Movie Success Prediction Model

May 2023

Meet Our Team

Brittni
Breese

David
Duran

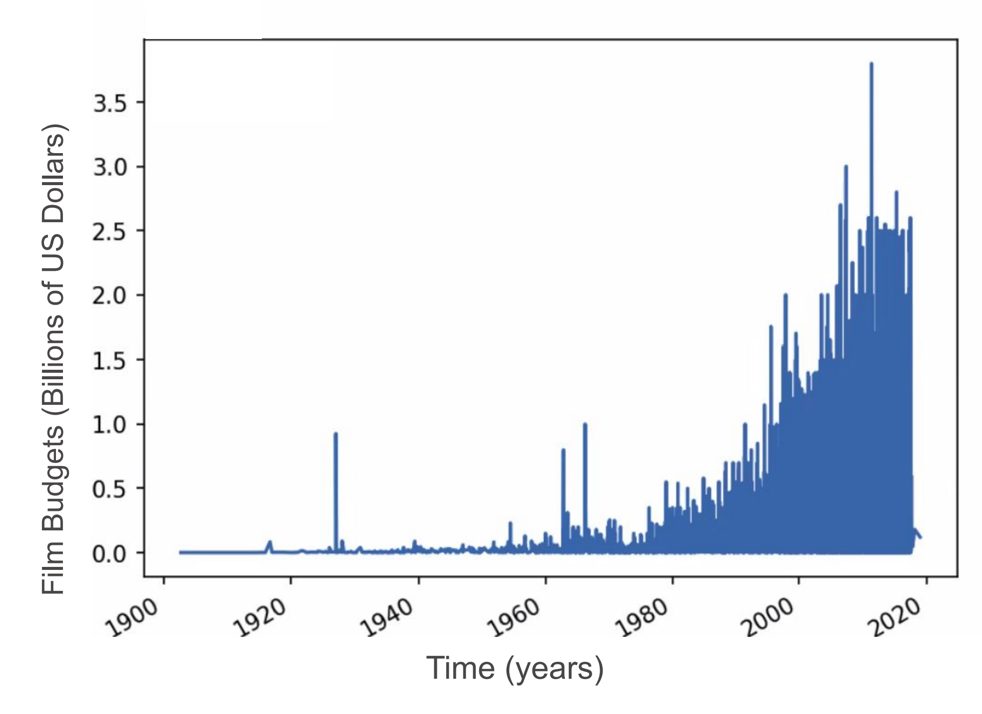
Nicole
Campos

Yash
Kansal

Vanessa
Anguiano

Johnny
Hollywood

Context & Objective



Question: What is the best way to fund movie projects while reducing financial risk?

Objective: Using data that tracks movie budget and revenue, we sought to develop a machine learning model that predicts whether a film would be successful.

Data Source: Kaggle

Dataset Overview

The Movies Dataset

- Found dataset on Kaggle
    - Source: TMDB Open API and GroupLens
- Dataset contains over 45,000 movies
- Original parameters state only movies released on or before July 2017 are included
    - We found outliers exceeding the 2017 limit
- Only used one file: movies_metadata.csv
    - Original dataset has 24 columns
    - Columns include features like release dates, revenue, budget, id, languages

EDA

Link

ETL Process

Data Extraction & Loading

- Downloaded CSV from Kaggle
- Used Pandas to import data in Jupyter Notebook
- Created dataframe and explored features

* Since the dataset was smaller, we decided storing the data in a database was unnecessary

Data Transformations

We excluded data that were unlikely to enhance model performance (at this time):

- Homepage
- Original Title
- Overview
- Belongs to Collection
- Tagline
- Video
- Poster
- IMDB ID
- Rows with N/As

We also developed qualifying criteria for movies with relevant data for our model:

- Full movie (60 minutes)
- IMDB Vote_count = 100+
- Budget: > $1M
- Status: Released
- Revenue: Not 0

Any data that did not fit the above criteria were removed.

Feature Engineering

In order to speed up data transformations and enhancing model accuracy, we simplified our data set and added new features:

Simplified Features

- **Language:** English vs. Foreign
- **Release Date:**
  - Pre-streaming (<2005) vs. Post-Streaming (≥2005)
  - Change the dates to month numbers (1-12)

Added Features

- **Anticipated Vote Rating:** Weights for every genre by vote rating to predict vote ratings
- **Anticipated Popularity:** Weights for every genre popularity to predict popularity
- **Target:** Net positive revenue vs. budget

Final Preprocessing Steps

	Method	Purpose
1.	`get_dummies()`	Converted categorical language variables into dummy/indicator variables (0 and 1) so the model does not assume correlation across the variables
2.	`train_test_split()`	Split the data into train and test sets, allowing model performance comparison on data that was not used to train the model
3.	`StandardScaler()`	Standardized variables in the same range (-1 and 1) and in the same scale so that no variable dominates other variables

Working with the Dataset

Dataset		Balanced Accuracy Scores (Random Forest Model)
Regular Train & Test Split (1: 2140, 0: 619)	....	0.50761
Oversampled Train & Test Split (1: 2140, 0: 2140)	....	0.62580
Doubled Train & Test Split (X -> 2X) (1: 4309, 0: 1209)	....	0.53062
Doubled & Oversampled Train & Test Split (1: 4309, 0: 4309)	....	0.79461

Model Development

Random Forest

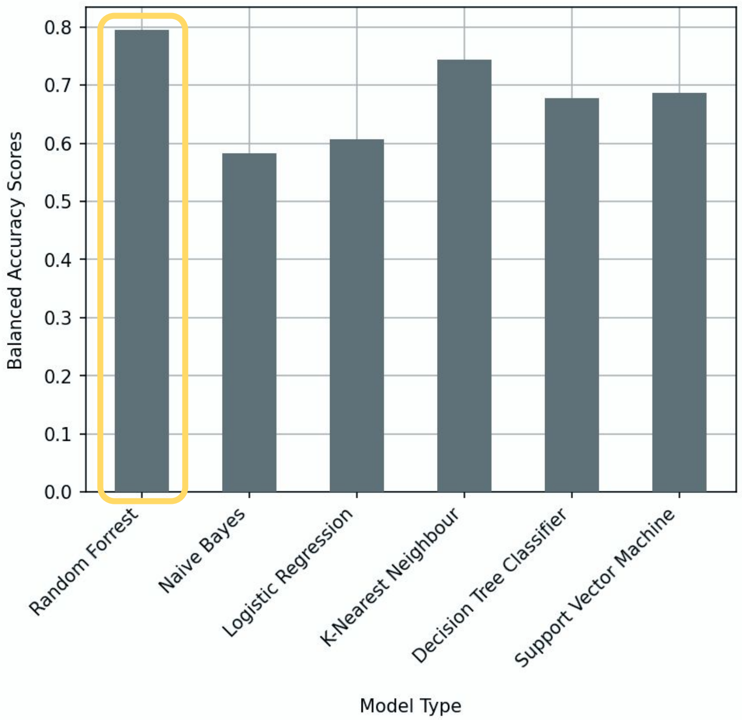
Naive Bayes

Logistic Regression

KNN

Decision Tree

Support Vector Machine

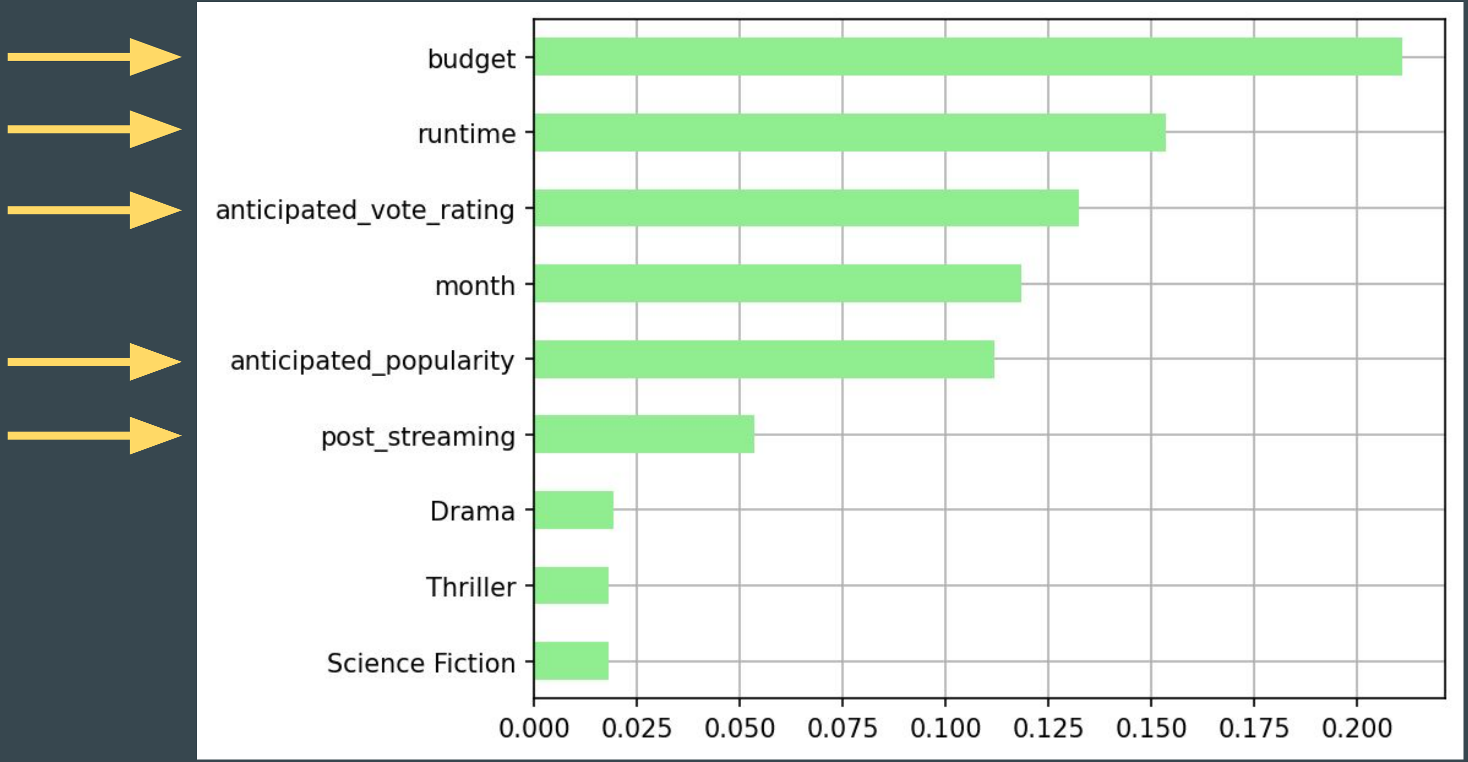


RF Balanced Accuracy Score: 0.79461

Final Model Performance

	Precision	Recall	f1-score	Support
Unsuccessful (0)	0.50	0.83	0.62	415
Successful (1)	0.94	0.76	0.84	1425
Accuracy			0.77	1840
Macro Avg	0.72	0.79	0.73	1840
Weighted Avg	0.84	0.77	0.79	1840

Understanding Feature Importance



DEMO

Challenges

- Limited dataset
- No MPA ratings (Motion Picture Association)
- Unable to use IMDB

Recommendations

- Improving the accuracy score
- A more detailed user app
- Sentiment analysis


TICKET