What Makes a High-rating Movie: Reviews Mining and Rating Prediction

https://github.com/chuhan98/BT5153_Group8

Abstract

In this project, we trying to use Machine Learning, Deep Learning and NLP models to understand and predict the movie rating in IMDB. The final selected model is Light GBM model with inputs of both basic features and text features. And the results shows that review sentiment score, revenue, budget and runtime are the most important features that impact movie rating.

1. Problem Definition

As a commodity of the art form, the film is an important part of modern cultural life. The global film and video market reached a value of nearly $59.14 billion in 2021, having increased at a compound annual growth rate. And the market size is expected to reach $114.93 billion by 2025 with the post-pandemic economic recovery, according to a new report by Grand View Research, Inc.

The study of the success of movies mainly goes through three stages. The first stage was in 1940s, mainly through analyzing the audience feedback collected by audience research institute; In the second stage, represented by Barry Litman, a large number of impact factors were added to conduct multiple linear regression models to predict the variables including movie box office. The third stage mainly used the massive content generated by netizens online as the main source of prediction.

Nowadays, the development of information platforms such as Twitter and YouTube allow audiences to express their views and opinions freely. And with the development of network films, public praise has become the main criterion for advertisers to judge a movie, which influences their investment. Information transparency has led to a strong correlation between film reputations and revenues. The audience's approbation degree with the film will be reflected through its rating, which is an important standard to measure the success of the film.

This project aims to predict the rating of films to quantify their popularity. We will use IMDb movie data to conduct machine learning models including classification models, ensemble methods like Decision Tree, Random Forest, Gradient Boost, XGBoost, and neural network, a deep learning algorithm that helps mimic the non-linear and complex patterns in calculations. And we will also use natural language processing methods to do reviews text mining and posters image processing to contribute to our prediction. Through conducting these methods, we can put forward suggestions on film production, marketing, and distribution.

2. EDA & Data Preprocessing

All data used in this project are from *Kaggle.com*[1], we combine IMDb Dataset and The Movies Dataset through *IMDb Ids* and combine Review Dataset through movie titles. The final dataset contains more than 45,000 movies from 143 countries. The objective of our project is to find the reasons behind a high rating and vote of movies in IMDb using all these available information as well as online movie reviews to predict ratings of new movies in IMDb.

Thus, there are five candidate variables that indicate the popularity degree of a movie, they are *averageRating*, *vote_average*, *numVotes*, *popularity* and *revenue*. We finally choose *averageRating* as model's target variable due to its normal distribution and easier interpretability. Besides, there are other basic information of movies including *type*, *genres*, *release date*, *directors*, *actors* and so on.

Text and image data also help to bring insights of a hit movie, in this project, we use NLP techniques to do the sentiment analysis of audience reviews and use Neural Network to process movie posters.

The table below shows information of variables used in this project.

[1] IMDB Dataset:

https://www.kaggle.com/ashirwadsangwan/imdb-dataset

The movies Dataset:

https://www.kaggle.com/rounakbanik/the-movies-dataset

IMDB Review Dataset:

https://www.kaggle.com/ebiswas/imdb-review-dataset

Table 1. Table of variables from Movie Dataset.

	Variables	Type
Target Variable	Average Rating	Float
Movie Title	Primary Title	String
	Original Title	String
Movie Related Information	Title Type	String
	Start Year	Int
	Release Date	Datetime
	Is Adult	Boolean
	Genres	String
	Runtime	Float
	Budget	Int
	Original Language	String
	Production Companies	String
	Production Countries	String
	Status	String
	Director	String
	Actor	String
Text Data	Overview	String
	Keywords	String
	Reviews	String
Image Data	Poster Path	String

2.1 Exploratory Data Analysis

After simple processing the whole dataset, we gain 9 numeric variables and 11 categorical variables excluding text and image data.

2.1.1 Numeric Data Analysis

The descriptive statistics table of numeric variables shows in appendix. We also find that a large part of movies doesn't have the data of budget and revenue, which means further data collection or fill-in techniques may be needed if we want to use these two variables.

We also draw the histograms of all numeric variables to check bias of data distribution. 99% Winsorize processing is used before drawing in order to remove outliers.

Finally, we draw a correlation matrix for the numerical variables seeking some correlation between variables. We can see relatively strong correlation between 6 numercial and averageRating.

2.1.2 Categorical Data Analysis

We visualize the categorical features using wordcloud and histgrams(appendix). We can see some distribution across different categories in the appendix.

2.2 Data Pre-processing

Among the whole dataset, 80% data (29760 movies) are used for training and 20% (7440 movies) for testing.

For all numerical features, we normalize both training and test dataset using mean and standard deviation of training dataset.

For categorical features, there exits so many classes for several features like directors and actors, so we decide to simplify classes as below before applying one-hot encoding.

Table 2. Categorical features remained for one-hot encoding.

Features	Classes
Title Type	movie, short, tv-Episode, tv-Mini-Series, tv-Movie, tv-Series, tv-Short, tv-Special, video
Start Year	pre2000, 2000-2010, post2010
Genres	Action, Adult, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Foreign, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, TV Movie, Talk-Show, Thriller, War, Western
Original Language	en, fr, it, ja, de, es, ru, hi, others
Production Companies	Paramount Pictures, Metro-Goldwyn-Mayer (MGM), Twentieth Century Fox Film Corporation', Warner Bros., Universal Pictures, others
Production Countries	US, GB, FR, CA, JP, IT, others
Status	Canceled, In Production, Planned, Post-Production, Released, Rumored
Director	John Ford, Michael Curtiz, Werner Herzog, Alfred Hitchcock, Woody Allen, Georges Méliès, Sidney Lumet, Jean-Luc Godard, Charlie Chaplin, Raoul Walsh, others
Actor	John Wayne, Jackie Chan, Nicolas Cage, Robert De Niro, Gérard Depardieu, Michael Caine, Burt Lancaster, Paul Newman, Bruce Willis, Barbara Stanwyck, others

After preprocessing, there are totally 96 features excluding text features used in the following models.

3. Movie Overviews and Reviews Analysis

3.1 Overviews and Keywords Analysis

In the text mining part, we have two types of data. One is the movie overview, and another is the movie keywords.

For overviews, we first use the Texthero package to lowercase and remove digits, URLs, punctuation, stop words, and HTML tags. lemmatization and stemming are also used to clean texts. In the model part, we conduct the pre-trained Distil BETR model without fine-tunes to conduct feature extraction. BERT is short for Bidirectional Encoder Representations from Transformers, which

indicates a transformer-based machine learning technique for natural language processing pre-training. After inputting sentences into BERT, the Bert Tokenizer will first break sentences into tokens, add 'CLS' and 'SEP' tokens, and then use their indexes in vocab to replace them. The Bert deep learning layers will finally extract features with a dimension of 768.

We first rated movies above 6.5 as high scores and those below 6.5 as low scores. By conducting Logistic Regression models using 768 features extracted by BERT, we can get out-of-sample accuracy of 67.87%.

For keywords mining, we first use the same text cleaning methods as the movie overview. After that, we conduct three methods, including the Bag of Word model, the One hot encoding method, and the TF-IDF model. Because there are too many words for movie overviews, we do not conduct these three models for it.

The bag of Word model is a straightforward way to turn a sentence into a vector representation, and it can convert sentences in the text to a word frequency matrix by counting the number of occurrences of each word. We also drop infrequent words that appear less than 10 times in total and the dimension of features is 2953. After setting movies above 6.5 as label=1 and those below 6.5 as label=0, we can get out-of-sample accuracy of 63.29%. According to the coefficients of LR, the top 5 words having a positive influence include 'anime', 'gojira', 'bell', 'innocence', and 'jazz'. And the top 5 words having a negative influence include 'carry', 'bigfoot', 'karate', 'bikini', and 'duringcreditssting'.

We also use One Hot encoding to get a matrix of 2953 dimensions of whether or not the frequent word appears in the sentence, which leads to the accuracy of 63.43%. And the top5 words having a positive influence include 'cinematic', 'samurai', 'phenomenon', 'anime', and 'classic'. And the top 5 words having a negative influence include 'escort', 'bat', 'bikini', 'pokémon', 'zomby', which are quite different from BOW.

TF-IDF is a commonly used weighting technique for text mining, which is used to assess the importance of a word to one of the documents in a corpus. And it is calculated by term frequency times inverse document frequency. After getting features with 2953 dimensions, we can get an accuracy of 64.82%. The top5 words having a positive influence on the LR model include 'anime', 'immigrant', 'history', 'samurai', and 'noir'. And the top 5 words having a negative influence include 'slasher', 'bikini', 'sex', 'zomby', and 'mutant'. From the influence of the word in the three models, we can see, that 'anime' and 'samurai' make a great contribution to high rates, and 'bikini' and 'zomby' make a great contribution to low rates. But because the matrix is too sparse, the influential words are not similar in different models.

To reduce feature dimensions, we also conduct principal component analysis on the TF-IDF matrix. The cumulative explained variance is shown in *Figure 1*. To make sure that the cumulative explained variance is larger than 60%, the first 473 PCs are chosen.

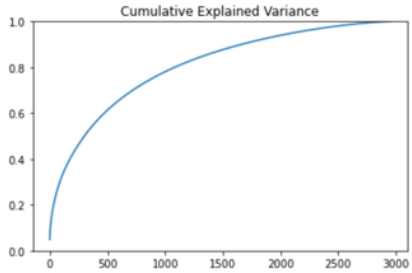


Figure 2. The cumulative explained variance of PCA

The out-of-sample performance accuracy using the LR model is 64.12%. The accuracy is slightly lower than LR using TF-IDF matrix and higher than LR using BOW and One hot encoding matrix. The total models' performance is shown in *Table 3*.

Table 3. LR models' performance using text features

		LR Accuracy	Feature Dimension
Overview	BERT	**0.6787**	**768**
Keywords	BOW	0.6329	2953
	OneHotEncoder	0.6343	2953
	TF-IDF	0.6482	2953
	TF-IDF& PCA	0.6412	473

Therefore, in classification model construction, we will mainly use features extracted from BERT.

3.2 Reviews Analysis

3.2.1 Sentiment Analysis

We conduct sentiment analysis for 2958353 movie reviews to extract emotional information from people's reviews on movies. We apply Valence Aware Dictionary (VADER), a module in NLTK, to complete the analysis. Meanwhile, the sentiment score generated by VADER can then be input as a feature to predict movies' ratings.

Although the standard criteria in VADER is 0.05, we choose a different threshold to classify reviews' sentiment type, which is 0.05 and 0.7(a compound score bigger than 0.7 indicates positive, and a compound score smaller than 0.05 indicates a negative paragraph), to extract more representative phrases and control the sample size. According to the distribution of sentiment score, we can see the mean is around 0.6. We believe most of the audience have a herd mentality and try to praise, so the extreme positive remarks can better reflect the advantages of the film.

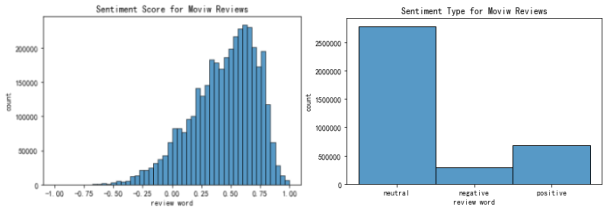


Figure 2. Sentiment Score and Sentiment Type

3.2.2 LDA

We utilize Latent Dirichlet Allocation (LDA) to summarize the topics in positive and negative reviews respectively. LDA builds a topic per document model and words per topic model, modeled as Dirichlet distributions. After modeling, a similarity-based optimal method for LDA is used to determine the number of topics and perform topic analysis. The specific steps are as follows:

1)Take the initial number of topics k values, get the initial model, and calculate the similarity (average cosine distance) between topics.

2)Increase or decrease the value of k, retrain the model, and calculate the similarity between topics again.

3)Repeat step 2 until the optimal k value is obtained.

From Figure3, it can be seen that for positive and negative comment data, the average cosine similarity between topics reaches the lowest when the number of topics is 9.

Therefore, the number of topics can be chosen as 9.

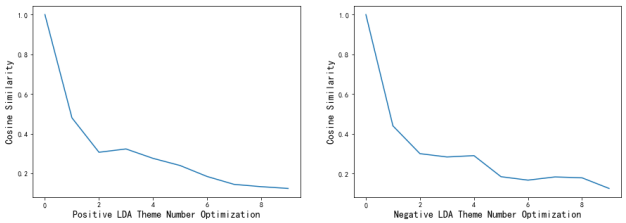


Figure 3. Cosine Similarity of Topics

Based on the result of LDA, we can see the negative comments description of the movie mainly deals with the subject words like war, space, etc., and little about the filming techniques of the movie. However, the positive comments mainly contain the actors' acting, storyline, theme (love, society), etc. These factors mainly influence people's positive comments on the movie.

Table 4. LDA Topics Sample

Negative topics		Positive topics	
American\ge rman\japane se\british\wh ite\army\hist ory\years\bl ack\battle	Action\lik e\fight\se ries\effec ts\time\wo rld\plot\s pace\alien	Character\ performanc e\role\cast\ michael\cha racters\act ion\screen	Life\people \world\lov e\story\re al\human\b eautiful\tr ue\society

4. Rating Prediction

In this part, we are trying to use machine learning and deep learning models to predict the average rating of each movie using both basic metadata and text features generated from NLP models.

We try both Regression and Classification models to find more insights. For Regression models, the target variable is average rating ranged from 0-10. For Classification models, we set target equals to 1 when average rating greater than or equal to 6.5, and 0 otherwise.

Also, for the input features, we try both basic features and basic features plus text features.

4.1 Linear Regression & Logistic Regression

For Linear Regression, the model performance is shown below.

Table 5. Linear Regression Performance.

	Basic Model		Basic & Overview & Review Model	
	Train MSE	Test MSE	Train MSE	Test MSE
Linear Regression	0.83	0.84	0.69	0.74

Both training and test MSE drop around 0.1 after adding text features.

For Logistic Regression, we can draw the confusion matrix to compare the model performance.

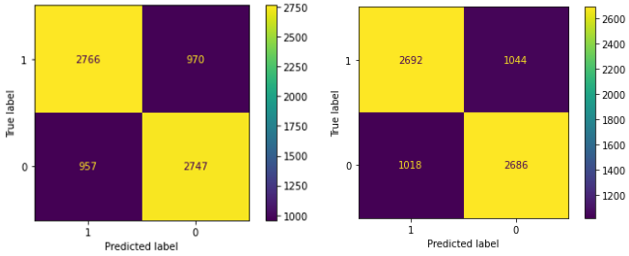


Figure 4. Test confusion matrix of Basic Model (left) and Basic & Overview & Review Model (right) using Logistic Regression.

The numbers of True Positive and True negative increase a lot after adding overview and review features. And both two models have a balanced TP, FP and Recall rate.

4.2 Tree Models

In this case, we adopt 3 different tree models, including the Decision Tree, the Light GBM, and the XGBoost.

Here are the performances of the tree regressors.

Table 6. Train / Test MSE for regression tree models.

	Basic Model		Basic & Overview & Review Model	
	Train MSE	Test MSE	Train MSE	Test MSE
Decision Tree	0.69	0.79	0.73	0.82
Light GBM	0.64	0.70	0.43	**0.62**
XGB	0.54	0.69	0.40	0.63

It can be found out that by adding more features, the regression performance is significantly enhanced. With features including basic, overview and review features, LightGBM model performs the best, with a minimum MSE of 0.62 on the test set.

To be more detailed, for this LightGBM regression model, the hyperparameters selected after tuning is: *bagging_fraction* = 1, *feature_fraction* = 0.8, *num_leaves* = 35.

For classification, we used the following tree models for prediction: random forest, XGBoost and Light GBM, since they can capture more complex relationship between features and variable and usually provide high performance.

We first used default hyperparameter and then tuned the best performing basic model Light GBM. The final hyperparameter used are listed as follows: colsample_bytree = 0.5, learning_rate = 0.1, max_depth = 10, min_child_samples = 15, min_child_weight = 0.001, n_estimators = 200, reg_alpha = 0.5, reg_lambda = 0.5, subsample= 0.3.

The results of the model performance are summarized in the section **4.5**.

4.3 Multilayer Perceptron (MLP)

Here are the hyperparameters, model structure, loss function and optimizer of regression and classification model.

Table 7. MLP hyperparameters.

	Regression Model	Classification Model
Model Structure	Input -> Linear (64) -> Output (1)	Input -> Linear (64) -> Output
Batch Size	16	16
Number of Epochs	20	5
Loss Function	MSE	Binary Cross Entropy
Optimizer	Adam	Adam

For MLP-regression model, we draw the loss curve during training process.

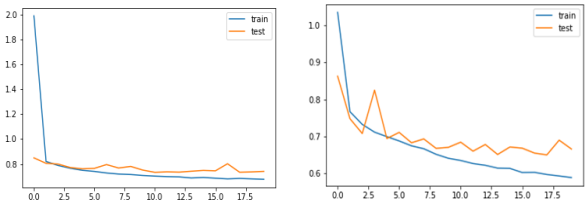


Figure 5. Training and Test MSE during training process of Basic Model (left) and Basic & Overview & Review Model (right) using MLP.

The overall MSE for both training and test achieve a lower level when adding overview and review features to the basic model. Test MSE of Basic Model only slightly drop during training process, while test MSE of Basic & Overview & Review Model shows an obvious decrease trend, which means MLP learns better after adding overview and review features.

The results of MLP-classification model show that although the accuracy results are similar in these two models, Basic & Overview & Review Model can gain higher True Positive rate and more balanced TP and FP rate in out-of-sample data.

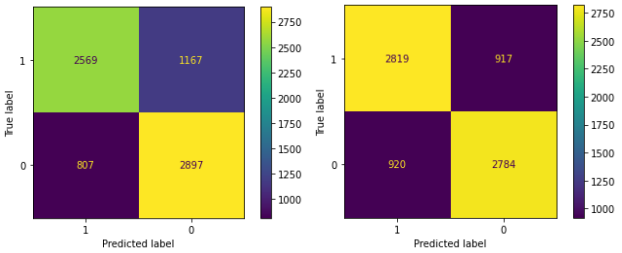


Figure 6. Test confusion matrix of Basic Model (left) and Basic & Overview & Review Model (right) using MLP.

4.4 Regression Models Comparison

Table 8. Train / Test MSE for regression models.

	Basic Model		Basic & Overview & Review Model	
	Train MSE	Test MSE	Train MSE	Test MSE
Linear Regression	0.83	0.84	0.69	0.74
Decision Tree	0.69	0.79	0.73	0.82
Light GBM	0.64	0.70	0.43	**0.62**
XGB	0.54	0.69	0.40	0.63
MLP	0.68	0.74	0.59	0.67

If we see the target variable as continuous, predicting the average rating of a movie will be a regression problem.

Compared to see it as a classification problem, it takes the order and the real meaning of the target number into consideration.

The regression model based on Light GBM performs the best, with an MSE of 0.62 on the test set. It uses basic data, features extracted from the overview and the review to train.

4.5 Classification Models Comparison

The distribution of high rating movie between the train and test dataset are shown in the figures below. We can see the distribution are balanced. We chose accuracy as evaluation metrics, since we care about the classification of both classes.

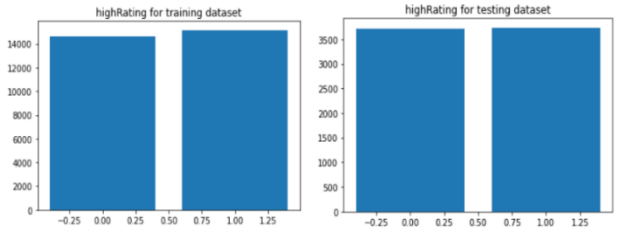


Figure 7. Distribution of High Rating Movie in Train / Test Dataset.

Table 9. Train / Test accuracy for classification models.

	Basic Model		Basic & Overview & Review Model	
	Train MSE	Test MSE	Train MSE	Test MSE
Logistic Regression	72.75%	72.28%	75.54%	74.10%
Random Forest	87.99%	74.46%	100%	70.71%
Light GBM	78.19%	74.81%	84.12%	76.76%
XGB	79.49%	74.77%	96.33%	74.48%
MLP	74.38%	73.49%	75.62%	75.31%

We can see that the Light GBM provide the highest test accuracy for both models with and without text feature after hyperparameter tuning.

The model accuracy is also improved overall after adding the text features.

5. Explainable AI & Insights

In this part, the model based on Light GBM with 865 features is further explore. We try to find out the key to high-rating movies by global interpretation as well as local interpretation. In global interpretation, feature importance scores and permutation feature importance are calculated to find out the most impactful features on a movie's average rating. In local interpretation, by conducing LIME and SHAP ratio, we try to find out why the movie with the highest average rating and the one with the lowest rating get their results respectively.

5.1 Global Interpretation

The feature importance figure shown in *Figure 8* is plotted based on the feature importance score of the Light GBM model. Permutation feature importance is shown in *Figure 9*. It can be discovered that the top features are very alike, especially the runtime, the sentiment score based on the review, and the revenue of the movie, which indicates that regardless of the methods chosen to calculate the feature importance, the result is robust and thus the movie makers surely should pay more attention to these top features.

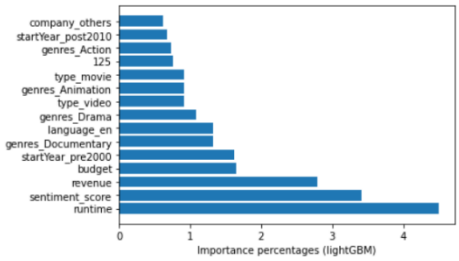


Figure 8. Feature importance

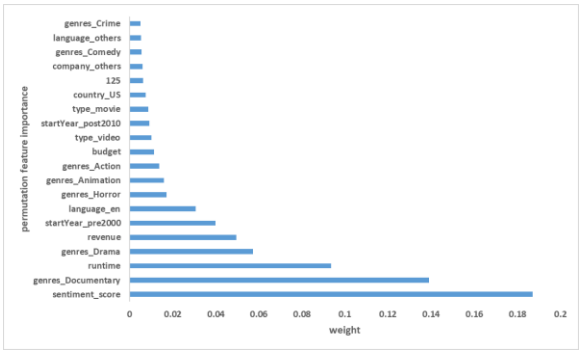


Figure 9. Permutation feature importance

For classification model, we used Light GBM which provides the best prediction accuracy for feature importance interpretation. We also used Permutation Feature Importance model, which shuffles the feature value within the dataset, calculate the performance loss happened due to shuffling, based on Light GBM for feature importance. The results of the two models are slightly different because of the different algorithm behind.

As we can see from the figures in appendix, for classification model without text features, **runtime**, **revenue**, **budget** and **startYear_pre2000** are important features for the high rating movie prediction. After adding text features, we can see that the **sentiment_score** start to play a prominent role in prediction, while the other featues remain as important features, but the importance level falls behind the sentiment_score. This is reasonable, since people leave positive comment of the movie will give a high rating to the movie.

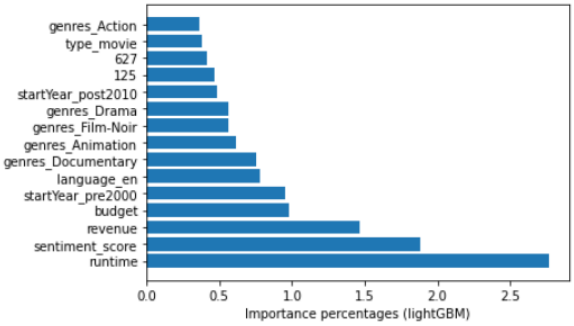


Figure 10: Feature importance for classification model with text feature using Light GBM.

Weight	Feature
0.0449 ± 0.0042	genres_Documentary
0.0290 ± 0.0043	sentiment_score
0.0281 ± 0.0067	runtime
0.0177 ± 0.0034	startYear_pre2000
0.0166 ± 0.0058	genres_Drama
0.0132 ± 0.0033	language_en
0.0117 ± 0.0028	revenue
0.0067 ± 0.0029	budget
0.0052 ± 0.0021	startYear_post2010
0.0048 ± 0.0021	genres_Film-Noir
0.0044 ± 0.0020	genres_Horror
0.0044 ± 0.0004	type_movie
0.0042 ± 0.0021	genres_Animation
0.0027 ± 0.0012	genres_Action
0.0023 ± 0.0023	497
0.0022 ± 0.0020	genres_Biography
0.0020 ± 0.0011	type_short
0.0020 ± 0.0018	genres_Comedy
0.0019 ± 0.0005	331
0.0018 ± 0.0009	592
… 845 more …	

Figure 11: Permutation feature importance for classification model with text feature using Light GBM.

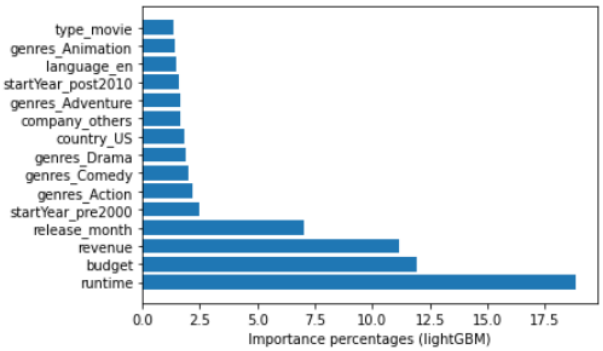


Figure 12: Feature importance for classification model without text feature using lightgbm

Weight	Feature
0.0596 ± 0.0026	genres_Documentary
0.0400 ± 0.0014	runtime
0.0206 ± 0.0031	revenue
0.0187 ± 0.0031	language_en
0.0170 ± 0.0044	genres_Drama
0.0137 ± 0.0056	startYear_pre2000
0.0136 ± 0.0057	budget
0.0135 ± 0.0034	genres_Horror
0.0095 ± 0.0032	country_US
0.0093 ± 0.0037	genres_Action
0.0073 ± 0.0043	startYear_post2010
0.0053 ± 0.0033	genres_Animation
0.0038 ± 0.0026	release_month
0.0035 ± 0.0016	genres_Film-Noir
0.0034 ± 0.0020	type_movie
0.0031 ± 0.0016	country_others
0.0030 ± 0.0027	genres_Thriller
0.0027 ± 0.0009	language_others
0.0024 ± 0.0016	genres_Biography
0.0023 ± 0.0017	company_others
… 76 more …	

Figure 13: Permutation feature importance for classification model without text feature using lightgbm

5.2 Local Interpretation

In local interpretation, we further look into the record with the highest ratings and the lowest ratings.

There are two movies get the highest average rating of 9.5 and we pick one between them.

Firstly, the SHAP value is calculated. SHAP value is used to interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value. The results are shown in *Figure 14*. It can be found out that the runtime, and a genre of documentary contribute a lot, and positively, to its high average rating.

Secondly, the LIME is adopted to try to explain the result generated by a black box. Before putting the sample into the LIME explainer, we should first make some adjustments to our dataset, and to be more detailed, we should have the dataset before we conduct one-hot encoding on the categorical features, since one-hot encoding is not allowed in the lime. However, the feature of genre is an exception, because a movie can be categorized into multiple genres at the same time, and this will not cause meaningless input. The result of LIME is shown in *Figure 15*. Still, it shows that the runtime, and a genre of documentary is very important in helping the movie achieve such a high rating.

Similarly, we also look into the movie with the lowest average rating of 1.1. The SHAP value is shown in *Figure 16*, and the LIME result is shown in *Figure 17*. This time, the results are not that similar. But they both suggest that movie makers should pay more attention to people's reviews, especially the sentiment implied in them.

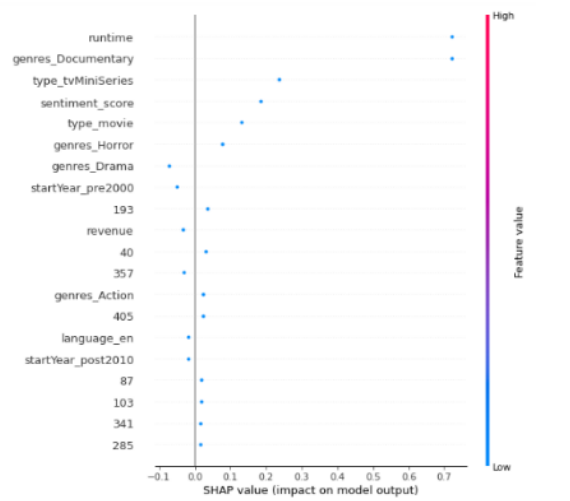


Figure 14. The SHAP value of the movie with the highest rating

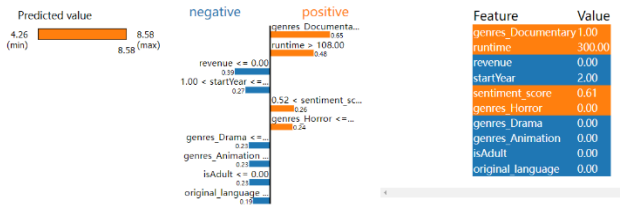


Figure 15. The LIME result of the movie with the highest rating

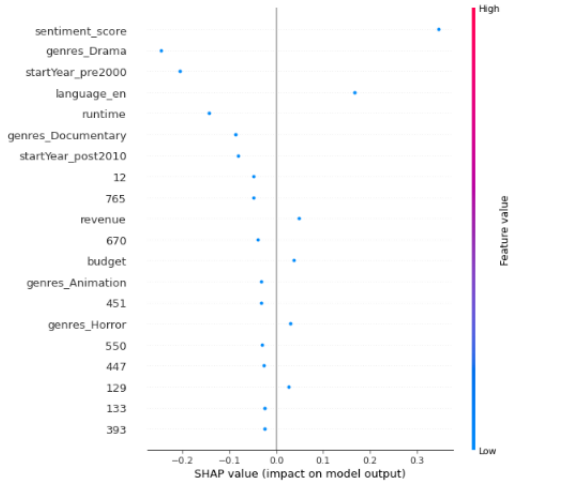


Figure 16. The SHAP value of the movie with the lowest rating

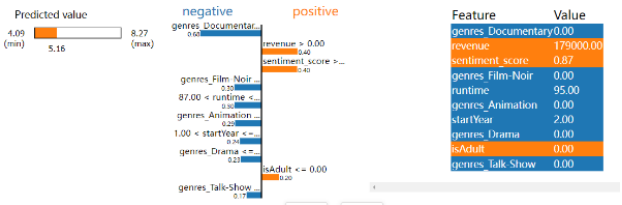


Figure 17. The LIME result of the movie with the lowest rating

6. Image Classification

In the IMDB dataset, there is a feature called *poster_path*, which contains the URLs for poster image of each film. We download poster images from https://image.tmdb.org/, and gather total 5713 poster images from valid URLs.

6.1 Exploratory Data Analysis

Among total 5713 posters, 57% of posters are from movies that have average rating greater than 6.5, and 43% of posters are from movies with average rating less than 6.5.

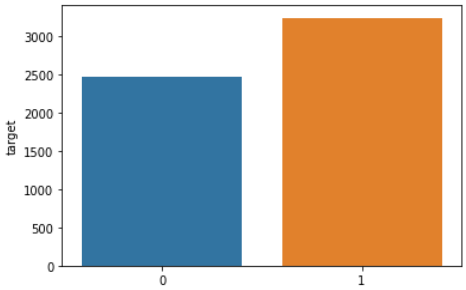


Figure 18. Class balance of poster dataset.

From Word Cloud we can found that most of posters belong to Drama and Comedy, and these movies are most produced in US.

Figure 19 Word Cloud of movie production countries and movie genres in poster dataset.

6.2 Data Pre-processing

In order to balance two classes and save ram, we extract a sub-dataset with 2000 images, 1000 images with positive targets and 1000 with negative targets.

For the target variable, we also treat them as previous classification models that target equals to 1 when average rating greater than or equal to 6.5, and 0 otherwise. For the image matrix, we also normalize pixel number to between 0 and 1 by dividing 255.

After that, we resize all images to (224, 224, 3) for the convenience of deep learning model pipeline, and finally gain a 4D Numpy array of shape (2000, 224, 224, 3).

Then, we use 80% of images as training set and 20% as test set.

6.3 Model Training and Selection & Insights

The models that we use to classify images are two CNNs with different structures, which are shown in **Appendix 4 and 5**. Both models share the same hyperparameters that $batch\_size = 16$ and $epochs = 20$.

First, we try CNN with one Conv2D layer and two Full-connected layers and the model turns out to be a random-guess model on test set, even though the training accuracy is close to 1. It seems that this model encounters an overfitting problem at beginning stage.

Then we try second CNN model with three Conv2D layers and two Full-connected layers. The model also experiences a high level of overfitting, but the test accuracy is a little bit better than the first CNN at 54.75%. As can be seen from the accuracy curve during training process as well as the Class Activation Map from Grad-Cam, the model actually doesn't learn the correct rules for prediction.

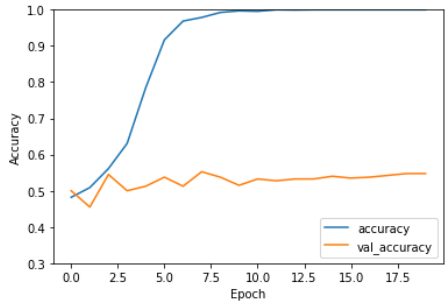


Figure 20. Training / Test accuracy during training process.

Figure 21. Class Activation Map for negative class.

Table 10. Train / Test MSE for classification models.

	Training Accuracy	Test Accuracy
CNN1	99.50%	50.25%
CNN2	100.00%	54.75%

Since the results are dissatisfied, we make final conclusion that movie posters may have little impact on people's rating in IMDB.

7. Conclusion

In this project, we trying to use Machine Learning, Deep Learning and NLP models to understand and predict the movie rating in IMDB. Here are the conclusions that we gain through this project.

(1) For overview feature extraction, features generated from BERT can achieve most accurate results.

(2) In review analysis part, the scores of the sentiment analysis are concentrated at 0.6, so most of the viewers in the sample have a positive evaluation of the movie. The most important concern in positive evaluation is actors' acting performance and storyline.

(3) In rating prediction part, among both regression and classification models, Light GBM can achieve smallest MSE and highest accuracy. Our final prediction model is Light GBM model with inputs of both basic features and text features.

(4) In interpretation part, review sentiment score, revenue, budget and runtime are the most important features that impact average rating.

(5) In the trial of image classification, the results shows that movie poster may has little impact on viewers' rating.

According to the conclusions, we have some strategy suggestions for movie companies that may help to gain a higher rating among viewers.

(1) Before movie comes out, improve movie budget and runtime can help to get higher rating.

(2) After movie comes out, higher review sentiment score and higher revenue can lead to higher rating. Besides, review sentiment score is highly related to actors' acting performance and storyline.

References

Sharda, R., & Delen, D. (2006). Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2), 243-254.

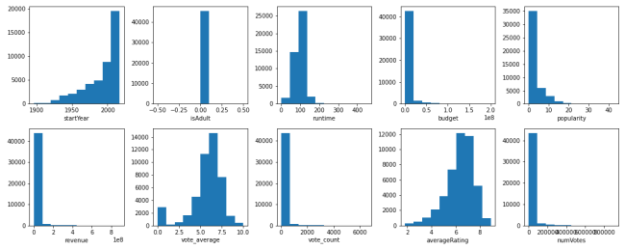
Shangbo, ZHENG Jian ZHOU. "Modeling on box-office revenue prediction of movie based on neural network." *Journal of Computer Applications* 34.3 (2014): 742.

Xuejun, WANG. The Brief Introduction of The History of Box Office Research. Chongqing University, 2015.

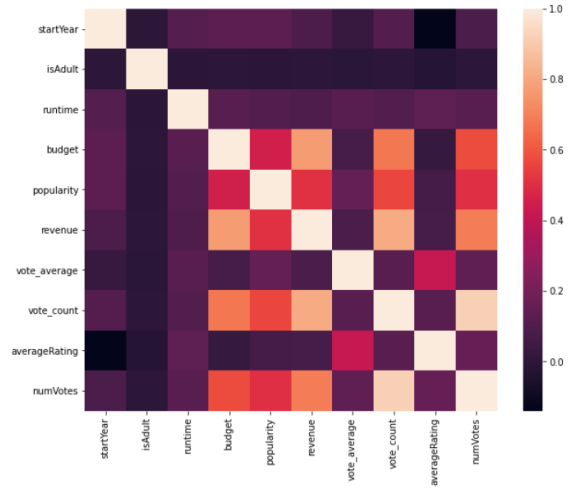
Appendix 1. Descriptive statistics table of numeric variables

	startYear	*runtime*	*popularity*	*Average Rating*
count	45,307	45,307	45,307	45,307
mean	1,991.86	98.10	2.93	6.32
std	24.00	35.34	6.01	1.14
min	1,874.00	0.00	0.00	1.10
25%	1,978.00	87.00	0.39	5.70
50%	2,001.00	96.00	1.13	6.50
75%	2,010.00	108.00	3.69	7.10
max	2,019.00	1,256.00	547.49	9.50

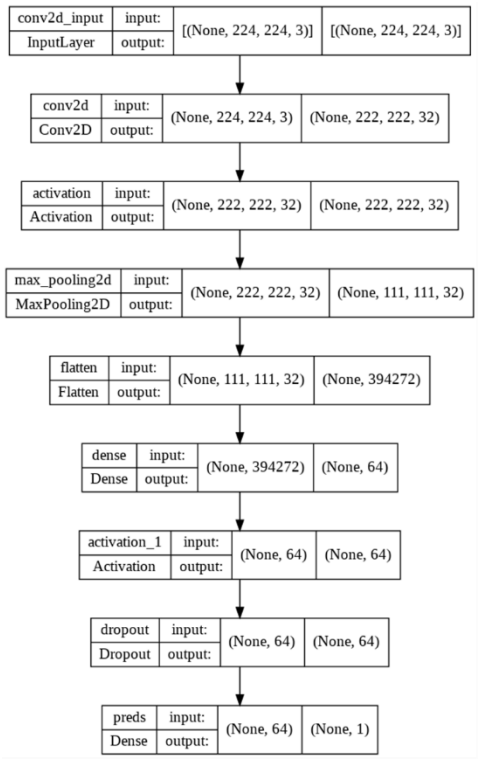
Appendix 2. Figure: histograms of all numeric variables



Appendix 3. Figure: correlation matrix among numeric variables



Appendix 4. Model Structure of CNN1



Appendix 5. Model Structure of CNN2