Adamya Nayyar

TMDB Box Office Collection: Prediction

Problem Statement

In a world where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever. But what movies make the most money at the box office? How much does a director matter? Or the budget? For some movies, it's "You had me at 'Hello.'" For others, the trailer falls short of expectations and you think "What we have here is a failure to communicate." In this, we're presented with metadata on over 7,000 past films from The Movie Database to try and predict their overall worldwide box office revenue.

Background

In this dataset, we are provided with 7398 movies and a variety of metadata obtained from The Movie Database (TMDB). Movies are labeled with id. Data points include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries.
We are training the model on 3000 movies and predicting the worldwide revenue for 4398 movies.

Many movies are remade over the years, therefore it may seem like multiple instances of a movie may appear in the data, however they are different and should be considered separate movies.

Goal

We want to predict the box office revenues for new movies with minimum error possible, for this we have to make 100s of new features based on the features given. After making the features, and training the date on various models, the best model for this project came out to be an ensemble model combination of XGBoost and CatBoost.

Data Wrangling

I started by checking the skewness of the columns and transformed the skewed features as this might get us better predictions. While the original data set contained 26 columns, and majority of them text dictionaries, it was important to extract information from those columns. So I enumerated through each column one by one and extracted significant information. I made WordClouds to get an intuition about the text columns. Extracted and converted the information in dummy variables.

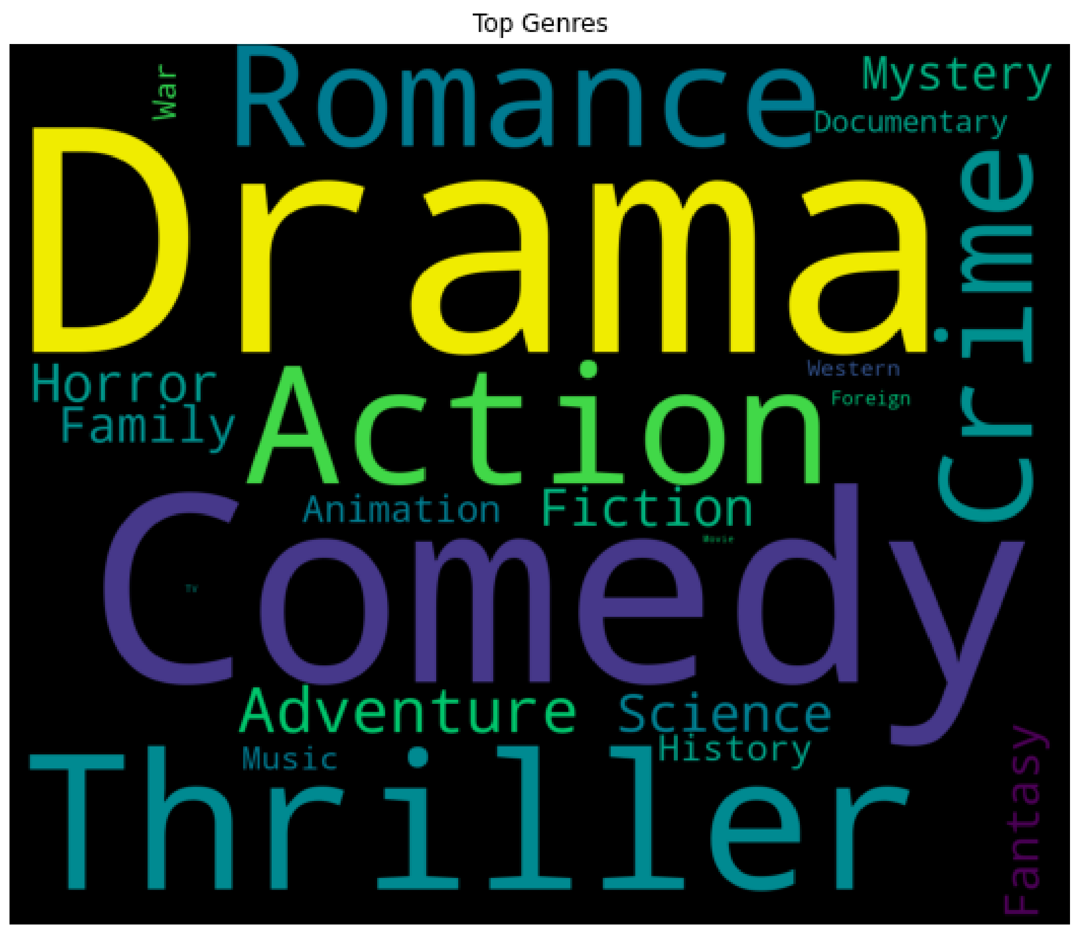


Figure 1: Word Cloud of Top Genres in our dataset.

Top Spoken Languages in Movies
English
Italiano
Français
Español
Deutsch
Русский
Português
Polski
Latin
svenska
Magyar
Český
ελληνικά
Dansk
suomi
Nederlands
Türkçe
Norsk
Kiswahili
Tiếng Việt
Română
Українська
Afrikaans

Figure 2: WordCloud of top languages in our dataset.

Most Common words in title
Last
Man
Love
La
II
de
Day
Life
Night
Boy
Movie
Dead
House
American
Legend
Girl
Men
III
City
Back
World
Le
Star
Part
Secret
Live
Dog
Red
Story
Lost
King
Big
Time
War
Kid
One
Black
Blood
Mr
El
Game
Death
High
Dark
Kill
Eye
Summer

Figure 3: WordCloud of Most common words in Title

After this I created datetime features as well as some additional ratio based features such as budget/popularity, budget/runtime, popularity/mean_year , inflation adjusted budget and many other features. I also looked for outliers and removed some of them after checking. By the time I was done with feature engineering I had a total of 122 features to train our model. The final shape of my training dataset was 3000 rows and 122 columns.

Exploratory Data Analysis

The main aim of EDA was to get an intuition about the behavior of our features with respect to the target variable, which was revenue in our case. Some features show visible relation with the revenue. Let’s check some below.

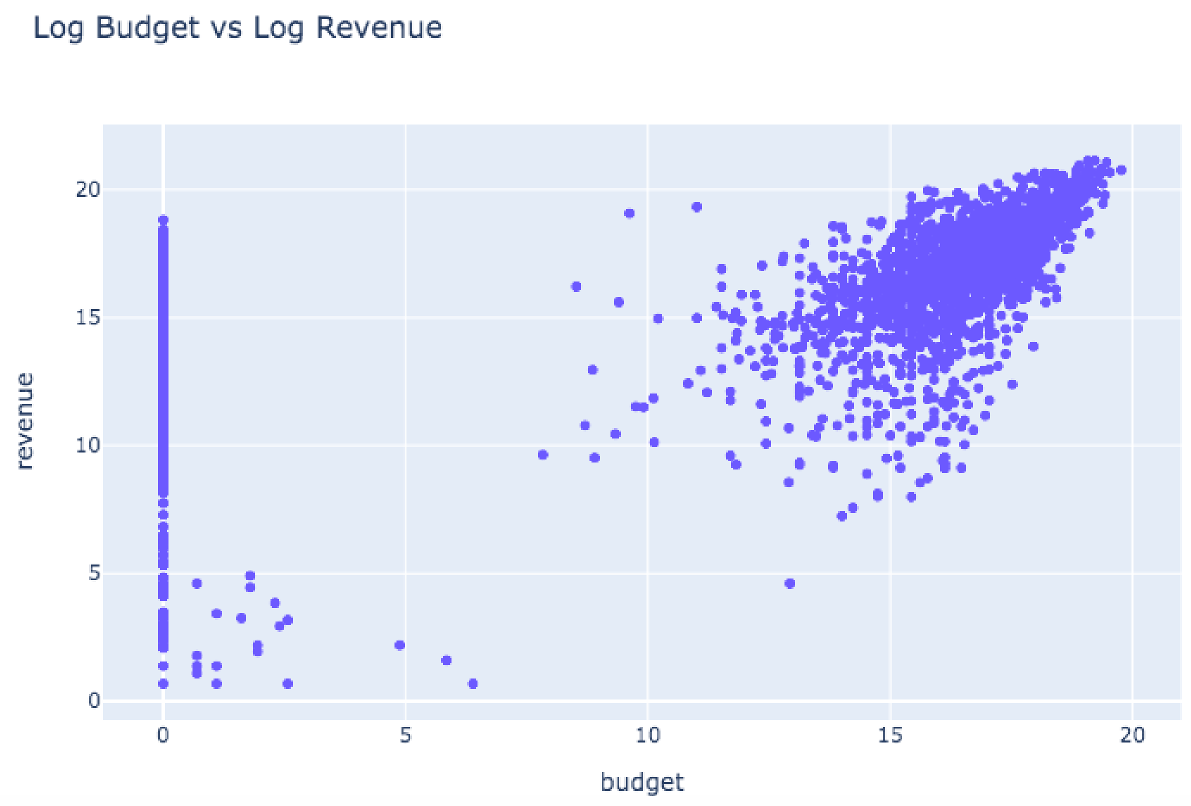


Figure 4 : Revenue vs Budget

We can clearly see that as the budget increases, the revenue also increases. This shows that budget might be an important feature while predicting revenue.

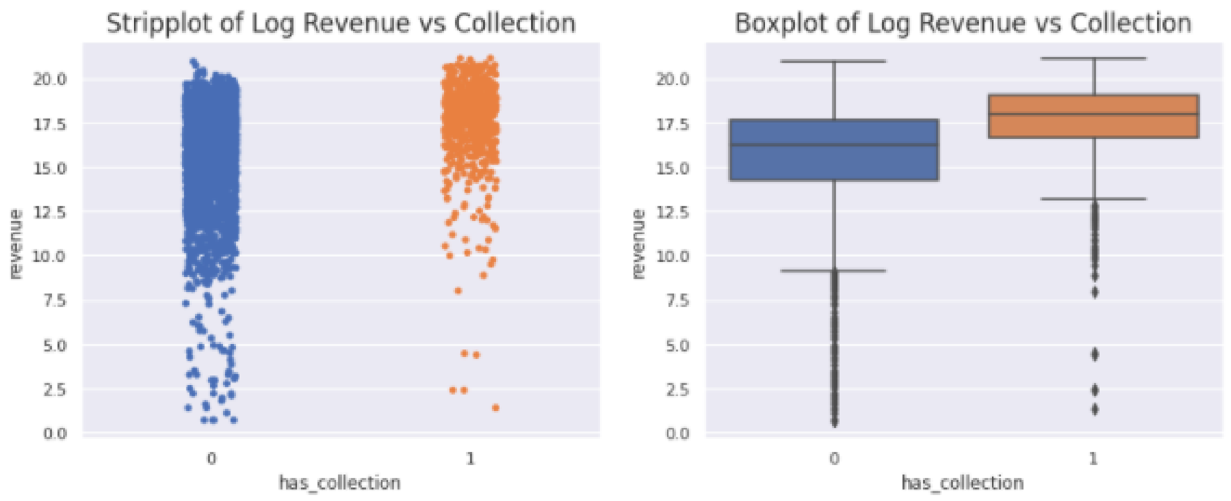


Figure 5 : Collection vs Revenue

The values of has_collection feature in the above figure shows that movies with collections (more parts) tend to earn much more than movies without collection.

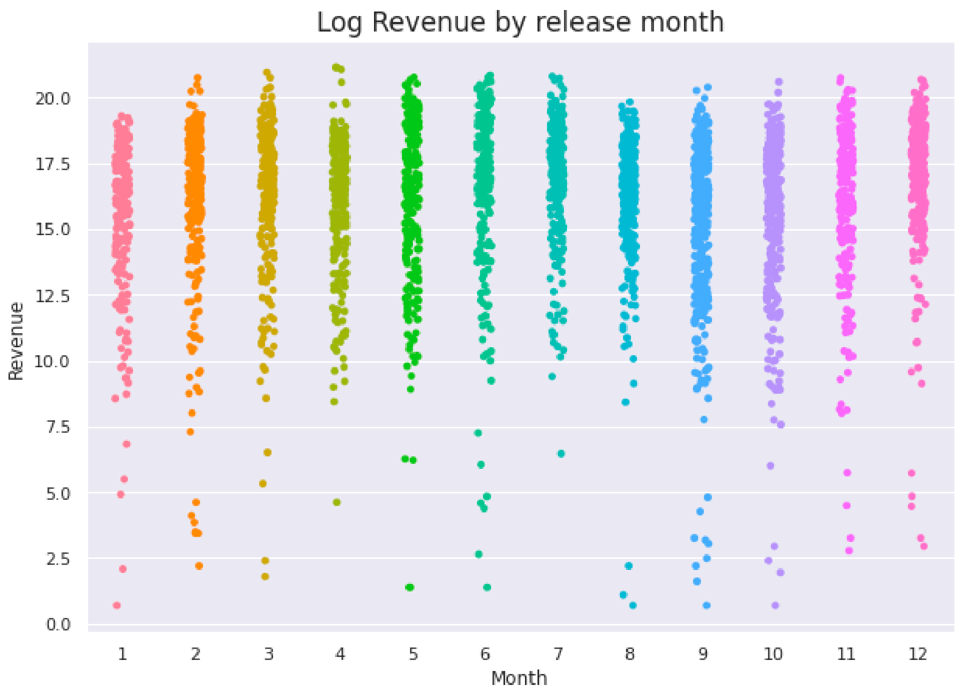


Figure 6: Revenue vs release month

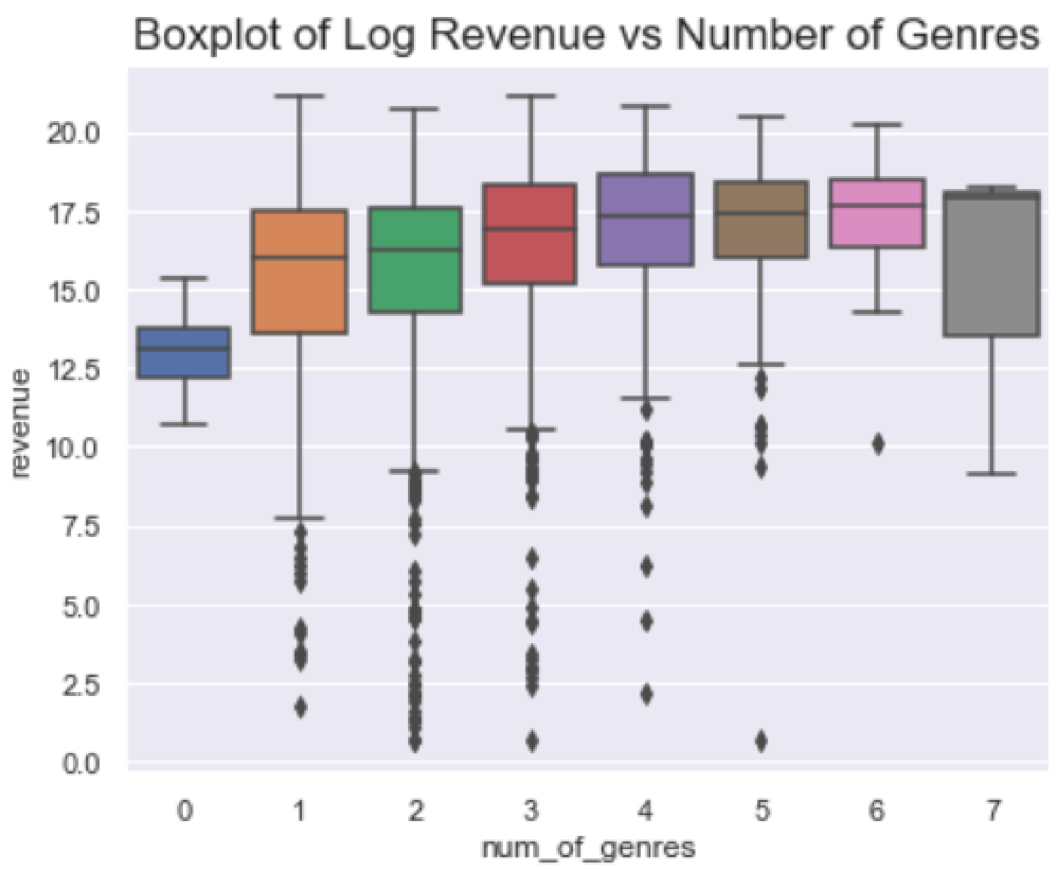


Figure 8: Revenue vs Number of Genres

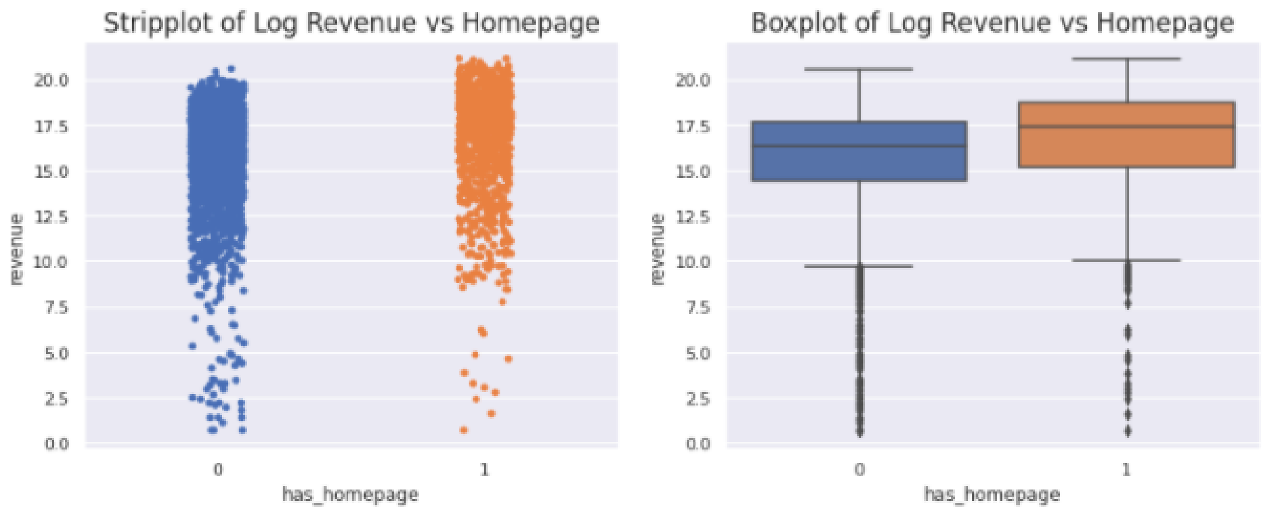


Figure 8: Revenue vs Homepage

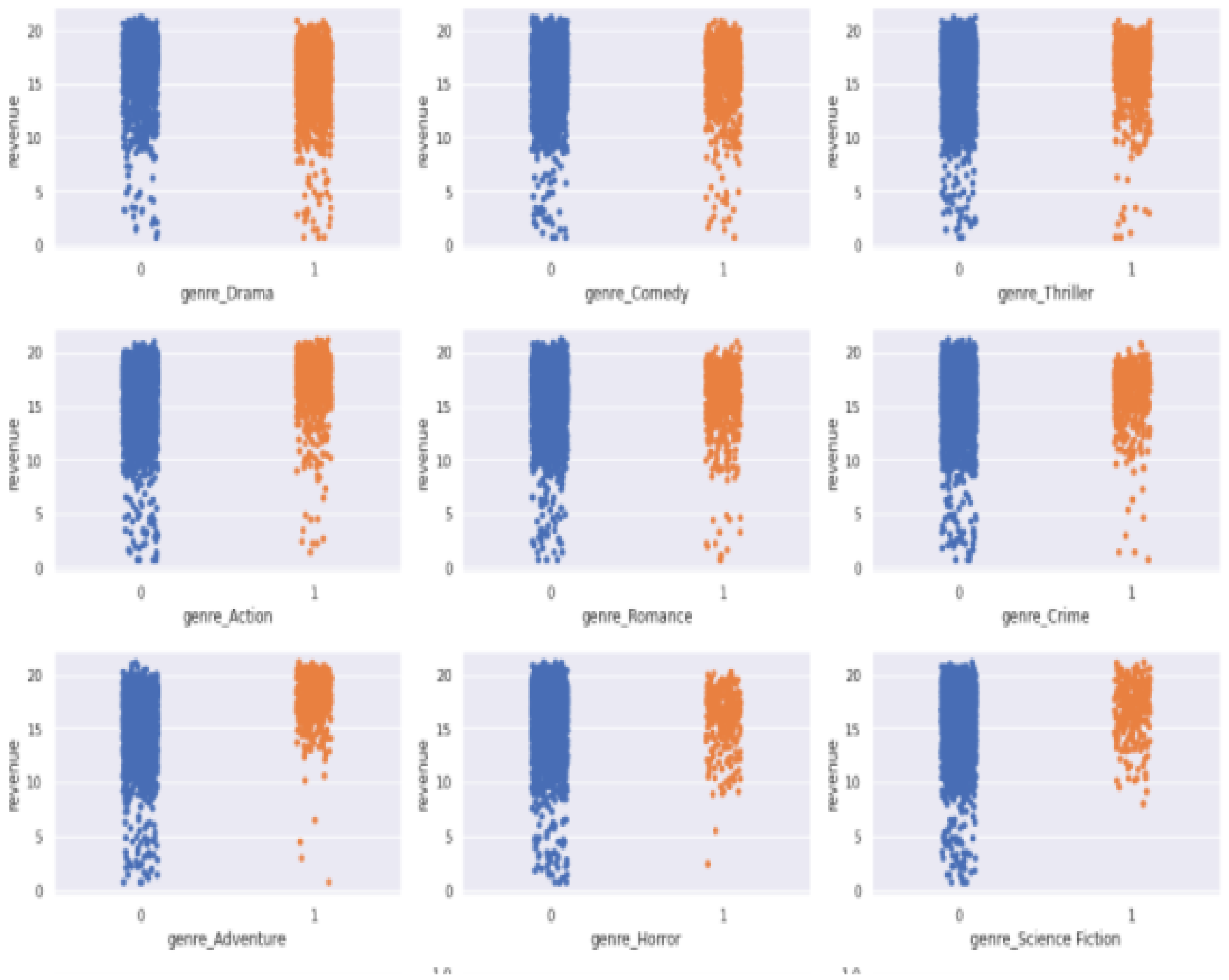


Figure 9: Revenue for Different Genres

We can clearly see that some genres have more revenue on average compared to others. Action, Crime, Adventure and Science Fiction genres have more revenues on average as compared to other genres.

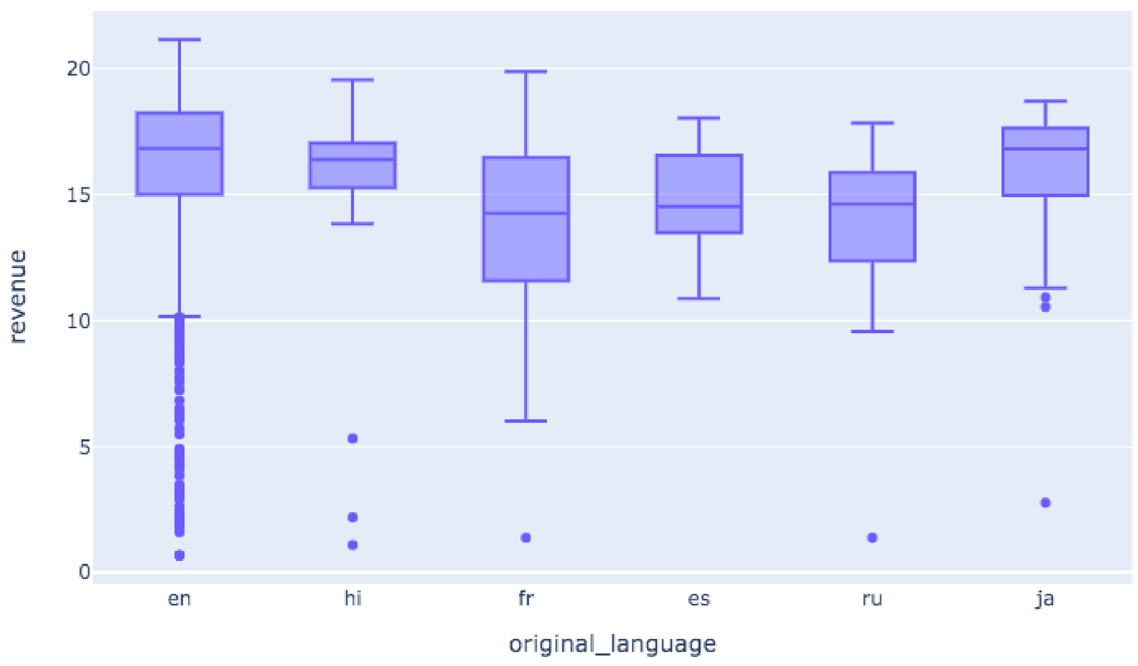


Figure 10: Revenue vs Top Languages

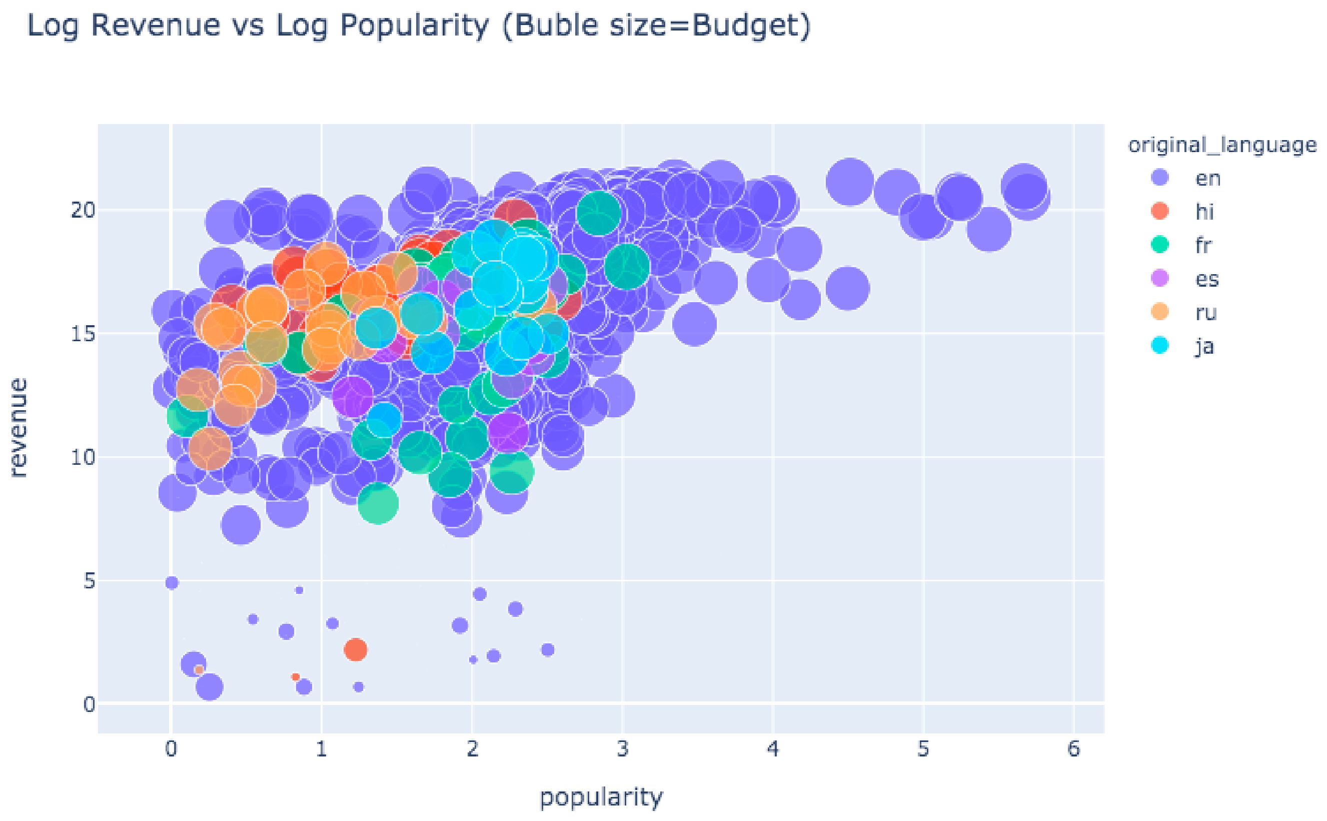


Figure 11: Revenue vs Popularity

Model Selection

After feature exploration, I splitted the train data into train and validation data. After that I fitted the training data on various regression techniques.

Here are the results of various regression techniques:

1) **Linear Regression:**
   Linear Regression R2 Score: 0.4896591352333747
   Mean Squared Error: 4.5802700017855065
   Root Mean Square Error 2.1401565367480733
   Cross-Predicted(KFold) R2 Score: -2.111470894867096

2) **Lasso Regression:**
   Lasso Regression R2 Score: 0.36465412087854543
   Mean Squared Error: 5.702180389236091
   Root Mean Squared Error 2.387923865879331
   Cross-Predicted(KFold) Lasso Regression Accuracy: -4.844751035714688

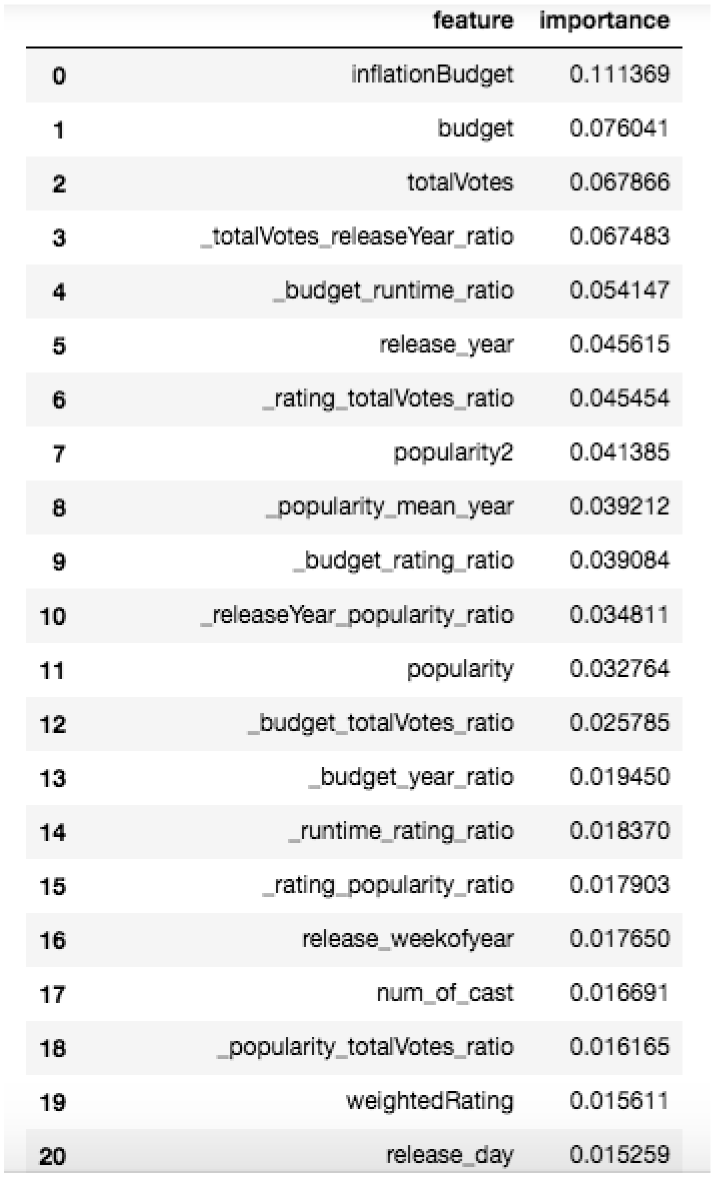
3) **Decision Tree Regressor:**
   Decision Tree R2 Score: 0.12200220450719623
   Mean Squared Error: 7.879962671945697
   Root Mean Square Error 2.807127120731389
   Cross-Predicted(KFold) Decision Tree Accuracy: 0.11756678423203948

4) **Random Forest Regressor:**
   Random Forest Regressor R2: 0.5337269640319187
   Mean Squared Error: 4.184764628367895
   Root Mean Square Error 2.045669726120982
   Cross-Predicted(KFold) Random Forest R2: 0.561212757747461

Random Forest looked like a better model than the other models so we hypertuned the random forest to get even less error.

5) **Randomized Search CV on Random Forest:**
   Mean Squared Error: 3.70034939086403
   Root Mean Square Error 1.923629223853711
   Random Forest Predict R2: 0.5877012692841159

Random Forest gave us very less error and we also checked the 20 important features given by the random forest.



	feature	importance
0	inflationBudget	0.111369
1	budget	0.076041
2	totalVotes	0.067866
3	_totalVotes_releaseYear_ratio	0.067483
4	_budget_runtime_ratio	0.054147
5	release_year	0.045615
6	_rating_totalVotes_ratio	0.045454
7	popularity2	0.041385
8	_popularity_mean_year	0.039212
9	_budget_rating_ratio	0.039084
10	_releaseYear_popularity_ratio	0.034811
11	popularity	0.032764
12	_budget_totalVotes_ratio	0.025785
13	_budget_year_ratio	0.019450
14	_runtime_rating_ratio	0.018370
15	_rating_popularity_ratio	0.017903
16	release_weekofyear	0.017650
17	num_of_cast	0.016691
18	_popularity_totalVotes_ratio	0.016165
19	weightedRating	0.015611
20	release_day	0.015259

Figure 12: Important Features by Random Forest

After Random Forest, We checked fitted the data on Xgboost, Catboost and Lightgbm to find the best model.

6) **XGBoost Regressor:**
Train-rmse: 0.502 valid-rmse:1.83
It performed even better on the Xgboost model. Below are the important features.

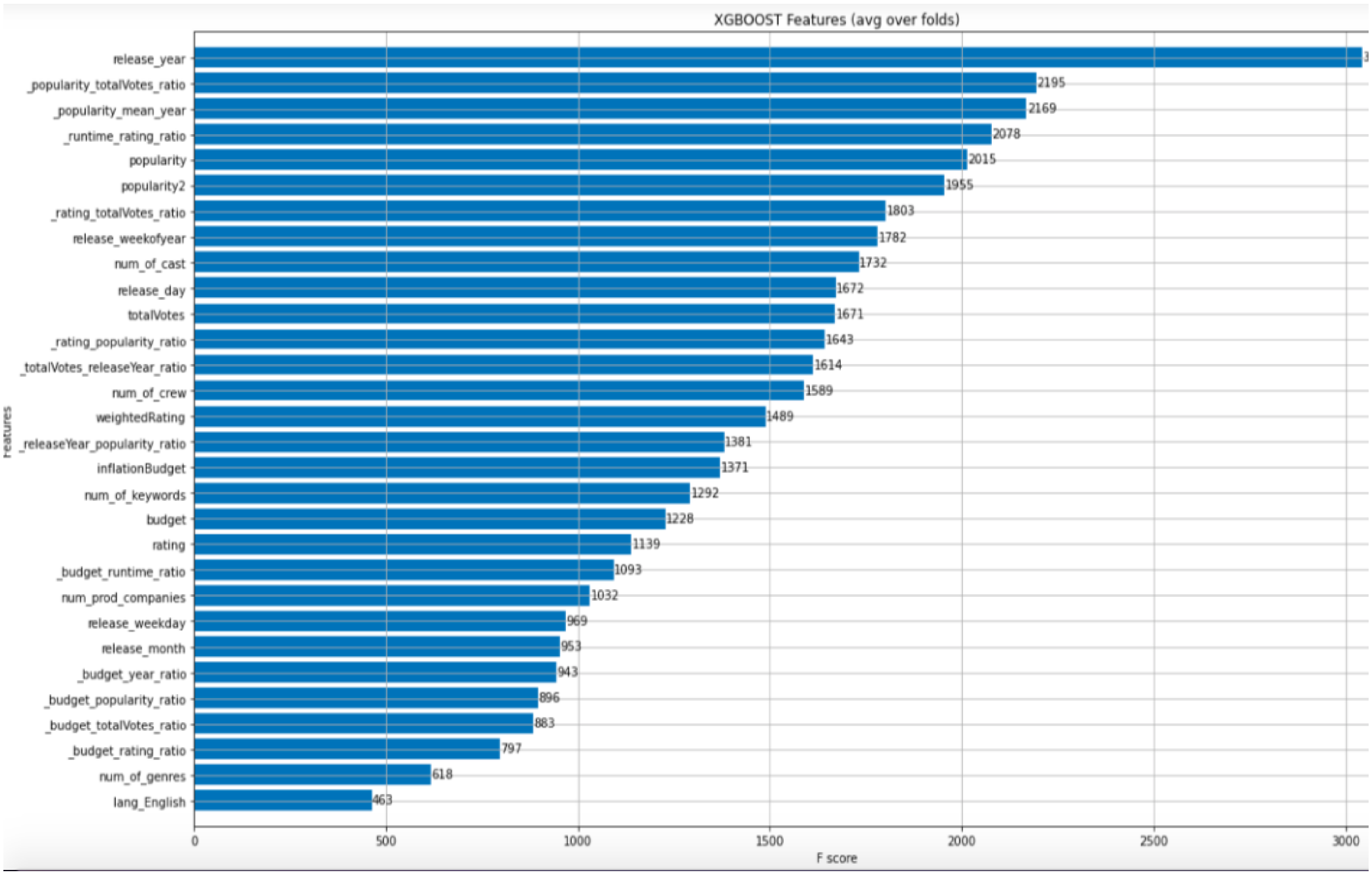


Figure 13: Feature Importance by Xgboost Model

Similarly I fitted the model on CatBoost and LightGBM only to realize that the rmse score of Xgboost and Catboost are very close and we can select the final model either of them as our final model.

7) **CatBoost Regressor:** rmse 1.81
8) **LightGBM** : rmse 1.95

For this project I selected prediction as,
final_pred = 0.3*(pred_by_xgb) + 0.7*(pred_by_catboost)

Conclusion

We started the project with data exploration and cleaning, we checked the skewness and transformed the features. Then we jumped straight to feature creation, we converted all the text features to usable features for our model, in this process we ended up creating about 122 features. Most of these features are redundant and after selecting our final model we can keep only the important features.

After feature engineering, we did Data Visualization and checked correlation between various features and our target variable 'Revenue'. We found various interesting correlation, budget and popularity looked like important features. After this step, we divided our data into a train and validation set for evaluating the performance of our models.

Finally we trained various models and checked their performance on training as well as validation data. We selected rmse and R2 score as our evaluation matrix. Random Forest model showed very less error and hence, we checked important features selected by Random Forest. H2o AutoML model gave us an even better performance, but since it is a blackbox model, we didn’t take it further. Atlast we tried Xgboost, Catboost and LightGbm models which gave us very less bias. Xgboost and Catboost gave us the least bias and hence we made the final predictions by the combination of these two models.