
Revenue (Categorized to three groups then six groups)

“Then we show the correlation between these new features and the revenue and choose most correlated features and train the model with them”

d. Model Training

(a) Regression Approaches

Used Model	Features	Results
Linear Regression	Cast_count Crew_count Popularity_mean_year budget_year_ratio	RMSE: 1.57
XGBoost Regression		RMSE: 1.75
Lightgbm Regression		RMSE :1.77
Linear Regression	Budget_runtime_ratio Crew count Popularity_mean_year Budget_year_ratio	RMSE: 1.69
XGBoost Regression		RMSE: 1.73
Lightgbm Regression		RMSE:1.90
Linear Regression	By Forward-Backward Feature Selection: Budget Popularity_mean_year Budget_year_ratio Production_companies_count Budget_runtime_ratio Has_homepage releaseYear_popularity_ratio2 cast_count	RMSE: 1.52
XGBoost Regression		RMSE : 1.59
Lightgbm Regression		RMSE : 1.74

(b) Classification Approaches

Used Model	Features	Results
SVM	Popularity_mean_year Budget (Categorized to **three** groups) Release_Year Production_companies_count Cast_Count Crew_Count Revenue (Categorized to **three** groups)	Accuracy: 72.6%
Decision Tree		Accuracy: 60.3%
SVM	Popularity_mean_year Budget (Categorized to **three** groups) Release_Year Production_companies_count Cast_Count Crew_Count Revenue (Categorized to **six** groups)	Accuracy: 56.6%
Decision Tree		Accuracy: 43.7%

7. Results and Evaluation.

a. Model accuracy on test and cross validation data.

Model Name	Accuracy / RMSE
Linear Regression by Forward-Backward Feature Selection (Final Model)	**RMSE: 1.52**
SVM	Accuracy: 72.6% (3*) Accuracy: 56.6 %(6**)
Decision Tree	Accuracy: 60.3% (3*) Accuracy: 43.7% (6**)
XGBoost	RMSE : 1.59
Lightgbm Regression	RMSE : 1.74

3* means revenue is divided into three categories
6**means revenue is divided into six categories

8. Unsuccessful trials that were not included in the final solution.

a. Classification: SVM and Decision Tree

In this model, we try to categorize some continuous values such as revenue and budget to categories like high , low , medium revenue and high , low , medium budget and split some features to be binary values such as home page to be has_home_page or not_has_home_page. In the beginning we try to use three categories only and the accuracy was:

- SVM Accuracy: 72.6% - Decision Tree Accuracy: 60.3%

Then six categories and the accuracy was:

- SVM Accuracy: 56.5% - Decision Tree Accuracy: 43.7%

b. Linear Regression

In this model, we try to predict revenue as a continuous value and we used the same features as SVM and DT and measure RSME and R Square. But also RSME is not so good.

c. XGBoost and Lightgbm

In this model, we try the features mentioned in Model Training Section but the RSME of Linear Regression using Forward Selection is smaller that XGBoost and Lightgbm.

9. Any Enhancements and future work.

Since our model accuracy is not too high we think of two enhancements which can be done in the future and may increase the model accuracy significantly.

- First is the data size, our dataset is just 17 megabyte (3000 rows) which is too small to reach high accuracy. So, we think if we can get larger data, our model accuracy will increase.
- Second is the data accuracy, out of our 3000 rows there are alot of data corrupted, for example, the budget column has 812 unknowns values. So we have to ignore these rows as budget is one of the strong features that affect the revenue.