GROUP 15

MOVIE SCHEDULE OPTIMIZATION USING SENTIMENTAL ANALYSIS OF MOVIE REVIEWS

TEAM MEMBERS

Cao, Jiali A0232321Y
Huang, Hai-Hsin A0231906J
Tsao, Kai-Ting A0231947Y
Mingxuan Yang A0231854E

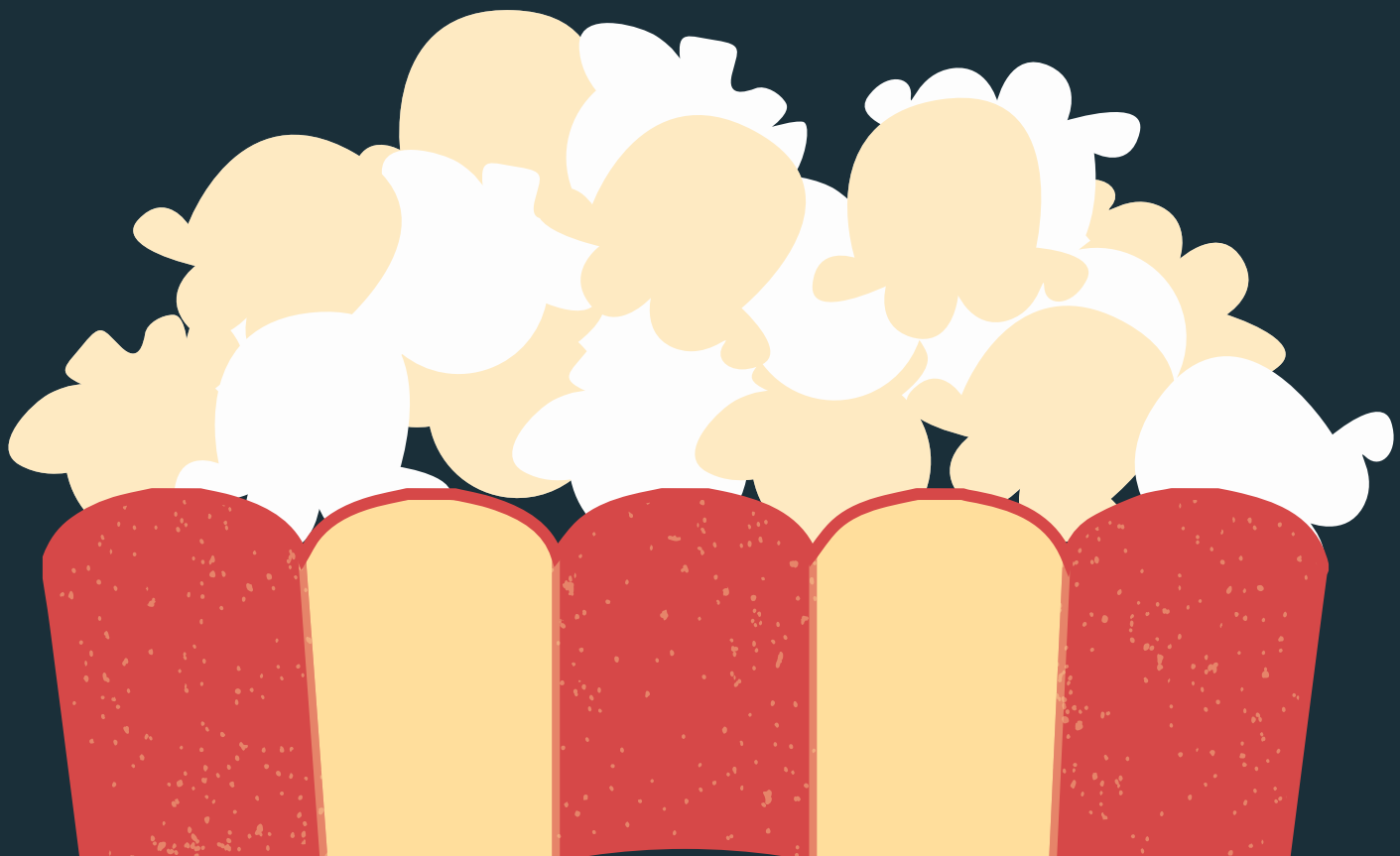

Table of Contents

Ⅰ Introduction (Motivation)

The majority of cinema income is from screening new release films. A hit film results in high ticket income and high concession sales, and vice versa. Therefore, film arrangement is a significantly essential task for cinemas. Traditionally, theaters would predict a high box office for a movie with famous producers, popular cast and high budget and allocate more screenings to these kinds of movies. This is also how audiences made their decision in the past. However, with the spread of SNS and online critics commentaries, people tend to search for comments on the Internet before buying a ticket. As a result, people's comments become more and more important to affect the revenue of a movie.

Ⅱ Problem Statement and Project Objective

Firstly, we would like to understand whether ranking on movie rating websites will affect potential audiences' willingness to watch the movie. If a movie has a high rank, it will attract more audiences to go to the theaters to watch this movie.

However, for newly released movies, the website should wait for days to conclude a rating, and cinemas apparently could not wait. They need to conclude how the first batch of audiences react on SNS and use this information to arrange their screenings.

So, the next step is to predict the rating. We decided to formulate a model to analyze audiences' sentiment by their text comments on the Internet. In this way, cinemas can easily gain a quick conclusion from people's comments and make decisions.

Ⅲ Methodology

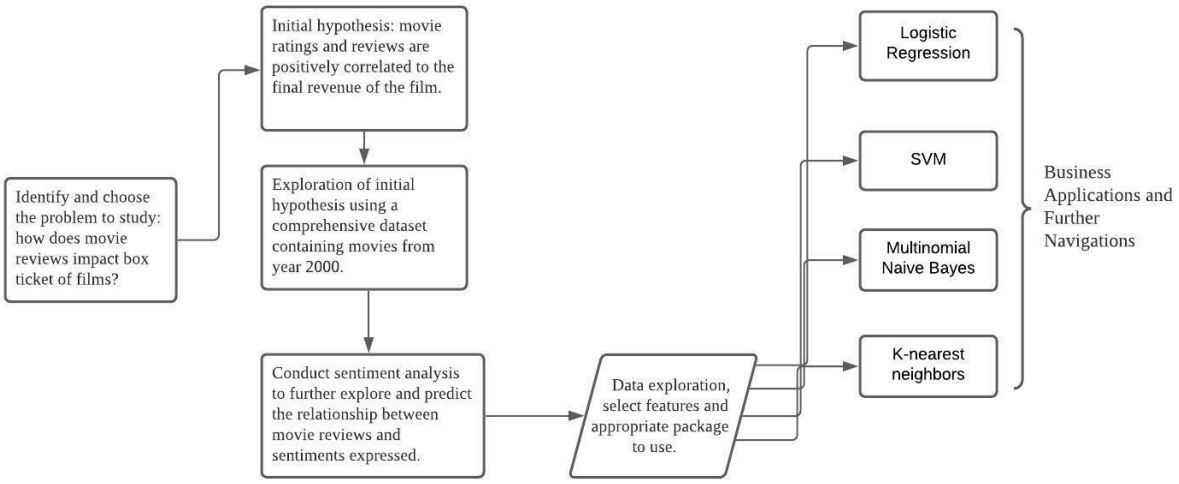

Figure 1: Workflow for the project

We firstly apply linear regression to gain the effects of Internet rating on movie revenue. Then we use TextBlob to explore the polarity and subjectivity of movie review comments. After that we evaluate the plain accuracy and confusion matrix of four models: Logistic Regression, SVM, Naive Bayes and K-nearest Neighbors Classifier, and choose Naive Bayes as our final model.

1. Relationship between Internet Rating and Movie Revenue

Introduction

To confirm our hypothesis that Internet comments are affecting the revenue of movies, we did a simple linear regression.

Dataset

Data Source:	TMDB and GroupLens
Time span:	2000-2017
Dependent variable:	revenue which measures movie box office
Independent variable:	vote_average which measures rating on the Internet
Control variables:	budget, genre, year, runtime and country
Data Size:	2561 rows

Table 1 Selected Regression result:

Model:	OLS	Adj.	R-squared:	0.501		
	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
genre[T.Comedy]	0.2335	0.0879	2.6556	0.008	0.0611	0.4059
genre[T.Crime]	-0.4547	0.1385	-3.2837	0.001	-0.7263	-0.1832
genre[T.Documentary]	1.4129	0.3891	3.6311	0.0003	0.6499	2.1759
genre[T.Drama]	-0.256	0.0889	-2.879	0.004	-0.4303	-0.0816
genre[T.Horror]	0.5615	0.1247	4.5018	0	0.3169	0.8061
genre[T.Western]	-1.1926	0.4647	-2.5663	0.0103	-2.1039	-0.2813
country[T.CH]	-1.4606	0.7036	-2.076	0.038	-2.8403	-0.0809
country[T.QA]	-2.777	1.4066	-1.9742	0.0485	-5.5353	-0.0187
country[T.ZA]	-2.0669	0.922	-2.2419	0.0251	-3.8748	-0.259
np.log(budget/1000)	0.9781	0.0266	36.7803	0	0.9259	1.0302
vote_average	0.4747	0.0393	12.0941	0	0.3977	0.5517

OLS Regression Result

The coefficient of vote_average is 0.4747 and p value is 0, which means vote_average has a significant positive correlation with movies' revenue. The result matches our hypothesis and we can move to the next step.

2. Sentimental Analysis using TextBlob

Introduction

We wanted to know whether the movie comments are consistent with the review scores, so we calculated the polarity scores to see if comments with positive scores are given higher review scores. Moreover, we calculated the subjectivity score to see if the subjectivity score is higher when the absolute value of the polarity score is high.

Dataset

Data source: Rotten tomatoes review
Time span: 2011 - 2020 (10 years range)
Data size: 59,498 rows
Data processing: clean up rows with null values

Package

We use TextBlob to analyze the polarity and subjectivity of movie review comments.

Sentimental Analysis Criteria

- Polarity ~ [-1,1], 0 is neutral, +1 is positive, -1 is negative.
- Subjectivity ~ [0,1], 0 is objective and 1 is subjective.

Feature engineering

Table 2 Features and feature description

Feature	Feature description
rotten_tomatoes_link	Movie identifier, to distinguish different movies.
top_critic	To identify if the comment is on the top of all critics.
review_type	This is calculated from review_rank. If the review_rank is better than or equal to B - , then it is classified as “Fresh”, while those with a review_rank of C + or below are classified as “Rotten”.
review_rank	The review_rank is the score that each reviewer gives to a movie, including A, A-, B+,B, B-, C+, C, C-, D+, D, D-, F.
review_content	Review_content is the comment given for each movie.
review_score	We calculated the review score from the review rank. The review score ranges from 12 to 1 and corresponds to the review rank A to F.
sentiment	We encoded the sentiment score from review_type. If the review type is “Fresh”, then the sentiment is 1. If the review type is “Rotten”, then the sentiment is 0.
polarity	We calculated the polarity score from review_content using the package TextBlob. The polarity score ranges from -1 to +1. -1 indicates negative, +1 indicates positive, and 0 indicates neutral.
subjectivity	We calculated the subjectivity score from review_content using the package Textblob. The subjectivity score ranges from 0 to 1, with 0 indicating that the comment is very objective and 1 indicating that the comment is very subjective.

Data Encoding

Encode review_rank to numeric values
Encode review_type to dummy variables

rotten_tomatoes_link	top_critic	review_type	review_rank	review_content	review_score	sentiment	polarity	subjectivity
m/0814255	False	Rotten	D+	The premise of Percy Jackson & the Olympians: ...	4	0	0.266667	0.916667
m/0878835	False	Fresh	A-	Funny at times, Please Give is a surprisingly ...	11	1	0.316667	0.700000
m/1000013-12_angry_men	False	Fresh	B	Lumet keeps things tense, sweaty, suspenseful ...	9	1	-0.083333	0.737500
m/1000013-12_angry_men	False	Fresh	A	This movie is a masterpiece. That term gets th...	12	1	0.620667	0.636000
m/1000079-20000_leagues_under_the_sea	False	Fresh	B+	A good one that could have been better with a ...	10	1	0.269167	0.433333

EDA and Preliminary Results

This scatter plot illustrates the polarity score distribution within each review score. As you can see, there are more data points that have a positive polarity score in higher review scores and there are less data points that have a positive polarity score in lower review scores. For example, At review score 12, which corresponds to review score A in the original dataset, we see there are comparatively less data points that have a negative polarity score.

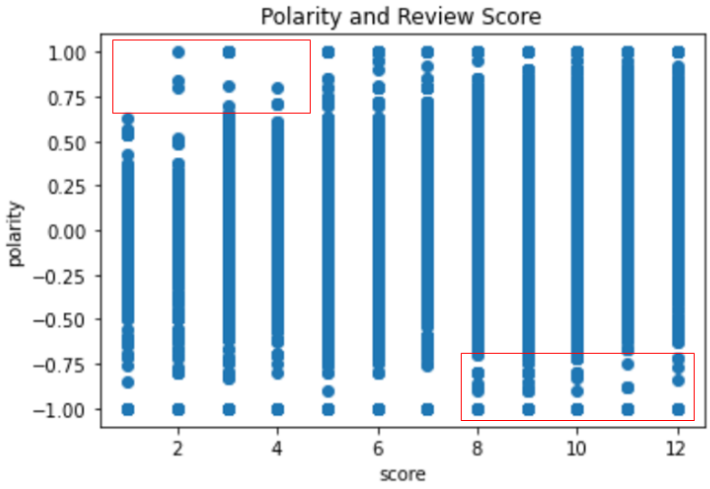

Figure 2. Polarity and Review Score

Furthermore, we compute the average polarity score in each review score range. Interestingly, the higher the review score, the higher the polarity score. We concluded that the polarity score, which is calculated from the review comments, and review score, which is ranked initially by commenters, have a positive correlation.

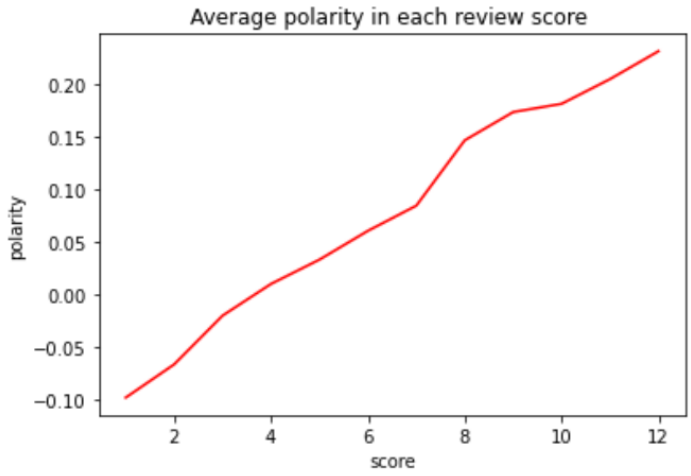

Figure 3. Average polarity in each review score

We also wanted to know if the polarity score calculated from movie comments has a significant effect on review score, so we ran an OLS regression model. The polarity score has a positive coefficient and a 0.0000 p-value, meaning that the polarity score has a significant effect on the review score and they are positively correlated.

Table 3 Regression model result

Model:	OLS	Adj. R-squared:	0.711
Dependent Variable:	score	AIC:	205112.6471
Date:	2021-11-17 18:48	BIC:	205148.6219
No. Observations:	59498	Log-Likelihood:	-1.0255e+05
Df Model:	3	F-statistic:	4.869e+04
Df Residuals:	59494	Prob (F-statistic):	0.00
R-squared:	0.711	Scale:	1.8394

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.1997	0.0100	521.0694	0.0000	5.1801	5.2192
polarity	0.6180	0.0209	29.5853	0.0000	0.5771	0.6589
sentiment	4.3658	0.0121	361.6722	0.0000	4.3421	4.3895
top_critic_dummy	0.0038	0.0137	0.2780	0.7810	-0.0231	0.0307

Omnibus:	749.086	Durbin-Watson:	1.617
Prob(Omnibus):	0.000	Jarque-Bera (JB):	740.345
Skew:	-0.252	Prob(JB):	0.000
Kurtosis:	2.789	Condition No.:	5

We also plot all the data points using their polarity score and subjectivity score calculated from the review comments. We concluded that more polar comments tend to be more subjective.

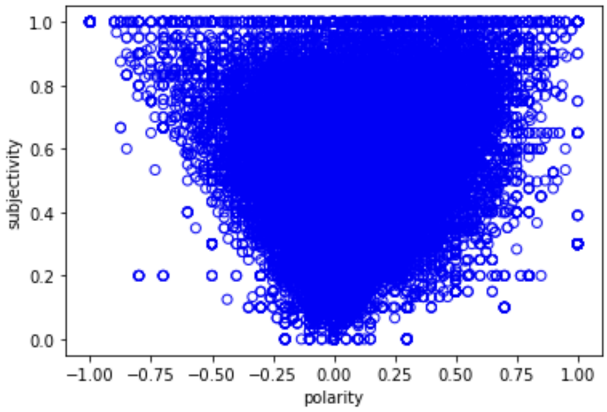

Figure 4. Polarity Score and Subjectivity Score

In this graph, we filtered out the comments that are too subjective (subjectivity score > 0.8) and the movies that have only one comment. Therefore, we have all the data points that have a subjectivity score <= 0.8 and have 2 or more review comments for each movie. Then, we calculated the polarity score of the review comments for each movie and plot against the review score. We saw a slightly positive correlation between the average polarity score for each movie and review scores.

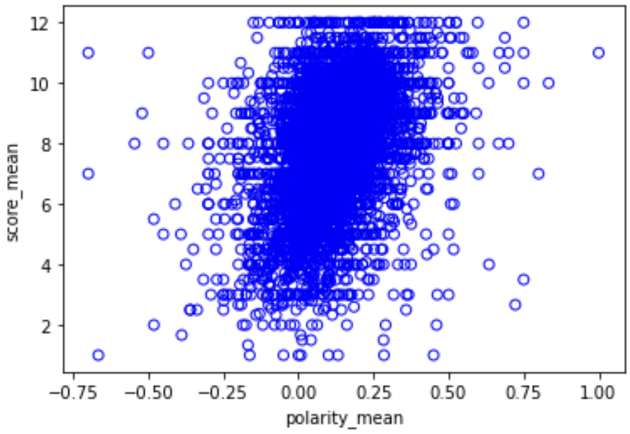

Figure 5. Polarity_mean and Score_mean

3. Predict movie sentiments with classification models

Introduction

Using the same dataset as sentimental analysis (Rotten tomatoes review), we ran several classification models to see if we can use movie comments to predict the sentiments of viewers. Sentiments are calculated from review type. If the movie has a review score greater than or equal to 9 (B-), then the

movie has a sentiment of 1, else 0. We input “movie comments” as our independent variables and used “sentiment” as our dependent variable.

Data Preprocessing method

How can we input review contents as independent variables? We first tokenized the sentences into single words. Second, we took out symbols and words without specific positive or negative meaning, such as “with”, “should”, or “these” from the review content. Then, we applied two algorithms: count vectorizer and tf-idf vectorizer, to convert review_content into vectors (meaningful representation of numbers) that can be put into our classification models.

The count vectorizer & the tf-idf vectorizer

The count vectorizer considers the frequencies of words in a sentence, while tf-idf vectorizer considers both the frequencies a word appears in a sentence and the number of sentences the word appears in.

Classification models’ results

We ran 4 classification models, including Logistic Regression, SVM, Multinomial Naive Bayes, and K-nearest neighbors classifier. The accuracy rates calculated from the count vectorizer (BOW) generally have a better performance. Overall, the Multinomial Naive Bayes model has a comparatively higher accuracy rate, so we chose it as our final model.

Table 4 Classification models’ result

Accuracy	**Count vectorizer for bag of words (BOW)**		**Tfidf vectorizer**	
	Test accuracy rate	**Training accuracy rate**	**Test accuracy rate**	**Training accuracy rate**
Logistic Regression	0.6479	0.9322	0.6458	0.6576
SVM	0.6458	0.8618	0.6458	0.6577
Naïve Bayes (Multinomial NB)	0.6568	0.9338	0.6473	0.9338
K-nearest neighbors classifier	0.6461	0.6577	0.6458	0.6577

Confusion Matrix

In the test dataset, there are 17,498 data points, with 11,300 true positive sentiments and 6,198 true negative sentiments. The data is unbalanced, with more positive sentiments (about 64.58% are positive sentiments). Therefore, we further looked at the confusion matrix on the models that are initially input with count-vectorizer vectors. In SVM, K-nearest neighbors classifier, and Logistic Regression models, almost all the sentiments are predicted as positive (sentiment = 1), and only the Multinomial Naïve Bayes model has a slightly more accurate prediction with 16356 positive and 1142 negative sentiment predictions.

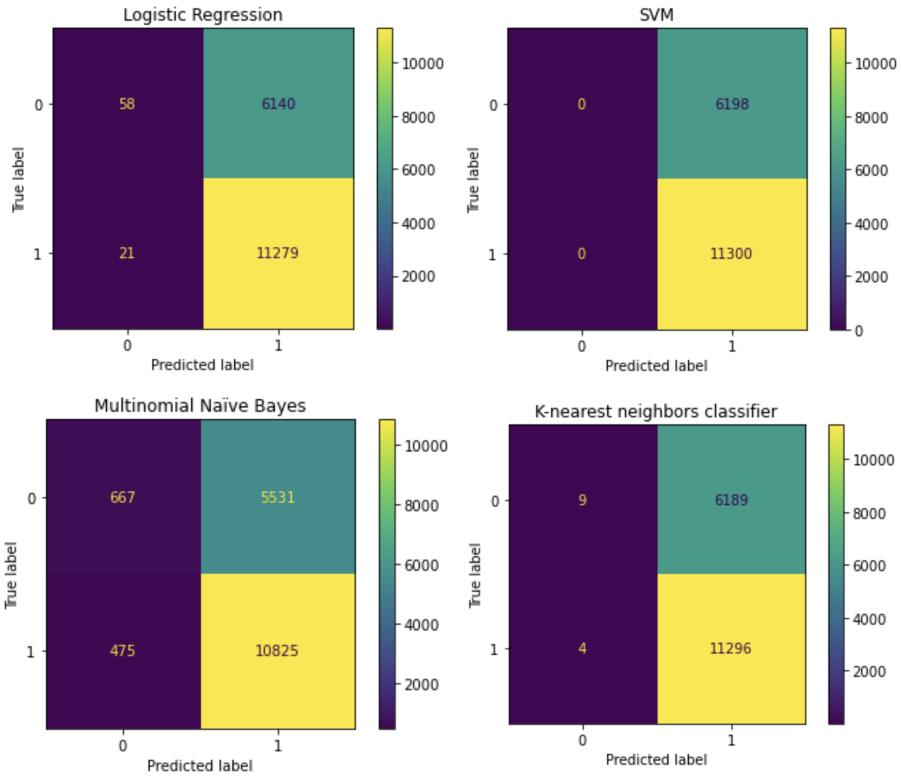

Figure 6. Confusion Matrix

Word Cloud

Moreover, we tried to understand the most frequent positive or negative review words used by reviewers. We used the “WordCloud” package and the result is shown in Figure 7 and 8.

Figure 7. Positive Review Words

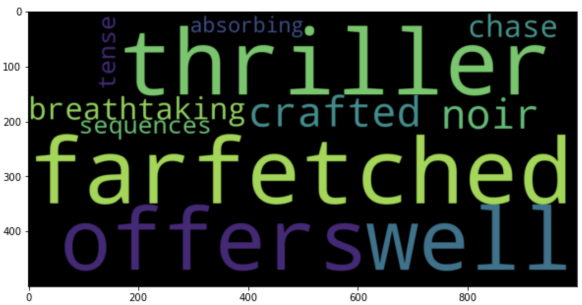

Figure 8. Negative Review Words

Ⅳ Business Application

In a real life setting, film investors and stakeholders often want to make the cinematic arrangements such as ticket prices, ticket availability, and most importantly, number of showtimes and locations of different cinemas the movie will be primarily aired in. It is crucial for them to understand the market demand for the film they invested in in order to maximize profit, and a common way to do that is through premieres and test screenings. We believe that stakeholders and investors could utilize the viewers' comments from the premieres and test runs, and perform sentiment analysis on them to extract the initial responses. In order to arrive at better decisions, we could combine viewers' demographics with their comments' sentiment analysis, and figure out which groups respond the best to the film, and which ones detest it or find it somewhat offensive. This way investors of the film could choose to make smarter decisions such as not airing it too much, air it in specific states where the demographics are mostly likely warm to the content, air it during times where the demographics would visit the cinemas the most, etc. The editors of the film could even conduct some further editing and blurring, to make sure that the movies are welcome to the wanted target audience, and therefore generate the most revenue out of each film.

Ⅴ Conclusion and Further Navigations

In conclusion, we have determined that the Multinomial Naive Bayes model has the most accurate predictions in comparison with the other models in the setting of predicting sentiments with movie comments. This could be used as a preliminary analytical tool for stakeholders to predict the gross box office of the film, and therefore better arrange the screening times and locations.

However, there exists a few areas that could be navigated further. To start off, we ruled out comments that we considered as "too subjective" and "too polarized", but there are films that have such a strong and unique characteristic that only appeal to a certain crowd. Some comments might love it and some might detest it, but both need to be included to arrive at an accurate and relevant prediction. This accuracy issue might also occur as we tokenized the words. We could easily break down a sentence into words that didn't convey its original implication. For example, a "huge mistake if you don't watch it over 100 times" comment would give the wrong idea. This could be further improved if we can incorporate more phrases or short sentences in the model to start with, and therefore interpreting a wider range of sentiments.

Another important issue to consider is that, during the initial analysis of our data, we recognized that sentiments are in values of only 0 and 1, that is, identifying emotions that are purely positive and purely negative. We could consider that words and emotions have different levels of positivity and negativity, and only considering a comment section's count of words and counts of sentences in which they appear could throw us off quite a bit. To further improve the accuracy rate of our classification models, we can add a more refined score to reflect comment's emotions, and eventually produce a movie classification that is more than just "good" and "bad".

Ⅶ Reference

[1] Rotten Tomatoes movies and critic reviews dataset
https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset?select=rotten_tomatoes_movies.csv

[2] The Movies Dataset
https://www.kaggle.com/rounakbanik/the-movies-dataset

[3] Python – Text Classification using Bag-of-words Model
https://vitalflux.com/text-classification-bag-of-words-model-python-sklearn/#:~:text=CountVectorizer%20%28sklearn.feature_extraction.text.CountVectorizer%29%20is%20used%20to%20fit%20the%20bag-or-words.text%20data%2C%20which%20can%20be%20documents%20or%20sentences.

[4] TF-IDF Vectorizer scikit-learn
https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a