Predict IMDB Rating Class with Machine Learning Algorithms

Jill Han

06/15/2020

IMDb: Ratings, Reviews, and Where to ...

imdb.com

1. Introduction

1.1 Data Description

This dataset contains the movies' basic information, online reviews, IMDB rating scores as well from the IMDb database. It contains 5043 observations and 28 variables. Those movies are released from 65 countries between 1916 and 2016.

1.2 Project Objective

Although IMDb rating is not absolutely accurate, it can be considered as a useful and informative tool to evaluate whether a film is successful. Which type of films are intended to be successful? What factors are crucial for a movie to get a higher IMDb rating score? The answers are going to be found by analyzing the variables of this dataset.

In this project, We are working on how to predict imdb_score. In reality, the levels of the score is more reasonable and interpretable. For example, the movies are always considered excellent when their imdb_score higher than 8. So the scores are classified into 4 levels and imdb_class is considered as the response variable. Several prediction models are built with different algorithms (Logistic, Random Forest Classifier, XGBoost Classifier, CNN)and are evaluated by the confusion matrix. The optimized model can successfully predict the quality of the movie with important features. It will be helpful for the film companies to get the trick of success of films.

2. Exploratory Data Analysis

2.1 Data Profile

2.1.1 Data Variables and Types

- Number of observations: 5043
- Number of features: 28
- Variable types
- Convert the object columns into categorical.

Data Types

Numeric	16
Object	12

Object Variables

#	Column	Type
0	color	object
1	director_name	object
2	actor_2_name	object
3	genres	object
4	actor_1_name	object
5	movie_title	object
6	actor_3_name	object
7	plot_keywords	object
8	movie_imdb_link	object
9	language	object
10	country	object
11	content_rating	object
Dtypes	Object (12)	

Numerical Variables

#	Column	Type
0	num_critic_for_reviews	float64
1	duration	float64
2	director_facebook_likes	float64
3	actor_3_facebook_likes	float64
4	actor_1_facebook_likes	float64
5	gross	float64
6	facenumber_in_poster	float64
7	num_user_for_reviews	float64
8	budget	float64
9	title_year	float64
10	actor_2_facebook_likes	float64
11	imdb_score	float64
12	aspect_ratio	float64
13	num_voted_users	int64
14	cast_total_facebook_likes	int64
15	movie_facebook_likes	int64
Dtypes	Numerical (16)	

2.1.2 Descriptive Statistics

Descriptive Statistics of Numerical Variables

	count	mean	std	min	25%	50%	75%	max
num_critic_for_reviews	4993.0	1.401943e+02	1.216017e+02	1.00	50.00	110.00	195.00	8.130000e+02
duration	5028.0	1.072011e+02	2.519744e+01	7.00	93.00	103.00	118.00	5.110000e+02
director_facebook_likes	4939.0	6.865092e+02	2.813329e+03	0.00	7.00	49.00	194.50	2.300000e+04
actor_3_facebook_likes	5020.0	6.450098e+02	1.665042e+03	0.00	133.00	371.50	636.00	2.300000e+04
actor_1_facebook_likes	5036.0	6.560047e+03	1.502076e+04	0.00	614.00	988.00	11000.00	6.400000e+05
gross	4159.0	4.846841e+07	6.845299e+07	162.00	5340987.50	25517500.00	62309437.50	7.605058e+08
num_voted_users	5043.0	8.366816e+04	1.384853e+05	5.00	8593.50	34359.00	96309.00	1.689764e+06
cast_total_facebook_likes	5043.0	9.699064e+03	1.816380e+04	0.00	1411.00	3090.00	13756.50	6.567300e+05
facenumber_in_poster	5030.0	1.371173e+00	2.013576e+00	0.00	0.00	1.00	2.00	4.300000e+01
num_user_for_reviews	5022.0	2.727708e+02	3.779829e+02	1.00	65.00	156.00	326.00	5.060000e+03
budget	4551.0	3.975262e+07	2.061149e+08	218.00	6000000.00	20000000.00	45000000.00	1.221550e+10
title_year	4935.0	2.002471e+03	1.247460e+01	1916.00	1999.00	2005.00	2011.00	2.016000e+03
actor_2_facebook_likes	5030.0	1.651754e+03	4.042439e+03	0.00	281.00	595.00	918.00	1.370000e+05
imdb_score	5043.0	6.442138e+00	1.125116e+00	1.60	5.80	6.60	7.20	9.500000e+00
aspect_ratio	4714.0	2.220403e+00	1.385113e+00	1.18	1.85	2.35	2.35	1.600000e+01
movie_facebook_likes	5043.0	7.525965e+03	1.932045e+04	0.00	0.00	166.00	3000.00	3.490000e+05

Descriptive Statistics of Numerical Variables

	count	unique	top	freq
color	5024	2	Color	4815
director_name	4939	2398	Steven Spielberg	26
actor_2_name	5030	3032	Morgan Freeman	20
genres	5043	914	Drama	236
actor_1_name	5036	2097	Robert De Niro	49
movie_title	5043	4917	Ben-Hur	3
actor_3_name	5020	3521	John Heard	8
plot_keywords	4890	4760	based on novel	4
movie_imdb_link	5043	4919	http://www.imdb.com/title/tt0232500/?ref_=fn_t...	3
language	5031	47	English	4704
country	5038	65	USA	3807
content_rating	4740	18	R	2118

Impression:

- Numerical variables about Facebook likes have large range from min to max.
- Categorical variables ‘color’, ‘language’, ‘country’, ‘content_rating’ have huge unique levels.

2.2 Data Cleaning

2.2.1 Duplicates Data

Remove the duplicates instances, after that, the dataset have 4998 rows.

2.2.2 Remove Redundant Variables

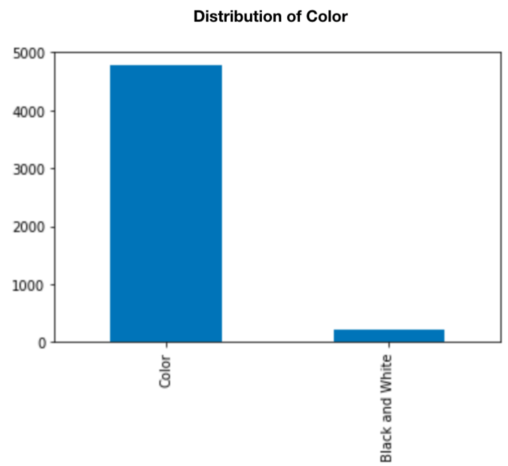
- “director_name", “actor_1_name", “actor_2_name", "actor_3_name", "movie_title", "plot_keywords", "movie_imdb_link" have too many levels.
- It is not helpful to use these variables in predictable models, especially in a small dataset. Thus, these variables are able to be removed.

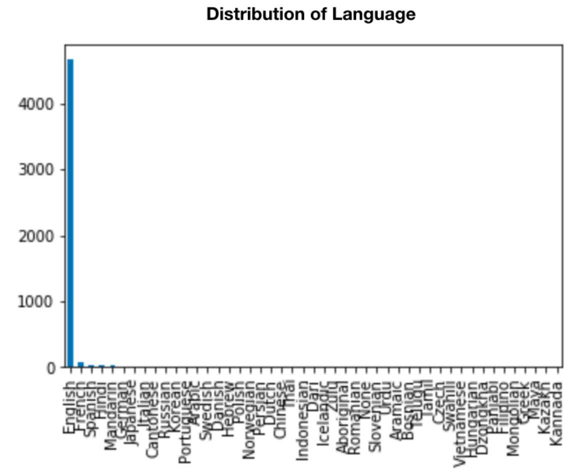
2.2.3 Color

- There are more than 95% of movies are colored. It indicates that this variable is almost fixed. Since there is no point to predictable models, I remove this predictor.

2.2.4 Language

- It is as same as the previous one. More than 93% are English movies. Remove 'Language' as well.





2.2.5 Country

- There are about 75% movies from USA, 9% from UK, and 16% from other 63 countries.
- Other countries are grouped as 'others' to reduce the number of levels.

Value counts of Original Country

USA	3773	
UK	443	
France	154	
Canada	124	
Germany	96	
	...	
New Line	1	
Indonesia	1	
Libya	1	
Kyrgyzstan	1	
Afghanistan	1	
Name: country,	Length:	65

Value counts of New Country

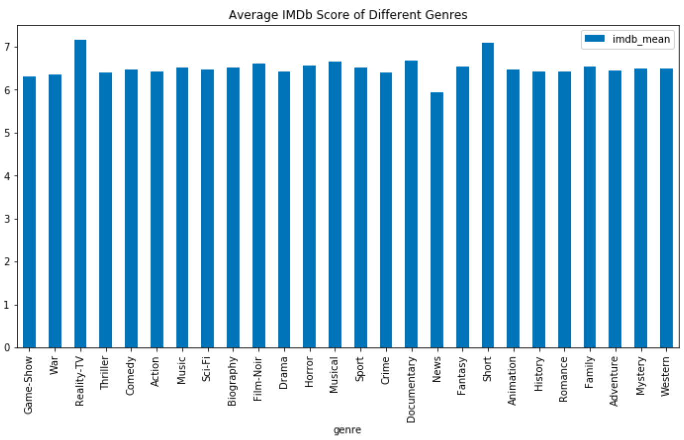
USA	3773	
UK	443	
Other_counties	782	
Name: country,	Length:	65

2.2.6 Genres

- A lot of values of this column combine with multiple genres.

- To analyze whether each genre is related to IMDb score (response variable), we separated the values by '|' first, then Calculate the means of imdb_score of each genre and plot it.
- The average IMDb scores of different genres are in the range from 6 - 8, which means the IMDb score is not sensitive to the feature “genres".
- So the variable “genres” is removed.

	imdb_score	genres
0	7.9	Action|Adventure|Fantasy|Sci-Fi
1	7.1	Action|Adventure|Fantasy
2	6.8	Action|Adventure|Thriller
3	8.5	Action|Thriller
4	7.1	Documentary



2.2.7 Missing Values

- We do not want to lose much data, especially for the variables that might be related to IMDb score.
- In this case, Even numbers of missing values of "gross", "budget", "aspect_ratio" and "content_rating" are quite high, we will not drop the missing values. We will impute Nas after splitting data into training and test sets.
- For other variables, the numbers of missing values are less, So that we drop Nas directly from the dataset.

Missing Values

#	variables	num_na	percentage
0	gross	874	17.486995
1	budget	487	9.743898
2	aspect_ratio	327	6.542617
3	content_rating	301	6.022409

4	title_year	107	2.140856
5	director_facebook_likes	103	2.060824
6	num_critic_for_reviews	49	0.980392
7	actor_3_facebook_likes	23	0.460184
8	num_user_for_reviews	21	0.420168
9	duration	15	0.300120
10	facenumber_in_poster	13	0.260104
11	actor_2_facebook_likes	13	0.260104
12	actor_1_facebook_likes	7	0.140056
13	num_voted_users	0	0.000000
14	cast_total_facebook_likes	0	0.000000
15	country	0	0.000000
16	imdb_score	0	0.000000
17	movie_facebook_likes	0	0.000000

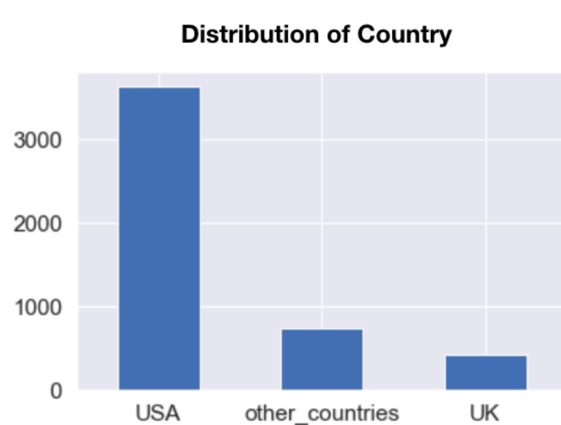
By now, we have 4814 rows left, only 5% of the observations which is acceptable.

2.2 Data Visulation

2.3.1 Univariate distributions

- Bar plot of important categorical variables

Country



Content_rating

- There are 3 categories which are unused and will be removed -- 'TV-MA','TV-Y','TV-Y7'.
- TV ratings are another rating system for TV. Since they are not related to movies, 'TV-G', 'TV-14','TV-PG' are going to be removed.
- The film rating systems are changed by years. The latest ratings including:'PG', 'PG-13', 'R',' NC-17', 'G' and other labels ('Not Rated', ‘Unrated').
- According to the changing history of rating systems, 'PG' replaced 'M', 'GP', 'NC-17' replaced ‘X’.

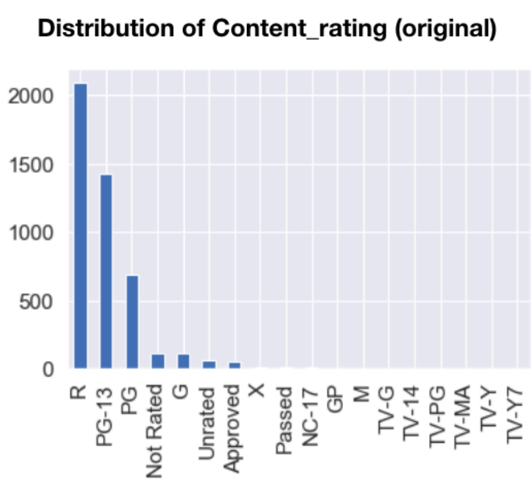
Value counts of Content_rating

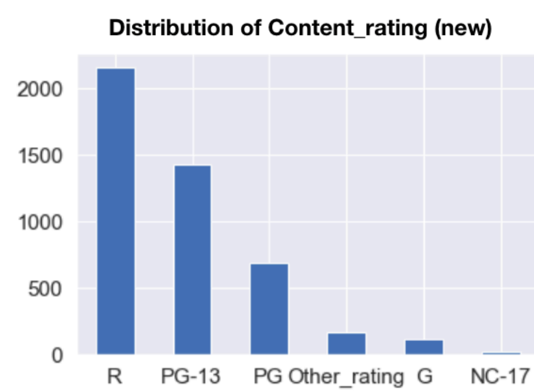
R	2091
PG-13	1429
PG	693
Not Rated	110
G	109
Unrated	58
Approved	55
X	13
Passed	9
NC-17	7
GP	6
M	5
TV-G	4
TV-14	3
TV-PG	1
TV-MA	0
TV-Y	0
TV-Y7	0

(https://en.wikipedia.org/wiki/Motion_Picture_Association_of_America_film_rating_system#:~:text=Rated%20G%3A%20General%20audiences%20%E2%80%93%20All,accompanying%20parent%20or%20adult%20guardian.)

- 'Approved' or 'passed' - They are labels before 1968. Based on whether they were 'moral' or 'immoral', films were approved or disapproved. Since 'R' has highest frequency, these two lables are replaced by 'R'.

(https://help.imdb.com/article/contribution/titles/certificates/GU757M8ZJ9ZPXB39#)

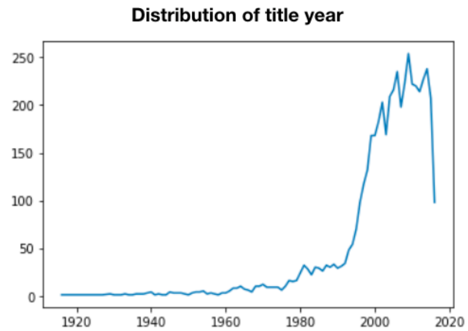




- Histogram of important numerical variables

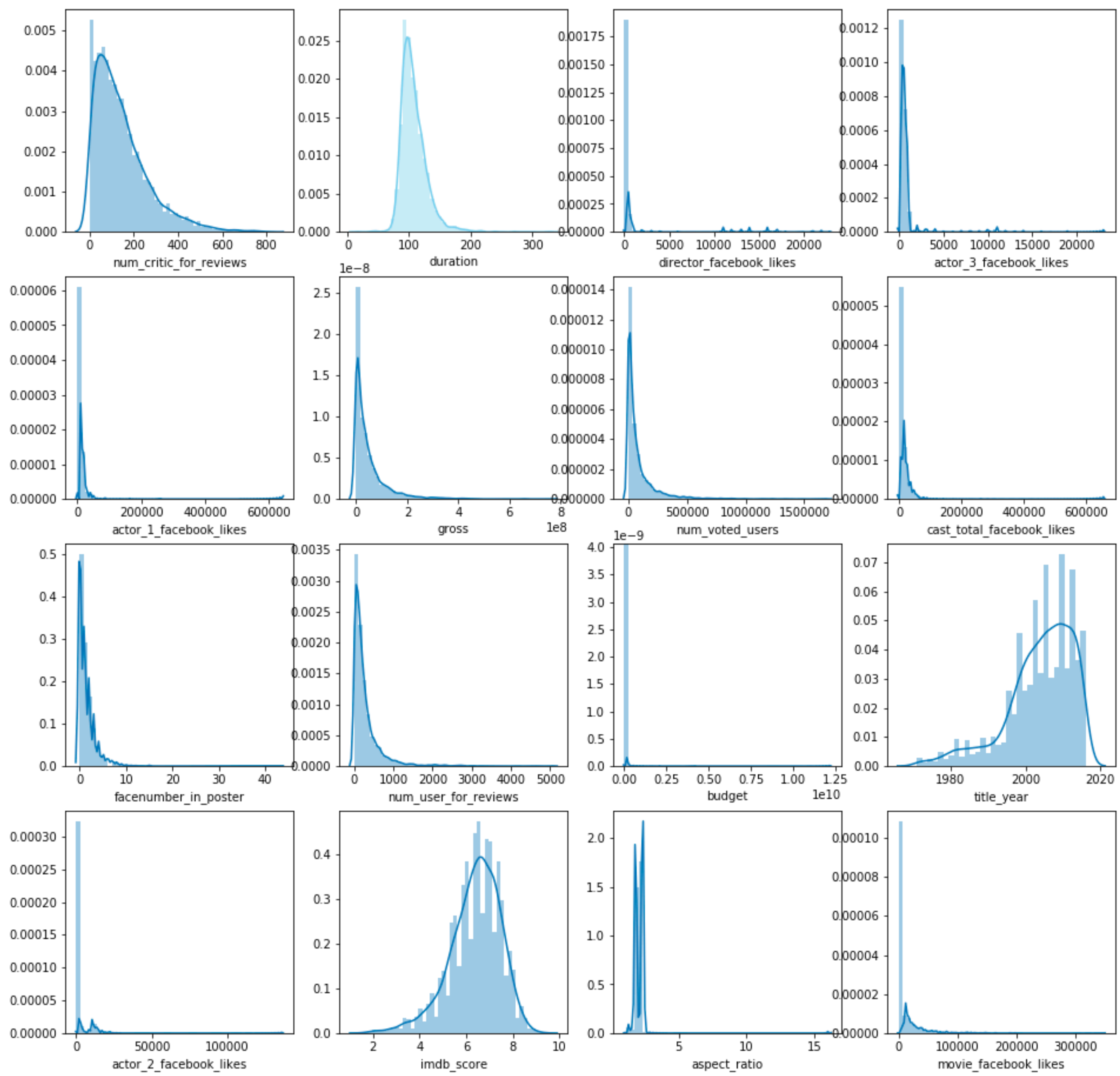
Title_year

- The number of movies before 1960 is: 76, percentage: 1.58.
- The number of movies before 1970 is: 156, percentage: 3.25.
- The number of movies before 1980 is: 279, percentage: 5.81.



- The movies released from 1916 to 1970 is only around 3% of the data, which indicated that these movies might not be typical and can be removed.

Distribution of other numerical variables

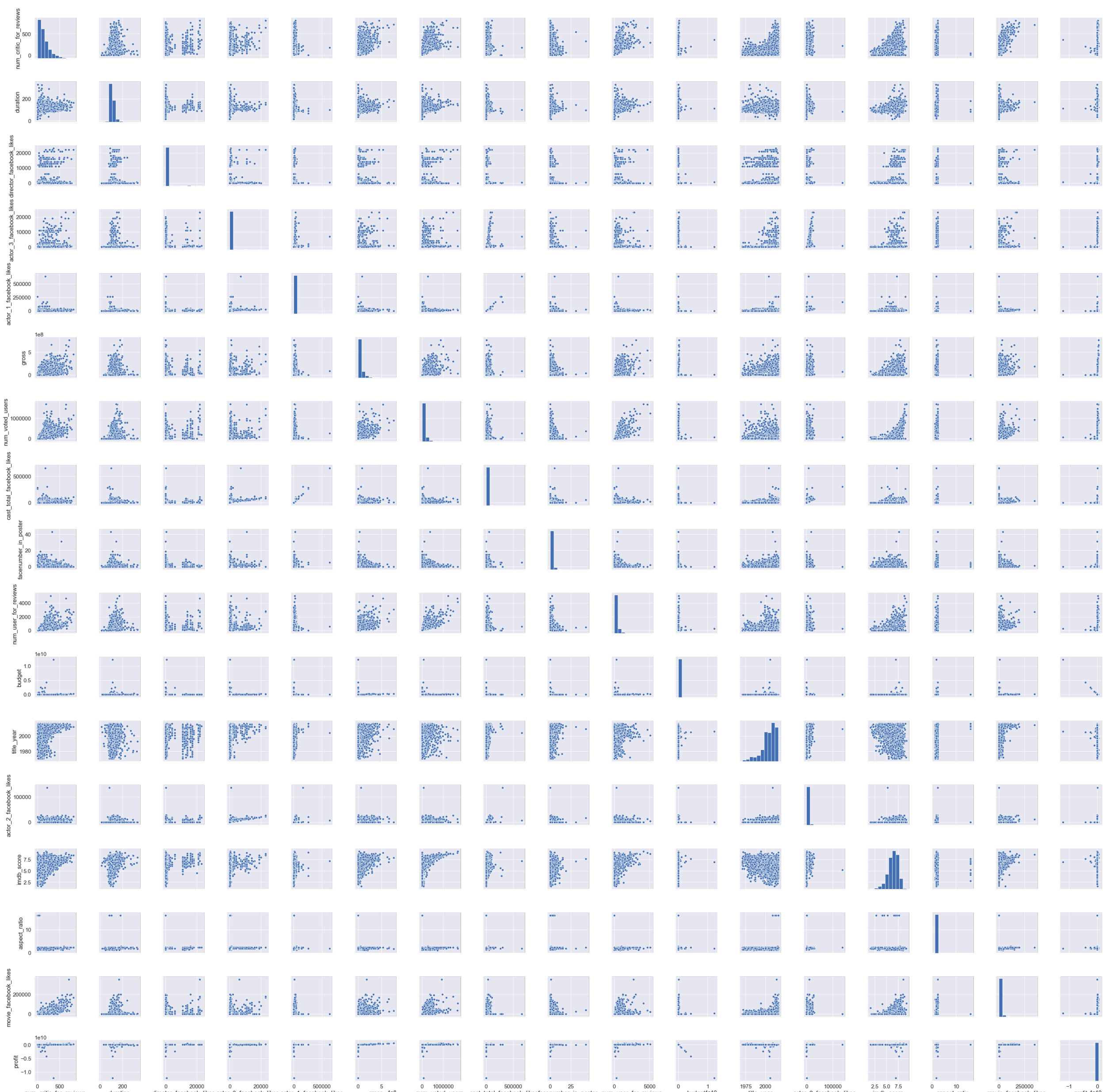


- A lot of distribution of numerical features are skewed, so that we will standardize these variables after splitting data into train and test.

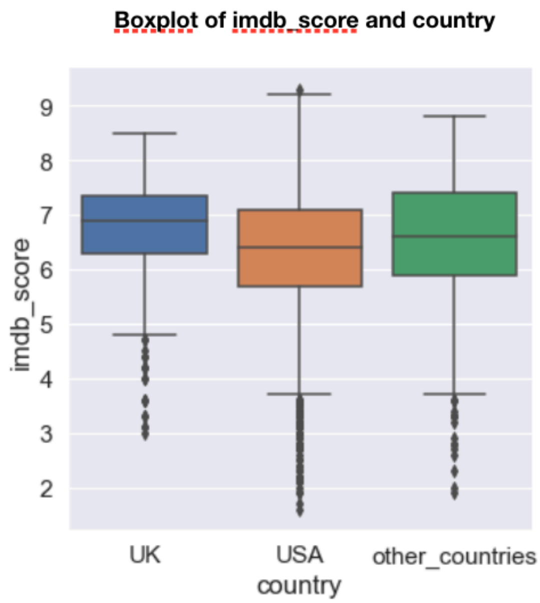
2.3.2 Bivariate Plots

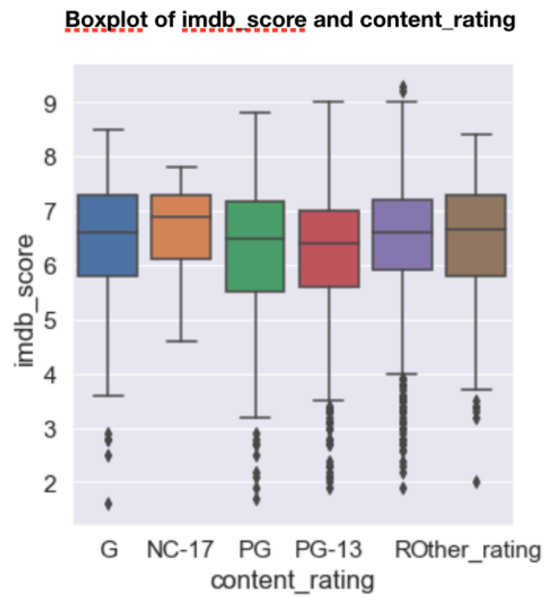
Scatter Plots of Pair Numerical Variables

Valuable insights: ‘Imdb_score’ is positively correlated with ‘num_critic_for _ revie ws’ , ‘gross', ‘num_voted_users', ‘facenumber_in_poster', ‘actor_2_facebook_likes'.

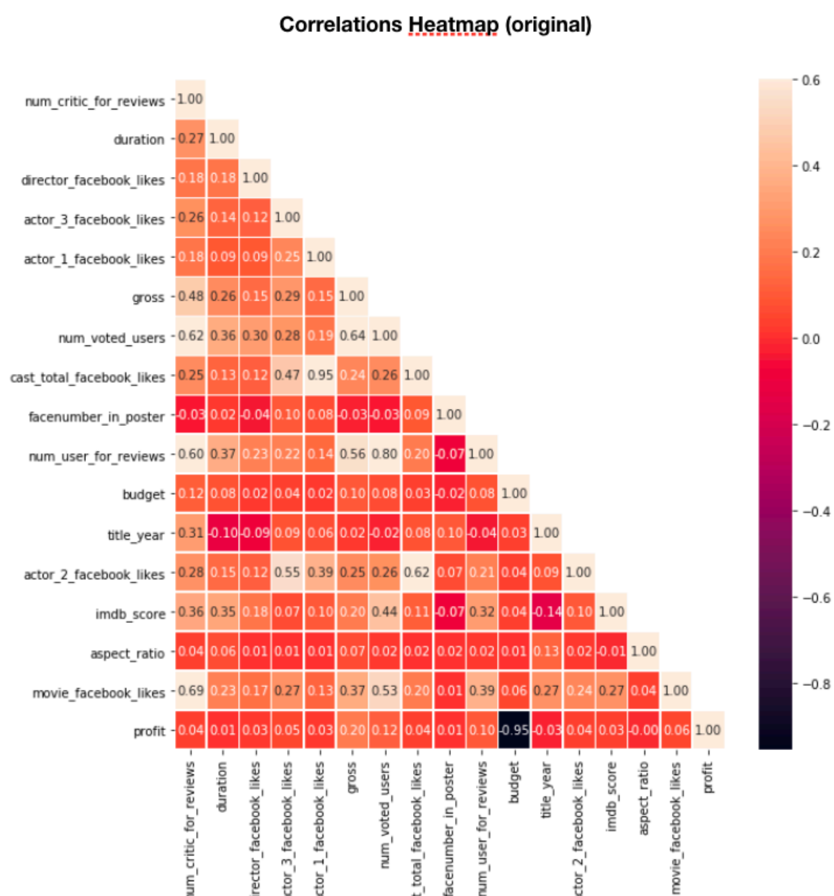


Bar Plots of Imdb_score and Categorical Variables





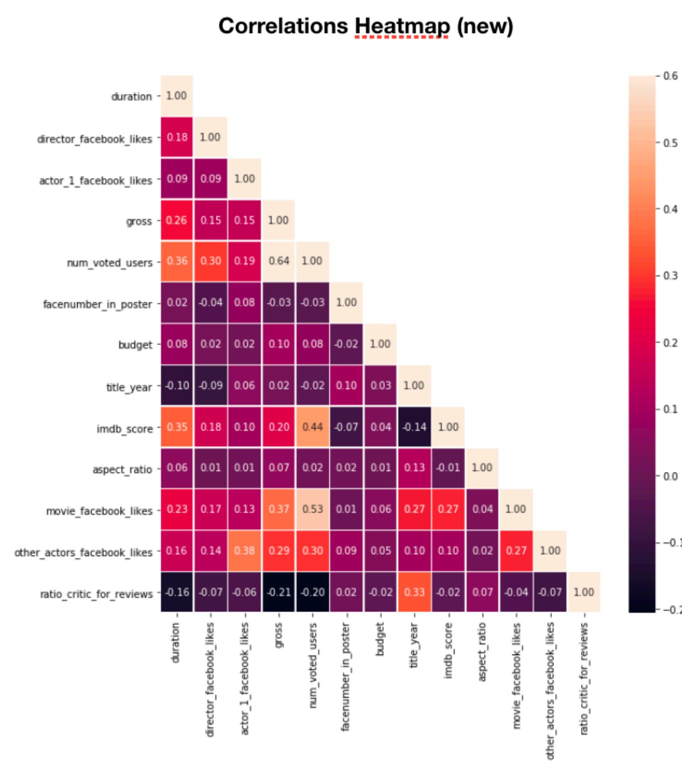
2.3.2 Correlation Heatmaps



Highly correlated variables

Based on the heatmap:

- 'actor_1_facebook_likes' and 'cast_total_facebook_likes' are highly correlated (corr=0.95). 'actor_2_facebook_likes' and 'actor_3_facebook_likes' are also correlated to 'cast_total_facebook_likes'. It is better to remove 'cast_total_facebook_likes', and combine actor 2 and actor 3 as 'other_actors_ Facebook _likes'.
- 'num_voted_users', 'num_user_for_reviews', 'num_critic_for_reviews' are also highly correlated each other. Based on the parrot, 'num_voted_users' should be kept because it might related with imdb_score. We can calculate the ratio of critic reviews from 'num_user_for_reviews', 'num_critic_for_reviews', which might be significant to IMDb scores. Then remove these two variables.



- 'movie_facebook_likes' is also highly correlated with 'num_critic_for_reviews'. As latter one would be removed after calculating the ratio of critic reviews, we will keep it temporarily and check it again. There is no corr value larger than 0.6.

By now, the data has 4650 rows and 15 columns. We just lost 8% data, which is acceptable.

3. Data Pre-processing

3.1 Bin Response Variable

This project is about to predict whether one movie is good or not so that the response variable should be binned into ranks.

Imbd_class Table

Imbd_scores	Level	Imbd_class
<4	Bad	0
4~6	Avarage	1
6~8	Good	2
8~10	Excellent	3

Imbd_class Value Counts

Imbd_class	Counts
2	3019
1	1256
3	230
0	145

4. Machine Learning Models

Features list: ['duration', 'director_facebook_likes', 'actor_1_facebook_likes', 'gross', 'num_voted_users', 'facenumber_in_poster', 'country', 'content_rating', 'budget', 'title_year', 'aspect_ratio', 'movie_facebook_likes', 'other_actors_facebook_likes', 'ratio_critic_for_reviews']

Numerical features list: ['duration', 'director_facebook_likes', 'actor_1_facebook_likes', 'gross', 'num_voted_users', 'facenumber_in_poster', 'budget', 'title_year', 'aspect_ratio', 'movie_facebook_likes', 'other_actors_facebook_likes', 'ratio_critic_for_reviews']

Categorical features list: ['country', 'content_rating']

4.1 Make a pipeline for data preprocessing

4.1.1 Split data into train and test sets

- Split data into train and test sets with ratio 4:1.
- Make sure to get a similar classes distribution.

Split Data Size

Train size	3720
Test size	930

Distribution of Imdb_class

Imdb_class	Train	Test
0	116	29
1	1005	251
2	2415	604
3	184	46

4.1.2 Make a pipeline for data processing

- Impute Nas: numerical variables with median, categorical variables with most_frequent.
- Standardize the numerical variables.
- One hot code the categorical variables.
- After data processing, we have 4 NumPy arrays, 21 predictors.

Shape of Train and Test Sets

X_train	(3720, 21)
y_train	(930, 21)
X_test	(3720,)
y_test	(930,)

4.2 Apply different machine learning algorithms to multi-class response.

Including hyperparameter tuning better estimator and cross-validation for avoiding overfitting.

- Logistic Regression
- Random Forest Classifier
- XGBoost Classifier
- Neural-network Model with Keras

4.3 Make a pipeline for data prediction

- Predict the test data using the estimator.
- Print the classification report.
- Plot the confusion matrix.

4.4 Interpretation of classification models results

4.4.1 Logistic Regression

- Logistic Regression model's best params: {'C': 0.1}.

- Logistic Regression best scores: 0.70.
- **Explation:**
  - This model is doing well on predicting ‘good’ level of imdb_class. It predicts 92% of true good movies and precision is 75%.
  - For ‘average’ and ‘excellent’ levels, this model just predicts 39% and 43% of true level separately. The precision is 63% and 71%. More instances of these levels are grouped into the good level.
  - The model fails to predict the ‘bad’ level. It predicts 19 true ‘bad’ movies as average, and 10 as good.

Confusion Matrix of Logistic Regression Model

	Precision	recall	F1-score	Support
Bad	0.00	0.00	0.00	29
Average	0.63	0.39	0.48	251
Good	0.75	0.92	0.83	604
Excellent	0.71	0.43	0.54	46
Accuracy			0.73	930
Macro avg	0.52	0.44	0.46	930
Weighted avg	0.69	0.73	0.69	930

CM Plot of Logistic Regression Model

4.4.2 Random Forest Classifier

- Random Forest Classifier model's best params: {'max_depth': 80, 'max_features': 3, 'n_estimators': 200}.
- Random Forest Classifier best scores: 0.74.
- **Explanation:**
  - This model is also doing well on predicting the ‘good’ level of imdb_class. It predicts 96% of true good movies but the precision is 74%.
  - For ‘average’ and ‘excellent’ levels, this model just predicts 36% and 35% of true level separately, which are worse than the logistic regression model. But precision is 73% and 89%.

- The model fails to predict the ‘bad’ level. It predicts 19 true ‘bad’ movies as good, and 10 as average.
- Although the model’s accuracy is higher, it groups more observations into the good levels.

Confusion Matrix of Random Forest Model

	precision	recall	f1-score	support
bad	0.00	0.00	0.00	29
average	0.73	0.36	0.48	251
good	0.73	0.96	0.83	604
excellent	0.89	0.35	0.50	46
accuracy			0.74	930
macro avg	0.59	0.42	0.45	930
weighted avg	0.72	0.74	0.70	930

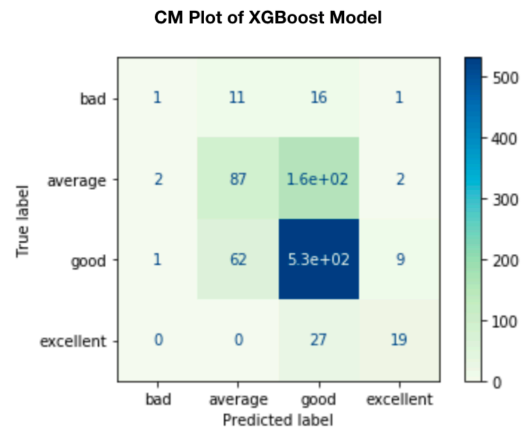
CM Plot of Random Forest Model

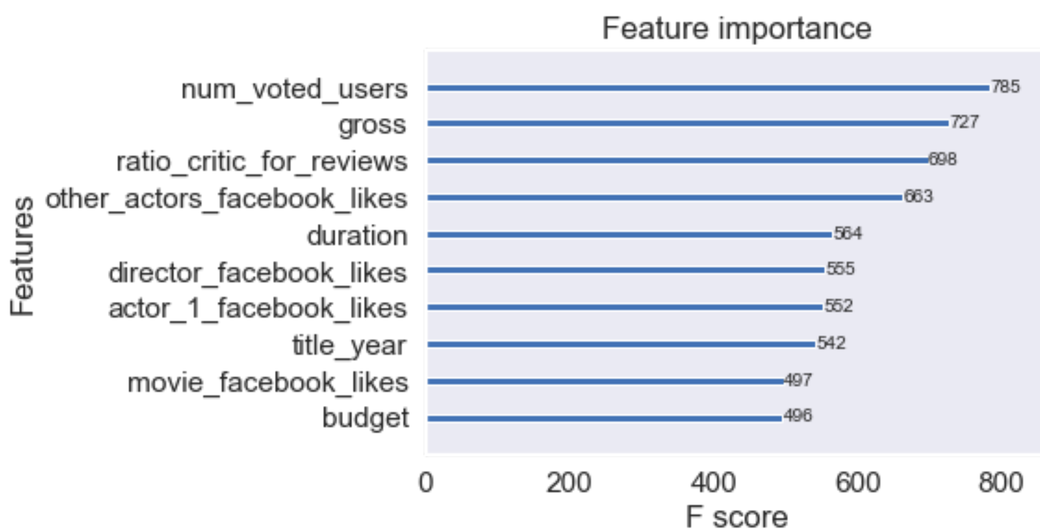
4.4.3 XGBoost Classifier

- XGBoost Classifier model's best params: {'colsample_bytree': 0.3, 'eta': 1, 'max_depth': 5}
- XGBoost Classifier model's best scores: 0.75.
- **Explanation:**
  - The training accuracy is 0.75 while predicting accuracy is 0.69. So this model is kind of overfitted.
  - The model can identify bad movies, although precision and recall are very low.
  - For average and excellent levels, the recall and precision are both lower than the Random Forest model.
  - It predicted 1 true ‘bad’ movies as ‘excellent’.
  - The top 8 importance features are: ‘num_voted_users’, ‘gross’, ‘ratio_critic_for_reviews’, ‘duration’, ‘director_facebook_likes’, ‘actor_1_facebook_likes’, ‘title_year’.

Confusion Matrix of XGBoost Model

	precision	recall	f1-score	support
bad	0.25	0.03	0.06	29
average	0.54	0.35	0.42	251
good	0.72	0.88	0.79	604
excellent	0.61	0.41	49	46
accuracy			0.69	930
macro avg	0.53	0.42	0.44	930
weighted avg	0.65	0.69	0.66	930





4.4.4 Neural-network Model with Keras

- Neural-network model's best params: {{'batch_size': 5, 'epochs': 100}.
- Neural-network model's best scores: 0.73.
- **Explanation:**
  - The training accuracy is 0.73 and predicting accuracy is 0.72. We need not worry about overfitted.
  - The model has a better ability to identify movies that are average or excellent than other models.
  - Although it cannot identify true bad movies precisely, it groups a major part of them into the average level, which is not far from reality.

- For average and excellent levels, the recall and precision are both lower than the Random Forest model.

Confusion Matrix of Neural-network Model

	precision	recall	f1-score	support
bad	0.30	0.10	0.15	29
average	0.58	0.49	0.54	251
good	0.77	0.86	0.81	604
excellent	0.76	0.54	0.63	46
accuracy			0.72	930
macro avg	0.60	0.50	0.53	930
weighted avg	0.70	0.72	0.71	930

CM Plot of Nerual-network Model

5. Conclusions

Accuracy for Different Models

Algorithm	Training Accuracy	Test Accuracy
Logistic Regression	0.70	0.73
Random Forest	0.74	0.72
XGBoost	0.75	0.69
Neural-network	0.73	0.72

Based on the accuracy table:

- Although the Random Forest model has the highest accuracy, it groups most observations into the good levels. This model cannot interpret the practical problem well.
- XGBoost has the highest training accuracy, but the test accuracy is lower than it so that this model is kind of overfitted.
- The difference between training and test accuracy of Neural-network is least. And the prediction is closer to reality than other models. Hence we conclude that the optimized model is nerual-network model.

- **User_vote, gross, ratio of critic reviews', duration, Facebook likes to director and actor_1** are very important variables, while **face number in post**, **content** and **country** are not so crucial to the quality of the movies.
- All of the models can not identify 'bad' level precisely. It is because the dataset is unbalanced. For improving the problem, more data are needed.