University of Southern California

ISE 540 Text Analytics

Let's Make a High-rating Movie

— Feature Analysis & Rating Prediction

Team member:

Pengfei Hu, Xiaoxuan Lu, Kexian Chen, Siewying Gong, Junzhuo Gu

Github Public Repo. *(Including README file)***:**

https://github.com/humphreyhuu/IMDb-FilmRating-NLP-Analysis

Data & Script Folder *(Based on Colab Environment)***:**

https://drive.google.com/drive/folders/1SE6hB4P7sIeb6JQVdFaitB58VTHPtYw_?usp=sharing

Medium Blog Post *(For report)*:

https://medium.com/@humphreyhuu/lets-make-a-high-rating-movie-1bdfbe636fec

Content

Executive Summary

Movies are like condiments in our life, we can live without them but it would be more enjoyable if we had them. Generally, there are over 3000 movies released over the world per year and the production company will be profitable if a movie gets 'popular'. However, it is not easy to produce a popular movie. In the current movie market, the highest profit of a movie is around $2 billion while the lowest can go down to $0.1 million. Even though the average profit of a movie is around $14 million, there are still more than 70% of movies that are losing money according to *Entertainment Industry Economics(Vogel).* This implies that there is a huge room for movie production companies to improve the profit. What makes a movie have a high box office? The celebrities, director, or the production company? With the data of past movies, we are trying to identify the important features that build up a popular movie and predict the rating of the upcoming movie.

With the question in mind, we started to gather the information we need to solve the problem. IMDB, one of the largest online databases for the media industry, is our main data source for this project and Wikipedia as a supplement data for names of directors, actors, writers, and production companies. We defined movies with 10,000 votes and had a rating of 5.0+ as popular movies. Since popular movies are usually associated with high ratings and box-office, we decide to set ratings as our 'y' and other possible important features(details in following) as x. We first use TF-IDF, LDA, and Doc2Vec to extract useful information as new features for the descriptive information we scrapped from Wikipedia. Then we fit them into OLS & Linear Regression as our baseline model to see how these important explicit factors influence the film rating. The next step is to compare and select models and feature sets. We choose the two lowest MSE & MAE as our final models and feature sets which are Gradient Boosting and Random Forest as regress, and (Basis + LDA),(Basis + LDA+TF-IDF) as feature sets. The last step is to tune the models to generate the best parameters. After finding the optimal parameters for the models, we use SHAP method (Lundberg, S. M., & Lee, S. I. , 2017) to illustrate the results of our finding. (Details will be talk about starts from next section)

As a result, our best model turned out to be the Gradient Boosting model with an adjusted R-square of 0.92 and 0.5656 for MSE. According to the SHAP method (Lundberg, S. M., & Lee, S. I. , 2017) , we also found that horror and drama movies that originated in the US tend to be more popular while documentary movies had a negative impact on rating in general. Directors that get awarded, also known as producer, and have directed a series of television or films are those who have a higher chance to produce a popular movie. Actors who have played a role in sitcom television, or acted in a series of movies or television are more likely to boost the popularity of the movie. During the past 3 years of pandemic, the movie industry was even in a deeper trough and people also got bored at home. The production companies need to generate profit from movies, and the audiences need movies like sugar to spice up their lives. With our model, the production company can better identify the important factors of the popular movies and produce more popular movies in a convenient way.

Proposed Approach

The objective in this project is to use natural language processing and machine learning models to explicitly discover the important features that lead to success of a movie and how they affect the movies' rating (y label). The flow chart below gives an overview of the methodology used in this project.

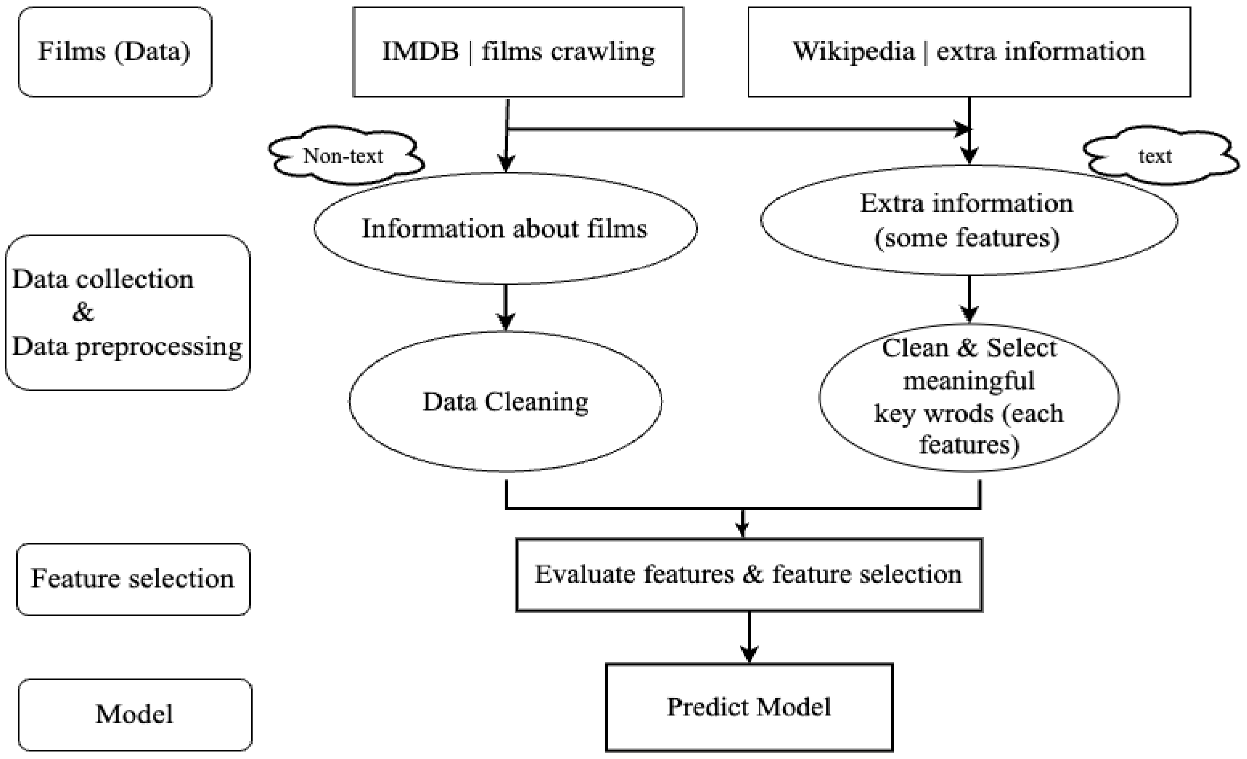

Films (Data)
IMDB | films crawling
Wikipedia | extra information
Non-text
text
Information about films
Extra information (some features)
Data collection & Data preprocessing
Data Cleaning
Clean & Select meaningful key wrods (each features)
Feature selection
Evaluate features & feature selection
Model
Predict Model

Figure. The general flow chart of project

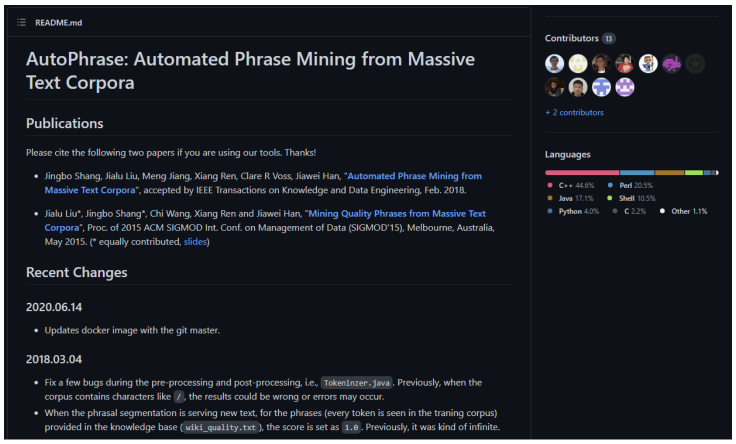
1. Data collection and preparation

Firstly, general information about 10158 films were scraped from **IMDB** (IMDB website as reference shows) as the initial dataset, 30 features in total. In order to eliminate the sampling bias, only films that are in 1900 - 2000 and have more than 10k votes were collected. Next step is to gather extra information about Writers, Directors, Actors and Production Companies from **Wikipedia**. We would find the introductions for each writer, director, actor and product company from their wikipedia pages as additional features in the initial dataset. And we make a mark here. The data in the initial dataset is defined as 'none-text' data and the data of people and companies' introduction is defined as 'text' data in following parts.

After collecting all data, data cleaning would be established. Most of the data cleaning could be done during the data scraping process, but there might still be some needs in fixing typos, unified represent forms and converting into correct data types. For example, we found the released date was stored as an object in the initial raw dataset, which needs to be transformed into datetime and split into year and month for further use. Besides, the genre of the first object is stored as [Drama, History, War], in order to use this kind of feature as a factor in the model, we need to transform them using the **dummy**

method. As for the text data, besides general cleaning like removing punctuation, numbers and stop words, we are supposed to summarize the long context into several keywords.

After reading many segmentation related articles, we decided to apply **Autophrase** (Jingbo Shang et al. 2018) on texting content, which is a predefined API on Github used for extracting phrases with meaning from the text. Autophrase using POS (part of speech) guided phrasal segmentation model and knowledge base to mining meaningful phrases from the text. Then among the phrases ranked by the autophrase, we would select words with high word frequency. We also tried this method on a text, it turns out that the selected words are meaningful ones which could be used as a candidate factor in the future models. See Figure 1 in Appendix A for instance. Therefore, we would apply this method on all Writer, Director, Actor, and Production company text contents.


README.md

AutoPhrase: Automated Phrase Mining from Massive Text Corpora

Publications

Please cite the following two papers if you are using our tools. Thanks!

- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "Automated Phrase Mining from Massive Text Corpora", accepted by IEEE Transactions on Knowledge and Data Engineering, Feb. 2018.
- Jialu Liu*, Jingbo Shang*, Chi Wang, Xiang Ren and Jiawei Han, "Mining Quality Phrases from Massive Text Corpora", Proc. of 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15), Melbourne, Australia, May 2015. (* equally contributed, slides)

Recent Changes

2020.06.14

- Updates docker image with the git master.

2018.03.04

- Fix a few bugs during the pre-processing and post-processing, i.e., `Tokeninzer.java`. Previously, when the corpus contains characters like `/`, the results could be wrong or errors may occur.
- When the phrasal segmentation is serving new text, for the phrases (every token is seen in the traning corpus) provided in the knowledge base (`wiki_quality.txt`), the score is set as `1.0`. Previously, it was kind of infinite.

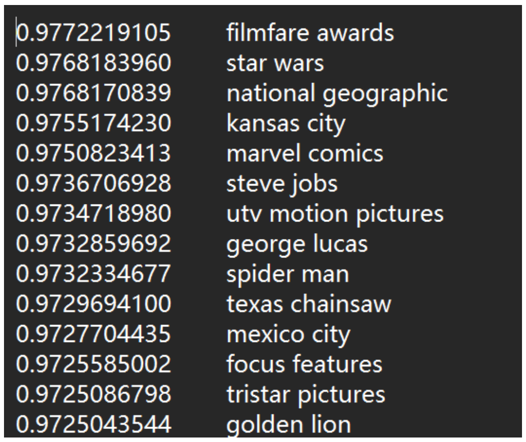

```
0.9772219105    filmfare awards
0.9768183960    star wars
0.9768170839    national geographic
0.9755174230    kansas city
0.9750823413    marvel comics
0.9736706928    steve jobs
0.9734718980    utv motion pictures
0.9732859692    george lucas
0.9732334677    spider man
0.9729694100    texas chainsaw
0.9727704435    mexico city
0.9725585002    focus features
0.9725086798    tristar pictures
0.9725043544    golden lion
```

Figure. AutoPhrase Description and Demo output with keywords + score
(URL: https://github.com/shangjingbo1226/AutoPhrase)

2. Statistical Analysis for feature selection

This step is using statistical tools to figure out whether the features are important for our label. Before statistical analysis, we could filter the features by the number of N/A values and the properties of the features. For instance, the features like 'User_reviews', 'Critic_reviews' would be removed, since they can only be obtained after the movie was released. Then we would apply the **OLS Regression Model** to make statistical analysis. Referring to the experiment result, we could filter all features whose p-values are smaller than the confident threshold 5%. Additionally, the features whose coefficients are not significant, we would regard them as baselines and find other methods to reconvert them as latent features before starting modeling.

3. Modeling

We would make a decision on whether to add new latent factors after analyzing the result of our OLS model. The detailed explanation would be made based on that result in the next part. And if we do need latent variables, we would consider the NLP technologies which would be explained as follows.

For more possible latent features by NLP method, more in detail, we would use **Topics Models (LDA)**, **Tf-idf encoding** method and **BERT-based sentences transformer** to extract our text from statistical and semantic perspectives, which could give us more useful information about text content. With the basic feature set of OLS regression in the previous part, there would be 8 different combinations of features used for each model.

Before starting modeling, the prepared dataset was broken down into 2 parts. The first part is about the Training set and Validation set, with 80% and 20% respectively, they account for a total of 70% of the whole dataset. The second part is the Testing set, which occupied 30% of the dataset.

Then we would test the predictive performance by **Linear Regression** and evaluate the models with their **MSEs** and **MAEs**. There are two possible situations we would meet in this process. The first one is: if the MSE/MAE is low enough, which means X and y are linear-related. In this case, we just need to add more latent features (assume they are needed) to improve the performance of linear regression. The second condition is that MSE/MAE is much higher than the allowable range (like MSE > 1,000), which means X and y might have non-linear relation, we would not only add more features but also change linear models to non-linear models for prediction. The **non-linear regression** we found are Gradient Boosting, Ada Boost, Random Forest, Elastic Net, and Support Vector Regressor.

Besides the modeling, we also want to figure out the importance of each feature and how they affected the y label, in positive direction or the negative. The first method is getting the parameters from the model which fit best directly. The other method we consider to use is concluding with a combined result which consists with the results from all models we used. In this case, we did research and decided to use the **SHAP Value** method (Lundberg, S. M., & Lee, S. I. , 2017) to describe these feature sets.

Detailed solution and experimental result is shown in the Experimental Result section. We would compare the metrics of the different models on the Testing set, the one with the lowest MSE was selected as the best.

Experimental Result

This part is mainly separated into two different points.

The first one is about feature selection by different explicit factors extracted from film's info and meaningful keyword extraction, and we use statistical inference method like OLS linear regression (provided by scipy.stats packages) to take a closer look for coefficient and testing value (Chi-square, F, p-value etc.) by each variables. We will explain how these important explicit factors have effects on film rating and demonstrate why the text info (meaningful keyword) is useful here according to the result.

Then we will focus on how to predict new launching films by quantitative or machine learning models, and we will solve some predictive problems like how to choose the suitable feature sets, how to evaluate models for prediction and related optimal configuration of hyperparameters by MSE. Here, we will add more latent factors encoded by BERT-based transformers, Topics Modeling (LDA) and TF-IDF encoding methods to get better predictions. Finally, we will decide the final model for prediction.

1. Feature Selection and Baseline Model

1.1 Feature selection I

After data cleaning and considering various factors for each column, we decided to use the factor shown in table 1 in Appendix B.

1.2 Statistic analysis : Linear Regression (OLS)

As we planned, we did ordinary least squares (OLS) on the features we selected before. Referring to the p-values for each column, it is not surprising to see that one or more values of each column (dummy columns) we selected contribute to the y label, as their p-values are smaller than the confident threshold 5%. Figure 2 in Appendix A shows part of the statistical result. The dummy variables show the importance of Genre, Country of origin, Language, Month of release date, director, actor. Because of space limitation, other dummy columns are not shown here. Also from the coefficients in result, we also got information about how features affect y labels. For example, the genre_Biography with a positive coefficient, means if the movie is about biography, it is likely to be a success. While the coefficient of genre_Action is negative, we can say the action movie might fail. Among the features list, we also found that the coefficient of director and actor are really small although their p values are smaller than 0.05. It means that they might be useful but not optimal as we expected. In order to get the well performed model, we might add other text related variables later. So now, we decided to regard these features as the baseline, the basis for our prediction model.

2. Data Modelling with Prediction

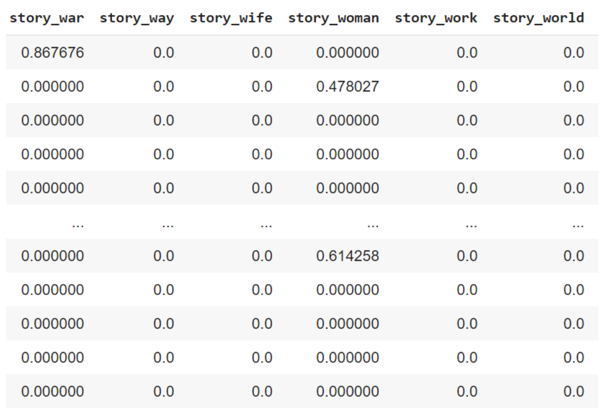
The target of prediction is to predict the film rating for each launching film, so the target variable (y) is 'Film Rating' here. As explained in the previous part, we would use the features from the last part as baselines, latent factors would be obtained byTF-IDF encoding, Topics Similarity and Doc2Vec..

2.1 More latent factors

We could get latent factors extracted from text by three different technologies. Here let's take a closer look at each method.

(a) TF-IDF - Encoding Features

We did TF-IDF encoding methods for synopsis of film's storyline (other than genre type), actor's description, director's description, writer's description and production company's description respectively. Before fitting the vectorizer, we mainly extracted the N. and NV. entities (through POS tagging) here and tokenized texts, and then we set the allowable number of features equals to 2000 and allowed the gram size for wordbags from 1 to 3 grams for fitting. Then we could get the TF-IDF feature vectors for each document and attribute. Here is a demo table displaying the final result of encoding.



story_war	story_way	story_wife	story_woman	story_work	story_world
0.867676	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.478027	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...
0.000000	0.0	0.0	0.614258	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0
0.000000	0.0	0.0	0.000000	0.0	0.0

Figure. Some result got from TF-IDF

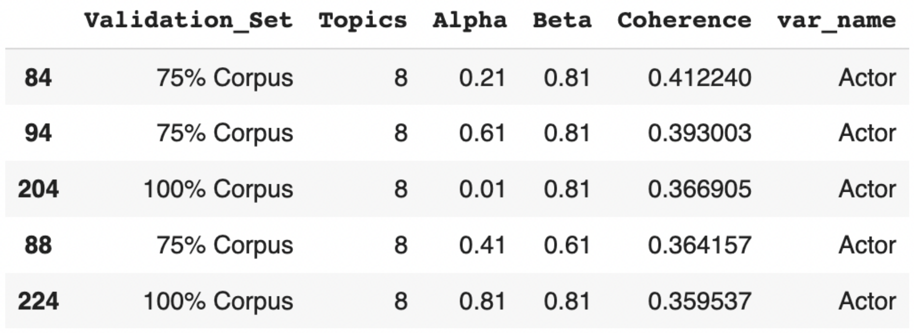
(b) Topic Modeling (LDA) - Topics Similarity Features

We constructed Latent Dirichlet Allocation (LDA) topic models here for film's actors, directors, writers and production companies respectively to get topics similarity for latent factors. For LDA, we completed it within three different steps. The first thing was that we needed to construct baseline models, then we needed to find optimal numbers of topics for each entity based on some quantitative metrics and our domain knowledge. Finally, we can configure the optimal parameter for LDA fitting text of each entity.

Here we took actor's text as an example:

- **Metric Selection:** We used C_v as our choice of metric for performance comparison. C_v measure is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity.
- **Hyperparameter Tuning:** We used a self-defined function which is similar to GridSearchCV to find C_v results for different combinations of hyperparameters. We could find the top 2-3 best configuration for text according to average C_v by # of topics. Then we could fit it and find out whether it can actually split topics clearly from semantic perspectives and our domain knowledge.
- **Get Final Topics Model & Topics Similarity:** Finally, we could get the optimal LDA model and related topics similarity for each film's actors, which can be considered as latent features for prediction.

Here is the part of tuning result displayed by top 5 configuration for actor's text:



	Validation_Set	Topics	Alpha	Beta	Coherence	var_name
84	75% Corpus	8	0.21	0.81	0.412240	Actor
94	75% Corpus	8	0.61	0.81	0.393003	Actor
204	100% Corpus	8	0.01	0.81	0.366905	Actor
88	75% Corpus	8	0.41	0.61	0.364157	Actor
224	100% Corpus	8	0.81	0.81	0.359537	Actor

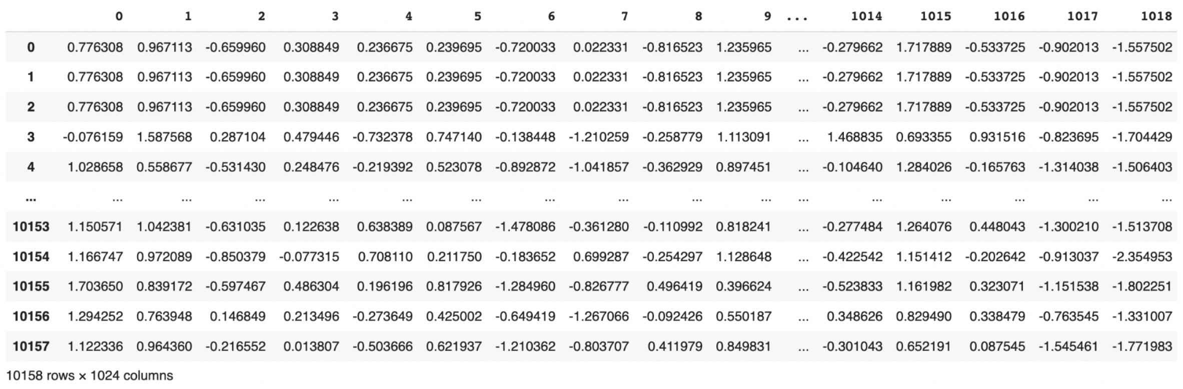
Figure. Hyperparameter Tuning Rank for Top 5 combination

Finally, we constructed LDA topic models with 8, 6, 4, 6 Topics for film's actors, directors, writers and production companies respectively, and we calculated the topics similarity for each film and each entity to get our desired latent features!

(c) Sentences Embedding - Doc2Vec Features

Here we used the BERT-based Transformer from "Hugging Face" websites to do this. This transformer can help us convert document's (film-related) sentences into vectors, which means the output vector can measure the sentence's meaning from semantic perspectives. Therefore, after embedding the text for each film and each entity (like actor, director etc.), we could get latent features with D * V shape. D here means the number of films, V here means the 1024 different semantic meanings.

After embedding descriptive text for each film's actors, directors, writers and production companies, we could get 4 * 1024 latent features finally. Here is the demo output of final transformation of actor's text:



	0	1	2	3	4	5	6	7	8	9	...	1014	1015	1016	1017	1018
0	0.776308	0.967113	-0.659960	0.308849	0.236675	0.239695	-0.720033	0.022331	-0.816523	1.235965	...	-0.279662	1.717889	-0.533725	-0.902013	-1.557502
1	0.776308	0.967113	-0.659960	0.308849	0.236675	0.239695	-0.720033	0.022331	-0.816523	1.235965	...	-0.279662	1.717889	-0.533725	-0.902013	-1.557502
2	0.776308	0.967113	-0.659960	0.308849	0.236675	0.239695	-0.720033	0.022331	-0.816523	1.235965	...	-0.279662	1.717889	-0.533725	-0.902013	-1.557502
3	-0.076159	1.587568	0.287104	0.479446	-0.732378	0.747140	-0.138448	-1.210259	-0.258779	1.113091	...	1.468835	0.693355	0.931516	-0.823695	-1.704429
4	1.028658	0.558677	-0.531430	0.248476	-0.219392	0.523078	-0.892872	-1.041857	-0.362929	0.897451	...	-0.104640	1.284026	-0.165763	-1.314038	-1.506403
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10153	1.150571	1.042381	-0.631035	0.122638	0.638389	0.087567	-1.478086	-0.361280	-0.110992	0.818241	...	-0.277484	1.264076	0.448043	-1.300210	-1.513708
10154	1.166747	0.972089	-0.850379	-0.077315	0.708110	0.211750	-0.183652	0.699287	-0.254297	1.128648	...	-0.422542	1.151412	-0.202642	-0.913037	-2.354953
10155	1.703650	0.839172	-0.597467	0.486304	0.196196	0.817926	-1.284960	-0.826777	0.496419	0.396624	...	-0.523833	1.161982	0.323071	-1.151538	-1.802251
10156	1.294252	0.763948	0.146849	0.213496	-0.273649	0.425002	-0.649419	-1.267066	-0.092426	0.550187	...	0.348626	0.829490	0.338479	-0.763545	-1.331007
10157	1.122336	0.964360	-0.216552	0.013807	-0.503666	0.621937	-1.210362	-0.803707	0.411979	0.849831	...	-0.301043	0.652191	0.087545	-1.545461	-1.771983

10158 rows × 1024 columns

Figure. Sentence Embedding Result of Actor's Text by Transformer

2.2 Model Evaluation & Selection

(a) Evaluation Methodology

Evaluation Metrics (For Regressor):

- For **Training**: (Adjusted) R-square;
- For **Validating & Testing**: MSE (*Negative MSE for GridSearchCV*) & MAE.

Evaluation Benchmark: Notice that when we compare and select models or feature sets, we mainly focus on MSE and MAE here, then we will choose the top 2 regressors and top 2 feature sets for the next step, and try to tune hyperparameters to get the optimal solution by lowest MSE. After deciding the final model, besides MSE & MAE, we would also check the adjusted R-square for it.

Dataset Splitting: 70% Training set: (including 20% validation set with 5-Fold CV) & 30% Testing set

(b) Regressor & Features Selection

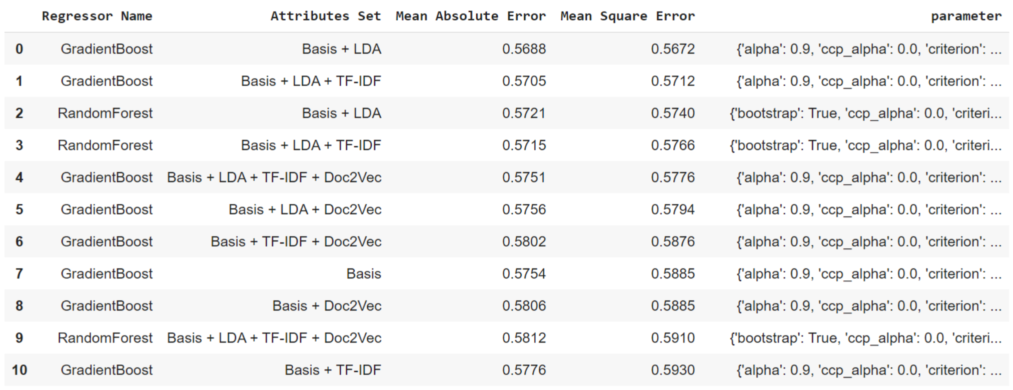
Regressor Candidates [totally 6 models]:

- Linear Regressor [Baseline]: OLS & Linear Regression
- Non-Linear Regressor: Gradient Boosting Regressor, Ada Boosting Regressor, Random Forest Regressor, Elastic Net and Support Vector Regressor.

Feature Set Candidates [totally 8 options]:

- Baseline: Basis feature set we got from previous part;
- Other: {Basis + TF-IDF}, {Basis + LDA}, {Basis + Doc2Vec}, {Basis + LDA + TF-IDF}, {Basis + LDA + Doc2Vec}, {Basis + TF-IDF + Doc2Vec}, {Basis + LDA + TF-IDF + Doc2Vec}

Therefore, there are **6 * 8 = 48** different combinations we can select for prediction, let's use testing MSE & MAE score to compare them and select candidates which might be more likely suitable for prediction! Here is the top 10 ranking result for these 48 situations by MSE.



	Regressor Name	Attributes Set	Mean Absolute Error	Mean Square Error	parameter
0	GradientBoost	Basis + LDA	0.5688	0.5672	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
1	GradientBoost	Basis + LDA + TF-IDF	0.5705	0.5712	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
2	RandomForest	Basis + LDA	0.5721	0.5740	{'bootstrap': True, 'ccp_alpha': 0.0, 'criteri...
3	RandomForest	Basis + LDA + TF-IDF	0.5715	0.5766	{'bootstrap': True, 'ccp_alpha': 0.0, 'criteri...
4	GradientBoost	Basis + LDA + TF-IDF + Doc2Vec	0.5751	0.5776	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
5	GradientBoost	Basis + LDA + Doc2Vec	0.5756	0.5794	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
6	GradientBoost	Basis + TF-IDF + Doc2Vec	0.5802	0.5876	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
7	GradientBoost	Basis	0.5754	0.5885	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
8	GradientBoost	Basis + Doc2Vec	0.5806	0.5885	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...
9	RandomForest	Basis + LDA + TF-IDF + Doc2Vec	0.5812	0.5910	{'bootstrap': True, 'ccp_alpha': 0.0, 'criteri...
10	GradientBoost	Basis + TF-IDF	0.5776	0.5930	{'alpha': 0.9, 'ccp_alpha': 0.0, 'criterion': ...

Figure. Top 10 best combination ranked by MSE

According to the rank, we can find that top 2 choices for models and feature sets:

- **Top 2 regressor:**
  - Gradient Boosting Regressor
  - Random Forest Regressor
- **Top 2 feature set:**
  - Basis + LDA
  - Basis + LDA + TF-IDF

2.3 Hyperparameter Tuning

Then we could do hyperparameter tuning based on the previous part to find the best model with optimal parameter configuration. Here we used the GridSearch method with 5-Fold CV for those two regressors and two feature sets, and we used **Negative MSE** as the core metric to rank different situations.

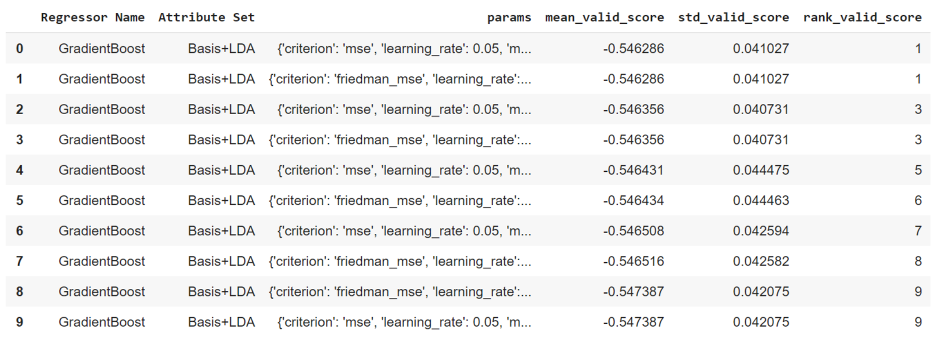
For **Gradient Boosting Regressor**, the parameter set for comparison is:

gradientboost_parameters = { 'learning_rate': [0.05, 0.1, 0.2, 0.5], 'n_estimators': [50, 100, 200], 'criterion': ['friedman_mse', 'mse'], 'min_samples_split': [2, 5, 10], 'max_depth': [3, 5]}

For **Random Forest Regressor**, the parameter set for comparison is:

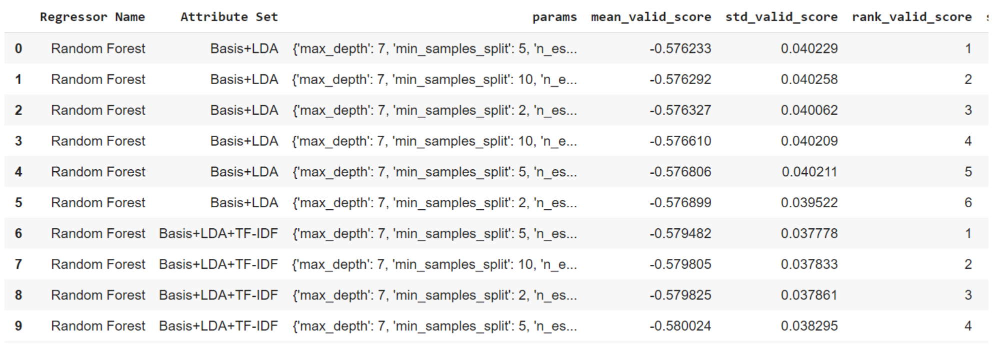
randomforest_parameters = { 'n_estimators': [50, 100, 200], 'criterion': ['squared_error', 'absolute_error'], 'max_depth': [3, 5, 7], 'max_features': ['sqrt', 'log2', 'auto'], 'min_samples_split': [2, 5, 10]}

Then we build these two regressors with different parameters to fit two feature sets {Basis + LDA} and {Basis + LDA + TF-IDF}. Here is the Top 10 rank for hyperparameter tuning results for Gradient Boosting and Random Forest Regressor respectively.



	Regressor Name	Attribute Set	params	mean_valid_score	std_valid_score	rank_valid_score
0	GradientBoost	Basis+LDA	{'criterion': 'mse', 'learning_rate': 0.05, 'm...	-0.546286	0.041027	1
1	GradientBoost	Basis+LDA	{'criterion': 'friedman_mse', 'learning_rate':...	-0.546286	0.041027	1
2	GradientBoost	Basis+LDA	{'criterion': 'mse', 'learning_rate': 0.05, 'm...	-0.546356	0.040731	3
3	GradientBoost	Basis+LDA	{'criterion': 'friedman_mse', 'learning_rate':...	-0.546356	0.040731	3
4	GradientBoost	Basis+LDA	{'criterion': 'mse', 'learning_rate': 0.05, 'm...	-0.546431	0.044475	5
5	GradientBoost	Basis+LDA	{'criterion': 'friedman_mse', 'learning_rate':...	-0.546434	0.044463	6
6	GradientBoost	Basis+LDA	{'criterion': 'mse', 'learning_rate': 0.05, 'm...	-0.546508	0.042594	7
7	GradientBoost	Basis+LDA	{'criterion': 'friedman_mse', 'learning_rate':...	-0.546516	0.042582	8
8	GradientBoost	Basis+LDA	{'criterion': 'friedman_mse', 'learning_rate':...	-0.547387	0.042075	9
9	GradientBoost	Basis+LDA	{'criterion': 'mse', 'learning_rate': 0.05, 'm...	-0.547387	0.042075	9

Figure. Top 10 Rank by Negative MSE among Gradient Boosting



	Regressor Name	Attribute Set	params	mean_valid_score	std_valid_score	rank_valid_score
0	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 5, 'n_es...	-0.576233	0.040229	1
1	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 10, 'n_e...	-0.576292	0.040258	2
2	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 2, 'n_es...	-0.576327	0.040062	3
3	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 10, 'n_e...	-0.576610	0.040209	4
4	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 5, 'n_es...	-0.576806	0.040211	5
5	Random Forest	Basis+LDA	{'max_depth': 7, 'min_samples_split': 2, 'n_es...	-0.576899	0.039522	6
6	Random Forest	Basis+LDA+TF-IDF	{'max_depth': 7, 'min_samples_split': 5, 'n_es...	-0.579482	0.037778	1
7	Random Forest	Basis+LDA+TF-IDF	{'max_depth': 7, 'min_samples_split': 10, 'n_e...	-0.579805	0.037833	2
8	Random Forest	Basis+LDA+TF-IDF	{'max_depth': 7, 'min_samples_split': 2, 'n_es...	-0.579825	0.037861	3
9	Random Forest	Basis+LDA+TF-IDF	{'max_depth': 7, 'min_samples_split': 5, 'n_es...	-0.580024	0.038295	4

Figure. Top 10 Rank by Negative MSE among Random Forest

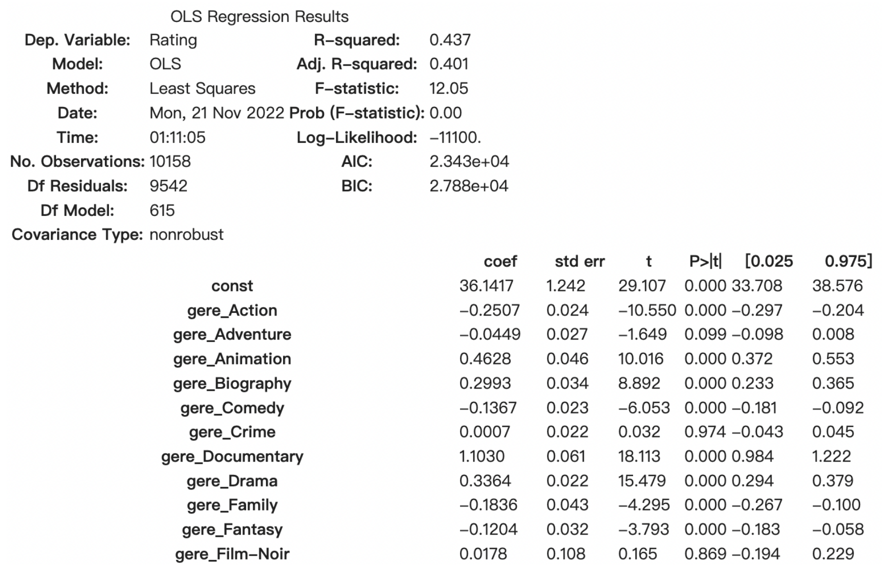
According to results, we could easily find that the Negative MSE (equivalent to MSE) of Gradient Boosting Regressor is significantly better than Random Forest. Therefore, the best model is Gradient Boosting Regressor fitting with [Basis+LDA] feature set and related optimal parameter is:

{'max_depth': 7, 'min_samples_split': 5, 'n_estimators': 200, 'criterion': 'squared_error', 'max_feature': 'auto'}

Discussion

1. Important Features

After Experimentation, several insights were generated. First of all, for the baseline model, although some features are important due to their low p-values, they might not perform well in prediction as we expected because of their low coefficients according to the OLS model implemented. Looking at the coefficients of each feature that generated from OLS models could give us a better understanding of this.



```
                          OLS Regression Results
    Dep. Variable: Rating                 R-squared:  0.437
            Model: OLS               Adj. R-squared:  0.401
           Method: Least Squares          F-statistic:  12.05
             Date: Mon, 21 Nov 2022 Prob (F-statistic): 0.00
             Time: 01:11:05          Log-Likelihood:  -11100.
 No. Observations: 10158                        AIC:  2.343e+04
     Df Residuals: 9542                         BIC:  2.788e+04
         Df Model: 615
  Covariance Type: nonrobust
                              coef     std err   t       P>|t|  [0.025   0.975]
        const                 36.1417  1.242     29.107  0.000  33.708   38.576
     gere_Action              -0.2507  0.024     -10.550 0.000  -0.297   -0.204
    gere_Adventure            -0.0449  0.027     -1.649  0.099  -0.098   0.008
    gere_Animation            0.4628   0.046     10.016  0.000  0.372    0.553
    gere_Biography            0.2993   0.034     8.892   0.000  0.233    0.365
     gere_Comedy              -0.1367  0.023     -6.053  0.000  -0.181   -0.092
      gere_Crime              0.0007   0.022     0.032   0.974  -0.043   0.045
   gere_Documentary           1.1030   0.061     18.113  0.000  0.984    1.222
      gere_Drama              0.3364   0.022     15.479  0.000  0.294    0.379
     gere_Family              -0.1836  0.043     -4.295  0.000  -0.267   -0.100
     gere_Fantasy             -0.1204  0.032     -3.793  0.000  -0.183   -0.058
    gere_Film-Noir            0.0178   0.108     0.165   0.869  -0.194   0.229
```

Therefore, before modeling, other texts factorize methods to be applied to the text content. We used TF-IDf, LDA and Doc2Vec to generate the latent variables for text separately. Then we combined them as different inputs to fit models. After fitting the best model, we can get feature importances for each variable. We decided to use the SHAP value to describe these feature sets, since it can show the positive or negative contribution or importance for each feature.

Positives related features:

- Gere_Drama: Genre of movie is drama
- OC_Uniter States: Original Country is US
- Actor_5: LDA actor topic #5
- Gere_Horror: Genre of movie is horror
- Director_5: LDA director topic #5

Negative related features:

- Year: Publish year
- Gere_Documentary: Genre is documentary
- IG_English: Language is English
- Director_6: LDA director topic #6

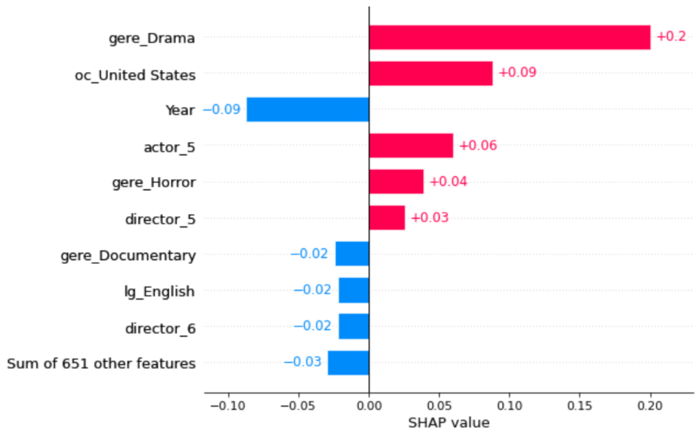


Figure. SHAP Value for top important features

2. Model Summary

We use our latest optimal predictive model to compare with the baseline model again. After several experiments, we came to the conclusion below:

Baseline model [OLS Linear Regression]

- For training: Adjusted R-square: 0.40;
- For testing: MSE: 2.79*e+18;

Best Non-linear model [Gradient Boosting]

- For training: Adjusted R-square: 0.92;
- For testing: MSE: 0.5656;

Looking at the statistics above, after experimenting and comparing result with diffent models, we found that Adding new latent factors (LDA Similarity) and using a non-linear model (Gradient Boosting) can increase the predicting performance significantly. The Gradient Boosting model has a significantly higher R-square and MSE than the baseline model. In order to evaluate whether the model works well, we used the model to predict several films and compare the result to their actual rating on IMDB. Below is a table of the predicted rating and the actual rating of the films:

Movie Name	Predicted Rating	Actual Rating
The Fourth Kind	5.9008	5.9
Naked Lunch	6.9009	6.9
The Swan Princess	6.4014	6.4
Insidious: The Last Key	5.6984	5.7
In the House	7.3982	7.4

As a result, our model predicted most of the film accurately, but there might be some differences due to bias.

Conclusion

To sum up the project. We scraped film data from IMDB with the size of 10 thousands. Then we scraped more detailed text content from Wikipedia to enrich the movies' features. After cleaning non-text data and phrase the text data, we applied statistical analysis to filter features. For better mining features and improving the performance, we used latent features by NLP technologies, such as tf-idf vectorization, LDA and Doc2Vec, to improve the performance of models and make the results of our project more convincing and expansive. Finally, we compared the metrics of 6 different regressors with 8 different sets of features. Our best predict model is Gradient Boosting regressor fitting "Base Features + LDA" features and its MSE is 0.5656. Based on it, the objective of our project was achieved.

Our project gave the views about what features are significant to the movies' rating scores and how they affect the ratings. From our result, the top significant features which are positive to ratings are drama and horror genres, original country as US, actors with characteristic in topic#5 and director in topic#5. Other features are with negative affects, the tops are the publish year (as year grows, the rating decreasing, see Appendix A Figure5), documentary genre, English language and director in top#6.

These features showed the audience preference and expectation for movies. Based on the result, the conclusion could be that, the anticipation for movies increasing year by year, audiences expect to see movies which could bring an immersive experience with high quality drama or horror content and English language is no longer the criterion for good movies and et.al. Using these fresh criteria for reference, it would be possible to predict whether a movie would get success before it is released. Besides, the product teams or companies could make a good movie by improving the positive features of the new movie.

Future Work:

Our project solved the problem as expected and could give some suggestions and views to predict the success of an unreleased movie. While there is still something that could be improved in the future.

On the data side, the information from IMDB may not be really objective or professional. The ratings are possible intentionally high or low. To exact the real information, researchers could scrape film information on two or more websites and combine them. Then there might be more available features to play with and the result would be more general. On the modeling side, we only did regression models for rating. While there are other interesting methods to apply. Researchers can group the ratings into high-rating, medium-rating, low-rating. Applying multiple classification models on them and figure out the critical factors for each group.

Lessons learned:

When we were trying to scrape data from IMDB and other movie rate websites, we realized that external data scraping is limited. For example, by the fact that we can't look up some specific information about a director or actor on a wiki which could be useful in our model, such as whether they've won any obscure film awards. So we decided to extend our data by scraping more detailed features from Wikipedia, and connect them with our IMDB data.

In the step of choosing important features, we used OLS for deciding features into our model. However, these features do not fit well in our model. We initially only adjusted these features to observe the fit of the model but the result did not improve significantly. After discussion within the group, we added latent features such as LDA into our model, finally improving the performance of our model.

Reference:

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "Automated Phrase Mining from Massive Text Corpora", accepted by IEEE Transactions on Knowledge and Data Engineering, Feb. 2018.

Kapadia, Shashank. “Evaluate Topic Models: Latent Dirichlet Allocation (LDA).” Evaluate Topic Models: Latent Dirichlet Allocation (LDA), Towards Data Science, https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda. Accessed 20 Nov. 2022.

Kelechava, Marc. Using LDA Topic Models as a Classification Model Input. https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28.

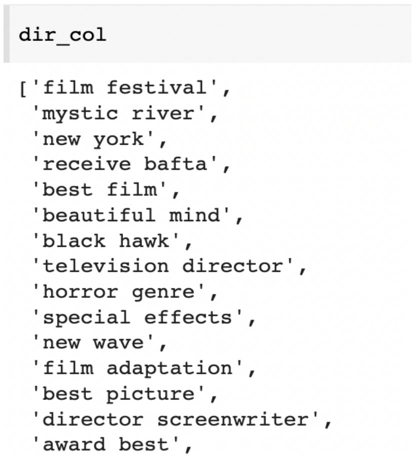
Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

“Sentence-Transformers/NLI-Roberta-Large · Hugging Face.”Sentence-Transformers/Nli-Roberta-Large · Hugging Face, https://huggingface.co/sentence-transformers/nli-roberta-large .

“Sort by Popularity - Most Popular Feature Films - IMDb.” IMDb, www.imdb.com/search/keyword/?title_type=movie Accessed 20 Nov. 2022.

“Unigram Tokenization - Hugging Face Course.” Unigram Tokenization - Hugging Face, https://huggingface.co/course/chapter6/7?fw=pt Accessed 20 Nov. 2022.

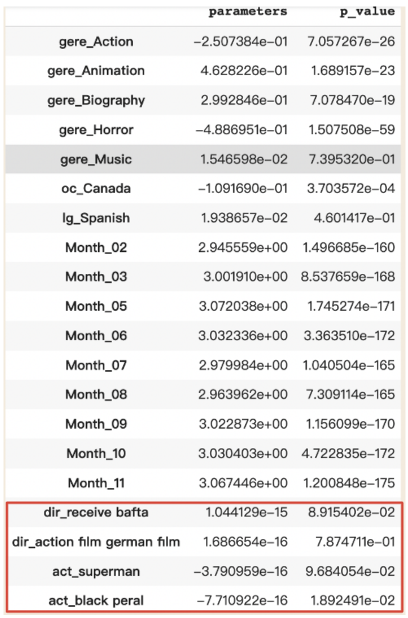
Appendix A



```
dir_col
```

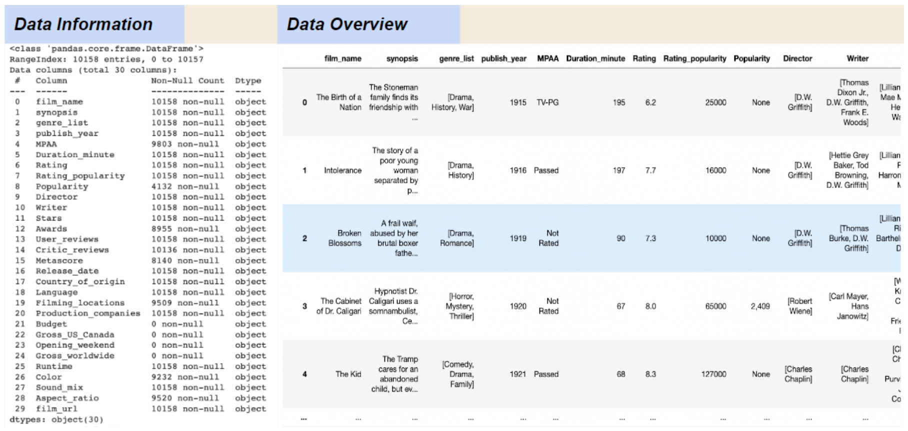
```
['film festival',
 'mystic river',
 'new york',
 'receive bafta',
 'best film',
 'beautiful mind',
 'black hawk',
 'television director',
 'horror genre',
 'special effects',
 'new wave',
 'film adaptation',
 'best picture',
 'director screenwriter',
 'award best',
```

Figure 1



	parameters	p_value
gere_Action	-2.507384e-01	7.057267e-26
gere_Animation	4.628226e-01	1.689157e-23
gere_Biography	2.992846e-01	7.078470e-19
gere_Horror	-4.886951e-01	1.507508e-59
gere_Music	1.546598e-02	7.395320e-01
oc_Canada	-1.091690e-01	3.703572e-04
lg_Spanish	1.938657e-02	4.601417e-01
Month_02	2.945559e+00	1.496685e-160
Month_03	3.001910e+00	8.537659e-168
Month_05	3.072038e+00	1.745274e-171
Month_06	3.032336e+00	3.363510e-172
Month_07	2.979984e+00	1.040504e-165
Month_08	2.963962e+00	7.309114e-165
Month_09	3.022873e+00	1.156099e-170
Month_10	3.030403e+00	4.722835e-172
Month_11	3.067446e+00	1.200848e-175
dir_receive bafta	1.044129e-15	8.915402e-02
dir_action film german film	1.686654e-16	7.874711e-01
act_superman	-3.790959e-16	9.684054e-02
act_black peral	-7.710922e-16	1.892491e-02

Figure 2



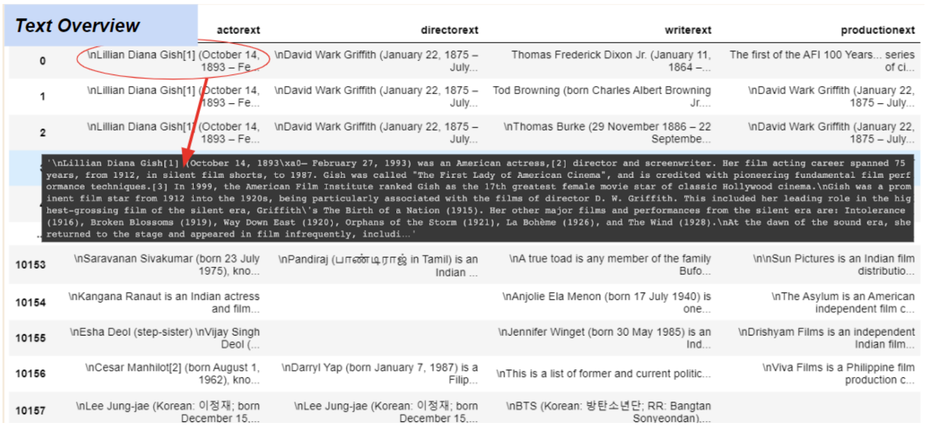
Data Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10158 entries, 0 to 10157
Data columns (total 30 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   film_name             10158 non-null  object
 1   synopsis              10158 non-null  object
 2   genre_list            10158 non-null  object
 3   publish_year          10158 non-null  object
 4   MPAA                  9803 non-null   object
 5   Duration_minute       10158 non-null  object
 6   Rating                10158 non-null  object
 7   Rating_popularity     10158 non-null  object
 8   Popularity            4132 non-null   object
 9   Director              10158 non-null  object
 10  Writer                10158 non-null  object
 11  Stars                 10158 non-null  object
 12  Awards                8955 non-null   object
 13  User_reviews          10158 non-null  object
 14  Critic_reviews        10136 non-null  object
 15  Metascore             8140 non-null   object
 16  Release_date          10158 non-null  object
 17  Country_of_origin     10158 non-null  object
 18  Language              10158 non-null  object
 19  Filming_locations     9509 non-null   object
 20  Production_companies  10158 non-null  object
 21  Budget                0 non-null      object
 22  Gross_US_Canada       0 non-null      object
 23  Opening_weekend       0 non-null      object
 24  Gross_worldwide       0 non-null      object
 25  Runtime               10158 non-null  object
 26  Color                 9232 non-null   object
 27  Sound_mix             10158 non-null  object
 28  Aspect_ratio          9520 non-null   object
 29  film_url              10158 non-null  object
dtypes: object(30)
```

Data Overview

	film_name	synopsis	genre_list	publish_year	MPAA	Duration_minute	Rating	Rating_popularity	Popularity	Director	Writer	
0	The Birth of a Nation	The Stoneman family finds its friendship with ...	[Drama, History, War]	1915	TV-PG	195	6.2	25000	None	[D.W. Griffith]	[Thomas Dixon Jr., D.W. Griffith, Frank E. Woods]	[Lillian Mae M He Wa
1	Intolerance	The story of a poor young woman separated by p...	[Drama, History]	1916	Passed	197	7.7	16000	None	[D.W. Griffith]	[Hettie Grey Baker, Tod Browning, D.W. Griffith]	[Lillian F Harron M
2	Broken Blossoms	A frail waif, abused by her brutal boxer fathe...	[Drama, Romance]	1919	Not Rated	90	7.3	10000	None	[D.W. Griffith]	[Thomas Burke, D.W. Griffith]	[Lillian Ri Barthel D
3	The Cabinet of Dr. Caligari	Hypnotist Dr. Caligari uses a somnambulist, Ce...	[Horror, Mystery, Thriller]	1920	Not Rated	67	8.0	65000	2,409	[Robert Wiene]	[Carl Mayer, Hans Janowitz]	[W K C Frie
4	The Kid	The Tramp cares for an abandoned child, but ev...	[Comedy, Drama, Family]	1921	Passed	68	8.3	127000	None	[Charles Chaplin]	[Charles Chaplin]	[Cl Ch Purv Co
...	...	...	...	...	...	...	...	...	...	...	...	

Figure 3



Text Overview

	actorext	directorext	writerext	productionext
0	\nLillian Diana Gish[1] (October 14, 1893 – Fe...	\nDavid Wark Griffith (January 22, 1875 – July...	Thomas Frederick Dixon Jr. (January 11, 1864 –...	The first of the AFI 100 Years... series of ci...
1	\nLillian Diana Gish[1] (October 14, 1893 – Fe...	\nDavid Wark Griffith (January 22, 1875 – July...	Tod Browning (born Charles Albert Browning Jr....	\nDavid Wark Griffith (January 22, 1875 – July...
2	\nLillian Diana Gish[1] (October 14, 1893 – Fe...	\nDavid Wark Griffith (January 22, 1875 – July...	\nThomas Burke (29 November 1886 – 22 Septembe...	\nDavid Wark Griffith (January 22, 1875 – July...
10153	\nSaravanan Sivakumar (born 23 July 1975), kno...	\nPandiraj (பாண்டிராஜ் in Tamil) is an Indian ...	\nA true toad is any member of the family Bufo...	\n\nSun Pictures is an Indian film distributio...
10154	\nKangana Ranaut is an Indian actress and film...		\nAnjolie Ela Menon (born 17 July 1940) is one...	\nThe Asylum is an American independent film c...
10155	\nEsha Deol (step-sister) \nVijay Singh Deol (...		\nJennifer Winget (born 30 May 1985) is an Ind...	\nDrishyam Films is an independent Indian film...
10156	\nCesar Manhilot[2] (born August 1, 1962), kno...	\nDarryl Yap (born January 7, 1987) is a Filip...	\nThis is a list of former and current politic...	\nViva Films is a Philippine film production c...
10157	\nLee Jung-jae (Korean: 이정재; born December 15,...	\nLee Jung-jae (Korean: 이정재; born December 15,...	\nBTS (Korean: 방탄소년단; RR: Bangtan Sonyeondan),...	

```
'\nLillian Diana Gish[1] (October 14, 1893\xa0– February 27, 1993) was an American actress,[2] director and screenwriter. Her film acting career spanned 75
years, from 1912, in silent film shorts, to 1987. Gish was called "The First Lady of American Cinema", and is credited with pioneering fundamental film perf
ormance techniques.[3] In 1999, the American Film Institute ranked Gish as the 17th greatest female movie star of classic Hollywood cinema.\nGish was a prom
inent film star from 1912 into the 1920s, being particularly associated with the films of director D. W. Griffith. This included her leading role in the hig
hest-grossing film of the silent era, Griffith\'s The Birth of a Nation (1915). Her other major films and performances from the silent era are: Intolerance
(1916), Broken Blossoms (1919), Way Down East (1920), Orphans of the Storm (1921), La Bohème (1926), and The Wind (1928).\nAt the dawn of the sound era, she
returned to the stage and appeared in film infrequently, includi...'
```

Figure 4

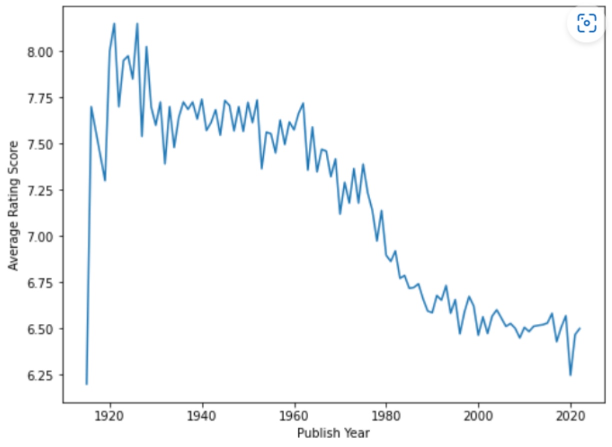


Figure 5

Appendix B

X	***Non-text features***	Genre, MPAA, Duration(minutes), Release_date, Country_of_origin, Language, Sound_mix, Release_location
	Text features	Director, Writer, Stars, Production_companies (the wiki-content taken in the previous step)
y	Rating	