**Data Analysis 094295 – H.W. 1**

Alexander Gofman 312884323 Tal Kaspani 204528673

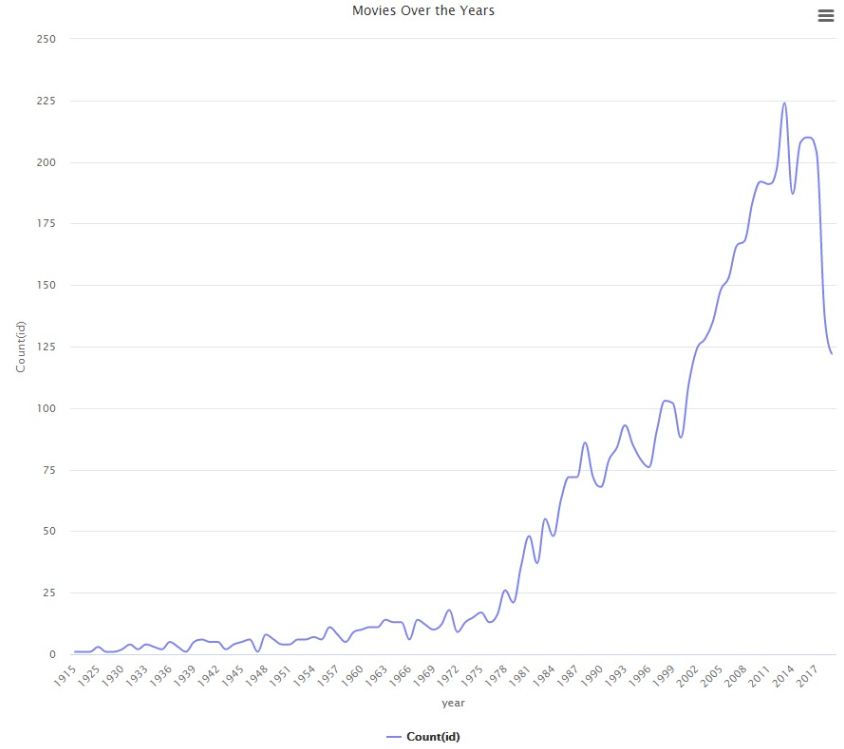


1. **Exploratory Data Analysis**
   1. Features

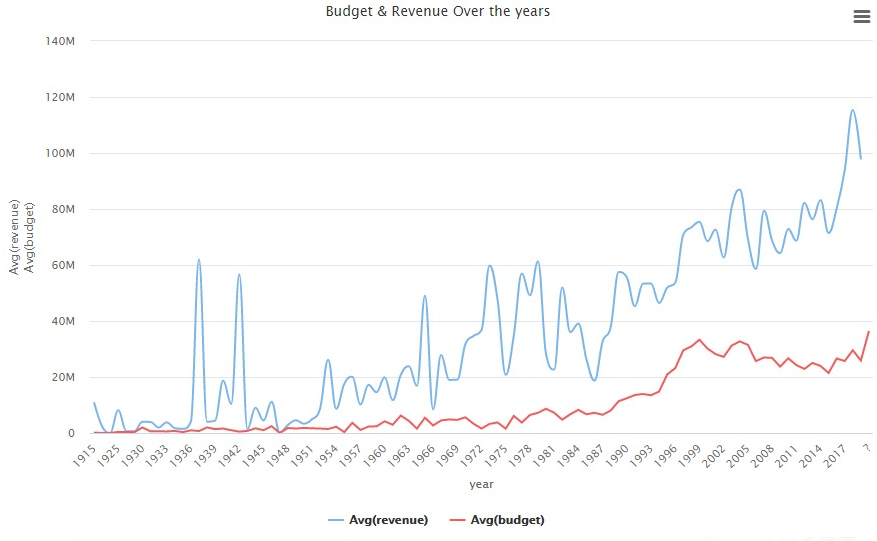
We will go over the features, divided by data type – most feature names are self-explanatory, and for those who are not we included a short explanation of their function in parenthesis. Video and status features both had a single value for the entire dataset so we omitted them.

* + 1. ID’s – id and imdb id.
    2. Numerical features – revenue (amount of money gained from the movie, in dollars), budget (in dollars), vote count, vote average (average score of movie), popularity (amount of votes), runtime (amount of time in theaters in days).
    3. Textual / Categorial – original language (two letters representing the language), original title, overview (textual, short paragraph), tagline and title.
    4. Links – backdrop path and poster path (picture links) and homepage (website link),
    5. Lists – production companies (id, logo path, name and origin country), production countries (shortened 2 letters name, name), spoken languages (shortened 2 letters name, name), keywords (id, name), cast (id, character, gender, name, order, path to profile picture), crew (id, department, gender, name, job, name, path to profile picture), belong to collection (id, name, poster path, backdrop path).
  1. Data Exploration

First, we can see the number of movies in the dataset increases very rapidly over the years – which means the dataset is unbalanced and most of it is in recent years.

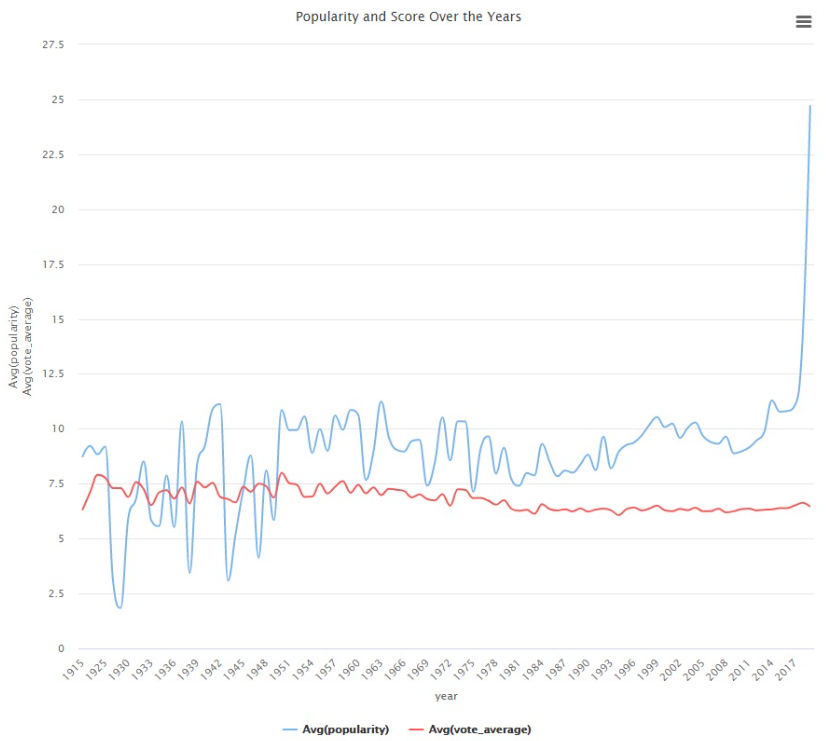


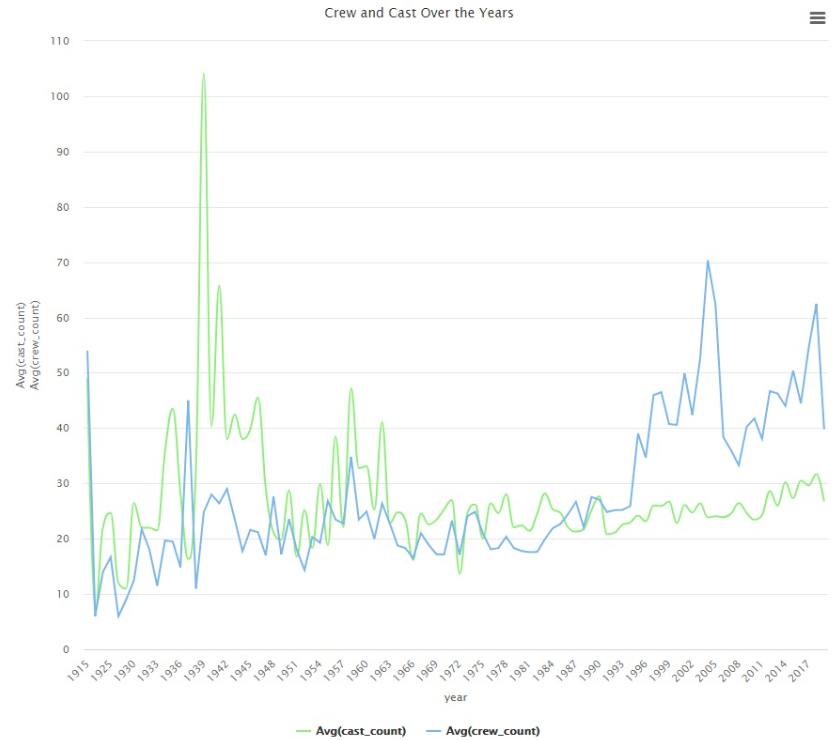
Checking the budget and revenue over the years, we saw that the revenue to budget ratio increases over the years. While the growth in the budget is solid, the revenue fluctuates a lot and is generally less stable.



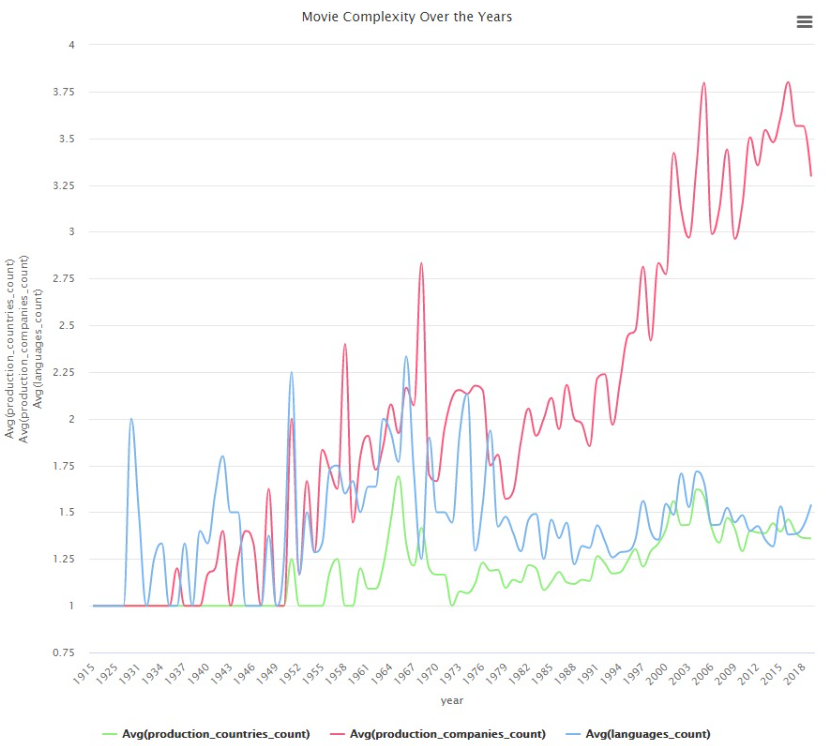
We noticed a few weird points in the early years – which we identified as movie that originated in the early years and were re-introduced in later years. We removed these examples as to not distract our classifier.

While the score average remains constant over the years, we can see the popularity is not related to it (unexpected result in our eyes).



We can see that there was a change over the years – around the 1980’s there was a massive increment in the amount of crew members, which until that point was similar to the amount of cast members. 

We can also see that the production of movies got relatively more complex – while the amount of languages spoken in a movie is relatively stable, the amount of countries and companies involved increased over the years. This supports the increase in budgets we saw in the beginning.



We can safely conclude that movies from different eras ‘behave’ differently – which we will address later in this document.

* 1. Missing Data

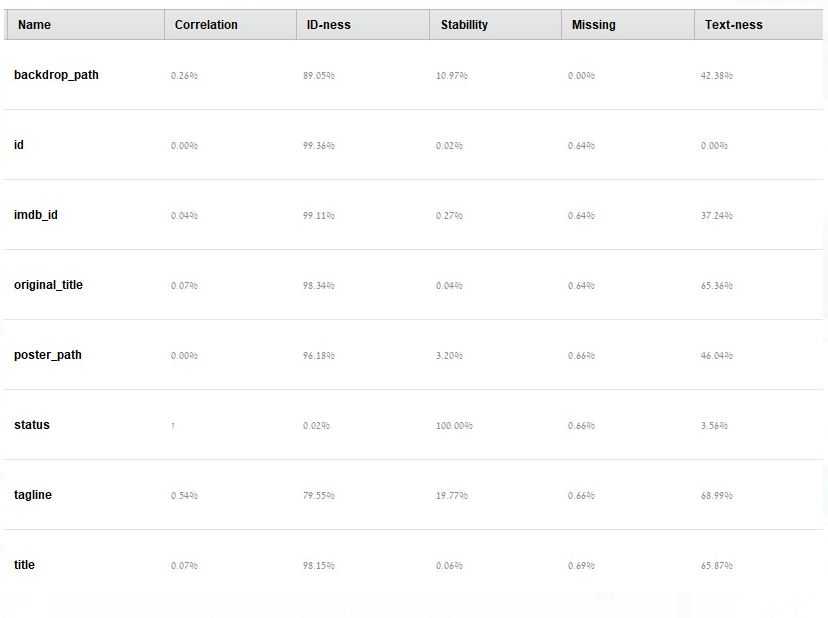
The only important attribute that we noticed was missing was the budget from some of the movies.

* 1. Data removal

From the entire dataset of 5000+ movies we decided to remove 93 movies which presents revenue values less than 1000 USD , which from our EDA process we assumed were outliers.

1. **Feature Engineering**
   1. Feature Selection

We used RapidMiner to help us with some of the decisions.



We did not use any of the above features – all of which were very id-like (many different single values), and status had a single value for the entire dataset.

We thought the rest of the features were important, but to use them we had to extract the information in them. We decided not to get into NLP techniques in order to extract information from the textual features, after a lot of thought on the matter – we thought it would be better to put our efforts into the other features.

* 1. Feature Transformation

The first thing we did was to extract information from all the lists – we created a feature that counted the items for each of the lists. Since we wanted to keep the information inside the lists but we didn’t want to have too many features, we used one-hot-encoding only on the most common items from each list (between 5%-10% most common items in each list).

Since we saw the release year was relevant to most features, we then created some statistical features from the numeric features – we created a median and an average for budget based on year of release and counted the amount of movies in the train set released each year.

We also created the collection size for each movie (0 for movies without a collection).

* 1. Imputation

We wanted to impute the missing budget. We went with a straightforward approach – since we saw that the budget is highly correlated to the year the movie was released in, we calculated the mean budget for every year based on our training set. We gave each record with a missing budget the calculated mean budget for year of release. If we did not have that particular year in our training set, we used the average budget of that decade instead.

* 1. Data Enrichment

In order to enrich our dataset, we tried to think where we can find a list of good movies and we thought of two places – the Oscars and the IMDB.

Oscasrs:

<https://datahub.io/rufuspollock/oscars-nominees-and-winners#data>

From the Oscars dataset we found, we extracted many new features. First thing we did was finding the relevant categories (those with a clearly defined winner). We then created lists of the winning actors, nominated actors, winning movies, and nominated movies – each list had the total number of wins/nominations in all categories for each entity (movie or actor). We used these lists to create 4 features describing the total number of movie/actor wins/nominations for each movie, counting the entire cast’s wins or nominations.

IMDB - "Fame score":

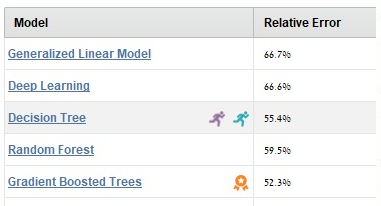
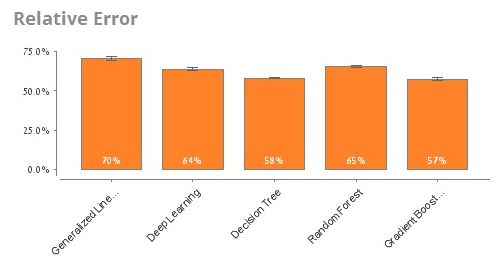
Top 1000 actors - <https://www.imdb.com/list/ls058011111/>

Top 1000 directors - <https://www.imdb.com/list/ls066140407/>

We decided to use the lists above to add a variation of fame/prestige score for every movie in our dataset. We did so by summing the total actors' positions on the "Top 1000 actors" list from the movie cast and summing the total crews' positions on the "Top 1000 directors" list from the movie crew. It is worth mentioning that the highest score for any given actor/director is limited by 1000 – the highest position possible.

1. **Prediction**
   1. Algorithms

We performed a quick test on the dataset with RapidMiner and tried a few different simple regressors: linear regression, a simple neural network, decision tree, random forest, and gradient boosted trees.



We got the best result from GBT, so we decided to test GBT along with an appropriate ensemble model (XGBoost) to see which has the best results.

* 1. Hyperparameters and Results

Since both algorithms were quick on our test set, we could test a large number of hyperparameters and choose the best result.

For GBT:

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Options | Best performance |
| Number of estimators | 500, 750 | 500 |
| Max depth | 10, 30 | 10 |
| Learning rate | 0.03, 0.05 | 0.03 |
| Min samples in a split | 2, 5 | 2 |

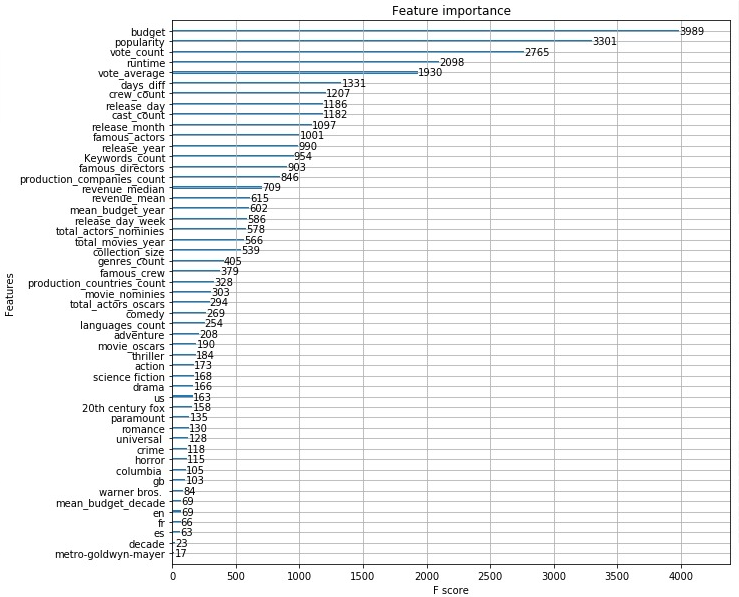
For XGBoost:

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Options | Best performance |
| Number of estimators | 100, 250, 500, 750 | 750 |
| Max depth | 7,10,12, 20, 30, 50 | 7 |
| Learning rate | 0.01,0.03, 0.05, 0.1 | 0.03 |
| Columns sampled for each tree | 0.1, 0.3, 0.5, 0.7, 0.9 | 0.9 |
| Alpha | 10, 50, 100 | 10 |

The best RMSLE score over the test set for GBT was: **2.313** and for XGBoost was: **2.273** so we settled on XGBoost.

* 1. Results Analysis

Feature Importance (best scoring model):



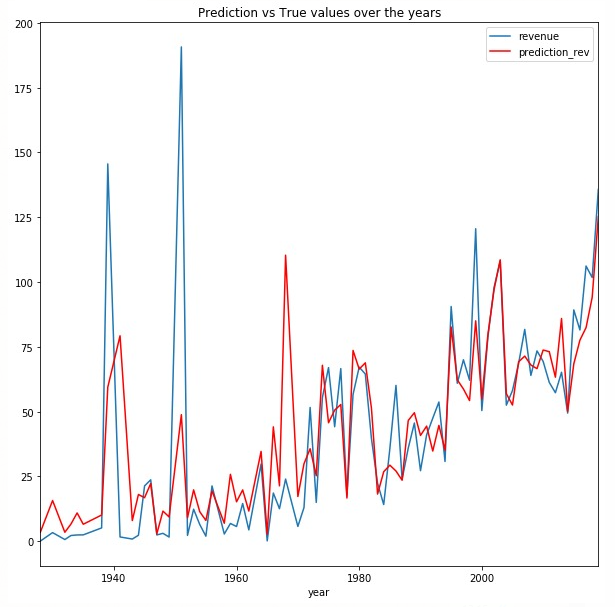
From the figure above we can conclude that the prediction is mostly determined by a small set of features including budget, votes and popularity.

Moreover, we can spot that the most useful features we added to the data were:

* Days\_diff – number of days between the movie release date to today.
* Crew\_count – number of crew members for a given movie.
* Cast\_count – number of total actors for a given movie.
* Famous\_actors – combined "fame score" of all actors in a movie.
* Famous\_directors – combined "fame score" of all crew members in a movie.

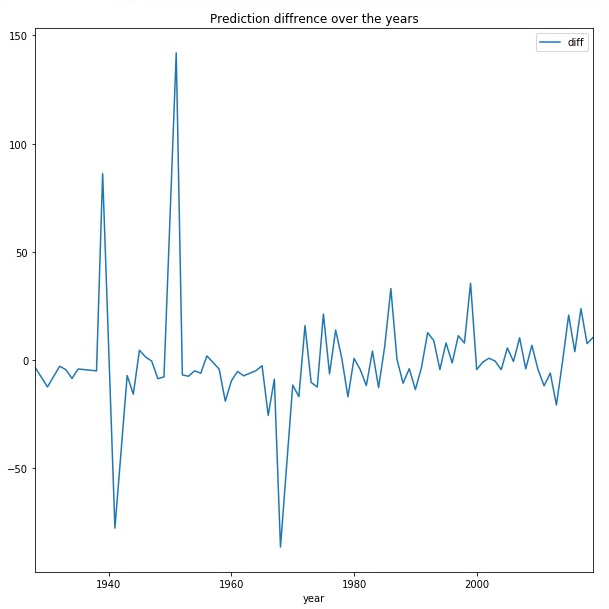
Prediction error analysis:

We compared our predictions on the test set to the actual values over the years:



We can see that our results are pretty good, marred by some outliers.

Since we saw that the values vary greatly based on the release year, we wanted to see if our model scales well with the change of years, so we measured the total error for each release year.



Same as our last comparison – many of our mistakes comes from old movies.

We assume these mistakes are the result of the data for movies from early years (before 1970) being scarce and features many deviations and extreme values in terms of budgets and revenue.