ISYE7406 Data Mining & Statistical Learning

Team Project: TMDB Movies Analysis

Introduction

- The film industry has been impacted significantly due to the global pandemic since 2020.
- With the film industry being challenged, it's crucial for investors to make decisions with minimum risk.
- In this study, we will focus on:
  - analyzing the historical data of the film industry
  - evaluating different classification models and providing our recommendations on which models could perform well when predicting if the film could be a success or not

Dataset Description

- “TMDB 5000 Movie Dataset” from Kaggle.
- It includes 4803 movies released from 1916 to 2017.
- The raw data contains 20 Variables:
  - Numerical explanatory variables: budget, id, popularity, release date, runtime, vote average, vote count
  - String explanatory variables: genres, homepage, keywords, original language, original title, overview, production companies, production countries, spoken languages, status, tagline, title
  - Response variable: revenue

Data Wrangling

#1.Drop columns that will not be explored

["homepage", "id", "original_language", "keywords", "release_date", "overview", "production_countries", "spoken_languages", "status", "tagline"]

#2. Drop the entries that has null values

#3. Drop the entries where “budgets==0”. Having 0 budgets is not possible and may simply because of the lack of information

Data Wrangling

#4. For variable “genres”, it contains a list of dictionaries as is shown below:

[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]

The name is separated into a list [Action, Adventure, Fantasy, Science Fiction], **and dummy variables are created as is shown in picture below:**

genres_Action	genres_Adventure	genres_Animation	genres_Comedy	genres_Crime	genres_Documentary	genres_Drama	genres_Family	genres_Fantasy
1	1	0	0	0	0	0	0	1

#5. For variable “production_companies”, similar operations are carried out as in #4 and dummy variables are created

Shape is (4800,40)

Data Wrangling

#6. Categorical variable “Success” is created, and this is the response variable for our studies. The successfulness of the movie is determined when the revenue is greater or equals 3*budgets [1]

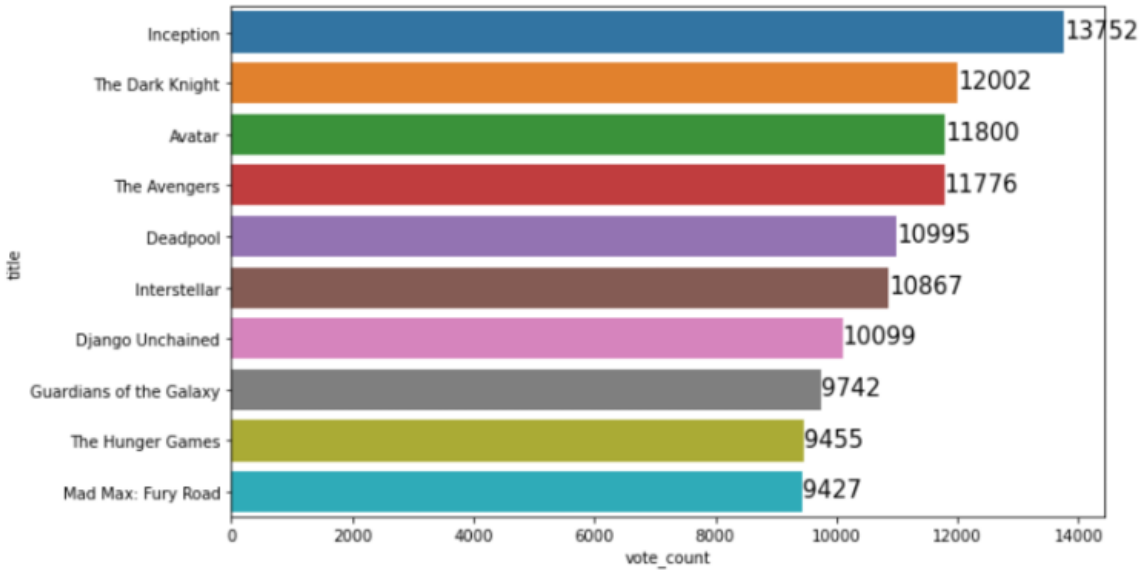
#7. Drop column “revenue” which is multicollinear with the variable “success”.

The dimension of the data is (3764, 4216).

#8. Randomly split the data with 80% for training data and 20% for testing.

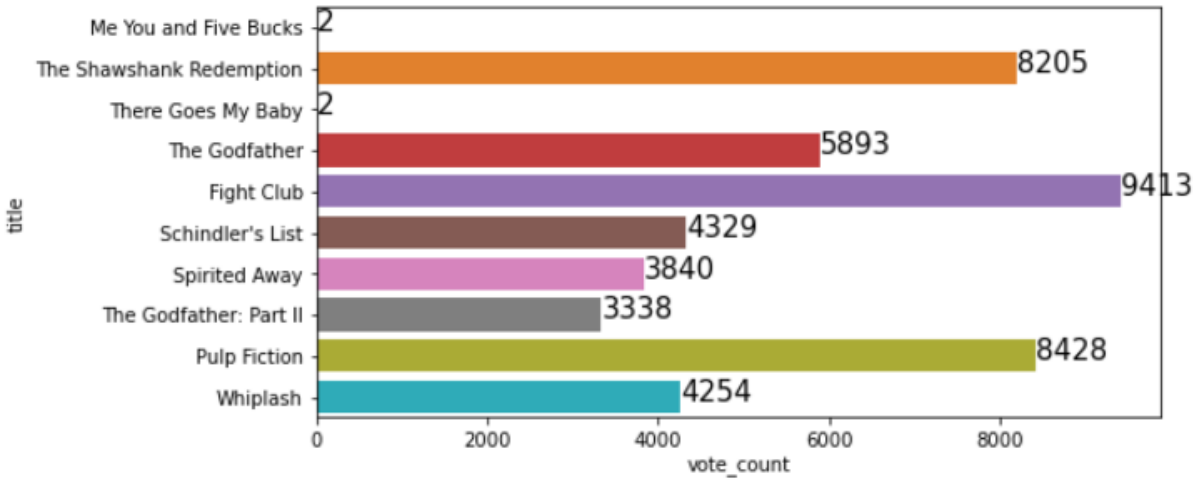
[1]https://gizmodo.com/how-much-money-does-a-movie-need-to-make-to-be-profitab-5747305

Top 10 movies with the highest number of votes

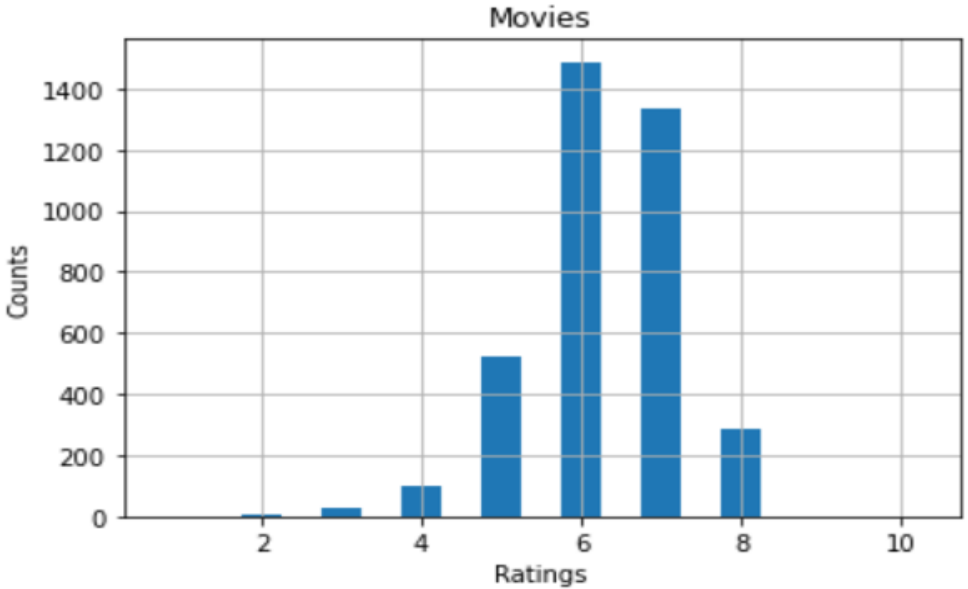


Top 10 movies with the highest average rating

-including number of reviews they received

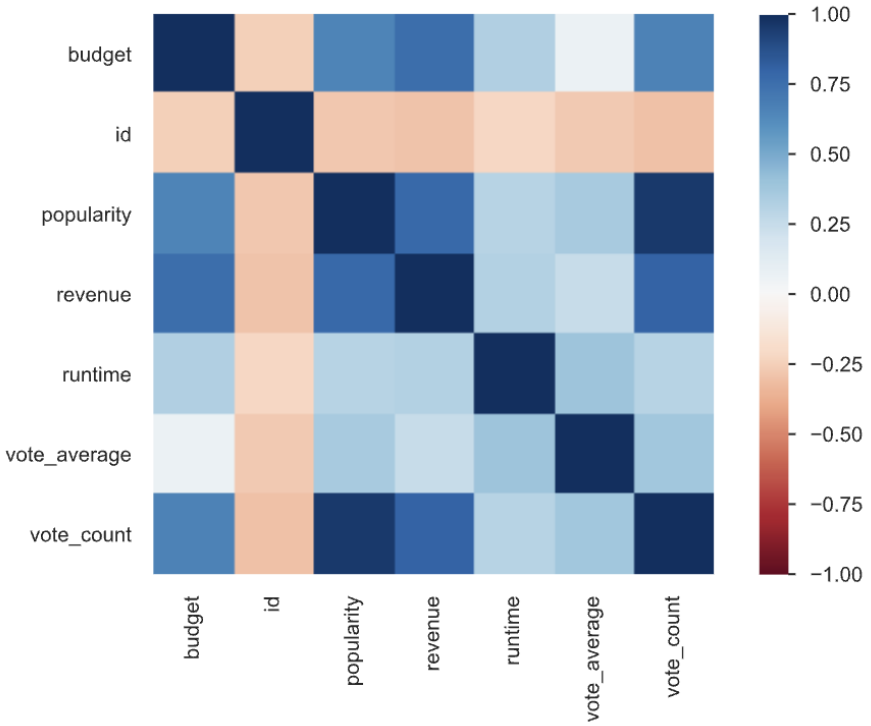


Histogram that shows review count 1-10



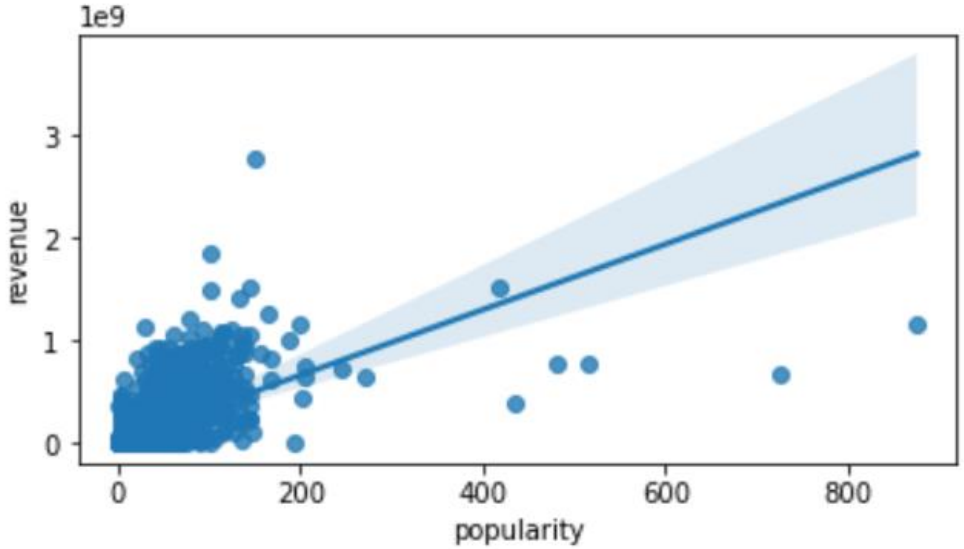
- Majority of the ratings are 6 and 7

Pearson Correlation



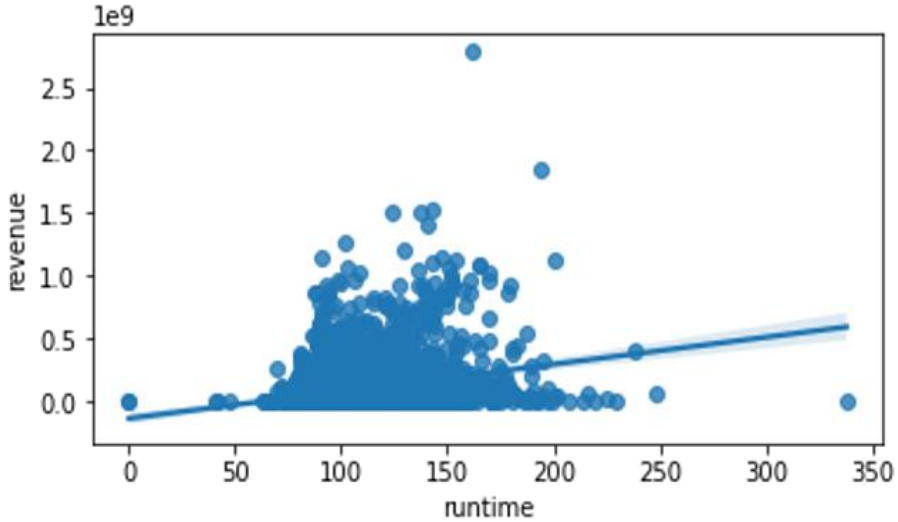
- These four variables: budget, popularity, revenue and vote_count are highly correlated with one another
- Runtime and vote_averge have low correlation with all the other variables
- Id variable has weak or no correlation with all the other variables

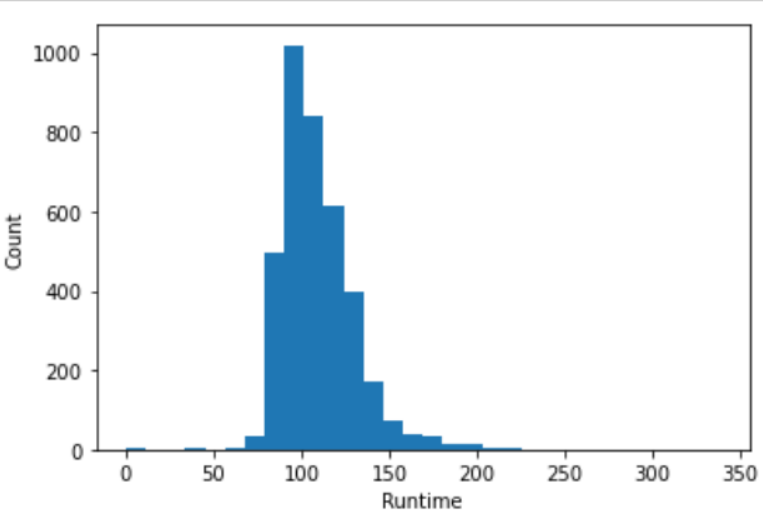
Correlation between popularity and revenue



- Strong positive correlation between popularity and revenue

Correlation between runtime and revenue

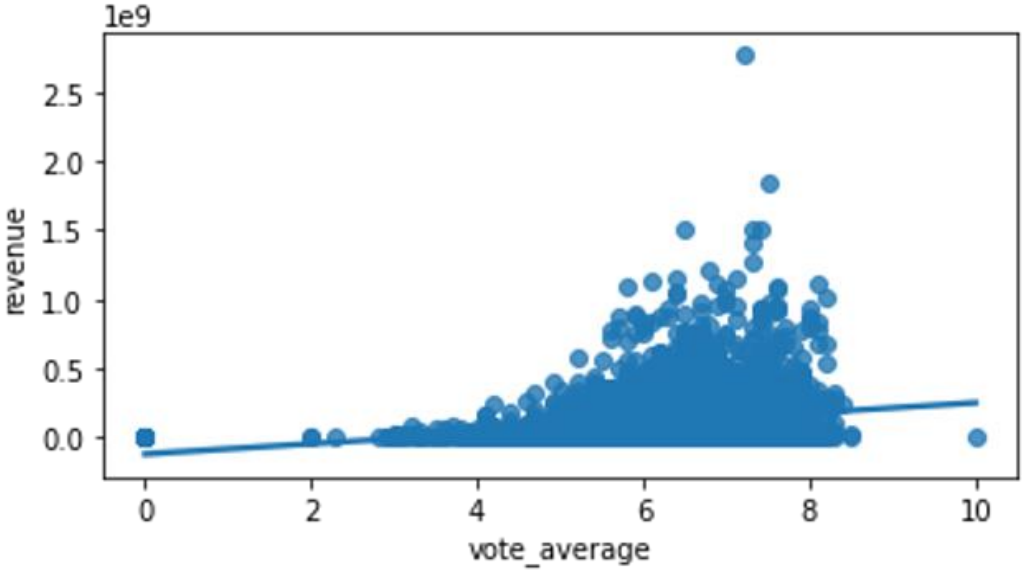




- Weak correlation between revenue and runtime

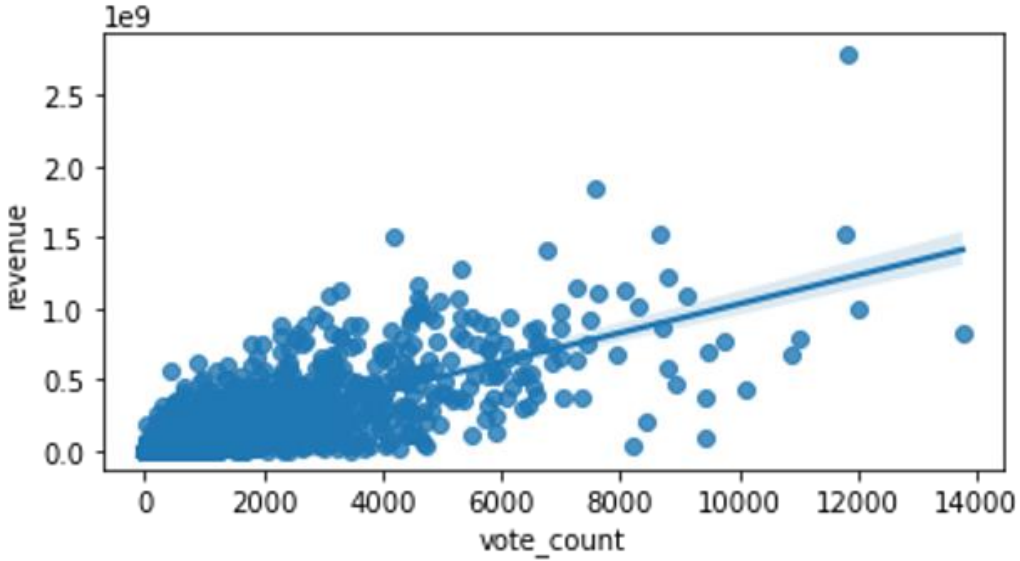
- Majority of the movie's runtime are between 100 minutes and 120 minutes

Correlation between vote average and revenue



- Weak correlation between revenue and vote_average

Correlation between vote count and revenue

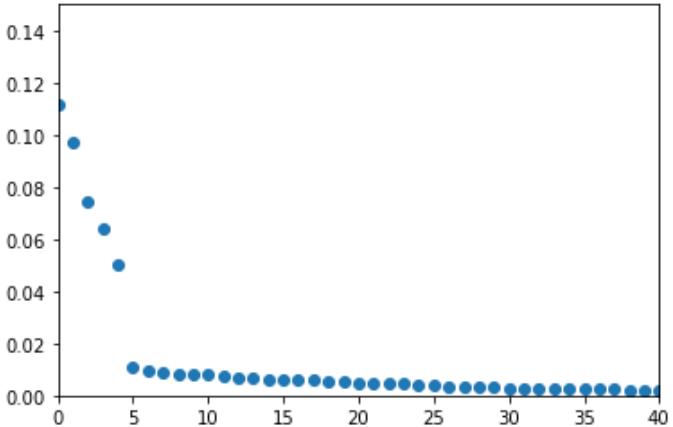


- Strong positive correlation between revenue and vote_count

Random Forest Classifier

Using RandomForestClassifier from sklearn.ensemble package, the training error for default parameter (n_estimators=100) is 0 and the testing error is 0.2164.

Below is the graph that shows the importance of the top 40 features.



- popularity, vote count, vote average, budget, runtime are the most valuable features.

Random Forest Classifier

For the convenience of computation, only the top 25 features are kept for further analysis. These features are ordered by their importance:

['popularity', 'vote_count', 'vote_average', 'budget', 'runtime',
'genres_Drama', 'genres_Comedy', 'genres_Action',
'genres_Thriller', 'genres_Romance', 'genres_Adventure',
'genres_Fantasy', 'genres_Science Fiction', 'productions_Paramount
Pictures', 'genres_Crime', 'productions_Universal Pictures',
'genres_Family', 'genres_Horror', 'genres_Mystery',
'genres_Music', 'productions_Warner Bros.',
'productions_Twentieth Century Fox Film Animation,
'productions_Columbia Pictures', 'productions_United Artists',
'productions_New Line Cinema']

Random Forest Classifier

Next a grid search for the optimal parameter was carried out using GridSearchCV from sklearn.model_selection, and the cross validations was set to 3.

The result is shown below:

Best accuracy: 0.787 Best params: {'max_features': 4, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 150}

The testing error using the the best parameters is 0.2058.

Gradient Boosting Classifier

Using GradientBoostingClassifier() from sklearn.ensemble package, the training error for default parameter (learning_rate=0.1) is 0 and the testing error is 0.2.

Next a grid search for the optimal parameter was carried out using GridSearchCV from sklearn.model_selection, and the cross validations was set to 3.

The result is shown below:

Best accuracy: 0.784 Best params: {'learning_rate': 0.45, 'max_depth': 8, 'n_estimators': 150}

The testing error using the the best parameters is 0.2138.

Simple Baseline Methods

One Split:

	Train Error	Testing Error
LDA	0.2405	0.2457
QDA	0.2710	0.2709
Naive Bayes	0.2933	0.2815
Logistic Regression	0.2288	0.2390
KNN (K = 31)	0.3819	0.3174
KNN (K = 59)	0.3613	0.3267
KNN (K = 101)	0.3427	0.3280
KNN (K = 201)	0.3414	0.3267
KNN (K = 301)	0.3414	0.3267

Simple Baseline Methods

CV with 100 iterations:

	Avg Train Error	Avg Testing Error
LDA	0.2429	0.2455
QDA	0.2643	0.2765
Naive Bayes	0.2897	0.2939
Logistic Regression	0.2299	0.2330
KNN (K = 31)	0.3710	0.3121

Findings

- The Random Forest method performed the best in terms of training and testing error. Not only does it give the lowest testing error, but it's also advantageous for several reasons. First, the tree-based methods do not require normality assumptions like LDA/QDA does. Nor does it require the observations to be independent of each other. Additionally, tree-based methods generally perform well with many features.
- However, the result from the Random Forest is not interpretable because Random Forest ran hundreds of models in the background and then used the calculation from those models to come up with the single result. In this regard, Logistic regression is more advantageous as it sacrifices a little accuracy in prediction, but is much easier to interpret.
- Five k values were used in KNN method, and it turns out that the model performed the best when k equals to 31. As k increases, the training error decreases and the testing error first increases and then decreases. The performance of the model didn't improve with k over 200.
- Compared with logistic regression, random forest and boosting methods, the LDA, QDA and Naive Bayes shows moderate performance in training and testing error. The normality assumptions for LDA, QDA and Naive Bayes may be one of the reasons why they don't perform as well.

Conclusion

- To conclude, we have taken a rather complicated dataset with multiple variables containing dictionaries and missing values cleaned it.
- We have created new variables to facilitate our studies.
- A total of 7 models were evaluated and compared. Cross validation was performed to reduce overfitting and gave a better approximation for model accuracy.
- We have found that Random Forest, Gradient Boosting and Logistic Regression show good predictive power for this particular dataset. The accuracy could reach as high as 79%.
- LDA, QDA, Naive Bayes, KNN models are not recommended for our purpose.