Movie data prediction
Technical report

Table of content

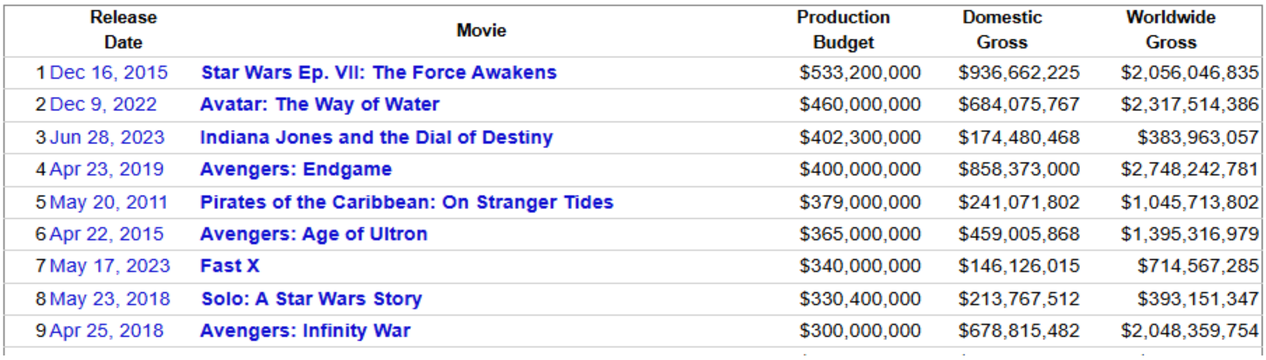
1. About the project

The idea was to gather data about movies and try to predict the profitability and success of the movies with features such as actors, directors, writers, release time, production budget, genre etc. We decided to use the US domestic box-office revenue as the measure for the profit of the movie. There were couple reasons for this decision. One was that if we would have used the gross revenue, that would include VFS, DVD sales and TV and streaming profits. That would give benefit for older movies, since they would have had more time to generate gross. Some movies might get shown in theatres multiple times during long periods of time, but we assumed, that those movies and the revenue they would generate from reshowing the movies, would be marginal. We also used only US domestic box office revenue, simply because that was available for us. So, the profit we have computed and used in this project, is not the overall profit. It still gives us a pretty good approximation of the success of the movie.

2. Data Collection

2.1 The Numbers

The Numbers (https://www.the-numbers.com/) is a database that holds production budget and box-office records for movies. This was the only page we were able to find that had production budgets readily available for our use. We asked for an API for this website, but their API was available only for payment. Luckily this website showed the data in tables with 100 entries at a time, so we were able to gather the data by copying and pasting. The website showed production budgets for 6531 movies, and we copied the information of all those movies. Like stated above, we wanted to use the box office revenue, and not the gross, so we could only use the *release date, movie name* and *production budget* from these tables.



	Release Date	Movie	Production Budget	Domestic Gross	Worldwide Gross
1	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	$533,200,000	$936,662,225	$2,056,046,835
2	Dec 9, 2022	Avatar: The Way of Water	$460,000,000	$684,075,767	$2,317,514,386
3	Jun 28, 2023	Indiana Jones and the Dial of Destiny	$402,300,000	$174,480,468	$383,963,057
4	Apr 23, 2019	Avengers: Endgame	$400,000,000	$858,373,000	$2,748,242,781
5	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$379,000,000	$241,071,802	$1,045,713,802
6	Apr 22, 2015	Avengers: Age of Ultron	$365,000,000	$459,005,868	$1,395,316,979
7	May 17, 2023	Fast X	$340,000,000	$146,126,015	$714,567,285
8	May 23, 2018	Solo: A Star Wars Story	$330,400,000	$213,767,512	$393,151,347
9	Apr 25, 2018	Avengers: Infinity War	$300,000,000	$678,815,482	$2,048,359,754

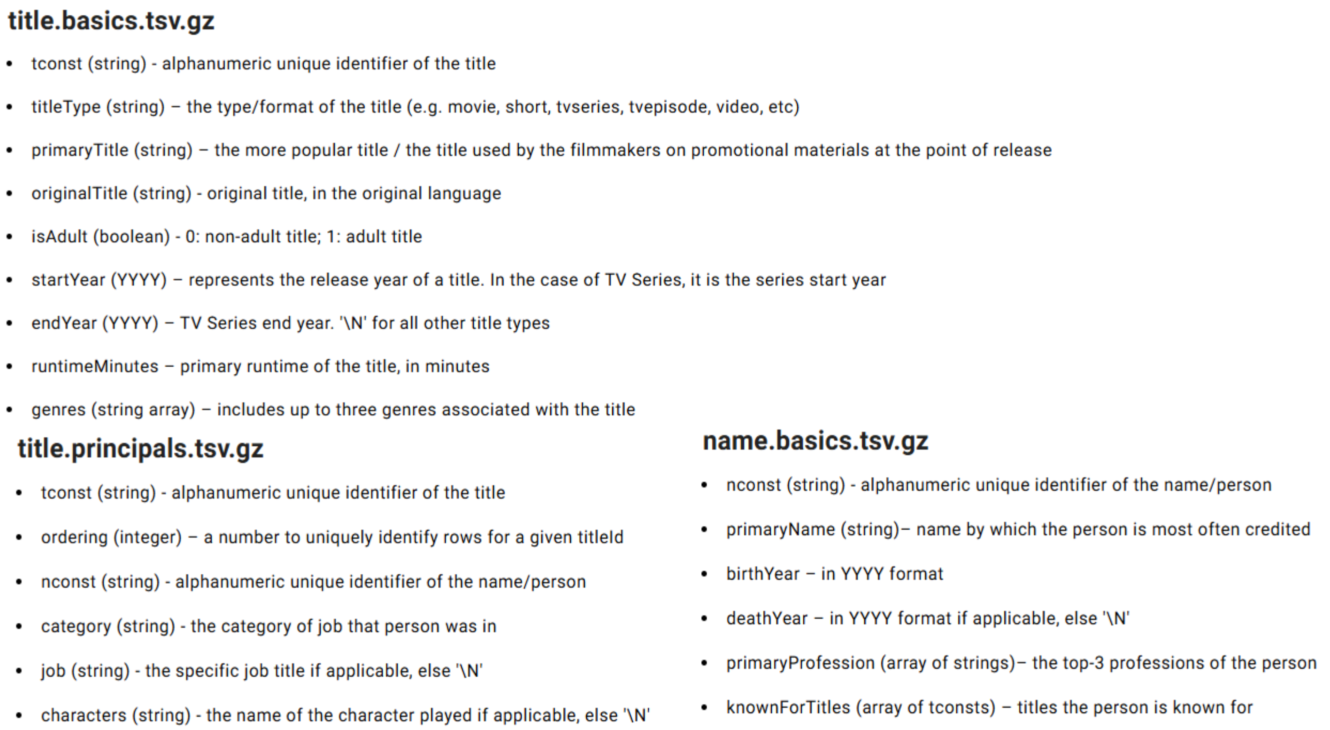
Example of data table from The Numbers.

2.2 OMDB

The open movie database (OMDB) offers a free API key with 1000 requests per day, and more for Patreon supporters. 100,000 API requests per day was only 1 EUR so we decided to take this option for easily accessing the data required for our project. The movie data could be requested using the movies' IMDB codes. We gathered information from OMDB with the API about those movie titles that were found in The Numbers dataset. More information about the gathering process of the IMDB codes and linking of datasets is included in chapter 3 of this report.

2.3 IMDB

Internet movie database (IMDB) had subsets of their data available for non-commercial use as .tsv files (https://developer.imdb.com/non-commercial-datasets/). We used three different datasets from the IMDB database.



title.basics.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) - original title, in the original language
- isAdult (boolean) - 0: non-adult title; 1: adult title
- startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) – TV Series end year. '\N' for all other title types
- runtimeMinutes – primary runtime of the title, in minutes
- genres (string array) – includes up to three genres associated with the title

title.principals.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- nconst (string) - alphanumeric unique identifier of the name/person
- category (string) - the category of job that person was in
- job (string) - the specific job title if applicable, else '\N'
- characters (string) - the name of the character played if applicable, else '\N'

name.basics.tsv.gz

- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)– name by which the person is most often credited
- birthYear – in YYYY format
- deathYear – in YYYY format if applicable, else '\N'
- primaryProfession (array of strings)– the top-3 professions of the person
- knownForTitles (array of tconsts) – titles the person is known for

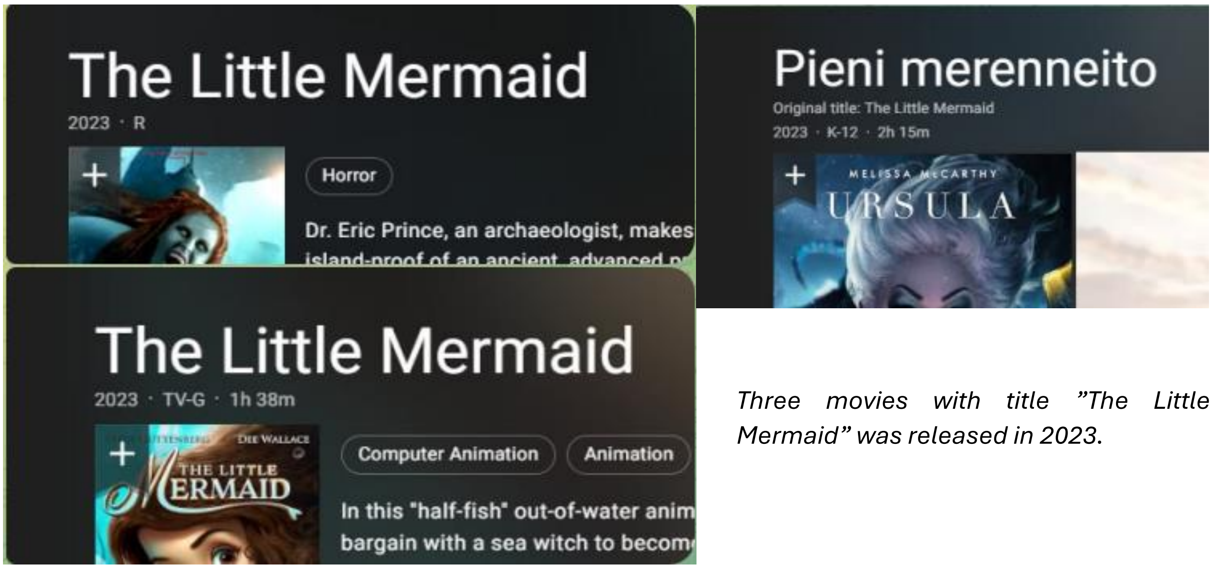
Datasets and their features from IMDB.

3. Data Wrangling/Preprocessing

3.1 Linking different datasets

The rest of the datasets had the imdb codes for each movie title, so we could use those to link the different datasets together, but the data from The Numbers didn't include the IMDB codes. For the start, we had to discard 190 titles out of the 6500 titles, because they had duplicated title name and release year, and we didn't have any means to link these to the right IMDB codes. For example, there were three different movies with the title "The Little Mermaid" released in 2023.

We used the title name to link the production budgets to the IMDB data. We also included the release year to the match, so movies with same titles but from different years wouldn't mix. We found a match for about half of the titles. The rest of the titles had different spelling. For example *Star Wars: Episode VII - The Force Awakens* in the IMDB data vs *Star Wars Ep. VII: The Force Awakens* in The Numbers data. We used SequenceMatcher from the difflib in Python with threshold 0.8 to match as many more titles as we could. In the end, we were able to match total of 5859 movie production budget to their IMDB codes. Rest of the linking between datasets was done with the IMDB code.


The Little Mermaid
2023 · R
+
Horror
Dr. Eric Prince, an archaeologist, makes
island-proof of an ancient, advanced
Pieni merenneito
Original title: The Little Mermaid
2023 · K-12 · 2h 15m
+
MELISSA McCARTHY
URSULA
The Little Mermaid
2023 · TV-G · 1h 38m
+
THE LITTLE MERMAID
Computer Animation
Animation
In this "half-fish" out-of-water anim
bargain with a sea witch to becom

Three movies with title "The Little Mermaid" was released in 2023.

3.2 Adjustment for inflation

To be able to compare the production budgets from different years. We computed inflation adjustments for all production budgets and box office revenues.

3.3 Average profits, average production budgets and number of films

For all actors, directors and writers, we computed the number of the movies they have been involved in and the average profit and average production budget of those movies. We then computed for every movie title the average of the average profits of the actors that were involved in that movie. This was also done for the average production budgets and number of films. This was repeated for the directors and writers as well. These averages were used in the prediction models to evaluate the crew, their experience and their effect on the success of the movie.

4. Methods used to process data

4.1 Visualization

For basic EDA we did simple visualizations. Scatterplots and histograms to see how different variables relate to each other and how the variables were distributed. Based on this we picked the most interesting relations between the variables to use in the final model and chose the most interesting findings to display on the website. We mainly used the matplotlib and seaborn libraries to make plots and histograms with colour coding.

4.2 Vectorization of keywords

Already in the beginning of the project we decided that we want to find out what keywords and tags are most related to the successful movies. We ended up using TF-IDF method to try to specify words in different documents, the documents being classes of movies with different success. We decided to use the profit ratio as the metric for success. So, we decided to divide the movies into six different categories based on their profit ratio. Original plan was to use the tags on the movies as well as the synopses, but the tags would have had to be scraped from web. We had hopes on using the tags from MovieLens research dataset but after combining the movies from that dataset with the movies we had the production budgets for, it would have reduced the final size too much. Eventually we decided to just use the synopses.

What we did was cluster the synopses together in each group, lowercased the text, removed the punctuation and stopwords and used the nltk library to stem the words. After that we could use the documents for the word vectorization. We used the TfidfVectorizer from the sklearn library for the method. The result was not impressive though, all the six classes of movies ended up having almost the same words as the most characterizing words even between the most profitable and least profitable classes. There were a few distinct words between the classes, so we ended up displaying the 30 most profitable and least profitable words on the website. The significance for the word vectorization in the overall project ended up being much less than we anticipated, however.

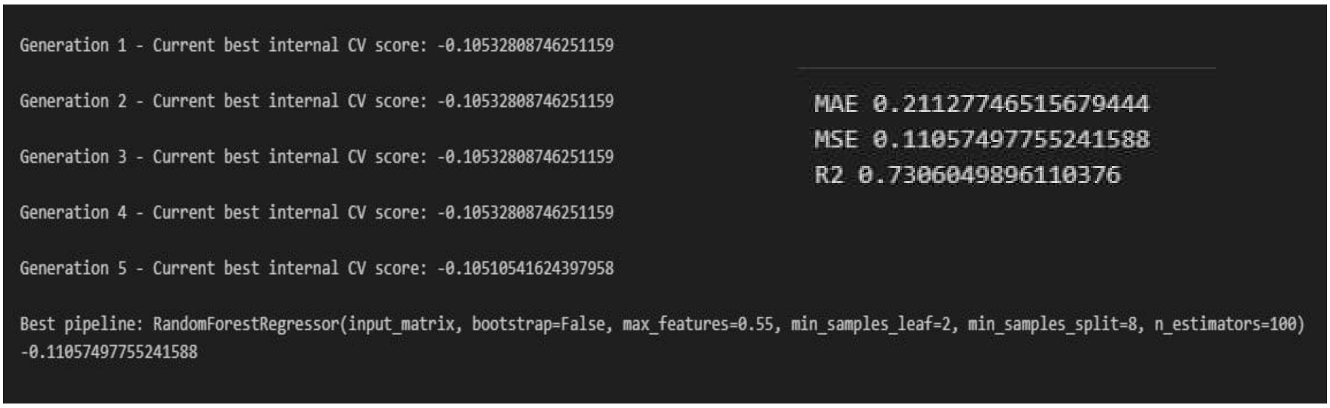
4.3 Clustering of actors, directors and writers

We used average production budgets, average profits and number of films of actors, directors and writers to place them in clusters using KMeans method from scikit-learn library. Before the clustering, we used StandardScaler to standardize the features. We computed the inertia of different number of clusters and used that to evaluate the optimal number of clusters.

4.4 Prediction model

We used the TPOTClassifier and TPOTRegressor to find optimal prediction models for our data. Features that were included in the data for the prediction models was inflation adjusted production budget, runtime, release month, genre (each title had 1 to 3 different genres selected), and the average production budgets, average profits and average number of films of the actors, directors and writers. For the prediction models, after removing all rows with missing cells, we had a dataset of 4919 movie titles. We tried the TPOTRegressor to find model to predict the inflation adjusted box office revenue, profit (box office / production budget) and the logarithm of profit. Best regression model was a random forest regressor that predicted the logarithm of the profit. The model had a R2 score of 0.73 and mean squared error of 0.11.

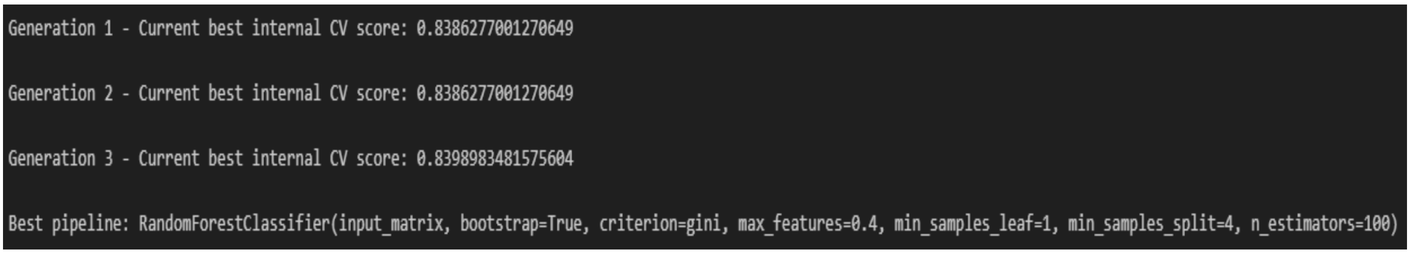
We also used the TPOTClassifier to find best model to predict if the movie would make profit or not (profit<1 or profit>1). The best model was a random forest classifier with accuracy of approximately 0.84. Although the R2 score was ok for the regressor model, we welt that the MSE was too big for any meaningful predictions (notice that it was for the log of profit). We felt that the accuracy of the classifier was decent enough, so we decided to use that as our predictive model.



```
Generation 1 - Current best internal CV score: -0.10532808746251159

Generation 2 - Current best internal CV score: -0.10532808746251159

Generation 3 - Current best internal CV score: -0.10532808746251159

Generation 4 - Current best internal CV score: -0.10532808746251159

Generation 5 - Current best internal CV score: -0.10510541624397958

Best pipeline: RandomForestRegressor(input_matrix, bootstrap=False, max_features=0.55, min_samples_leaf=2, min_samples_split=8, n_estimators=100)
-0.11057497755241588
```

```
MAE 0.21127746515679444
MSE 0.11057497755241588
R2 0.7306049896110376
```

Best regressor model from TPOT.



```
Generation 1 - Current best internal CV score: 0.8386277001270649

Generation 2 - Current best internal CV score: 0.8386277001270649

Generation 3 - Current best internal CV score: 0.8398983481575604

Best pipeline: RandomForestClassifier(input_matrix, bootstrap=True, criterion=gini, max_features=0.4, min_samples_leaf=1, min_samples_split=4, n_estimators=100)
```

Best classifier model from TPOT.

5. Web application

To both present our findings and to demo the prediction model we set up, we created a simple web application using Python's Flask library. It includes plots we want to showcase, an interactive plot generator based on Python's Matplotlib library for plots that were not included in the showcase, and lists of the vectorized keywords mentioned in the previous chapter. A tool to predict whether a movie with certain parameters (directors, writers, actors, genre, budget, runtime and release month) could turn in a profit was also included and utilized the aforementioned prediction model. We used ngrok (https://ngrok.com/) to launch the web application online.

6. Added value and conclusions

Making movies is a form of art. Making movies is also a hard business. In the current capitalistic worldview, we must accept the fact that movies must make profit. We believe that we succeeded in our task to create a model for predicting the movie profit. In the beginning the idea was to predict how much profit the movie would produce but the timeline and data restrictions allowed us to create a binary model that will predict if your movie brings in profit or not. We believe our model still succeeded in the primary goal: added value to movie makers, whether it would be movie studios, producers or directors. The model succeeds to give easily interpretable results from clear input values, such as budget, actors, director and so on. It became an easy-to-use decision-making tool. We believe it makes good predictions for producing more profitable movies or in general movies, that the consumers would like to watch more.

We had hoped that we get more information or value from the keywords and word vectorization but due to the timeframe and data restrictions we ended up not getting much out of that approach. We had hoped that we could extract relevant actors, genres and tags to characterize successful movies. That ended up not happening in this project.

Future developments in this project would include deeper text classification/word vectorization exploration to characterize successful movies better. It would also be interesting addition to analyse the advertising side of movies. It is clearly a variable that associates with making a profit from a movie.