BRAINSTATION DATA SCIENCE

PREDICTING SUCCESS STATUS BASED ON FILM SCRIPT TEXT

EVALUATING EXISTING SCRIPTS & FILM PERFORMANCE TO PREDICT SUCCESS

Introduction

Greenlighting film scripts requires experts, time, multiple opinions and money. These processes are still integral to the actual greenlighting of a film script and cannot be done away with, but could there be a way to develop a heuristic using data science that aids in filtering scripts based on the script and other related data ?

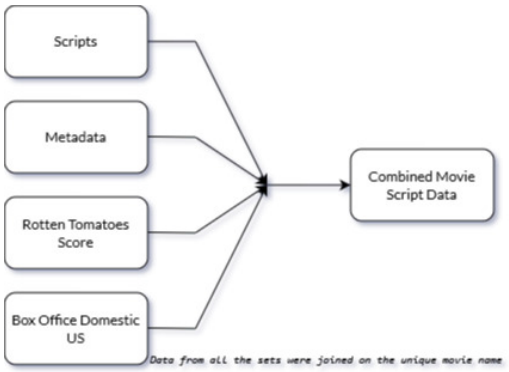
Data Source

Movie scripts - Scraped from Actorpoint

Movie metadata - The Movies Dataset-Kaggle

Rotten tomatoes - Scraped from Rotten Tomatoes

Box Office data - BoxofficeMojo Kaggle



The Score

A combined score was calculated based on the box office performance, imdb score and rotten tomatoes scores. The method used to calculate the score is detailed below.

I started out with trying to predict the movie score as a regression problem but given the size of the dataset, did not get favourable results. In order to simplify the problem I converted it to a classification problem with a success score of 55 as the threshold.

Data

After processing the scripts we got the following derived columns using the nltk tokenizer

Column	Description
movie_name	Name of the movie.
script	The entire script of the movie.
spoken languages	Languages spoken in the movie.
genres	All the genres that the movie falls into.
is_success	Whether the movie is / isn't a success (success_score >55).
success score	Calculated success score.
belongs to collection	Whether the movie belongs to a franchise of movies.
year	Year of release.
prop_count	Count of proper nouns.
verb_unique_percent	Percentage of unique verbs.
noun_unique_percent	Percentage of unique nouns.
verb_percent	Percentage of verbs.
noun_percent	Percentage of nouns.
adj_percent	Percentage of nouns.
prop_nunique	Count of unique proper nouns.
adj_unique_percent	Percentage of unique adjectives.
adv_unique_percent	Percentage of unique adverbs.

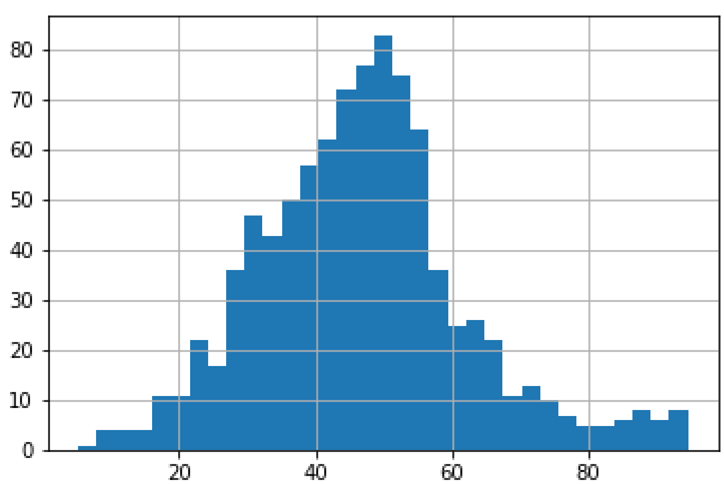
Methodology

All the above features were converted to numerical by using the one hot encoder method and were used to predict whether the movie would be successful based on the script features. We used Logistic regression, Decision trees and K Nearest neighbors to predict accuracy of the model.

EDA

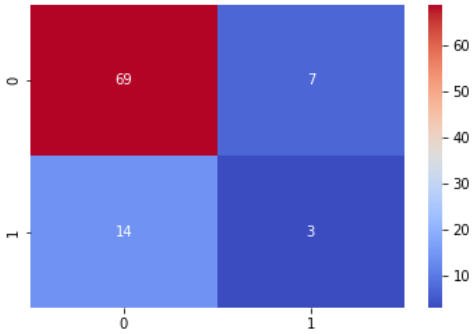
WordClouds were used to visualize the most common words by genre. e.g. ‘Horse’ was a very common word in westerns whereas one of the most common words in action scripts was ‘eye’. This gave me enough indication to support my hypothesis that different kinds of movies use different topic and they provide enough signal to distinguish between the genres/topics.

I also looked at the distribution of success scores to make sure the scores are not too skewed towards one end of the spectrum or the other. As expected, we find the bulk of the movies in the middle of the distribution.



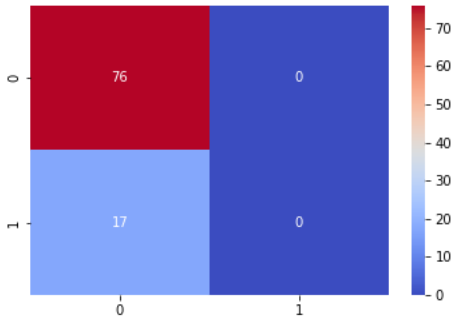
In order to combine the different data sources together, I needed to use a partial string matching function since there was no other common key between the data sources and the movie names were sometimes slightly different across datasets. For this I used Levenshtein distance matching from fuzzywuzzy library to match with the most similar movie name across datasets.

Modelling results



Logistic Regression

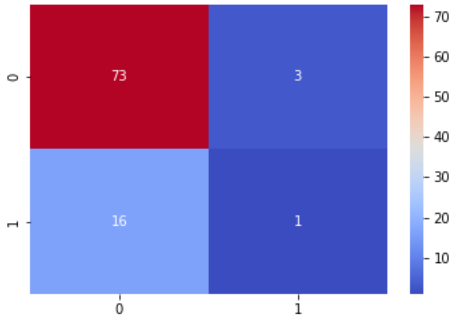
The Logistic regression model gives us 7 false positives, 14 false negatives, 69 true negatives and 3 true positives. This model uses only the numerical features.



K Nearest Neighbours

Based on this confusion matrix, it shows that the KNN model is heavily biassed towards predicting that the movie won't be a success based on the script features. This model uses only the numerical features.

This isn't a very good model given our current data and target variable.



Decision tree

The Decision tree model with 3 decision trees gives us 3 false positives, 16 false negatives, 73 true negatives and 1 true positive. This model uses only the text features based on word counts in the script.

Conclusion

We can conclude that there is a positive indication towards our ability to predict whether a script would make a successful movie or not but we need to supplement our data with more scripts. The current dataset is too small to catch a pattern in the successful scripts.

Scraping the scripts from different sources or even across the same website ran into issues because of different formatting and different HTML tags being used and needed a lot of manual checking to fix in order to get uniformity in the data. I envision continuing to work on this project and create a clean dataset of movie scripts that are uniform and ready to process and make it publicly available.

I would also like to experiment with a model that uses both text and numerical features together and once I have enough movie scripts, use some deep learning techniques to process them.