Rotten Tomatoes Review Discrepancy

Benjamin Raible
Matrikelnummer 6413200
b.raible@student.uni-tuebingen.de

Daniel Kerezsy
Matrikelnummer 6300575
daniel.kerezsy@student.uni-tuebingen.de

Andreas Kotzur
Matrikelnummer 6479284
andreas.kotzur@student.uni-tuebingen.de

Abstract

This project investigates the discrepancies between official critics' and the audience's ratings for movies on Rotten Tomatoes. We want to find out if there are any regularities and if these are enough to predict the scores (or their discrepancy/divergence). This project covers the data gathering and the evaluation with Null-Hypothesis Tests, Linear Regression, Decision Trees and MLPs. Results show that there is not enough information to infer any significant results.

1 Introduction

There are a lot of websites rating the quality/perception of movies and TV series. One of the most popular ones is the US-American website https://www.rottentomatoes.com [1] which allows officially registered critics and the general audience to rate a movie resulting in a score between 0% and 100% for both voter categories. It is a relatively well known phenomenon that the opinions of these two parties often diverge and sometimes even by a huge amount. This observation can be made every so often and has thus obviously led to many discussions [2][4][5].
We want to know if there are any commonalities between the movies where ratings diverge and thus need a database where all this information is combined. We gather information about movies from the 1950s up to 2022 using Wikipedia lists and Rotten Tomatoes. We did not use IMDb since for us relevant data like the 'budget' or the 'box office' is hidden behind a paywall (IMDb Pro).

2 Creating the Database

Since we could not find a compact and accessible database containing the data we need, the first logical step was to create our own. This requires gathering data at first, repairing as much as possible and finally filtering unusable or wrong entries.
We use Python 3.10, Scrapy and Pandas for this process.

2.1 Collecting the data

We start by initially crawling through Wikipedia collecting URLs to entries of a specific movie and their corresponding year of release. These movies are listed under sublists, containing all movies worldwide released in that year. Note that the year under which the movie is listed does not necessarily

represent the actual time of release.
In a second crawler, this URL will then be used to gather information such as the name of the movie and the further more detailed information.
These Wikipedia entries won't contain a link to their corresponding entry on Rotten Tomatoes. To acquire this entry without needing several steps (search, compare, filter) to eventually get to the corresponding entry, we used an heuristic approach.
We found out that Rotten Tomatoes filters their URL entries before redirecting to the appropriate URL and resource. Using this information, we took the name of the movie, we could simply append the name of the movie to the URL. This works pretty well but is susceptible to remakes or different movies with the same name, e.g. Batman (1943), Batman (1966) and Batman (1989)[3]. Again we use heuristics to acquire the correct entry by simply appending the year to the query as in "<movie_name>_<yyyy>". If we don't get a result we try querying the film title without this appendix. Both data entries from Rotten Tomatoes (if any) and Wikipedia are then combined in our unrefined database. This way we acquired 19366 entries.

2.2 Cleaning the data

In order to verify/clean our data from the applied heuristic, we extracted the release date gathered from Wikipedia and compared it with the date gathered from Rotten Tomatoes. We assume that a movie with the same name would not be released in the same year and thus removed all entries where the two different dates diverged resulting in a loss of 222 entries.
One of our main goals for the data extraction was to gather information about the Box office (revenue) and the budget. Since our list contains movies from all around the world beginning from the 1950s, we have a lot of different currencies subjected to different inflation rates within our database. Furthermore, some of the Wikipedia entries already contain the converted currencies inclusive the inflation rate. In order to have a common ground, we convert all entries to US-Dollar with the inflation being accounted up to December 2022. We consider 15 different currencies, the release date and the potentially already converted values plus their corresponding date.
To calculate the inflation rate we use the average Consumer Price Index (CPI) for all items obtained from a database by the International Monetary Fond (IMF). Using a dataset containing 52 different currencies which is also based on IMF data, we can convert all currencies into US-Dollar.

3 Processing the data

3.1 Hypothesis tests

First we decided to perform hypothesis tests to see, if our parameters influence our targets. Audience and critics score on rotten tomatoes measure the percentage of people who submitted a positive review. It makes sense to measure the distance between this scores. This extends to their discrepancy. Therefore our target is metric. This now allows us to find the correct hypothesis tests.

Parameters with multiple groups For parameters that divide the data into multiple groups the corresponding hypothesis test is a simple analysis of variance combined with a post hoc test. In our case such parameters are the genre, streaming supplier and the release month.
For our simple analysis of variance the two hypothesises are as follows:

H0: The mean scores of all groups are equal.

H1: At least one of the mean scores of the groups differs.

We set our significance level to 0.05.

In figure (1) we see that for all three we reject the zero hypothesis for each of these three scores.
To further analyse between the means of which groups we see a significant difference in there mean, we use post hoc tests. The post hoc test uses a student t-test to test the differences between each pair of groups. The hypothesises for the student t test are as follows:

H0: The mean scores of the two groups are equal.

H1: The mean scores of the two groups are different.

	genre	suppliers	month
audience_score	0.0e+00	2.4e-27	2.3e-14
critics_score	2.1e-259	5.1e-13	2.0e-32
divergence_score	1.7e-77	9.9e-04	9.1e-21

Figure 1: p-values of a simple anova

	genre	suppliers	month
audience_score	84.74%	76.19%	50.00%
critics_score	77.89%	57.14%	62.12%
divergence_score	62.11%	38.10%	54.55%

Figure 2: percentage of the pairs of groups with significant deviation in mean messured by a student-t test

For a significance level of 0.05 we get that the following percentages of the groups reject the zero hypothesis (3). From this we can assume that genres, suppliers and release month offer use useful information for predicting our 3 target scores.

Metric parameters For metric parameters correlation is usually a good score to look at, when deciding whether a parameter contains useful information for our prediction or not. We used the pearsonr test, that has zero correlation as its zero hypothesis and a no-zero correlation as its alternative hypothesis which returns p-values as depicted in figure (3).

For year and film-length we reject the zero hypothesis for all 3 target scores. For the other parameter we only have enough evidence to reject the zero hypothesis for some of them. Since all of the parameters have a significant correlation to at least some of the target scores, we will include them in out training.

3.2 Regression

As stated in the abstract, our goal was to predict the critics' score, the audience's score or the difference between the former (i.e. divergence). We used several different methods to transform our input parameters with significance to one of the three possible targets. All of the tested methods resulted in a bad fit.

LinearRegression We tried fitting the data with a simple linear regression model from the sklearn library. This generally resulted in the model predicting the mean of the targets for the training samples, whereas the Audience Model was the only one which could apply this to the test samples as well. Both Divergence and especially the Critics Model performed conclusively worse (1).

	Training score	Test score	Training MAE	Test MAE
Critics Model	0.057	-945.89	22.83	39.68
Audience Model	0.08	0.077	17.15	17.18
Divergence Model	0.01	-6.88	16.39	17.25

Table 1: Linear Regression Scores and Mean Absolute Errors (MAE)

CatBoost CatBoost works on the theory of decision trees. It uses many weak models sequentially, to create a strong model. To achieve our target we used the Regressor (`CatBoostRegressor`) model from the CatBoost library. We see that the model suffers from overfitting (4). We tried to combat this be introducing L2-Regularizers, randomness in the choices on the decision-tree of CatBoost and adjusting the learning rate. Neither of them helped.

Neural Networks with PyTorch Using PyTorch we created a model with three Hidden Layers (hidden width = 2 times input width) using Tanh, Leaky ReLU and ReLU as activation functions. It

	year	length	budget	box_office	proportional_profit
audience_score	3.3e-06	1.2e-257	2.4e-05	3.0e-01	3.6e-08
critics_score	2.1e-48	2.2e-53	7.7e-02	4.4e-01	1.3e-01
divergence_score	3.6e-10	6.0e-14	4.7e-01	3.3e-02	2.4e-02

Figure 3: p-values of the pearsonr test

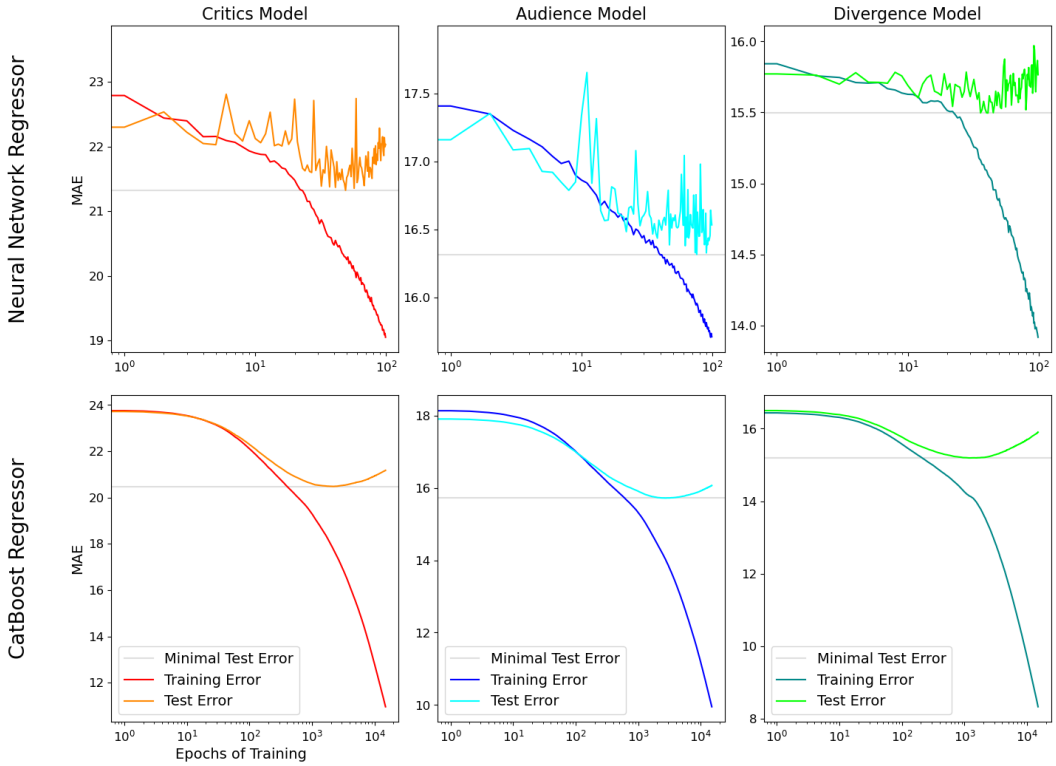

Figure 4: Mean Absolute Errors during training of Neural Network- and CatBoost Regressors

was not possible to use a higher Batch size since then our model would only generalize to predict the mean (similiar to linear Regression) thus the results depicted in figure (4) are created not using batches at all. But similarly to the CatBoost Regressor, only with slightly higher mean errors and less iterations, we achieve overfitting of the training samples and slight reduction in the test error.

Conclusion We conclude that probably our parameters are not strong enough to predict either the score or the divergence. Our relatively low quantity of data (8000 training samples) is most likely too sparse for our objective and thus the algorithm only overfits without any significant generalization. But we could also observe that the predictions for the audience's score were more accurate, indicating that the audience score is possibly most influenced by the parameters we chose. There are more parameters such as production company, gender ratio of cast and producers that could be included already in our gathered data. Many more can be found by scraping more sources of data. The number of samples is way harder to increase, because the number of films released with scores on rotten tomatoes is limited.

References

[1] Fandango. About rotten tomatoes®. https://www.rottentomatoes.com/about.

[2] H. G. Noolan Moore. The biggest divides between audience and critic scores on rotten tomatoes. https://www.looper.com/136024/the-biggest-divides-between-audience-and-critic-scores-on-rotten-tomatoes/.

[3] Reddit. Batman in film. https://en.wikipedia.org/wiki/Batman_in_film.

[4] Reddit. Critics vs audience. https://www.reddit.com/r/ExpectationVsReality/comments/d042sa/critics_vs_audience/.

[5] Reddit. Rotten tomatoes: Critic vs. audience score [oc]. https://www.reddit.com/r/dataisbeautiful/comments/o1xw4w/rotten_tomatoes_critic_vs_audience_score_oc/.