

Predicting movie success

Group 12

Liang Shuyi

Lin Shuyan

Nguyen Thi Le Van

Widya Tantiya Yutika

Yen Yun-Hsuan

Contents

[Introduction 1](#_Toc139200571)

[Business Problem Statement 1](#_Toc139200572)

[Dataset and Preprocessing 1](#_Toc139200573)

[Overview of Dataset 1](#_Toc139200574)

[Supplementary Datasets 1](#_Toc139200575)

[Preprocessing for Model Training 2](#_Toc139200576)

[Topic Modelling 2](#_Toc139200577)

[Profitability Prediction 2](#_Toc139200578)

[Logistic Regression 3](#_Toc139200579)

[Decision Tree 4](#_Toc139200580)

[Comparison between Models 6](#_Toc139200581)

[Revenue Prediction 6](#_Toc139200582)

[Linear Regression 6](#_Toc139200583)

[Neural Network 7](#_Toc139200584)

[Comparison between Models 8](#_Toc139200585)

[Conclusion 8](#_Toc139200586)

[References 9](#_Toc139200587)

[Appendix 1 – Description of Datasets and Treaments 10](#_Toc139200588)

# Introduction

## Business Problem Statement

In 2021, the film industry was estimated to be US$50.3 billion, and it is expected to continue to show strong growth as the industry develops in regions outside the USA (Grand View Research, 2022). The COVID-19 pandemic has changed the way people consume movies; less people are going to the cinemas and streaming services have become more popular, changing the traditional revenue model of box office earnings (Grand View Research, 2022). A movie is a significant investment; most major Hollywood productions cost at least US$100 million, and movie budgets having increasing over time (Mueller, 2019). Faced with the changes in the industry, increased competition and costs, movie producers and makers adapt to ensure their products are profitable.

This project will develop machine learning models to predict the success of a movie, measured in profitability (binary classification) and revenue (regression). The models will be trained using metadata that should be available prior to production, such as storyline, production company, cast and budget. While not all these factors may be decided at the point of deciding whether to greenlight a proposal, movie makers can use the model to estimate the success of a potential movie with varying levels of investment to make data-driven decisions on the investment required to make the proposal successful.

While predicted profitability could be derived from the revenue prediction, we have developed models for both as variance in predicted revenue could make profitability prediction from the revenue unstable.

# Dataset and Preprocessing

## Overview of Dataset

The main dataset is a snapshot of movie metadata and ratings from TMDB and GroupLens obtained from Kaggle (<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>). It covers over 45,000 movies released on or before July 2017. 3 files were used: movies\_metadata, credits and keywords. Details on the variables and processing are in Appendix 1.

The tables below summarise the treatment of each variable. Variables that are only available after production and release (e.g. user ratings) were not used in the models.

## Supplementary Datasets

Additional datasets were used to supplement the main datasets for feature engineering purposes.

The budget and revenue data were not adjusted for inflation (TMDB, n.d). As there are movies from many different countries in the dataset, global inflation rate was used to adjust budget and revenue to 2018 dollars. The global annual inflation rate from 1980-2021 was obtained from <https://www.macrotrends.net/countries/WLD/world/inflation-rate-cpi#:~:text=World%20inflation%20rate%20for%202021,a%200.25%25%20increase%20from%202017/>

The dataset is not large enough to support one-hot encoded variables for all cast and crew. As such, only the top 100 actors/actresses (<https://www.imdb.com/chart/starmeter/>) and top 100 directors (<https://www.imdb.com/list/ls026411399/>) were considered in the model.

## Preprocessing for Model Training

The original dataset consists of more than 45,000 movies, but only 5,381 movies had valid budget and revenue information. On top of that, we excluded movies produced on or before 1980 as their information may not be relevant anymore, and the inflation rate data starts from 1980. The dataset has 4,835 unique movies.

Categorical and JSON variables were processed into one-hot encoded variables, with variables with more than 20 unique categories simplified into the 10 most frequent categories and an others category. The reduction in categories was done as the dataset is not large enough to support so many explanatory variables. Numerical variables with scales that are too different from the others were transformed (budget and revenue) or standard scaled (runtime and total cast). The detailed treatment of each variable is described in Table 1 above. Text variables were aggregated into a single document for each movie and processed using topic modelling into 10 one-hot encoded variables. The process is detailed in the section on topic modelling.

A 70:30 train/test split as adopted using random sampling.

## Topic Modelling

With only over 4000 records, we cannot vectorise all the words into high dimensional TFIDF matrix. Thus, topic modeling is implemented to reveal the underlying relationship among movies using overview, title and keywords, within a reasonable number of features. As BERTopic outputs many topics and generates an “empty” topic (Robinson, 2022), it is not suitable for our purposes. Latent Dirichlet Allocation (LDA) and Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) models were trained. The keyword lists of each topic were manually inspected to check if topics are distinct.

The 3 text variables were combined. Pre-processing of text data includes tokenization, lemmatization, remove stop words. A bigram model was also used to retain the meanings of linked words. Finally, a bag-of-words structure with corpus and dictionary are obtained and used as input to train the LDA and GSDMM models.

LDA topic modeling requires a pre-determined number of topics. Using the Elbow method on coherence score against number of topics, we chose 10 topics. Topics generated by LDA had a high degree of overlap with many words occurring in multiple topics. The reason could be due to short text content; the longest document was only 175 words. The relatively short text hampers the performance of LDA as it lacks co-occurrence of words within the documents.

LDA Topic Modelling Results

A picture containing text, font, screenshot, handwriting

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

In contrast, GSDMM is designed to work on shorter texts (Yin & Wang, 2014). It works by grouping documents based on similar words. Each document is initially assigned to a random topic out of k topics. The algorithm then iteratively assigns each document to a new topic that either (a) has more documents than its current topic, or (b) has documents with similar keywords (Hannachi et al., 2021). After sufficient iterations, the documents will be “optimally” distributed among topics such that topics are distinct, cohesive and no topic will have too few documents. The topics generated by GSDMM topics were more interpretable with distinct non-overlapping keywords. GSDMM topics will be used as features for subsequent machine learning tasks.

A picture containing screenshot, graphics, rectangle, design

Description automatically generatedGSDMM Topic Modelling Result

A picture containing text, font, screenshot, line

Description automatically generated

# Profitability Prediction

Profitability is defined as a binary outcome: profitable (1) when budget is less than revenue, and not profitable (0) when budget is equal to or greater than revenue. Investors can use these models to predict whether a potential movie will breakeven before considering return on investment from the revenue models in the next section of the report.

Model performance will be evaluated using accuracy, ROC, AUC, precision, recall, and F1 score. These metrics provide a comprehensive assessment of the model's overall accuracy and ability to discriminate between profitable and non-profitable movies.

## Logistic Regression

The unconstrained model with all features and model with the 10 most important features (chosen by recursive feature elimination) appeared to perform well with accuracy of ~70%. However, this was because the models over-predicted the profitable class which formed 70% of the dataset. It was poor at predicting non-profitable movies, which would cause investors to lose money if deployed.

Regression models are sensitive to multicollinearity. Language and production country features were highly correlated and excluded production country from subsequent models.

To prevent overfitting, we also trained L1 and L2 regularized models. GridSearch CV was used to find the best combination of solver and regularization parameter C. The L2 regularised model gave the best result with high accuracy and AUC of 0.72. However, recall for the not-profitable class was still poor.

A picture containing text, screenshot, font, number

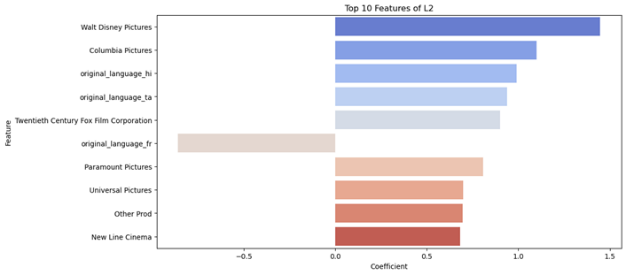
Description automatically generatedTable 1: Performance of Logistic Regression models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | | No constraints | RFE | L1 | **L2**  **(C=1, Liblinear)** | **SMOTE**  **(best)** |
| **Train Accuracy** | | 0.7 | 0.68 | 0.7 | **0.71** | **0.66** |
| **Test Accuracy** | | 0.69 | 0.67 | 0.69 | **0.69** | **0.65** |
| **[0]** | **Precision** | 0.57 | 0.63 | 0.59 | **0.6** | **0.49** |
| **Recall** | *0.31* | *0.05* | *0.30* | **0.32** | **0.60** |
| **F1-score** | 0.39 | 0.09 | 0.40 | **0.41** | **0.54** |
| **[1]** | **Precision** | 0.72 | 0.67 | 0.7 | **0.72** | **0.77** |
| **Recall** | 0.8 | 0.99 | 0.89 | **0.90** | **0.69** |
| **F1-score** | 0.79 | 0.8 | 0.80 | **0.8** | **0.73** |

To improve prediction on the negative class, we also trained models with SMOTE oversampling method and class weight adjustment. This improved prediction ability for the not-profitable class (higher recall and F1 score) at the expense of slightly reduced accuracy in the profitable class. Nonetheless, the SMOTE model would be preferred for this use case as it would be more important for investors to identify projects that will not breakeven.

Considering the difficulty in estimating budgets during the initial stages of movie production, we trained L2 regularization models without budget to better align with real-world business scenario. It had similar performance and top features as the L2 model without budget.

The figures below show the top 10 features for the L2 and SMOTE models. Many of the important features were shared between the two models. Both models emphasize the significance of movie production companies, particularly Walt Disney Pictures, in predicting profitability.

A picture containing text, line, number, plot

Description automatically generated

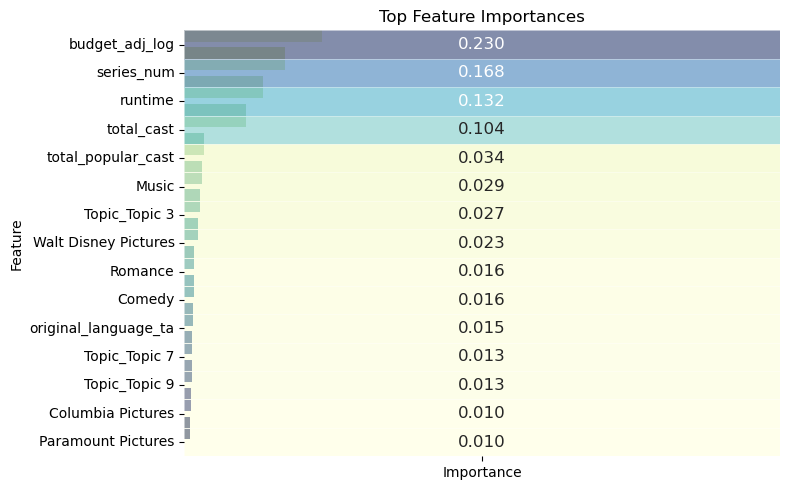
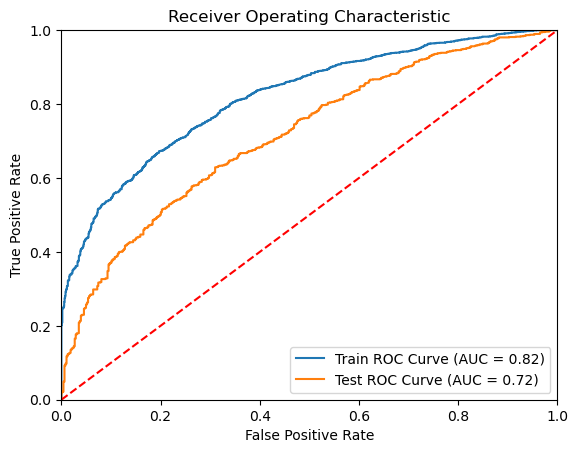
## Decision Tree

The hyperparameters tested were split criterion (Gini, Entropy), splitter (best, random), max depth, min samples split, min sample leaf. In addition, ensemble models were also trained (using grid search for hyperparameter tuning of number of estimators, algorithm and learning rate). Since the data set is relatively small, stratified K-fold cross validation was used to efficiently utilize the amount of data and enforce class ratio in each portion, providing more robust estimate of model performance.

Table 2: Performance of decision tree models

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Metric** | | No constraint | Random Search | Grid Search | Bayesian Optimization | Random Forest | **Ada Boost**  **(best)** | Gradient Boosting |
| **Train Accuracy** | | 1 | 0.7037 | 0.7081 | 0.7037 | 0.8256 | **0.7592** | 0.7530 |
| **Test Accuracy** | | 0.6141 | 0.6763 | 0.6673 | 0.6763 | 0.7019 | **0.6970** | 0.6977 |
| **[0]** | **Precision** | 0.42 | 0.55 | 0.51 | 0.55 | 0.66 | **0.58** | 0.60 |
| ***Recall*** | *0.41* | *0.17* | *0.20* | *0.17* | *0.23* | ***0.34*** | *0.30* |
| **F1-score** | 0.41 | 0.26 | 0.29 | 0.26 | 0.34 | **0.43** | 0.40 |
| **[1]** | **Precision** | 0.71 | 0.69 | 0.69 | 0.69 | 0.71 | **0.73** | 0.72 |
| **Recall** | 0.72 | 0.93 | 0.90 | 0.93 | 0.94 | **0.88** | 0.90 |
| **F1-score** | 0.71 | 0.79 | 0.78 | 0.79 | 0.81 | **0.79** | 0.80 |

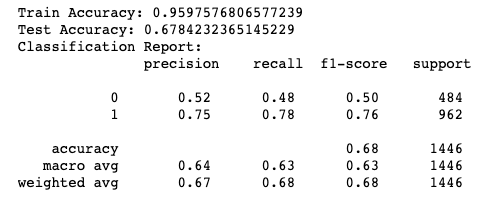
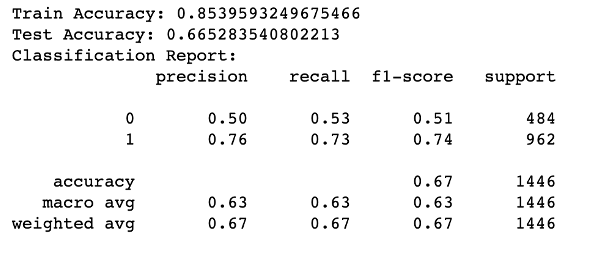
The ensemble models performed slightly better than the standalone models. The Ada Boost model as the model with highest accuracy of 0.69. However, there was a gap between train and test accuracy, indicating slight overfitting. Similar to the logistic regression model, the decision tree was much better at predicted the profitable class than the non-profitable class.



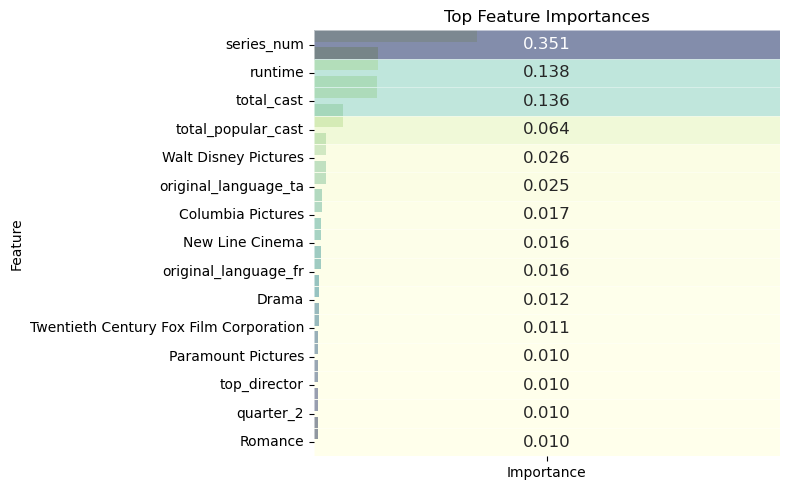
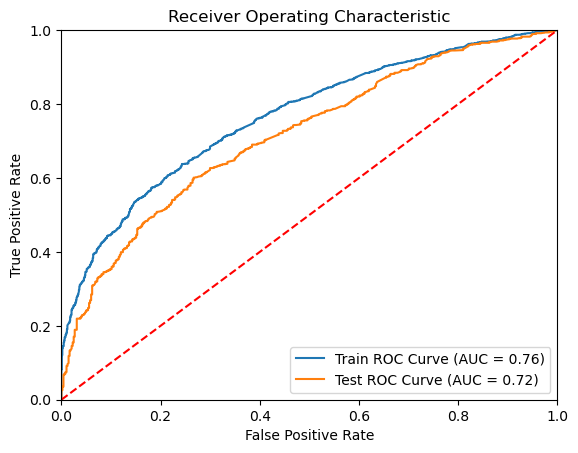
The Ada boost model has a Test AUC of 0.72 (on the low side compared to Train AUC of 0.82). The top important features include budget, number in series, run time, total cast, total popular cast.

To address the class imbalance, we used SMOTE to oversample on train data set to boost performance on class [0]. We achieved higher precision and recall for class [0] at the expense of overall. However, both models had a big gap between train and test accuracy, indicating overfitting.

**Oversampling for Ada Boost**  **Oversampling for Gradient Boosting**



Budget was the most important feature in the best decision tree, but it is one of the most difficult features to obtain before production. As such, we also built an Ada boost model without budget. Contrary to expectations, it still achieves the same accuracy of 0.697. Number of movies in series becomes the most important feature.



## Comparison between Models

Both the logistic regression and decision tree models had similar performance. The SMOTE logistic regression was slightly better at predicting the not-profitable class. Both models still had similar overall accuracy even after removing budget, meaning that they are somewhat robust to uncertain budget.

# Revenue Prediction

The revenue models used mean squared error (MSE) or root mean squared error (RMSE) as the cost function during training. Final evaluation on the test dataset used RMSE, MSE, and adjusted R2. MSE/RMSE measures prediction error using the average squared difference between the predicted and actual revenue. Lower values indicate better accuracy.

On the other hand, adjusted R2 measures the proportion of the total variation in revenue that can be explained by the independent variables, which provides insight on how well the model is able to capture the underlying patterns of the data. Higher adjusted R2indicates that the model can explain most of the variation in revenue. It was chosen as an additional performance measure because MSE/RMSE is dependent on the scale of the target variable while adjusted R2can be directly compared between models.

## Linear Regression

For linear regression, we conducted feature selection using backward elimination and trained lasso and ridge regularization models to avoid overfitting. Performance is summarized below.

Table 3: Performance of linear regression models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Description** | **Mean Revenue** | **Std Dev Revenue** | **RMSE**  **Train** | **Adjusted**  **R2 Train** | **RMSE**  **Test** | **Adjusted**  **R2 Test** |
| Removal of highly correlated variables | 7.521 | 1.166 | 0.724 | 0.599 | 0.776 | 0.538 |
| **Use variables from Backward Elimination (Best Model)** | **7.521** | **1.166** | **0.729** | **0.598** | **0.782** | **0.542** |
| Regularization: Lasso  (GridsearchCV best alpha: 0.001) | 7.521 | 1.166 | 0.726 | 0.597 | 0.781 | 0.533 |
| Regularization: Ridge  (GridsearchCV best alpha:10) | 7.521 | 1.166 | 0.725 | 0.598 | 0.778 | 0.536 |
| Remove budget column | 7.521 | 1.166 | 0.919 | 0.356 | 0.945 | 0.316 |

Most of the models had similar performance in terms of RMSE except the model without budget which performed very poorly. Since it is essential to understand how well the independent variables collectively explain the variation in revenue, model with highest R2 is chosen as the best model (backward elimination).

A picture containing text, screenshot, line, plot

Description automatically generatedA picture containing text, screenshot, diagram, line

Description automatically generated

As shown from the plot above (from best model), the model is good for predicting higher revenue movies, but performs poorly for low revenue movies. This may be because data on low revenue movies is limited.

A picture containing screenshot, diagram, line

Description automatically generatedA picture containing text, screenshot, diagram, font

Description automatically generated

The most important feature in predicting revenue is budget. The other top 10 features are mostly from production companies and language (this is also observed for model without budget). Budget, production companies and language are essential features to predict movie revenue for linear regression.

## Neural Network

Fully connected neural network can also be used for predicting continuous target variables. They have the advantage over traditional linear regression models in that they can learn non-linear relationships. They are also more robust against multicollinearity.

To test different network architectures and hyperparameters, the test dataset was further divided using random sampling to get a 70:10:20 train/validation/test ratio. Model architecture was between 2-5 layers with a decreasing number of neurons in each layer, and activation functions tried were ReLu, leaky ReLu and linear. Other hyper parameters tested were batch size and number of epochs.

Initial networks had the issue of not learning further after the first epoch regardless of architecture or hyperparameters. It was determined that this was due to incorrect specification of inputs with some of the numerical inputs being in a different scale from the others. As such, numerical variables that were not already scaled in the pre-processing step were standard scaled (budget, revenue, top100 cast).

Table 4 summarises performance of models with different input variables. The final architecture chosen was a relatively simple one with only one hidden layer of 30 neurons.

Table 4: Performance of neural network models

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Test RMSE** | **Test adj-R2** |
| Remove production country | 0.426 | 0.538 |
| Remove topics & production country | 0.426 | 0.543 |
| Remove topics | 0.423 | 0.540 |
| **Linear regression backward elimination vars (Best Model)** | **0.425** | **0.558** |
| Best model, remove budget | 0.624 | 0.350 |
| Remove production country & budget | 0.467 | 0.493 |

## Comparison between Models

The neural network is slightly better at predicting revenue than the linear regression (adjusted R2 of 0.558 >0.542), although both models would be considered underfitted.

Budget was one of the most important explanatory variables in the linear regression model. While the neural network performs better when budget is included, the improvement in performance is not sizeable. If the user is unsure of the estimated budget, the neural network without budget can still provide an estimate. Nonetheless, prediction from both models would likely have a large confidence interval, especially for the neural network result which is both log transformed and standard scaled.

# Conclusion

The team has developed 4 predictive models to predict both profitability and revenue and found that the models are generally underfitted. This could be because of missing explanatory variables, poor quality of input or that the dataset is too small. Budget and revenue information on TMDB are user submitted, and many studios do not disclose the production or marketing budget and revenue (Zipin, 2021). Coupled with the evolution in movie revenue models, the input data may not be accurate enough to train effective models.

In all models, involvement of one of the top 10 production companies was a significant positive explanatory variable. Successful production companies likely already have a good method for predicting the success of a potential movie, hence their movies are more likely to succeed, and they would likely not have much use for our prediction models. Nonetheless, the models could be useful for smaller producers with less resources at their disposal. Future work could focus more on smaller scale productions which have less certain outcomes.

Budget was a key explanatory variable in the revenue models and removal resulted in significantly worse predictions. However, the profitability models were relatively robust to removal of budget. This means that predicting return on investment is highly dependent on accurate estimation of budget. In any project, not just moviemaking, actual expenditure will likely differ greatly from estimated budget. When testing various budget levels, there must be sanity checks on the feasibility of the proposed budget to get a more meaningful revenue prediction. Nonetheless, the profitability models can still give an indication on whether the move will breakeven even without budget.

# References

Grand View Research. (2022). *Movies and entertainment market size, share & trends analysis report by product (movies, music & videos), by region, and segment forecasts, 2022 – 2030*. Grand View Research. <https://www.grandviewresearch.com/industry-analysis/movies-entertainment-market>

Hannachi, S., Najar, F, Bouguila, N. (2021). Short text clustering using generalized Dirichlet multinomial mixture model. Communications in Computer and Information Science (Vol. 1371). <https://link.springer.com/chapter/10.1007/978-981-16-1685-3_13>

Mueller, A. (2022, July 31). *Why movies cost so much to make*. Investopedia. <https://www.investopedia.com/financial-edge/0611/why-movies-cost-so-much-to-make.aspx>

Robinson, D. (2022, July 7). *Is LDA topic modeling dead?* Towards Data Science. <https://towardsdatascience.com/is-lda-topic-modeling-dead-9543c18488fa#:~:text=From%20a%20practical%20standpoint%20BERTopic,less%20resource%20intensive%20than%20LDA>.

The Movie Database (TMDB). *Movie bible*. The Movie Database. <https://www.themoviedb.org/bible/movie#59f73b759251416e71000007>

Yin, J. & Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Digital Library. <https://dl.acm.org/doi/abs/10.1145/2623330.2623715>

Zipin, D. (2021, December 18) *How exactly do movies make money?* Investopedia. <https://www.investopedia.com/articles/investing/093015/how-exactly-do-movies-make-money.asp>

# Appendix 1 – Description of Datasets and Treaments

**movie\_metadata**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Type** | **Description (TMDB, n.d)** | **Treatment** |
| 1 | adult | Categorical | 0/1 for adult movie | Not used |
| 2 | belongs\_to\_collection | Text | Name of franchise | * Binary indicator for franchise * Number of movies in franchise |
| 3 | budget | Numerical | Budget (user submission) | Adjusted to 2018-dollars and log10 transformed |
| 4 | genres | Json | Genres | One-hot encoding (multi-label) |
| 5 | homepage | Text | Movie homepage | Not used |
| 6 | id | Numerical | Identifier | Not used |
| 7 | imdb\_id | Numerical | IMDB identifier | Not used |
| 8 | original\_language | Categorical | Original language | One-hot encoding (top 10 by frequency) |
| 9 | original\_title | Text | Original movie title | Not used |
| 10 | overview | Text | Official movie description | Used in topic modelling |
| 11 | popularity | Numerical | Lifetime popularity score | Not used |
| 12 | poster\_path | Text | Link to poster | Not used |
| 13 | production\_companies | Json | Production companies | One-hot encoding (top 10 by frequency, multi-label) |
| 14 | production\_countries | Categorical | Production country | One-hot encoding (top 10 by frequency) |
| 15 | release\_date | Numerical | Release date | One-hot encoding for release quarter |
| 16 | revenue | Numerical | Budget (user submission) | Adjusted to 2018-dollars and log10 transformed |
| 17 | runtime | Numerical | Length in minutes | Standard scaled |
| 18 | spoken\_languages | Json | Language spoken | Not used |
| 19 | status | Categorical | Release status | Not used |
| 20 | tagline | Text | Poster promotional text | Used in topic modelling |
| 21 | title | Text | English title | Used in topic modelling |
| 22 | video | Categorical | Indicator for other media | Not used |
| 23 | vote\_average | Numerical | Average of user rating | Not used |
| 24 | vote\_count | Numerical | No of user ratings | Not used |

**credits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Type | Description (TMDB, n.d) | Treatment |
| 1 | id | Numerical | Movie identifier | Not used |
| 2 | cast | Json | Actor name and other metadata, character played | * Actor names matched with top100 actor list to generate number of top100 actors in movie * Total cast count (standard scaled) |
| 3 | crew | Json | Crew member name and role | Director name matched with top 100 director list to get binary indicator for top100 director |

**keywords**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Type | Description (TMDB, n.d) | Treatment |
| 1 | id | Numerical | Movie identifier | Not used |
| 2 | keywords | Json | Keywords describing movie | Used in topic modelling |