Predicting Movie Ratings with Multimodal Data

Yichen Yang
yyang9@stanford.edu

Ruoyun Ma
ruoyunma@stanford.edu

Min Haeng Cho
minhaeng@stanford.edu

1. Introduction

Can we predict the success of a movie based on information about the film available prior to theatrical release? To answer this question, we use open-source multimodal data released by IMDB and TMDB, such as movie posters and synopses. By applying image processing to visual data extracted from posters and natural language processing to textual data, we aim to predict movie ratings.

Although predicting movie ratings has been an active research area, a rather unexplored avenue of research in this application is the exclusive use of information available prior to theatrical release as features to predict movie ratings. Moviegoers base their decisions whether to watch a pre-released movie on limited information about the film, such as posters, synopsis, genre, cast and crew. Therefore, it behooves researchers to ask whether it is possible to predict movie ratings by only using information available prior to theatrical release as features, and if so, which features have the strongest predictive power. To predict movie ratings, we conduct an ablation study of various visual and textual features and evaluate their performance in the prediction accuracy. Specifically, we use linear and ridge regression, decision trees and random forest, SVR and neural networks as input algorithms.

Our motivation for exploring this question is twofold. First, we aim to provide consumers with film recommendations by predicting movie ratings accurately prior to theatrical release. Second, we hope to offer insight on the determining factors of film ratings that will guide producers through film promotion.

2. Related Work

Previous research has shown remarkable interest and advances in predicting movie ratings. As part of the 2009 Netflix Prize competition, an open contest for the best algorithm to predict user ratings for films streamed on the platform, researchers utilized singular value decomposition (SVD) to predict users' movie ratings based on their previous ratings history [1]. By utilizing a Bayesian approach, they effectively mitigated overfitting in SVD. In a different context, researchers utilized both surface and textual features such as the number of tweets as well as Youtube comments for each film to predict ratings [2]. Despite strong predictive performance, the model cannot predict movie ratings prior to release, since features from social media are extracted only after theatrical release, and is thus limited. Others developed a supervised latent Dirichlet allocation model (LDA), shown to have more predictive power than unsupervised LDA, to predict ratings from movie reviews [3]. Others used data mining to build a model on interesting relations between different attributes and assign a weight for each feature in every movie, enhancing prediction accuracy [4]. Most recently, researchers extracted and utilized visual features from movie trailers to predict ratings. [5]. Although results are preliminary, the approach is novel and we thus adopt this method to extract visual features from posters. Indeed, research in predicting movie ratings using state-of-the-art techniques and various features has been active and ongoing.

3. Dataset and Features

We utilized open-source data, "Movie Genre from its Poster Dataset" [6] and "The Movie Dataset" [7], from Kaggle. We selected posters, synopses, cast, crew, runtime and genre as input features and IMDB film ratings as the prediction objective.

We split features into three categories: images (posters), text (synopses), and others (cast, crew, genre, and runtime). For posters, we transformed the pixel dimensions from 900x600 to the input resolu-

Table 1. Processed Data by Groups

Data	Type	Dimension	Example
Genre	Categorical	23	Action
Runtime	Numerical	1	100 (minutes)
Actors	Categorical	1603	Robert Downey Jr.
Director	Categorical	492	Steven Spielberg
Poster	Numerical	13	Number of faces = 1
Synopses	Categorical	3884	"innocence"

tion 224x224 for ResNet34. Besides feeding pixels into our model, we manually extracted 13 visual features. Specifically, we considered posters in both RGB format and HSB format, and extracted the mean and standard deviation of red, green, blue, hue, saturation, brightness (as suggested by [5]), as well as the number of human faces using openCV (as suggested by [8]). For synopses, we used spaCy[9] for tokenization and only kept words that appeared in at least 20 movies. For cast and crew, we extracted the main director and top three leading actors for each movie, and kept directors and actors involved in at least 5 movies in our dataset. The result is shown in Table 1.

After filtering out movies released before 1980 and whose original language is not English, we had 19429 movies in our sample. We then did a train-validation-test split at the 70%-15%-15% ratio, with 13600 training, 2914 validation and 2915 test data points. We standardized all the numerical features, performed feature selection and hyperparameter tuning using the validation set, and did model selection with the test set.

4. Methods

4.1. Linear Regression Models

For our baseline algorithm, we used linear regression, which models the relationship between independent and dependent variables by finding a linear correlation and minimizing the sum of the squares of the differences between predicted and actual values. It is among the most common, fundamental and simple methods used to solve regression problems.

To mitigate overfitting that might arise from fitting a simple regression model to our data, we also used ridge regression. This regularization method adds an extra l_2-norm of the parameter to the cost function to penalize large regression coefficients. In general, ridge regression helps solve multicollinearity, or high inter-correlations among features, and efficiently improves model prediction accuracy.

4.2. Decision Trees and Random Forest

Since a nonlinear model could provide a better fit to our data, we also experimented with decision trees and random forest, an ensemble learning method that aggregates outputs from a multitude of decision trees. A decision tree contains internal nodes corresponding to input features, and leaves, which represent the output value following certain paths from the root. Random forest utilizes a bagging strategy by splitting input features into a random subset while selecting the best feature for each node of a decision tree. It is extensively used for non-linear problems due to strong stability and efficient reduction of overfitting.

4.3. Support Vector Regression

Support Vector Machine (SVM) is commonly used in many machine learning problems as baselines. SVM defines a margin from the hyperplane, where points inside the margin incur loss, while Support Vector Regression (SVR) defines a margin of ϵ-distance from the hyperplane such that data points inside the boundary are error-free. For our project, we used the radical basis function as the kernel.

4.4. Neural Networks

To effectively capture information from the multimodal data, we used convolutional neural network (CNN), used to scale down the magnitude of patterns from original inputs, as a primary tool to analyze posters and perform natural language processing on synopses. We did word embedding on synopses and applied kernels on the embedding. We also applied a residual neural network ResNet 34 [10], known for mitigating the problem of vanishing gradients via shortcuts between layers, to poster data.

4.5. Feature Importance

To evaluate the predictive power of different features, we employed permutation feature importance (FI) [11], a technique that measures the importance of each feature. The intuition is that if a feature is not useful for predicting an outcome, then permuting its values will not result in a significant reduction in a model's performance. For each feature, permutation

FI is defined by:

$$\text{FI} = \text{err}_{\text{perm}} - \text{err}_{\text{orig}}$$

where err_{orig} is the baseline model error with all the original features, and err_{perm} is the model error when a certain feature is permuted.

Also known as Random Forest FI, mean decrease impurity (MDI) is another FI metric in a random forest model that computes the extent to which each feature decreases the weighted impurity on average while training the model. We evaluated each individual feature and group of features on the train set using MDI FI and permutation FI, respectively.

5. Experiments and Results

5.1. Metrics

We used the Mean Square Error (MSE) and R^2 as our metrics. MSE was also used for training and R^2 for representing the proportion of the variance for the predicted film ratings explained by the features.

5.2. Individual Models

We first assessed individual contributions of the three feature categories–text, images, others–in predicting film ratings. When we trained word embedding and a CNN model (Kernel size = 2,3,4, 100 kernels each) on the synopses as our only feature, we had an R^2 of 0.136. This implies that synopses can be used to predict movie ratings. However, when we trained a ResNet34 model only on 224x224 posters, we either had overfitting (small regularization) or saw no significant improvement over simply predicting movie ratings using the mean (big regularization). Thus, a complicated model is not suitable for predicting movie ratings with posters. We therefore manually extracted 13 visual features instead of directly feeding the posters to our model. Combining these 13 visual features into a linear regression model, we had an R^2 of 0.015, similar to the result in [5]. Thus, visual features in movie posters are capable of predicting movie ratings. Table 2 displays the results of all the individual models.

5.3. Combined Models

We set the results from linear regression as our baseline, and used validation data to tune hyperparameters.

Table 2. Preliminary results on text and image data

Data	Method	Valid MSE	Valid R^2
Overview Only	Word Embedding + CNN	1.2900	0.136
Poster Only	ResNet34	Tend to overfit	Tend to overfit
Extracted Features from Poster	Linear Regression	1.4699	0.015

Table 3. Model MSE and R^2

Method	Test MSE	Test R^2
Linear Regression	0.9302	0.3745
Ridge Regression	0.8775	0.4099
Decision Tree Regression	0.8959	0.3975
Random Forest Regression	0.8546	0.4253
Support Vector Regression	0.8542	0.4256
Neural Network	0.8765	0.4109

Additionally, we used L2 regularization on linear regression, and results show that the best alpha is 10. For decision tree regression, the optimal maximum depth was 8. For random forest regression, we set the maximum depth to be 32 and used 100 trees. For support vector regression, the optimal penalty parameter was 1 and the optimal ϵ was 0.1. Finally, we combined results from the CNN for the textual features and non-textual features into 5 fully connected layers for our final neural network. The regularization techniques we used for the neural network are dropout, L2 penalty, and early stopping. The results are shown in Table 3.

We found that all three feature categories contribute to the prediction of IMDB scores. Our current results show that random forest regression and support vector regression have the best performance, which explain the 42 percent of variance in IMDB movie scores.

6. Discussion

6.1. Best Model

Adding L2 regularization improved regression performance. SVR and random forest regression have similar R^2, but SVR is significantly slower and harder to interpret. Considering accuracy, efficiency and interpretability, random forest was our best model.

Compared to the previous related work discussed in Section 2, which used movie trailers and genre to predict film ratings and achieved an MSE of 0.88 [5], we have a smaller MSE because our model had more fea-

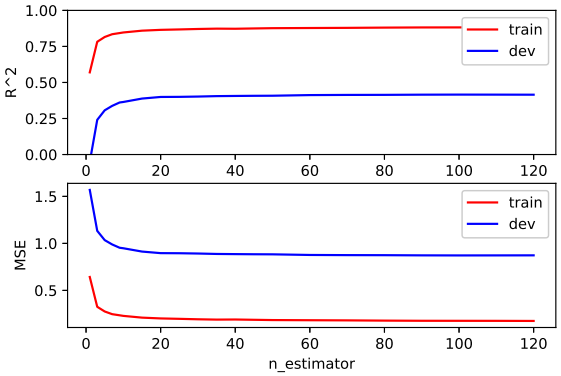

Figure 1. Model performance of different number of trees

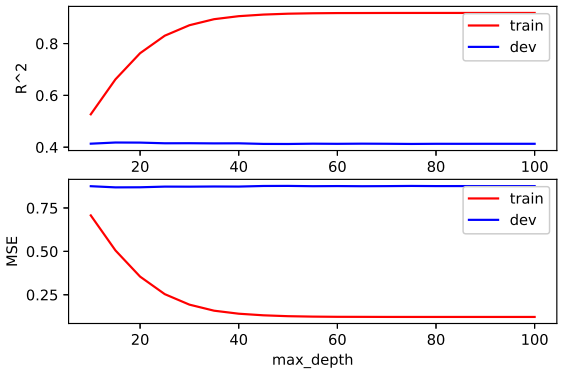

Figure 2. Model performance of different max depth

tures. However, compared to the social media model which had an MSE of 0.2 [2], our model performed worse. This could be because we exclusively used as features information available before theatrical release, whereas the social media model used features such as social media comments, which can only be retrieved after the movie release.

6.2. Effect of Parameter Tuning

We discuss the effect of two key parameters in the random forest model: number and maximum depth of trees. Figure 1 shows that as the number of trees (n_estimator) increased (max depth = 32), the train and valid sets showed the same trend in R^2 and MSE. The performance curves also flattened out after 20 trees.

Figure 2 (number of trees = 100) shows that the performance of the train set improved as the maximum depth increased, while the valid set line almost remained unchanged. When we reduced the number of trees, the valid set and train set lines moved in opposite directions. The fewer trees we used, the sharper the change in R^2 and MSE for the dev set. That is, the bagging strategy with 100 trees in the forest effectively mitigates or even eliminates high variance.

Due to a trade-off between training time and model performance, 100 and 32 are reasonable values for the number of trees and maximum depth, respectively.

6.3. Feature Contributions

For Random Forest FI (Figure 3), certain genres, such as documentary, horror and drama, have high FI. Runtime is the second most important feature that can be used to predict movie ratings. Furthermore, manually extracted visual features are also important, including standard deviations for RGB and HSB channels. By contrast, none of the features associated with actors, directors or synopses appear in the 20 most important features.

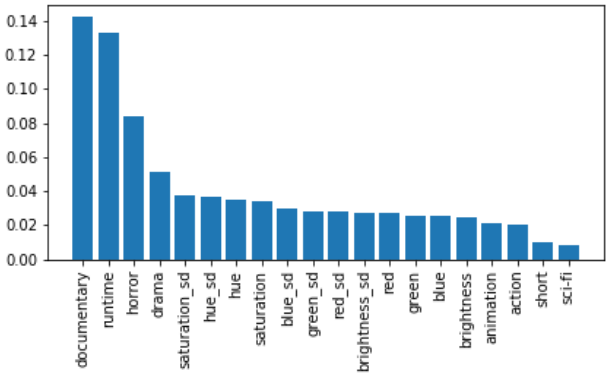

Figure 3. Feature importance using Random Forest

Table 4 shows the top 5 permutation features in descending order for four feature groups. FI determines the magnitude, not the direction, of effect on movie ratings. Since random forest FI considers the importance of each individual feature, it assigns low scores to sparse feature columns. Thus, we used permutation FI to determine the predictive power of feature groups by permuting all feature columns for every feature group and calculating the difference of the new and original full-model MSE (Figure 4).

As with Random Forest FI, genre, runtime and manually extracted visual features had the highest FI. By contrast, actors, directors and synopses had low to negligible importance.

Table 4. Feature Importance Ranking (Top 5)

Poster	Genre	Actor	Director
hue_sd	documentary	Stephen Baldwin	Tyler Perry
hue	horror	Thomas Kretschmann	Woody Allen
saturation	drama	Lauren Bacall	Craig Moss
blue_sd	animation	Manisha Koirala	Uwe Boll
green_sd	action	Angie Everhart	Steven R. Monroe

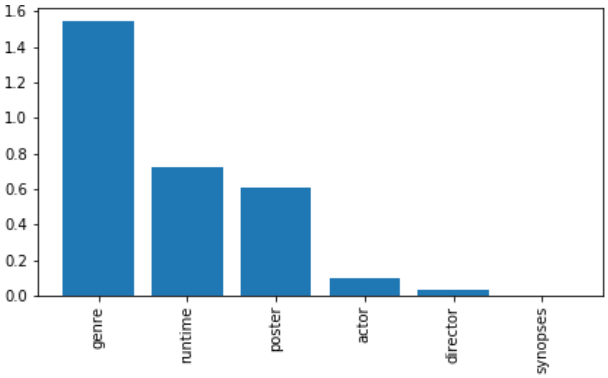

Figure 4. Permutation feature importance

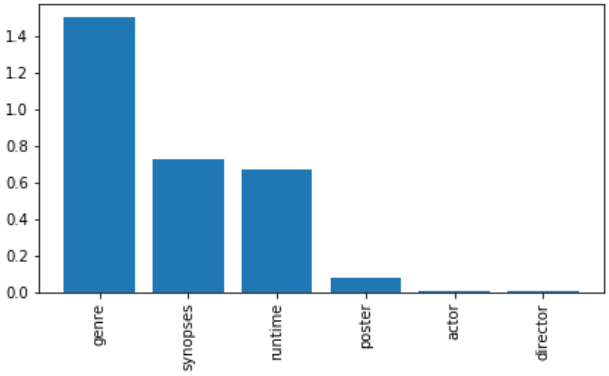

Figure 5. Permutation feature importance of nested random forest

Table 5. Comparison of Regression and Classification

Method	Test MSE	Test R^2
Classification (5 classes)	0.8722	0.4135
Classification (10 classes)	0.8649	0.4184
Classification (20 classes)	0.8608	0.4212
Regression	0.8546	0.4253

6.4. Nested Model

Although CNN with synopses as the sole feature can reach R^2 of 13.5%, synopses alone have negligible FI in random forest. A possible explanation is that using vocabulary dictionary failed to capture sequential information in synopses.

We used CNN to extract and feed 300 features from synopses into the final random forest model. Yet our resulting nested model was 1% lower in R^2. This could be because CNN models tend to overfit, causing our nested model to overfit. After using the features extracted by CNN, synopses and posters had higher and lower permutation FI, respectively.

6.5. Classification Perspective

In addition to regression, we framed the film rating prediction task as a classification problem. We discretized movie ratings uniformly into 5, 10 and 20 classes, and chose a central value for each interval to represent each class. We then applied a random forest classification model to the data using the same parameter settings as in regression.

Test MSE and R^2 are shown in table 5. The higher the number of classes, the higher the R^2 and lower the MSE on the test data. We also tested other classification algorithms with corresponding regression models (e.g. SVM for SVR, Logistic Regression for Linear Regression). For all the models, regression outperformed classification because discretizing data renders some information missing from the original data.

7. Conclusion and Future Work

All three feature categories contribute to the prediction of movie ratings. By feeding information available prior to theatrical release into a random forest model, we could explain 42% of variance in English-language movie ratings, better than other models that depend only on information prior to theatrical release. Among all the features, genre had the highest predictive power for film ratings. Certain combinations of genre like horror and action could lead to bad ratings, while others like documentary and history could lead to good ratings. Thus, production companies and consumers should consider various combinations of genres for movie production and choice.

Adding new features, such as movie trailers, could help improve our film rating prediction. Another area for future research is the development of a sophisticated nested model to better combine information from all three feature categories.

8. Contributions

Yichen Yang and Ruoyun Ma co-wrote the report and ran experiments. Min Haeng Cho performed literature review and co-wrote and finalized the report. We all created the poster together. The GitHub Link for codes is https://github.com/JeffJeffy/CS229Project.

References

[1] Y. J. Lim and Y. W. Teh, "Variational bayesian approach to movie rating prediction," in *Proceedings of KDD Cup and Workshop*, vol. 7, 2007, pp. 15–21.

[2] A. Oghina, M. Breuss, M. Tsagkias, and M. De Rijke, "Predicting imdb movie ratings using social media," in *European Conference on Information Retrieval*. Springer, 2012, pp. 503–507.

[3] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.

[4] J. Ahmad, P. Duraisamy, A. Yousef, and B. Buckles, "Movie success prediction using data mining," in *Institute of Electrical and Electronics Engineers*, 2017.

[5] F. B. Moghaddam, M. Elahi, R. Hosseini, C. Trattner, and M. Tkalčič, "Predicting movie popularity and ratings with visual features," in *2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*. IEEE, 2019, pp. 1–6.

[6] KaggleInc, "Movie genre from its poster," https://www.kaggle.com/neha1703/movie-genre-from-its-poster.

[7] Kaggle, "The movies dataset," https://www.kaggle.com/rounakbanik/the-movies-dataset.

[8] C. Sun, "Predict movie rating," https://nycdatascience.com/blog/student-works/web-scraping/movie-rating-prediction/, 2016.

[9] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[11] A. Fisher, C. Rudin, and F. Dominici, "Model class reliance: Variable importance measures for any machine learning model class, from the "rashomon" perspective," *arXiv preprint arXiv:1801.01489*, 2018.