Movie Box Office Failure Prediction

Hongjiao Zhang; Jiao Wang; Zijin Gao; Ruize Hou; Curtis Chen

Background

In this project, we will be combining the MovieLens, IMDb, Kaggle Revenue datasets to obtain movie features, including ratings, revenue, and metadata, and use these features to predict whether a movie will be a box office failure or not. The questions we are trying to solve with this project are:

- What are the variables/features that determine whether a movie will be a box office failure?
- Can we accurately predict a movie's box office performance based on its features?
- How does a movie's budget impact its box office performance?
- Can we identify genres that are more likely to be box office failures?
- How can studios and investors optimize their investments in movie projects to maximize their returns?

Why are you trying to answer these questions (what's the *value* you would bring)? What's the impact?

- We could identify successful genres and provide insights into audience preferences, helping studios understand what kind of movies audiences prefer and optimize their production strategies accordingly.
- Our predictions can help studios optimize their investments by reducing the risk of a movie being a box office failure.
- We can can provide insights into the impact of marketing expenses on a movie's box office performance and help studios plan their marketing campaigns more effectively.
- By identifying factors that contribute to a movie's success, our analysis can help studios improve the return on investment for movie projects.

Background

Is there any relevant work in your area that is useful? Relevant work could be an existing research/systems in the same domain/similar domains.

There are several relevant works in our area of focus, we hope to use more up to date data and more novel technical approaches to achieve better performance.

- Predicting movie box office success using multiple regression and SVM
- Predicting movie success with machine learning techniques: ways to improve accuracy
- A Machine Learning Approach to Predict Movie Revenue Based on Pre-Released Movie Metadata
- Finding Nemo: Predicting Movie Performances by Machine Learning Methods
- Box office forecasting using machine learning algorithms based on SNS data
- Movie Box office Prediction via Joint Actor Representations and Social Media Sentiment

Dataset

We combined these three datasets to obtain all the required movie features. We stored the dataset locally on our computer.

1. MovieLens dataset https://grouplens.org/datasets/movielens/
    a. Size: 25 million ratings and one million tag applications applied to 62,000 movies by 162,000 users
2. IMDb dataset https://www.imdb.com/interfaces/
3. The Movies Dataset https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=movies_metadata.csv
    a. Metadata on over 45,000 movies. 26 million ratings from over 270,000 users.

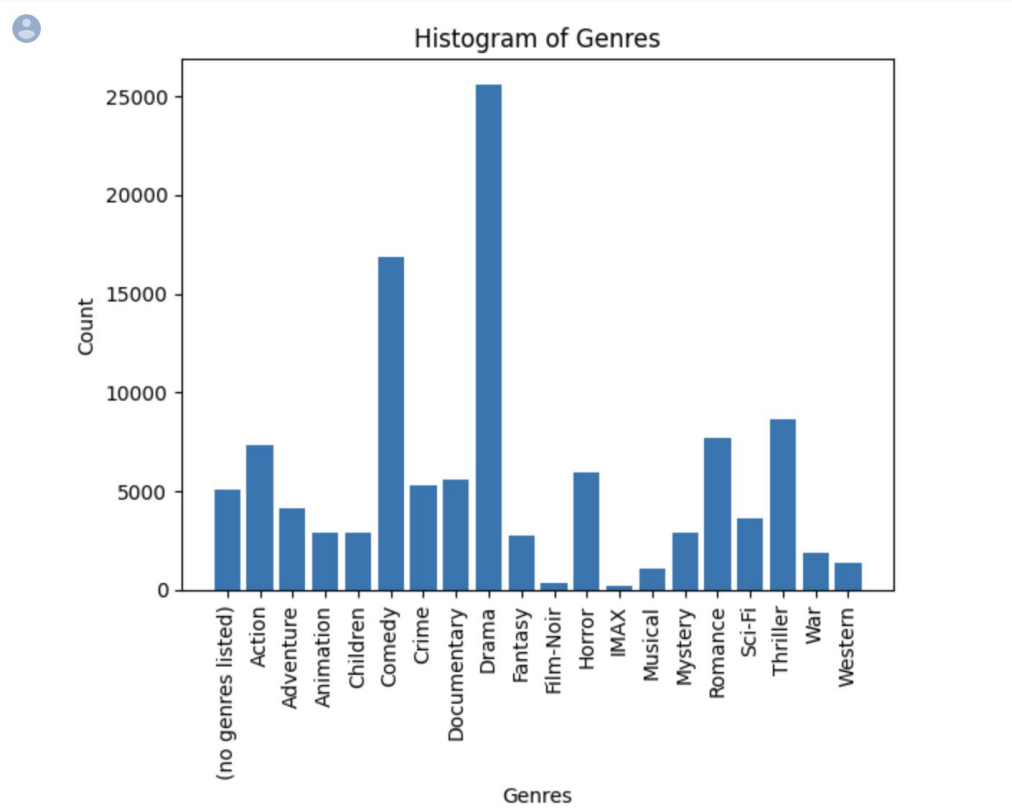
Data Cleaning

1. Movie item data cleaning:

```
# drop the complete dupicates
movies = movies[np.logical_not(movies.duplicated())]

# check for duplicating ids
movies['id'].duplicated().sum()

# investigate duplicate ids
duplicated_id = movies['id'][movies['id'].duplicated()]
movies[movies['id'].isin(duplicated_id)].sort_values('id')

# we only see minor differences between these duplicates, so we drop the ones with bigger index
movies = movies[np.logical_not(movies['id'].duplicated())]
```

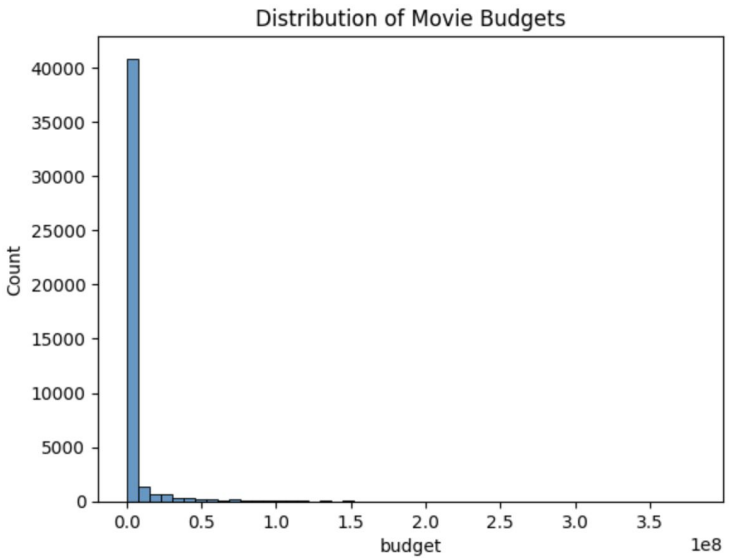
Data Cleaning

2. Movie genres

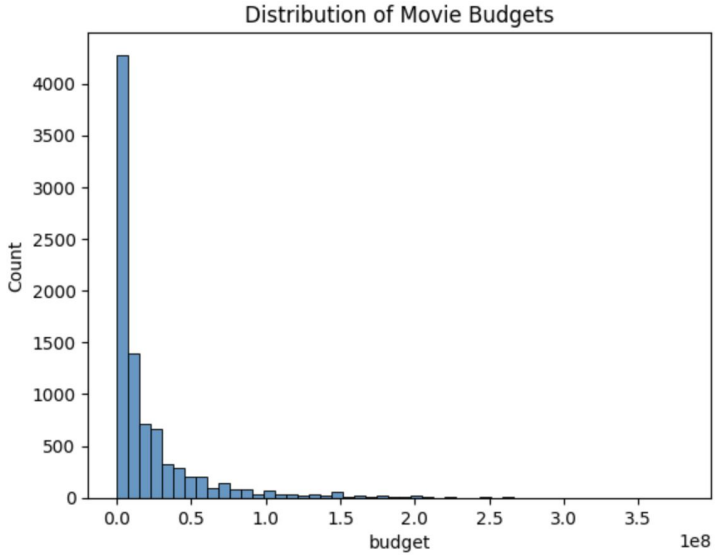


Data Cleaning

3. Movie Budgets



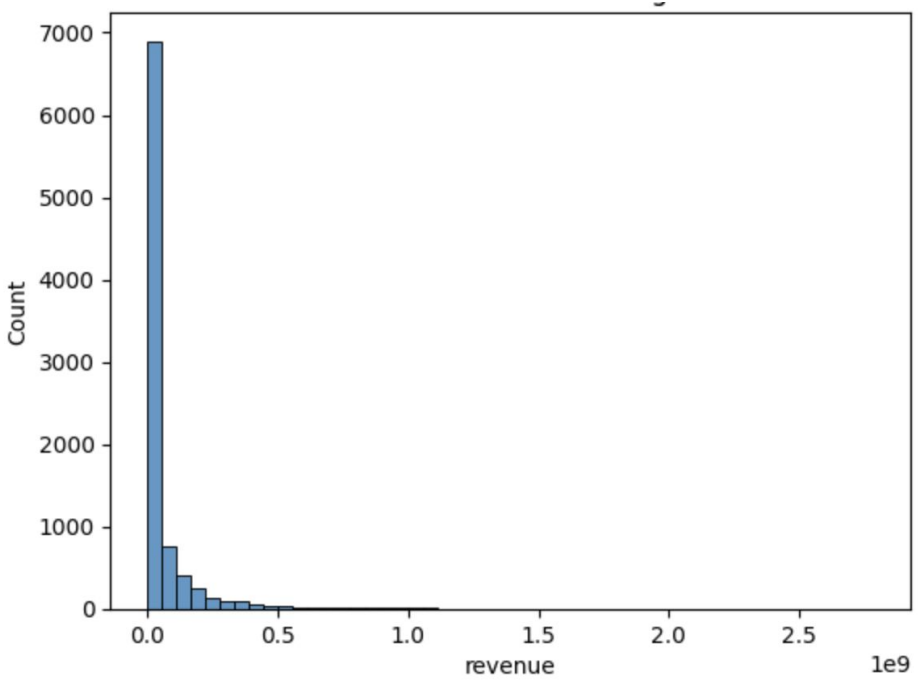
before



after

Data Cleaning

4. Movie Revenue



Data Cleaning

5. Movie profit data construction (Feature Engineering)

- Profit = Revenue - Budget
- ROI (Return on Investment) = (Revenue - Budget) / Budget
- failure_roi = (ROI < 0) ? 1: 0
  - Creates a new feature “box_office_failure_roi” that assigns a value of 1 to movies with a negative ROI and a value of 0 to movies with a non-negative ROI.
- Failure = (profit < 0) ? 1 : 0

```
# make a new column failure to classify whether a movie fails (if profit >= 0 then not fail)
movies_withBudget['failure_profit'] = np.where(movies_withBudget['profit'] < 0, 1, 0)
movies_withBudget['roi'] = (movies_withBudget['revenue'] - movies_withBudget['budget']) / movies_withBudget['budget']
movies_withBudget['failure_roi'] = np.where(movies_withBudget['roi'] < 0, 1, 0)
movies_withBudget['failure'] = np.where(movies_withBudget['profit'] < 0, 1, 0)

# let us only keep language dummy with more than 30 observations
language_dummies_selected = ['language_cn','language_da','language_de', 'language_en', 'language_es', 'language_fi', 'language_fr
language_dummies = language_dummies[language_dummies_selected]
```

Planned Analysis (from old slides)

Various machine learning algorithms for box office failure prediction:

- Logistic Regression: To model the probability of box office failure as a function of the input features. It is well-suited for binary classification problems and can handle both categorical and continuous input features.
- Random Forests: To build an ensemble of decision trees that aggregate the input features and predict the box office success/failure. It can handle both categorical and continuous input features and can capture non-linear relationships and interactions between the input features and the target variable.
- Neural Networks: To build a multi-layered network that learns the input-output relationships through backpropagation. It can handle both categorical and continuous input features and can capture complex non-linear relationships and interactions between the input features and the target variable

Planned Analysis (from old slides)

Issues might arise:

- There are some numerical values like ‘budget’ that are missing. We intend to use other features like ‘year’, ‘genre’ to interpolate these missing values.
- The IMDb dataset may be biased towards certain types of movies or genres, which may not be representative of the entire movie industry. To address these potential biases, we will supplement the IMDb dataset with the box office revenue data.

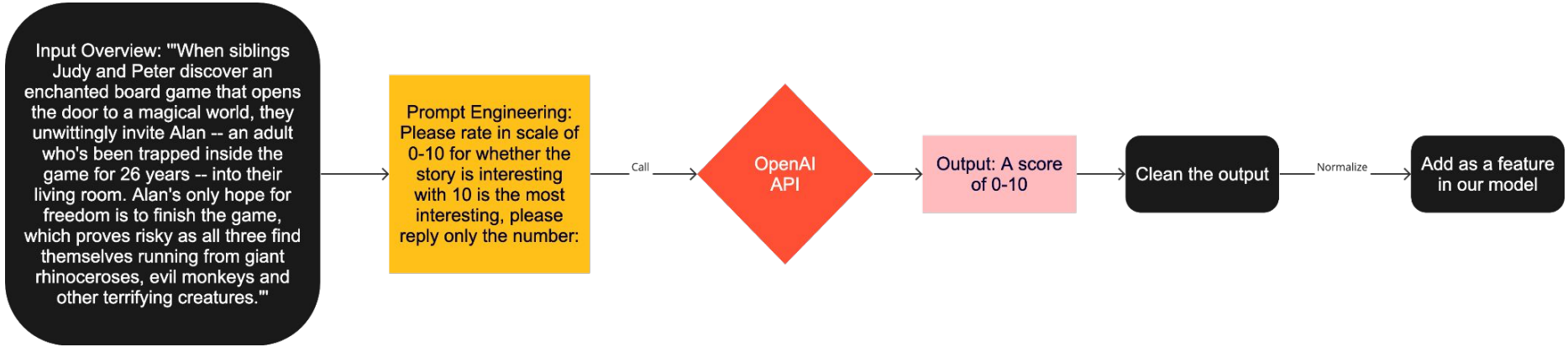
Feature engineering

- Create new features from the existing data that may be more informative. Examples of such features include user ratings, user demographics, temporal trends, movie metadata, such as its title, genre, release year, director/actor popularity, and budget-to-revenue ratio.

Inference

- What analysis will we conduct?
    - To build a model that can predict whether a given movie will be a box office failure or not.
- How do we make sure that the analysis will generalize to new data (i.e. make sure we am not overfitting)?
    - Cross-Validation, Regularization, and Model Selection.
- If we are using a machine learning algorithm, what is our training/testing/CV plan?
    - Data preparation, feature engineering, model selection, cross validation, model evaluation, result interpretation.
- If there is any tool you expect to use for processing/analysis, please include it here.
    - Python, PyTorch, other library tools and data visualization tools.

Using LLM for analyzing Movie Overviews



Model Training

- Logistic Regression
  - To model the probability of box office failure as a function of the input features. It is well-suited for binary classification problems and can handle both categorical and continuous input features.
- Random Forest
  - To build an ensemble of decision trees that aggregate the input features and predict the box office success/failure. It can handle both categorical and continuous input features and can capture non-linear relationships and interactions between the input features and the target variable.
- XGBoost
  - XGBoost is an efficient machine learning algorithm that uses gradient boosting techniques to build an ensemble of decision trees. It iteratively adds trees to the model, minimizing a given loss function while incorporating regularization to avoid overfitting. It handles both regression and classification tasks well.

Model Training - Using Meta data

Logistic Regression

	precision	recall	f1-score
0	0.69	0.51	0.59
1	0.68	0.82	0.75
accuracy			0.69
macro avg	0.69	0.67	0.67
weighted avg	0.69	0.69	0.68

Test Accuracy: 0.6869597895527997
Test f1-score: 0.7478050257341811
Train Accuracy: 0.7028507005959092
Train f1-score: 0.7673098751418843

Random Forest

	precision	recall	f1-score
0	0.69	0.55	0.62
1	0.70	0.81	0.75
accuracy			0.70
macro avg	0.70	0.68	0.68
weighted avg	0.70	0.70	0.69

Test Accuracy: 0.6989853438556933
Test f1-score: 0.7525486561631141
Train Accuracy: 0.7721050088581092
Train f1-score: 0.8193078789426638

XGBoost

	precision	recall	f1-score
0	0.70	0.59	0.64
1	0.71	0.80	0.76
accuracy			0.71
macro avg	0.71	0.69	0.70
weighted avg	0.71	0.71	0.70

Test Accuracy: 0.7076287110108982
Test f1-score: 0.7556532663316582
Train Accuracy: 0.7216943147044612
Train f1-score: 0.7755844155844157

Model Training - Using Meta data + GPT rating

Logistic Regression

	precision	recall	f1-score
0	0.64	0.78	0.70
1	0.60	0.43	0.50
accuracy			0.63
macro avg	0.62	0.61	0.60
weighted avg	0.63	0.63	0.62

Test Accuracy: 0.629973474801061
Test f1-score: 0.5044404973357016
Train Accuracy: 0.6478953356086462
Train f1-score: 0.5067729083665338

Random Forest

	precision	recall	f1-score
0	0.65	0.78	0.71
1	0.61	0.46	0.53
accuracy			0.64
macro avg	0.63	0.62	0.62
weighted avg	0.63	0.64	0.63

Test Accuracy: 0.6392572944297082
Test f1-score: 0.5261324041811847
Train Accuracy: 0.8196814562002275
Train f1-score: 0.75893536121673

XGBoost

	precision	recall	f1-score
0	0.67	0.74	0.70
1	0.61	0.52	0.56
accuracy			0.65
macro avg	0.64	0.63	0.63
weighted avg	0.64	0.65	0.64

Test Accuracy: 0.6458885941644562
Test f1-score: 0.5601317957166392
Train Accuracy: 0.7440273037542662
Train f1-score: 0.6700879765395894

Model Training: Using Meta data + post release data

Logistic Regression

Training Accuracy: 0.9245495495495496
Testing Accuracy: 0.9121621621621622

	precision	recall	f1-score
0	0.91	1.00	0.95
1	0.62	0.05	0.09
accuracy			0.91
macro avg	0.76	0.52	0.52
weighted avg	0.89	0.91	0.88

Random Forest

Training Accuracy: 1.0
Testing Accuracy: 0.9245495495495496

	precision	recall	f1-score
0	0.93	0.99	0.96
1	0.68	0.30	0.41
accuracy			0.92
macro avg	0.81	0.64	0.69
weighted avg	0.91	0.92	0.91

XGBoost

Training Accuracy: 0.9439752252252253
Testing Accuracy: 0.9245495495495496

	precision	recall	f1-score
0	0.93	0.99	0.96
1	0.68	0.30	0.42
accuracy			0.92
macro avg	0.81	0.64	0.69
weighted avg	0.91	0.92	0.91

Model Training - Random Forest

- Random Forest on box failure ROI

```
Training Accuracy: 1.0
Testing Accuracy: 0.9206081081081081
              precision    recall  f1-score   support

           0       0.64      0.26      0.37       159
           1       0.93      0.99      0.96      1617

    accuracy                           0.92      1776
   macro avg       0.78      0.62      0.67      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91272523 0.93243243 0.92398649 0.92117117 0.92905405]
Mean Cross-Validation Score: 0.9238738738738739
```

Model Training - Random Forest

- Random Forest on profitability

```
Training Accuracy: 1.0
Testing Accuracy: 0.9245495495495496
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      1617
           1       0.68      0.30      0.41       159

    accuracy                           0.92      1776
   macro avg       0.81      0.64      0.69      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91385135 0.9329955  0.92454955 0.92117117 0.92792793]
Mean Cross-Validation Score: 0.924099099099099
```

Model Training - Logistic Regression

- Logistic Regression on box failure ROI

```
Training Accuracy: 1.0
Testing Accuracy: 0.9206081081081081
              precision    recall  f1-score   support

           0       0.64      0.26      0.37       159
           1       0.93      0.99      0.96      1617

    accuracy                           0.92      1776
   macro avg       0.78      0.62      0.67      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91272523 0.93243243 0.92398649 0.92117117 0.92905405]
Mean Cross-Validation Score: 0.9238738738738739
```

Model Training - Logistic Regression

- Logistic Regression on profitability

```
Training Accuracy: 1.0
Testing Accuracy: 0.9245495495495496
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      1617
           1       0.68      0.30      0.41       159

    accuracy                           0.92      1776
   macro avg       0.81      0.64      0.69      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91385135 0.9329955  0.92454955 0.92117117 0.92792793]
Mean Cross-Validation Score: 0.924099099099099
```

Model Training - Xgboost

- Xgboost on box failure ROI

```
Training Accuracy: 0.9439752252252253
Testing Accuracy: 0.9239864864864865
              precision    recall  f1-score   support

           0       0.67      0.30      0.42       159
           1       0.93      0.99      0.96      1617

    accuracy                           0.92      1776
   macro avg       0.80      0.64      0.69      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91722973 0.93637387 0.92849099 0.92286036 0.93130631]
Mean Cross-Validation Score: 0.9272522522522522
```

- Xgboost on profitability

```
Training Accuracy: 0.9439752252252253
Testing Accuracy: 0.9245495495495496
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      1617
           1       0.68      0.30      0.42       159

    accuracy                           0.92      1776
   macro avg       0.81      0.64      0.69      1776
weighted avg       0.91      0.92      0.91      1776

Cross-Validation Scores: [0.91722973 0.93637387 0.92792793 0.92286036 0.93130631]
Mean Cross-Validation Score: 0.9271396396396396
```

Validation Plan

- If you are doing machine learning, what metric (e,g, accuracy, area under the ROC) are we going to use on my algorithm?
  - Due to primary exploration of our dataset, we found that the Failure and Non-failure ratio are 89% and 11% (measured by if revenue covers the budget).
  - Since this is an imbalanced dataset, we cannot rely on accuracy as our primary metric. Instead, we will use metrics such as precision, recall, F1-score, and AUC-ROC to evaluate the performance of our algorithm.
- What is considered to be success in this project? Is it quantitative or qualitative?
  - We will incorporate multiple ways to evaluate the success of our prediction model. We will randomly set aside a validation set of movies preferably from within a year, and we will also try web scraping to enrich the validation set if needed.
  - Most importantly, we want to focus on correctly identify those movies that might fail to protect movie investors. Due to this goal, we want to build a model that has a high recall score on the positive (failure) class. This is because we want to correctly identify as many movie failures as possible.
- If you were presenting this to our manager, how would you prove the value of this analysis?
  - Our model would be most valuable for investors and movie production companies to maximize their investment in movies. They would be able to make better decisions on whether to invest in/ produce a movie based on our model prediction.
  - We can demonstrate the value of this analysis by showing the potential savings that can be achieved by identifying movie failures early on.
- What visuals might be useful in this analysis, that you would be able to extract from your data?
  - We will present in figures the outcome of the prediction model in terms of true positive, true negative, false negative and false positive rates, as well as the correlation between some of the most influential features we discovered and the prediction result.

References

- Harper, F. M., and J. A. Konstan. "The MovieLens Datasets: History and Context." ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, 2015, Article 19, doi:10.1145/2827872.
- Asano, Yohei, and Yoichi Motomura. "Large-Scale Movie Review Dataset for Sentiment Analysis." Proceedings of the 2015 International Conference on Asian Language Processing (IALP), IEEE, 2015, pp. 118-121, doi:10.1109/IALP.2015.7519394.
- Maltby, Richard. "Predicting box-office success: do critical reviews really matter?" Journal of Cultural Economics, vol. 24, no. 2, 2000, pp. 135-161.
- Narayanan, M., et al. "Box office prediction of movies using machine learning." 2019 IEEE 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 577-580.