Box Office Prediction

Bhaskar Gupta - 2019237
Krishna Jalan - 2021001
Robin Garg - 2019092
Utkarsh Dubey - 2019213

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation

The motion picture industry is a multibillion-dollar business, and there is a massive amount of data related to movies available over the internet.

Motivation

Box office revenue prediction is an important problem in the film industry that governs financial decisions made by producers and investors. Predicting the box office profits of a movie prior to its worldwide release is a significant but difficult problem that needs an advance of intelligence.

This project proposes a decision support system for the movie investment sector using machine learning techniques.

Literature Review

1. Predicting Box Office Revenue for Movies

   by **Matt Vitelli**

   Used two different models to predict Box office revenue, first was linear classifier with softmax activation function and second was two layer neural network with tanh activation function.

   Author used given features as well as extracted his own features to predict the revenue. By extracting new features he was able to predict data with more accuracy.

Literature Review

2. A Machine Learning Approach to Predict Movie Box-Office Success

   Used sentiment analysis(Microsoft Azure text analysis of IMDB reviews), Support Vector Machine, Neural network analysis to predict revenue.
   They found pre and post release, both the features are important for prediction.
   Budget, number of screens where movie is released dominated. Finally figuring out that budget, IMDb votes and no. of screens are the most important features.

Dataset Description

1. Our dataset contains over 10,000 movies having details of each movie like title, genre, budget, cast, crew, Release_Date, etc. Dataset is gathered from different source to create a larger training and testing dataset. (TMDB official dataset and web scraping TMDB API to get newer datasets).
2. Dataset contains 22 distinct features where we targeted revenue generated for our prediction.

#	Column	Non-Null Count	Dtype
0	TMDb_Id	10649 non-null	int64
1	IMDb_Id	10578 non-null	object
2	Title	10649 non-null	object
3	Original_Title	10649 non-null	object
4	Overview	10609 non-null	object
5	Genres	10580 non-null	object
6	Cast	10596 non-null	object
7	Crew	10636 non-null	object
8	Collection	10648 non-null	object
9	Release_Date	10646 non-null	object
10	Release_Status	10648 non-null	object
11	Original_Language	10649 non-null	object
12	Languages_Spoken	10565 non-null	object
13	Runtime	10634 non-null	float64
14	Tagline	7862 non-null	object
15	Popularity	10649 non-null	float64
16	Rating_average	10649 non-null	float64
17	Rating_Count	10649 non-null	int64
18	Production_Companies	10307 non-null	object
19	Country_of_Origin	10521 non-null	object
20	Budget	10648 non-null	float64
21	Revenue	10648 non-null	float64

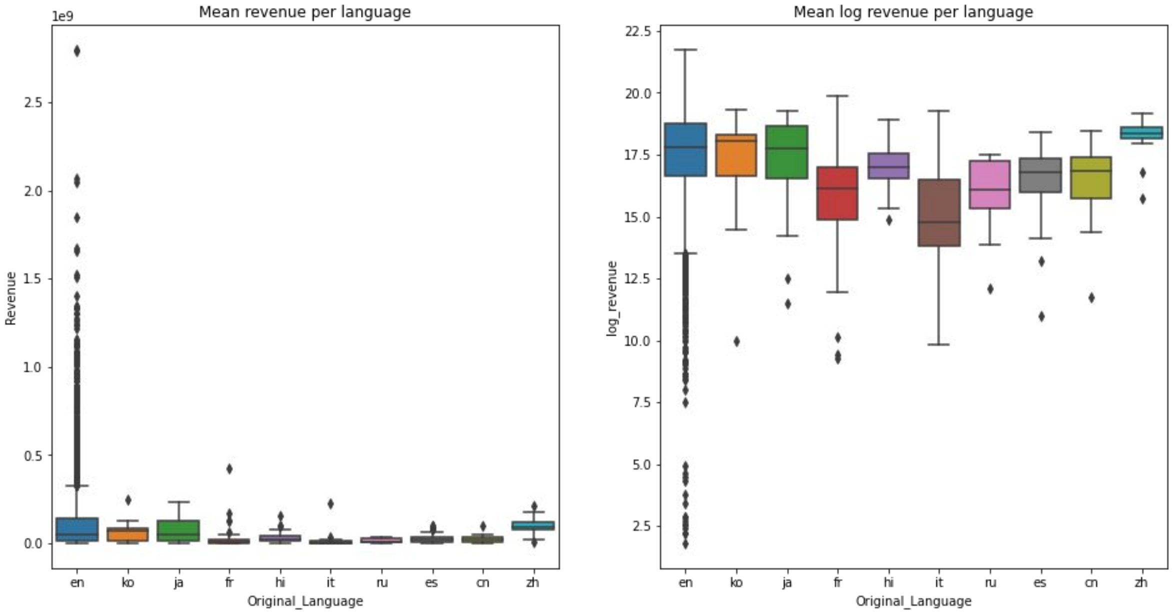
3. We tried finding relations between the collected features and revenue.

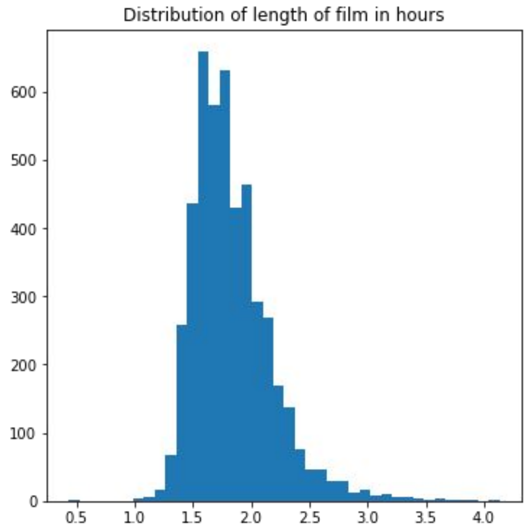
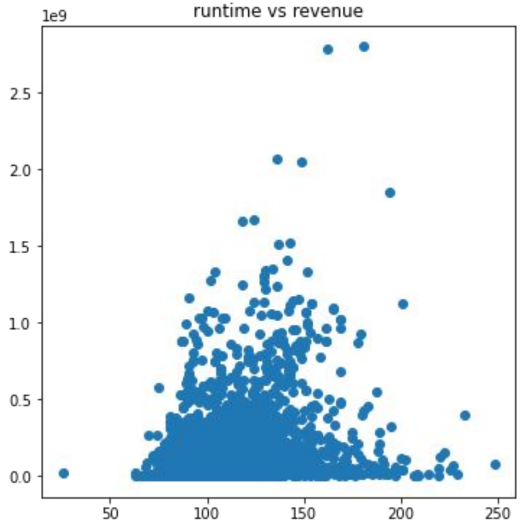
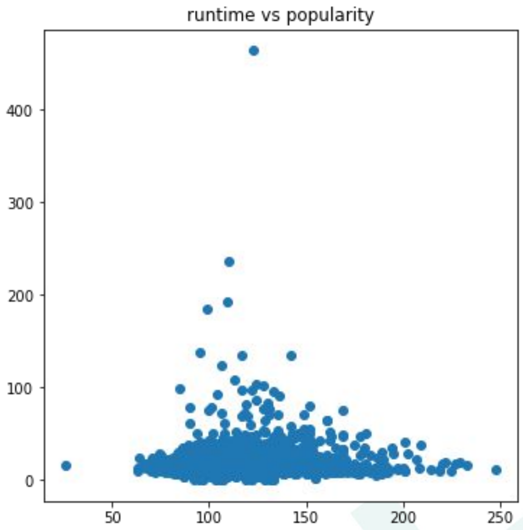
Dataset Cleaning/Preprocessing

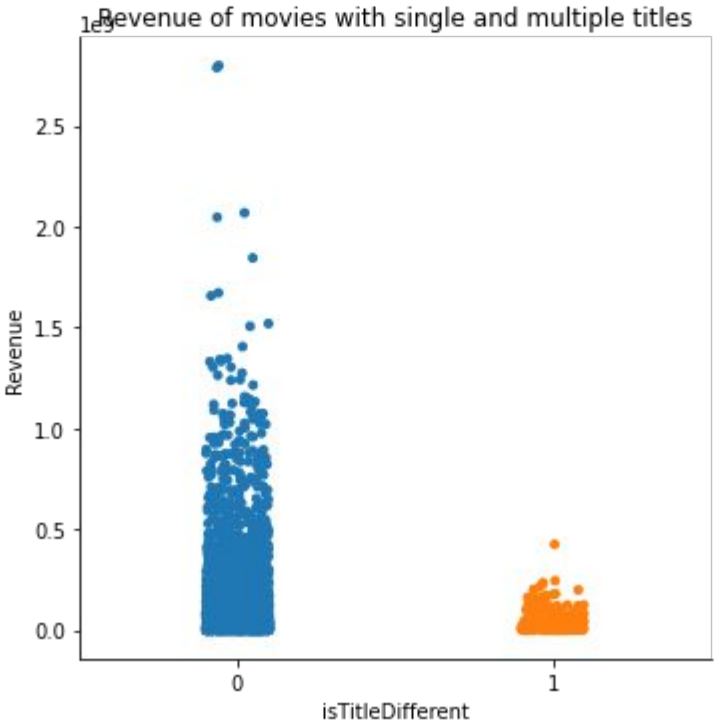
1. Remove invalid dataset (i.e. budget=0 or revenue=0 or rating_count=0 … etc).
2. Convert labelled data into binary features. (i.e. origin language, production countries)
3. Used one-hot encoding to create another feature using the top 10 casts by revenue generated by their movies.

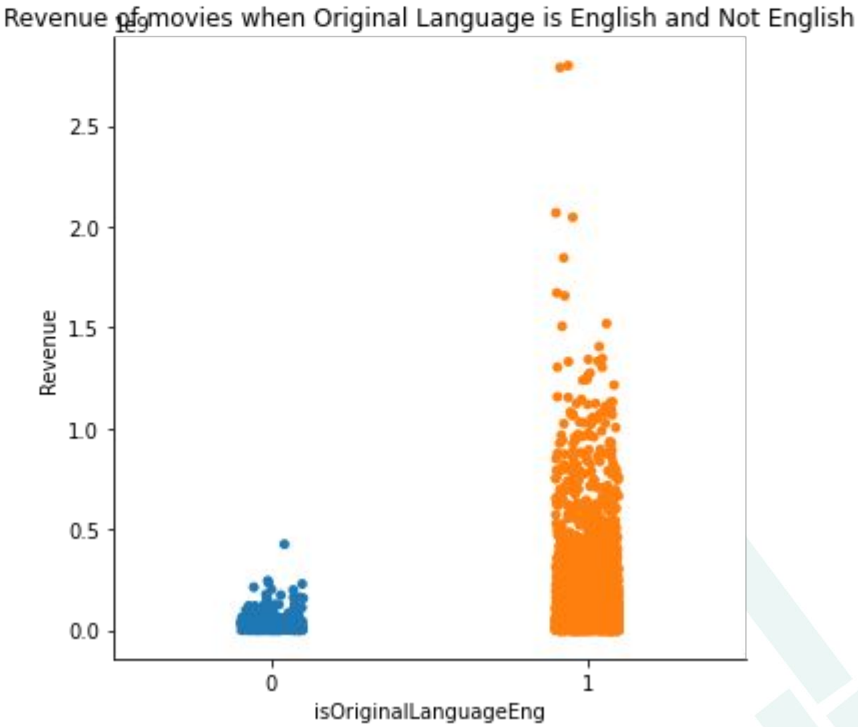
	Runtime	Popularity	Rating_average	Rating_Count	Budget	Revenue
count	10634.000000	10649.000000	10649.000000	10649.000000	1.064800e+04	1.064800e+04
mean	102.584258	13.249832	6.316687	995.059348	1.834690e+07	5.371899e+07
std	26.549647	10.225099	1.327804	1957.076797	3.508205e+07	1.420160e+08
min	0.000000	0.600000	0.000000	0.000000	0.000000e+00	0.000000e+00
25%	91.000000	9.453000	5.800000	151.000000	0.000000e+00	0.000000e+00
50%	101.000000	11.406000	6.500000	323.000000	2.433500e+06	1.502982e+06
75%	115.000000	14.052000	7.100000	873.000000	2.100000e+07	4.237419e+07
max	400.000000	463.487000	10.000000	25159.000000	3.870000e+08	2.797801e+09

Data Visualization



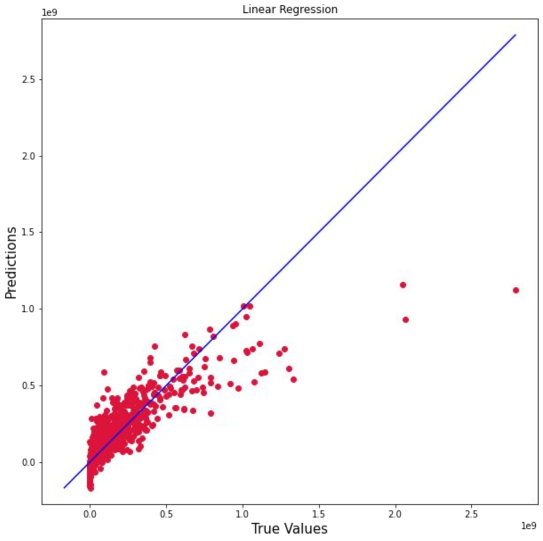

Distribution of length of film in hours
0
100
200
300
400
500
600
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

runtime vs revenue
1e9
0.0
0.5
1.0
1.5
2.0
2.5
50
100
150
200
250

runtime vs popularity
0
100
200
300
400
50
100
150
200
250


Revenue of movies with single and multiple titles
1e9
2.5
2.0
1.5
1.0
0.5
0.0
Revenue
0
1
isTitleDifferent


Revenue of movies when Original Language is English and Not English
1e9
2.5
2.0
1.5
1.0
0.5
0.0
Revenue
0
1
isOriginalLanguageEng

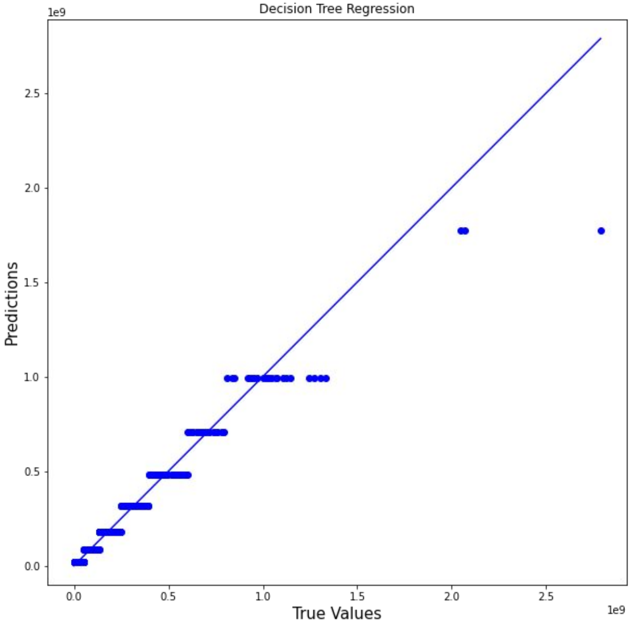
Linear Regression

1. Assuming a linear relationship between the input variable and single output variable("Revenue"), we applied linear regression on the dataset with 66-33 train-test data split.
2. We have trained the model with 2857 input samples with 147 features and tested it with 1539 test samples.
3. Linear Regression RMSE, MAE - (100858863.15014496, 62278929.10013399)
4. R-square score: 0.73 (testing data)



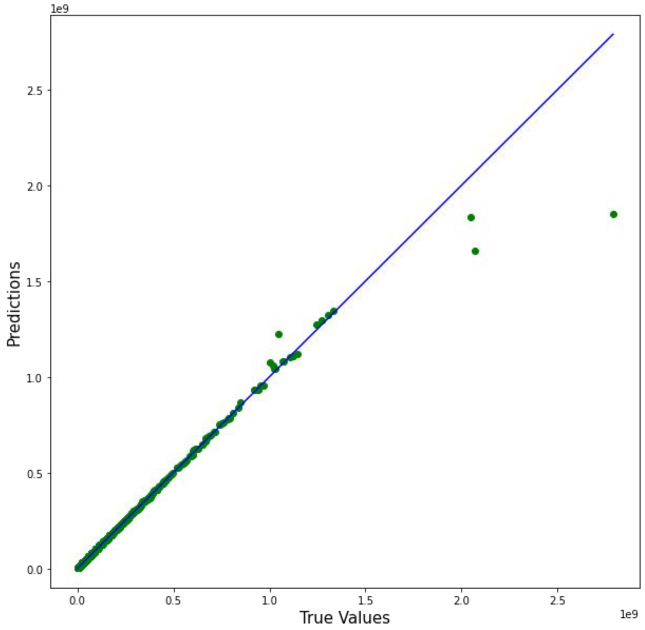
Decision Tree

1. Applied Decision Tree Regressor on the given dataset and used 66-33 train-test split.
2. Decision tree is used as a baseline for future comparison with different models.
3. Decision tree Regressor RMSE, MAE - (4207113, 22058687)
4. R-square score: 0.95 (testing data)



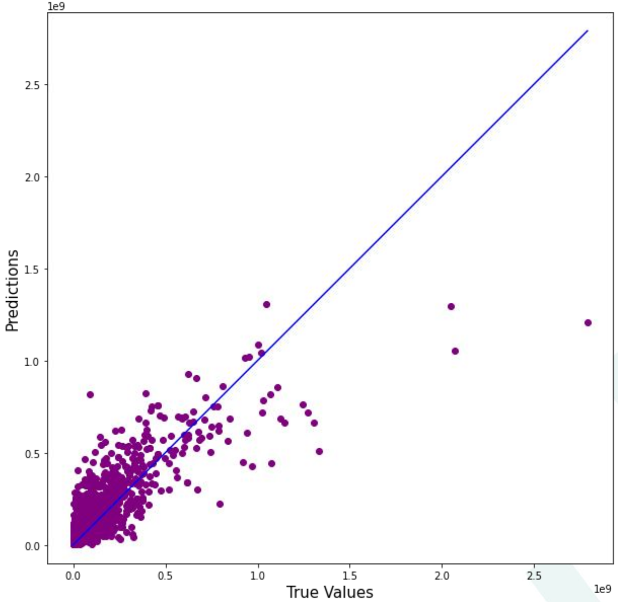
Random Forest

1. Applied Random Forest Regressor on the given dataset with 66-33 train-test split.
2. With max depth of 5 to reduce the overfitting problem to achieve better results compared to the decision tree regressor model.
3. Random Forest RMSE, MAE - (27441463, 3361622)
4. R-square score: 0.98 (testing data)



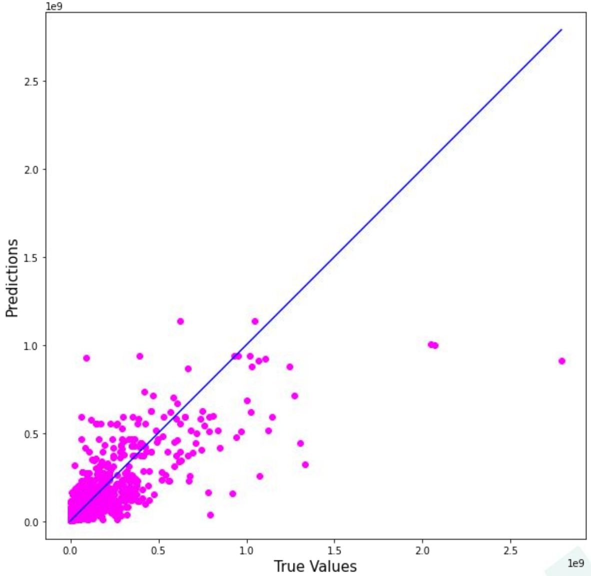
MLP Regressor

1. Applied MLP Regressor with one input-layer, two hidden-layers [147, 74] (number of input feature, mean of input feature and output feature) and one output layer with a number of neurons = 1 (regression model) with activation function as “linear”. Model is trained for 500 epochs.
2. With max depth of 5 to reduce the overfitting problem to achieve better results compared to the decision tree regressor model.
3. Random Forest RMSE, MAE (11668440, 66859636).
4. R-square score: 0.64 (testing data)



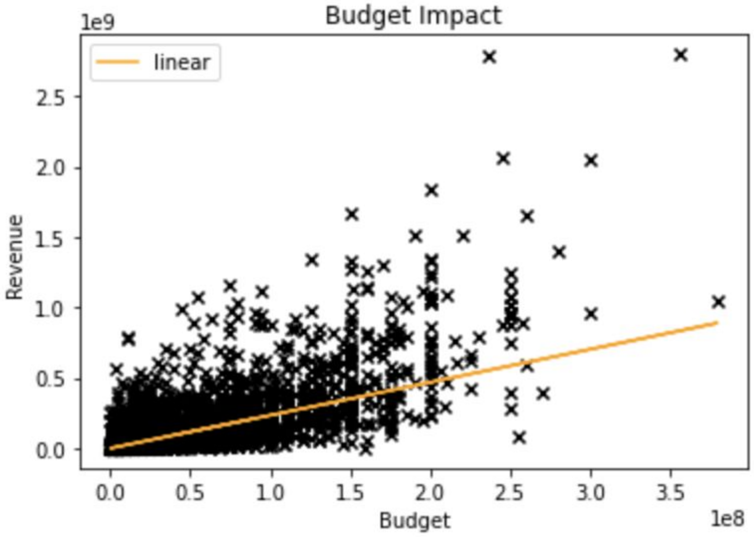
KNN Regressor

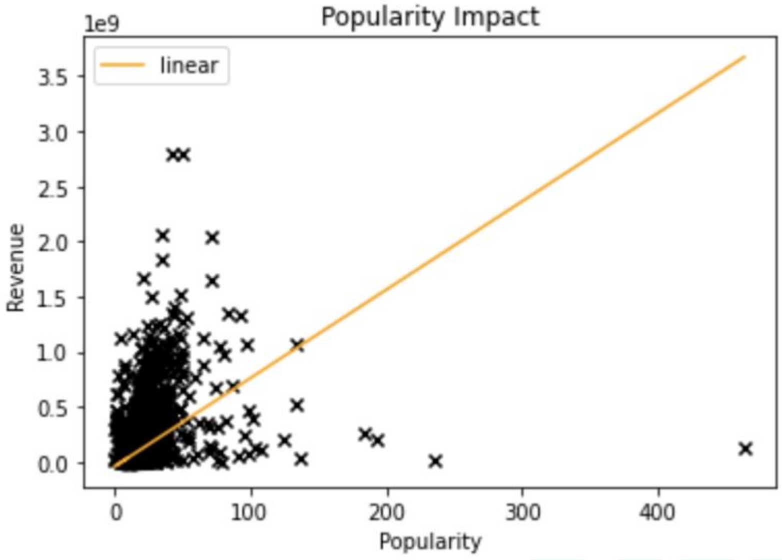
1. Applied K Neighbors Regressor on the given dataset and used a 66-33 train-test split. For the hyperparameter, we used n_neighbors=10 to best fit the dataset (obtained by testing multiple values of n)
2. With max depth of 5 to reduce the overfitting problem to achieve better results compared to the decision tree regressor model.
3. Random Forest RMSE, MAE (126378561, 64130231).
4. R-square score: 0.60 (testing data)



Result/Analysis

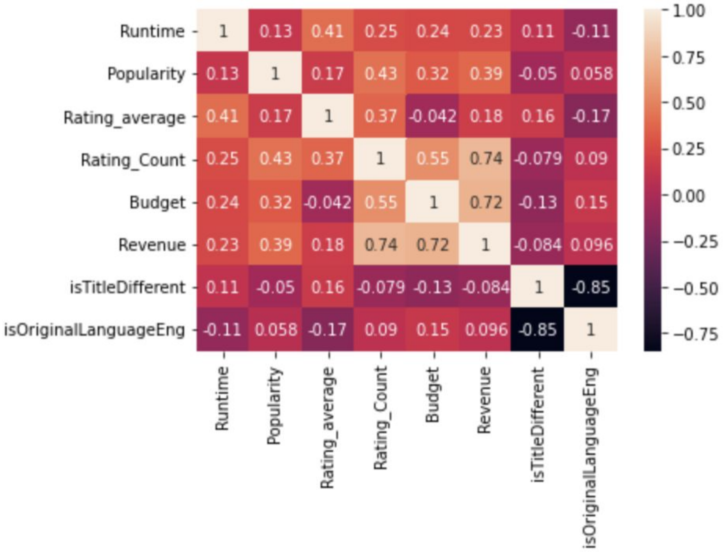
1. The linear regression model under-fitted the dataset as the testing score is 0.60 and the training score is 0.70.
2. The decision tree regressor overfitted the whole dataset as the testing score is 0.51 and the training score is 1.0 but after hyperparameter tuning and setting max_depth=3 score came out to be 0.95 , 0.96 respectively.
3. The best fit model comes to be random forest with 0.98 with testing data and 0.99 with training data.



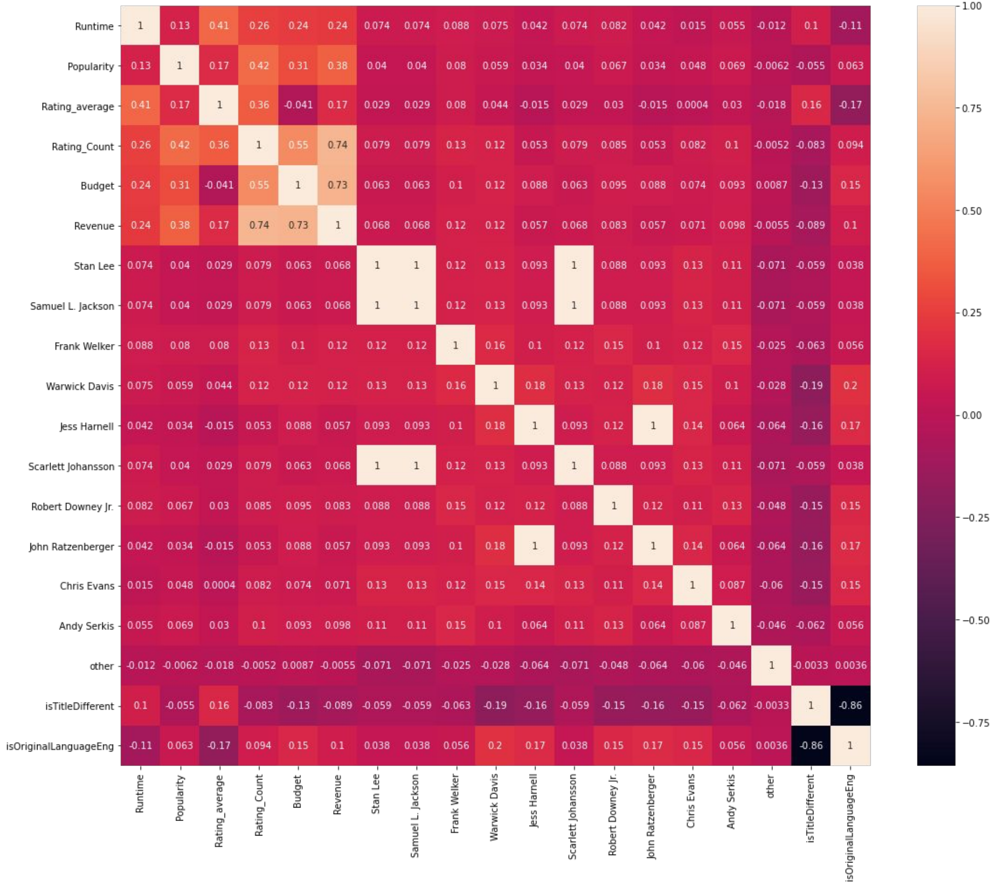


Conclusion

1. Revenue shows a positive correlation with “rating_count”, “popularity” and “budget” and a negative correlation with “IsTitleDifferent”.
2. The best model came out to be random forest with the r2 score of 0.98 followed by decision tree with an r2 score of 0.95 then followed by linear regression with an r2 score of 0.71 then MLP Regressor with an r2 score of 0.64 last but not the least KNN with r2 score of 0.60
3. Need to apply hyperparameter tuning to avoid overfitting and underfit to improve the performance measure of each model.



Correlation



Future Work

1. Future work can include applying the Deep Learning model to the given dataset for better Results.
2. We can include features like production company, crew members and do more processing on the cast members and expand our model to run on more features.
3. The features like Title, Overview got dropped due to textual data; we can apply NLP to process the data.
4. Finally, the dataset was limited, and we further dropped many rows during preprocessing; we can include more movies to improve the training of the current model.

Individual Team Member Contribution

1. **Krishna Jalan** - Reducing noise from data, Linear/MLP/KNN Regression, Analysis of results/models using different evaluation methods.
2. **Robin Garg** - Literature review, Data Preprocessing of grouped data, Model Selection, Random Forest, Future Work.
3. **Utkarsh Dubey** - Data Preprocessing and Data scraping, walking over different available datasets, Feature Selection, Parameter Selection for models, Decision Tree.
4. **Bhaskar Gupta** - Data Extraction from TMDb API, Data Visualisation, Analysis of data on Graphs, Ridge/Lasso Regression, Verification of Accuracy.

Thank you!