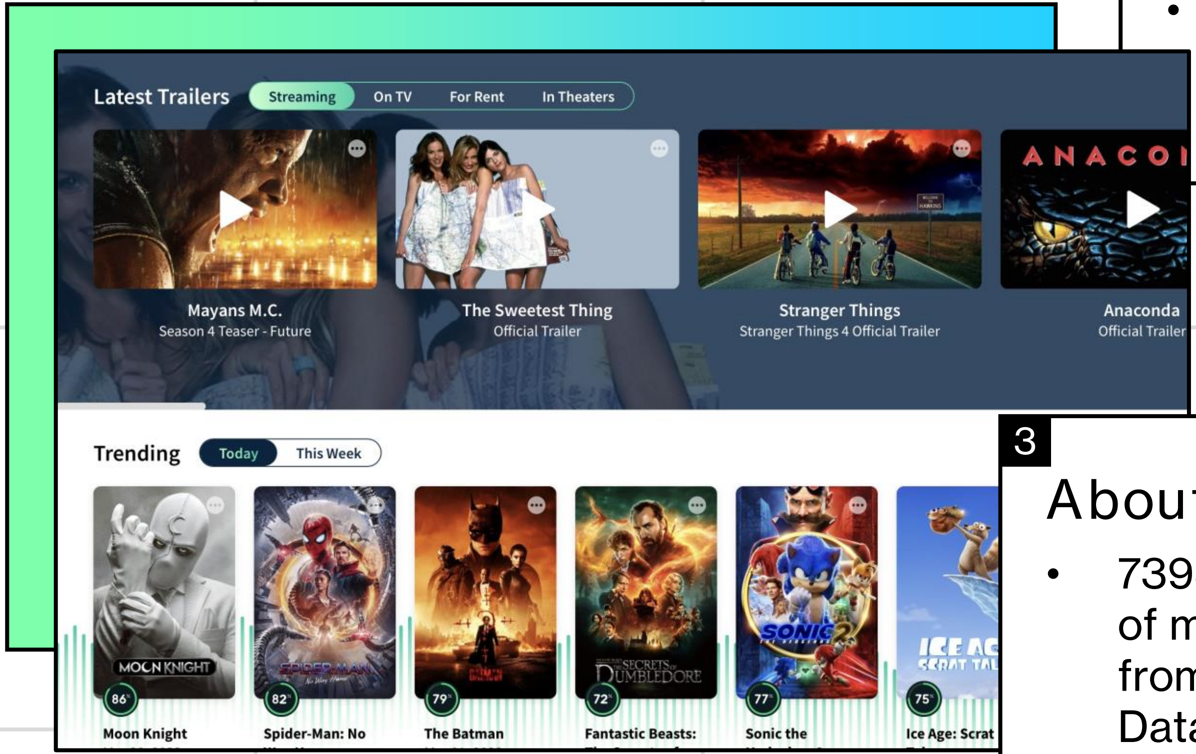
EDA and Revenue Prediction of TMDB Box Office Dataset

SDSC2102 Group Presentation

Du, Junye
56641800
Wu, Jianrui
56641885
Yang, Wentao
56643528
Zhou, Xin
56644501

Introduction



1

Background

- Booming film industry
- Numerous factors that make contributions
- Hard to predict-TMDB Box Office Prediction Challenge

2

Objectives

- Evaluate the contribution of each factor to a movie
- Explore possible methods to predict the revenue of a movie
- Predict the overall revenue (4398 movies)

3

About our dataset

- 7398 movies and a variety of metadata obtained from The Movie Database (TMDB)
- Data points including cast, crew, etc are provided to evaluate the revenue

The Problems

1

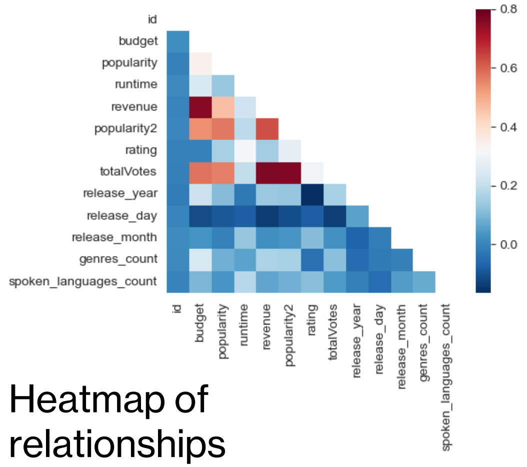
Mess metadata

Filter the data during data cleaning.

2

Prediction with numerous data points

Explore and identify strong relationships between points and find effective data.



Heatmap of relationships

3

Quantifiable expected outcomes and unquantifiable data points

Efficient tools are used to make points quantifiable and to measure the contribution of points “director” and “actors”.

director	Steve Pink
actors	[Rob Corddry, Craig Robinson, Clark Duke]
director	Garry Marshall
actors	[Anne Hathaway, Julie Andrews, H√©ctor Elizondo]

Unquantifiable attributes

LightGBM

XGBoost

CatBoost

Efficient tools we used

Preprocessing

1

Various redundant and unnecessary data

Filter and extract useful information from the messy dataset. Unwanted columns are dropped.

2

Large number of missing values in different columns

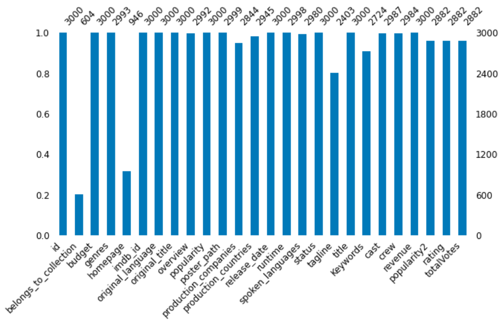
Fill them with supplementary materials found online, or with the mean of its relative column for consistency.

3

Unstandardized dates

Format them properly.

Missing values (distance between the bars and the top) in training data – quite a lot



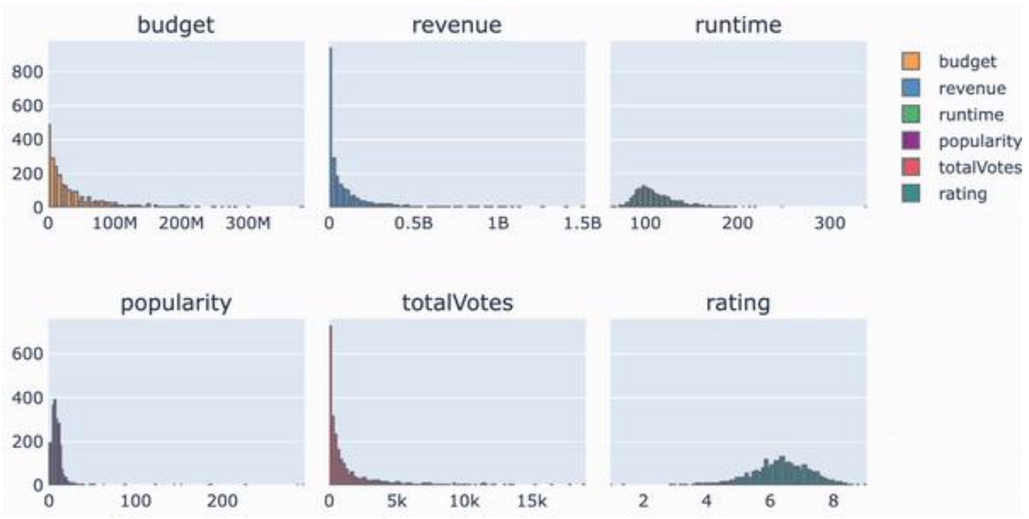
id	**1**
budget	**14000000**
genres	**[Comedy]**
original_language	**en**
original_title	**Hot Tub Time Machine 2**
popularity	**6.575393**
production_companies	**Paramount Pictures**
production_countries	**United State of America**
runtime	**93.0**
spoken_languages	**English**
status	**Released**
title	**Hot Tub Time Machine 2**
revenue	**12314651**
popularity2	**10.4**
rating	**5.0**
totalVotes	**482.0**
director	**Steve Pink**
actors	**[Rob Corddry, et al.]**
release_year	**2015**
release_day	**4**
release_month	**2**
genres_count	**1**
spoken_languages_count	**1**

Dataframe after preprocessing

Our Analysis

Distributions of important numeric features

1. The number of films decrease as budget, revenue, popularity, total votes increase. Also, the number of films show an approximate normal distribution over runtime and rating.
2. Most films' budgets are below $100m, and revenues are below $0.5b. Budget is positively correlated with revenue to some extend.
3. Features other than budget do not show clear relationships with revenues, their values are concentrated in a certain range.



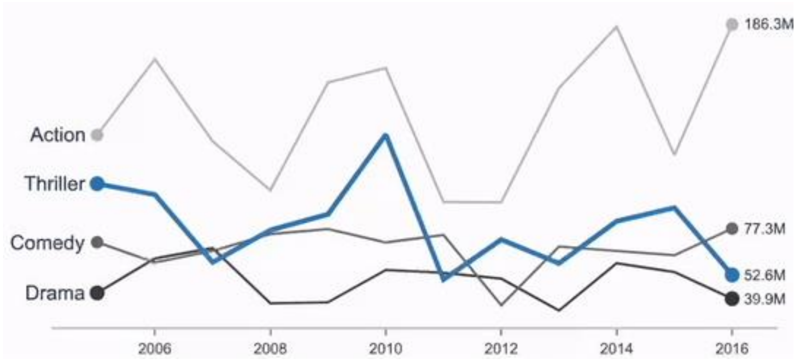
Histograms

Scatter plots

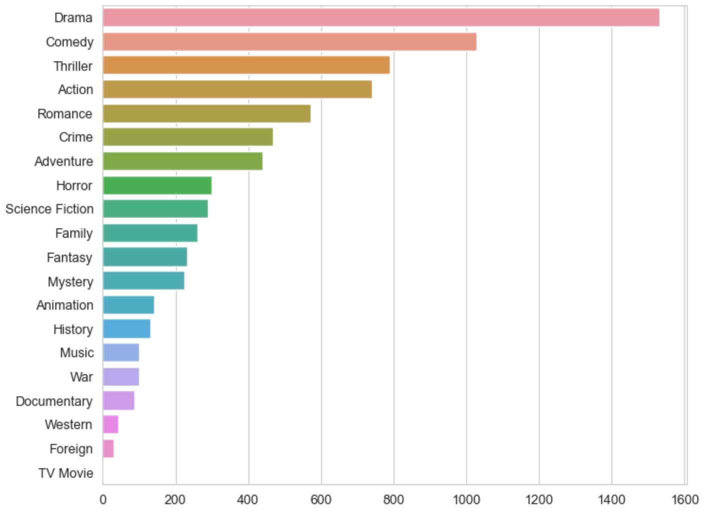
Our Analysis

Distributions of important categorical features

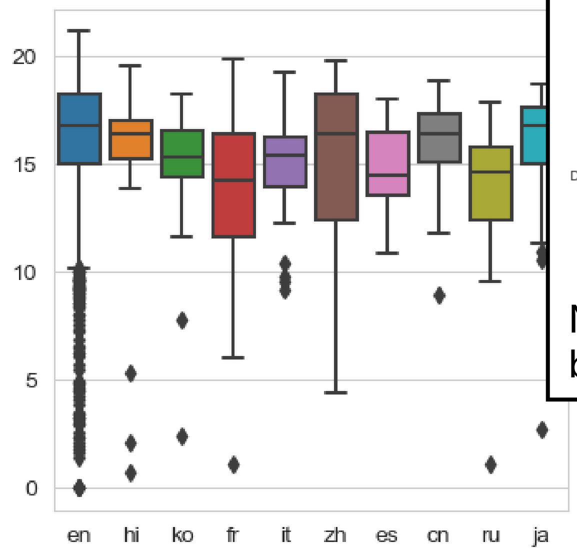
1. The top 4 movies genres including overlapping are drama, comedy, thriller and action. The mean revenue of action movies rises rapidly after 2000 and surpasses that in other three genres whose mean revenue keeps fluctuating.
2. There are more English movies than languages, and more English movies with highest revenues than others. However, The median revenue of Japanese, Chinese and Hindi is almost the same with English.



Movies' mean revenue of top 4 genres over the years



Number of movies belonging to each genre



Log revenue vs. original language

Our Method

1

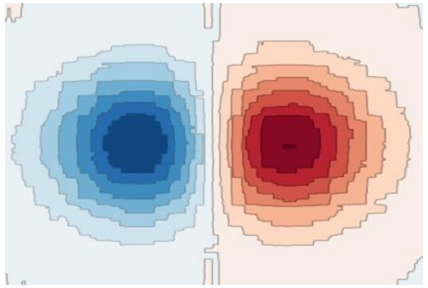
Tree-based methods: good interpretability

A good measure of each variable’s importance: the total amount that the loss function is decreased due to splits over a given predictor, averaged over all sub-trees.

2

Boosting: high accuracy

Trees are sequentially grown using information from previously grown trees. By fitting small trees to the remained loss, we slowly improve the model in areas where it does not perform well.



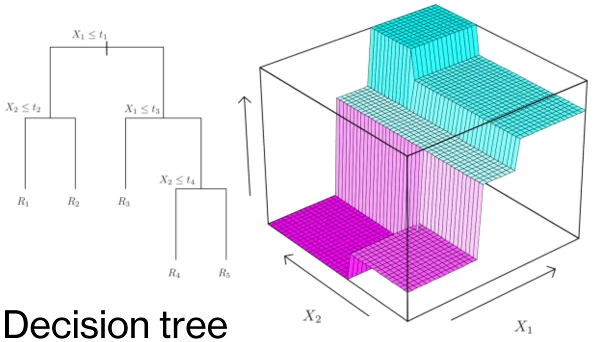
Boosting
– reducing the bias

predictors randomly selected

bootstrap

majority vote

Bagging; random forests
– reducing the variance

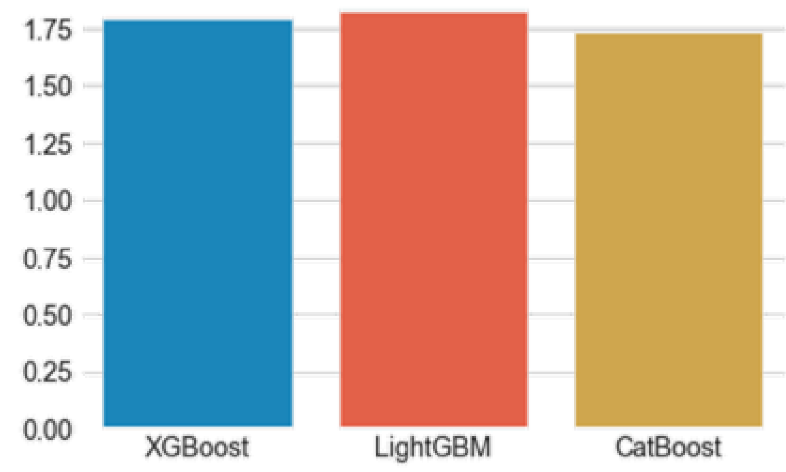


Decision tree
– highly interpretable

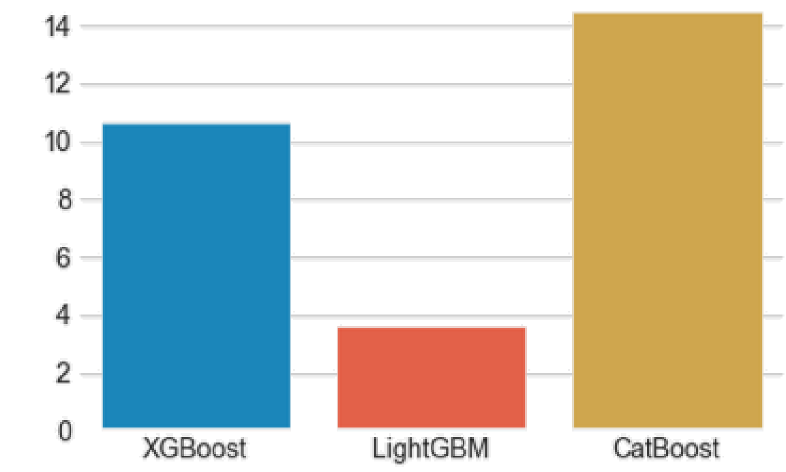
Our Method

Different implementations, different performance

XGBoost, LightGBM and CatBoost are three famous implementations of the (improved) gradient boosting algorithm. Here, the parameters are set to make them give similar cross validation RMSEs to each other, and LightGBM is the fastest to fit. Hence, in the following part we will explore LightGBM and our dataset further.



RMSE



Fit time, seconds

Interpretation

Feature importance

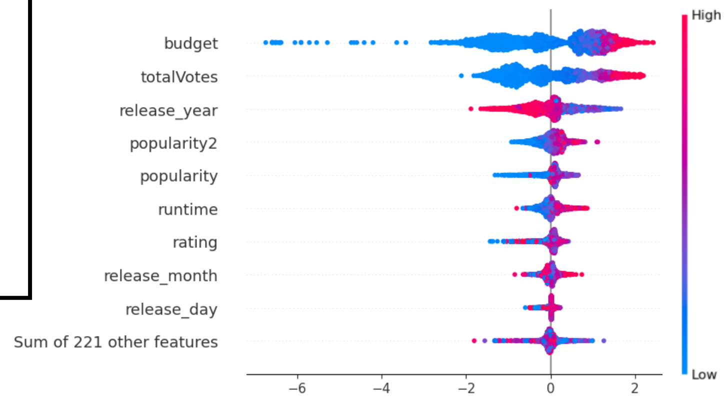
1. Weight of different features: sorting the features according to their weights
2. Feature density scatterplot: summarizing the effect of all features
3. Global effect plot: visualizing all the training set predictions selecting 100 samples

Weight	Feature
0.4171	budget
0.1602	totalVotes
0.0908	release_year
0.0779	popularity2
0.0567	popularity
0.0488	rating
0.0480	runtime
0.0293	index
0.0261	release_month
0.0140	release_day
0.0097	original_language_en
0.0083	genres_count
0.0059	original_language_fr
0.0030	id
0.0025	spoken_languages_count
0.0013	original_language_ja
0.0004	original_language_ru
0.0002	original_language_hi
0.0001	original_language_es
0	19_companies

Weights of each feature

SHAP

A powerful tool



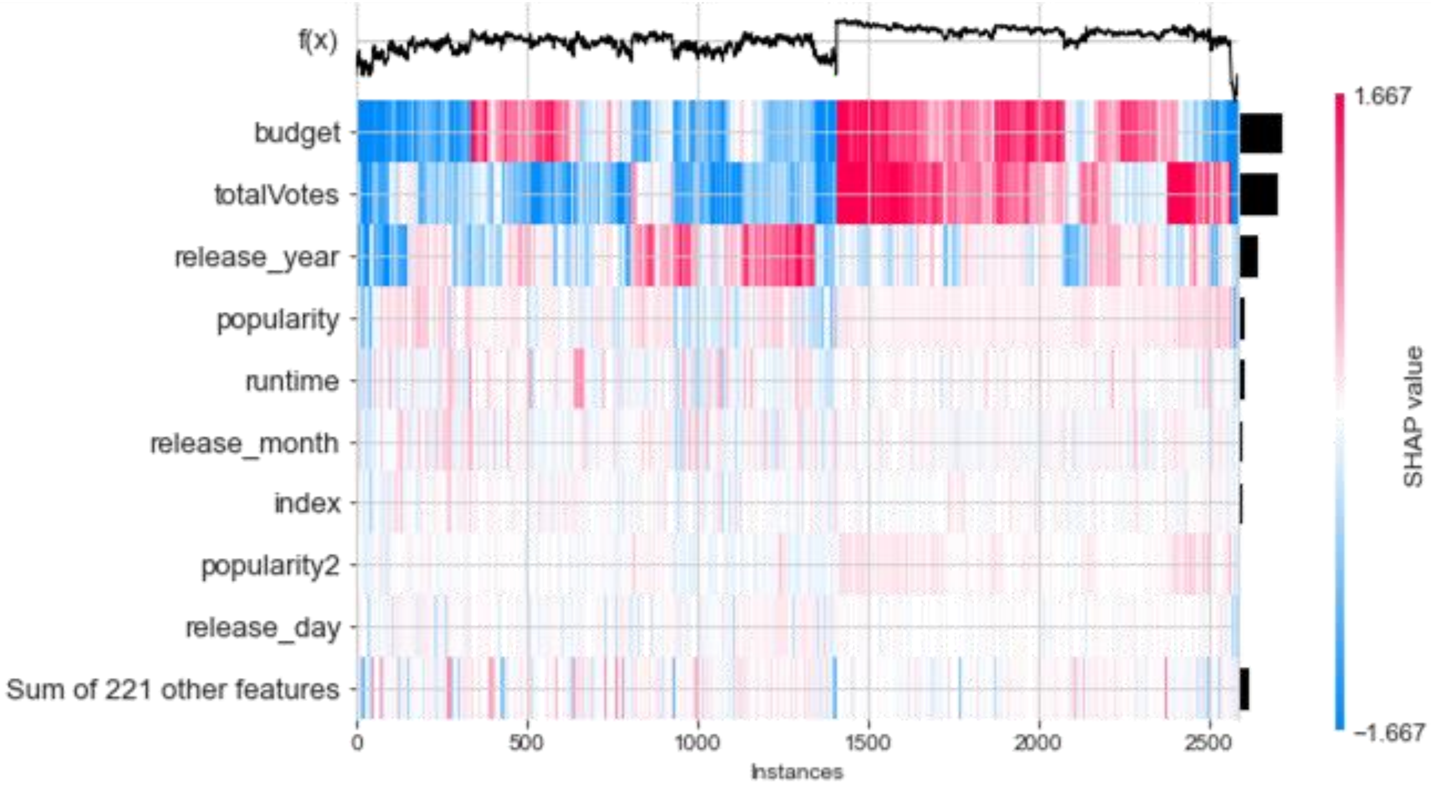
SHAP values (impact on model output) of each feature

Similar samples

Interpretation

Feature distribution under macroscopic sample clustering

1. Sample arrangement: sorting the samples via hierarchical clustering
2. Sample selection: selecting high quality data basing on color concentration

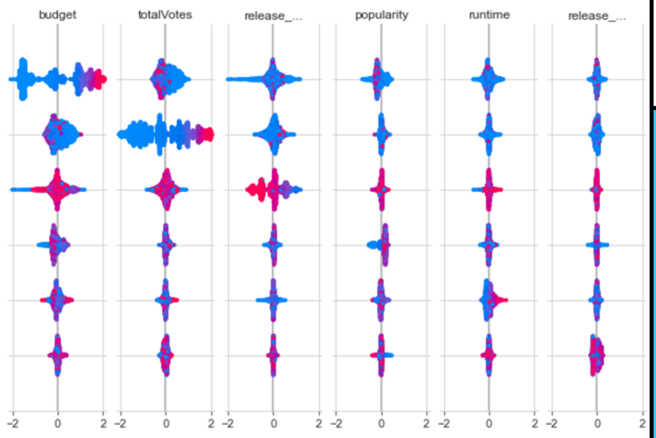


Heat map of feature distribution under macroscopic sample clustering

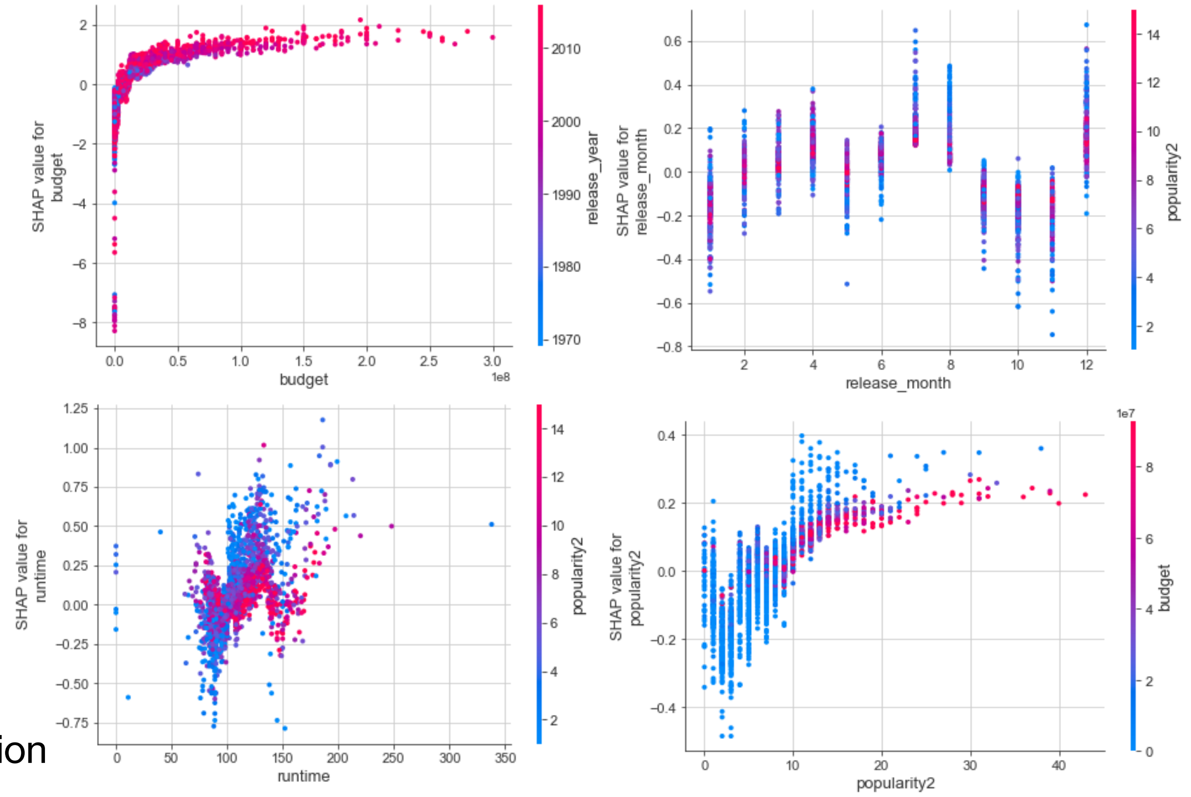
Interpretation

Interaction between features

1. Budget and release_year
2. Release_month and popularity
3. Popularity and budget



SHAP interaction values



SHAP interaction plots

Thank you

Reference

Bugaj, M., Wrobel, K., & Iwaniec, J. (2021). Model Explainability using SHAP values for LightGBM predictions. *2021 IEEE XVIIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. https://doi.org/10.1109/memstech53091.2021.9468078

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R* (1st ed. 2013, Corr. 7th printing 2017 edition). Springer.

Slundberg/shap: A game theoretic approach to explain the output of any machine learning model. (n.d.). GitHub. https://github.com/slundberg/shap

Welcome to LightGBM's documentation! — LightGBM 3.1.1.99 documentation. (n.d.). Welcome to LightGBM's documentation! – LightGBM 3.1.1.99 documentation. https://lightgbm.readthedocs.io/en/latest/

XGBoost documentation — xgboost 1.6.0 documentation. (n.d.). XGBoost Documentation – xgboost 1.6.0 documentation. https://xgboost.readthedocs.io/en/stable/