## Is a rating worth a thousand words?

## Introduction:

The success of a box-office movie is a complex thing to predict, and of great interest for the investors in the movie industry. However, the box office revenues of a movie is hard to predict, and seem to predict on many interacting factors like global economic trends, characteristics of the movie and the general public’s reception [1]. The movie market is very volatile and creating a good model for predicting movie success is therefore important to the movie market [2].

Large databases of movie information are stored online, where Internet Movie Database (IMDb) is the most prominent. This database stores box office success, information about the movies characteristics, and serves as a platform for consumers to rate and review movies in their own natural language. Natural language is highly dense in information about a movie that numerical genre identifications and ratings might not completely capture, however it has also been historically computationally harder to extract this information. But with the open access to pretrained transformer models it is way more feasible to extract useful information from natural language data. This paper will therefore investigate if a single natural language review can replace the mean numerical ratings of thousands of people when predicting box office success.

## Methods

This paper will use an XGBoost-model as the baseline prediction model. The baseline prediction model will fit its’ decision trees on basic tabular movie information downloaded from IMDb and box office revenue time series data. The scraped time series box revenue information is publicly available in a GitHub repository continuously updated by *Tom 'tjwaterman99' Waterman*[3]. This method is expected to perform well as it combines the unique characteristics of the movie with the global tendencies in the movie market. The timeseries data is pre-processed and combined with the tabular IMDb data, this is done by creating aggregated summations of previous time-intervals for each date. If a movie is released on the 9th of March 2017, then the model will have access to the average worldwide revenue for each day in the period 2nd to 8th of march, the average revenue for the month of February, January and December, and the average revenue for the year 2016. This should give the model access to global tendencies of movie revenue at release time. The IMDb data is meant to give a tabular overview of the movie-specific characteristics, and overview of this data can be found in the appendix as table 1. A subset was created without the variables averageRating and numVotes to create two distinct conditional datasets. These two variables were chosen to remove information about public reception of the movie to simulate a pseudo pre-release condition. This condition will be referred to as the no rating-score condition.

### Boosting of decision trees:

As the project utilizes a wide array of information types it was decided to use decision trees as our base learner, given their versatile nature and their historical effectiveness in movie predictions[2]. To accommodate for their weakness an ensemble boosting method was deployed, more specifically an XGBoost model. Gradient boosting is an ensemble learner that combines multiple weak-learner decision trees into a strong predictive model [4]. It is a boosting algorithm for regression. Gradient Boosting additively approximates F(x) as a weighted sum of decision trees where is the weight of the tree . Each tree is fitted to the pseudo residuals.

### XGBoost

The XGBoost [5] decision tree ensemble algorithm that is based on the gradient boosting algorithm. It is highly efficient and flexible. It utilizes the same ideas as gradient boosting but is much faster using second-order approximations. XGBoost has proven itself as the award-winning algorithm in multiple machine learning and data mining competitions[6]. This also includes performing well on times series data [7].

### Word embeddings:

XGBoost can’t fit decision trees on natural language like a movie review. The review therefore needs to be transformed into a numeric representation; these are referred to as word embeddings. This numeric word embedding is an ordered vector compromised by several hundred numbers representing a dimension of the word each. This means that the words exist in a high-dimensional space where two words that are close to each other are more similar in meaning. Historically word embedding has been decontextualized word embeddings known from Bag of Words model such as Word2Vec. These gives a word the same embedding independent of its surrounding words[8]. This means that for the two sentences ‘I love the movie’ and ‘I didn’t love the movie’ the word ‘*love*’ would result in the same embedding. However, this embedding idea still performs surprisingly and is the groundwork for future models. Contextualized Word Embeddings is created with deep learning architectures using decontextualized word embeddings, word tokenisation and word position as input [9]. The word embeddings are able to influence each other through self-attention, a machine learning framework introduced in the paper *Attention is all* *you need*.[10].

### Bidirectional Encoder Representations from Transformers (BERT)

A diagram of a network structure

Description automatically generatedThe deep learning structure used to transform linguistic information into numerical information in this project is called Bidirectional Encoder representation from transformers (BERT). BERT is a pre-trained easily deployable transformer available in R [11] by connecting to HuggingFace's transformer library [12] through the text-package [13]. Transformer-models’ structure is comparable to models such as Long Short-Term Memory models, in that they are both based on recurrent neural networks[14]. However, in contrast to LSTM models where only the previous state can affect the next sequential word, in self-attention models like BERT all words are connected and able to influence each other[13]. The BERT model used in this paper is the ‘bert-base-uncased’ has 12 attention layers of 768 dimensional weights as seen in figure 1. Each layer transforms the matrix of text-embeddings by applying weights to the previous layer, thus transforming what the model pays ‘attention’ to.

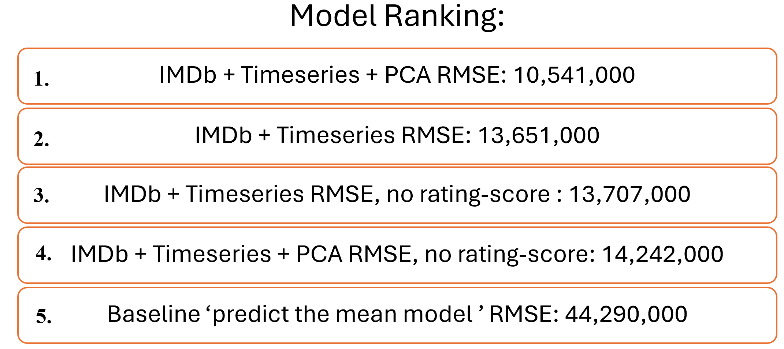
Figure 1: Structure of bert model

In an attempt at making transformer model more interpretable and understandable the of each layer has been investigated [15], and it was found that lower levels of the model encoded semantics while higher layers of the model encoded more abstract relationships and grouping of answer candidates. When predicting human behaviour, the later layers has ben found to be the most influential, and only the 11th attentional layer was extracted to make the project computationally feasible, the 11th layer is the model standard as well. This results in a matrix, where n is the input tokens of the string. This needs to be reduced to one sentence-level embedding, even though each set of token-embeddings are generated in the context of the sentence. This is necessary to keep input dimensions evenly sized across reviews when fitting the XGBoost model and to reduce computational load. The sentence embedding was defined as the additive sum of all embedding-vectors, i.e. the mean of each dimension. This number of dimensions was still too computational heavy to fit a model to, with the available computational power, so the dimensions was then reduced via PCA. It was found that 99% of variance could be expressed as 50 principal components. The PCA-dimensions are conditionally included in the model, resulting in 4 total models as shown in figure 2. The PCA-dimensions are high-level representations of a singular review, so sampling biases are expected to represent the individual reviews in this relatively A diagram of a model construction

Description automatically generatedsmall dataset. The data embedded in the was scraped using the ‘*rvest*’-package in r, the scraping script can be found in the project GitHub linked in the appendix.

Figure 2. Workflow of BERT-XGBoost model fit and evaluation

## Results

Each of the four models, conditioned on inclusion of PCA-dimensions and inclusion of numerical rating of movies, were all hyper-parameter tuned on 5-fold cross validation on 80to ensure optimal parameters for the task. From the tuning results the optimal model was chosen and tested on the withhold testing data. The models were then using RMSE as the scoring variable. Variance importance plots of each model can be seen in appendix. The best performance was found in the model that had access to all the data. However, surprisingly the model with access to PCA-dimensions without rating-scores seems too overfit to the PCA-dimensions and have reduced performance as a result. This could be a result of the models of the 11th layer of the BERT-model not representing semantic enjoyment but a more abstract scale, that only has predictive power in relation to the enjoyment-semantics expressed in the rating. All the models greatly outperformed the baseline mean model, that used mean of all movies to predict the box office revenue of new movies. The no rating condition also loses predictive power in the non-PCA condition as expected.

## Conclusion

A BERT-XGBoost ensemble workflow was used to predict the box office success of movies. By scraping IMDb and BoxOfficeMojo a comprehensive dataset was created. A BERT-model was used to embed the natural language into a XGBoost fit’able format that could improve prediction of the model. Future research could investigate if pre-release social media discussion could replace post-release reviews in predictive models. Future projects should also investigate if keeping an lower-level embedding together with the upper-level embedding used in this project could help baseline model perform better, by keeping embedding information that on average represent more semantic sentiments. Another research possibility would be to prompt each string with a preface such as: ‘What is the enjoyment rating of this review’ before embedding each review. A transformer-XGBoost is concluded to be a promising ensemble method with plenty of ways to improve.

## References:

[1] S. M. R. Abidi, Y. Xu, J. Ni, X. Wang, and W. Zhang, ‘Popularity prediction of movies: from statistical modeling to machine learning techniques’, *Multimed Tools Appl*, vol. 79, no. 47, pp. 35583–35617, Dec. 2020, doi: 10.1007/s11042-019-08546-5.

[2] K. Lee, J. Park, I. Kim, and Y. Choi, ‘Predicting movie success with machine learning techniques: ways to improve accuracy’, *Inf Syst Front*, vol. 20, no. 3, pp. 577–588, Jun. 2018, doi: 10.1007/s10796-016-9689-z.

[3] T. Waterman, ‘tjwaterman99/boxofficemojo-scraper’. Apr. 14, 2024. Accessed: Apr. 15, 2024. [Online]. Available: https://github.com/tjwaterman99/boxofficemojo-scraper

[4] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, ‘A Comparative Analysis of XGBoost’, *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[5] ‘XGBoost | Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining’. Accessed: Apr. 13, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2939672.2939785

[6] Q. Hoang, ‘Predicting Movie Genres Based on Plot Summaries’. arXiv, Jan. 15, 2018. Accessed: Apr. 14, 2024. [Online]. Available: http://arxiv.org/abs/1801.04813

[7] L. Zhang, W. Bian, W. Qu, L. Tuo, and Y. Wang, ‘Time series forecast of sales volume based on XGBoost’, *J. Phys.: Conf. Ser.*, vol. 1873, no. 1, p. 012067, Apr. 2021, doi: 10.1088/1742-6596/1873/1/012067.

[8] T. Mikolov, W. Yih, and G. Zweig, ‘Linguistic Regularities in Continuous Space Word Representations’, in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, L. Vanderwende, H. Daumé III, and K. Kirchhoff, Eds., Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 746–751. Accessed: Apr. 15, 2024. [Online]. Available: https://aclanthology.org/N13-1090

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. arXiv, May 24, 2019. doi: 10.48550/arXiv.1810.04805.

[10] A. Vaswani *et al.*, ‘Attention Is All You Need’, Jun. 2017.

[11] ‘R Core Team (2020). — European Environment Agency’. Accessed: Jan. 04, 2023. [Online]. Available: https://www.eea.europa.eu/data-and-maps/indicators/oxygen-consuming-substances-in-rivers/r-development-core-team-2006

[12] T. Wolf *et al.*, ‘HuggingFace’s Transformers: State-of-the-art Natural Language Processing’. arXiv, Jul. 13, 2020. doi: 10.48550/arXiv.1910.03771.

[13] O. Kjell, S. Giorgi, and H. A. Schwartz, ‘The text-package: An R-package for analyzing and visualizing human language using natural language processing and transformers’, *Psychological Methods*, vol. 28, no. 6, pp. 1478–1498, 2023, doi: 10.1037/met0000542.

[14] S. Hochreiter and J. Schmidhuber, ‘Long Short-Term Memory’, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] B. Van Aken, B. Winter, A. Löser, and F. A. Gers, ‘How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations’, in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing China: ACM, Nov. 2019, pp. 1823–1832. doi: 10.1145/3357384.3358028.

## Appendix figures:

|  |  |  |
| --- | --- | --- |
| Feature | Description | Scope |
| Total revenue | Box office revenue of each movie | 95% int: |
| Total Theatres | Amount of theatres displays of the movie | 95% interval: 330, 174,198 |
| Category | The thematical category of the movie | 18 one-hot encoded variables |
| runtimeMinutes | Length of movie in minutes | 95% interval: 85, 143 |
| averageRating | Average IMDb rating of viewers | 95% interval: 4.5, 7.8 |
| numVotes | Number of ratings on IMDb | 95% interval: 1820, 518200 |
| release\_year | Year of first release | 2000 - 2024 |
| release\_month | Week of the year of first release date | 1 - 52 |

Table 1: Tabular data downloaded and cleaned from imdb, 95% interval assumes normality of data.

### Variance Importance Plots:

A screenshot of a graph

Description automatically generated

Total\_theaters is expected to be correlated to a “cost of producing movie”, that was locked behind a paywall, and therefore not removed. When removed from dataset the ‘numVotes’ is the most important factor. Likely facebook likes, google searches or other popularity score is imagined to be able to replace it.