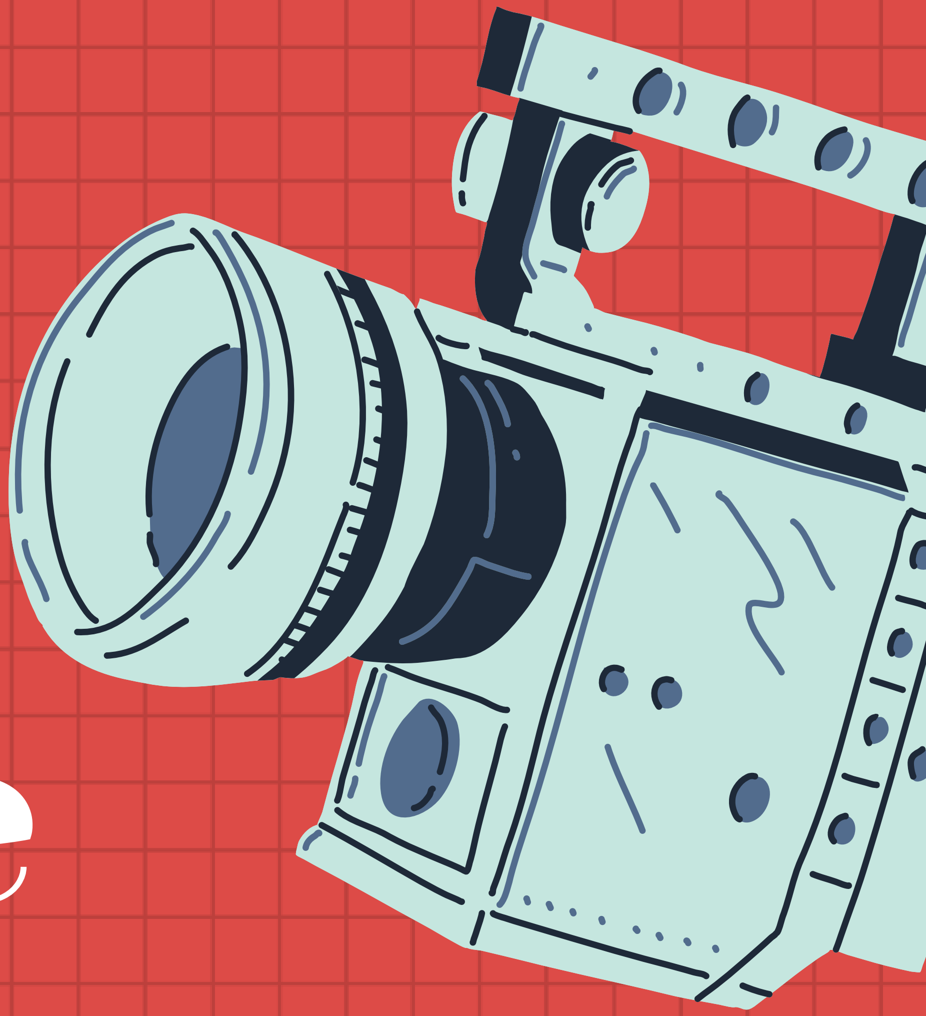
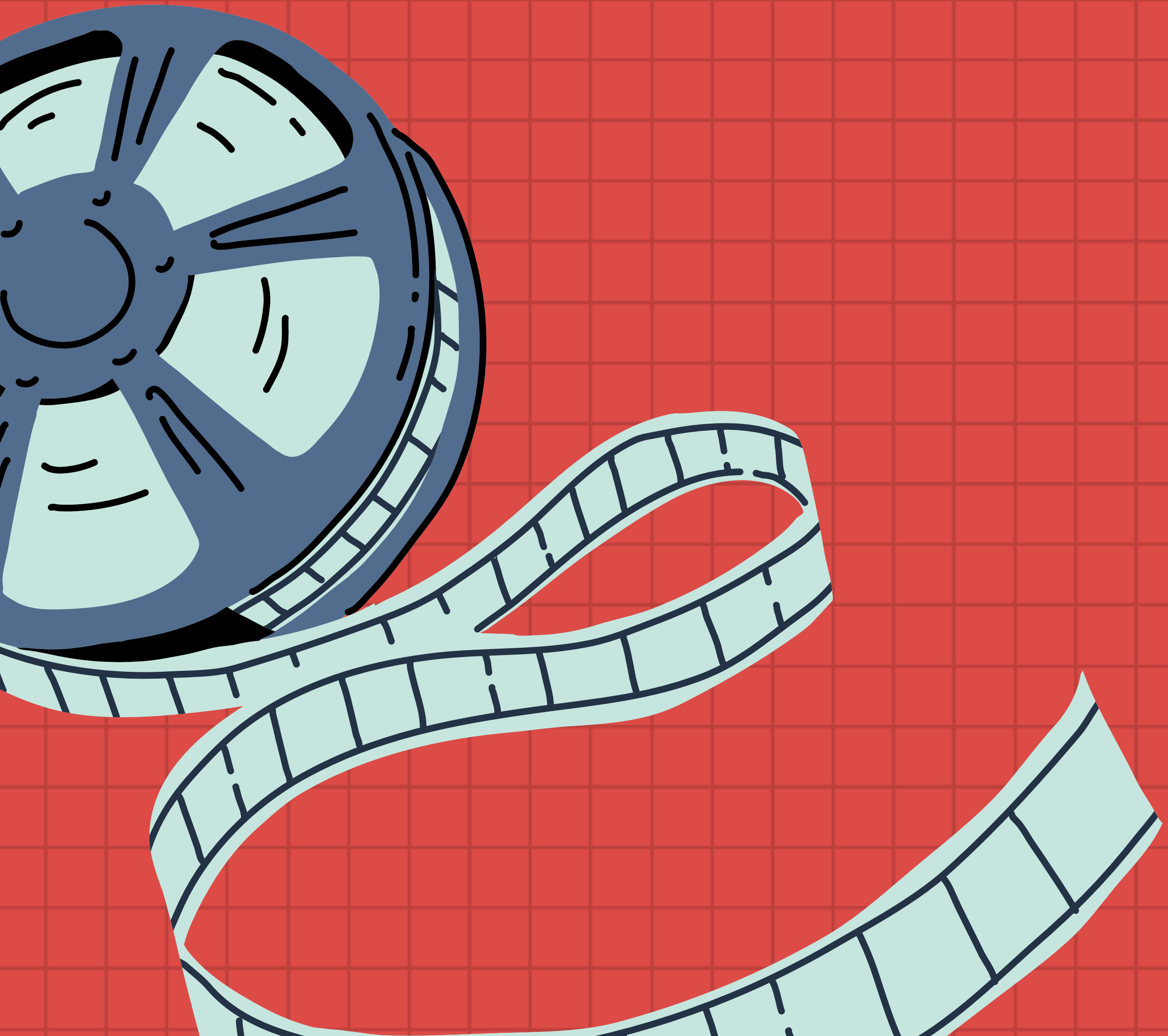




Data Science Capstone

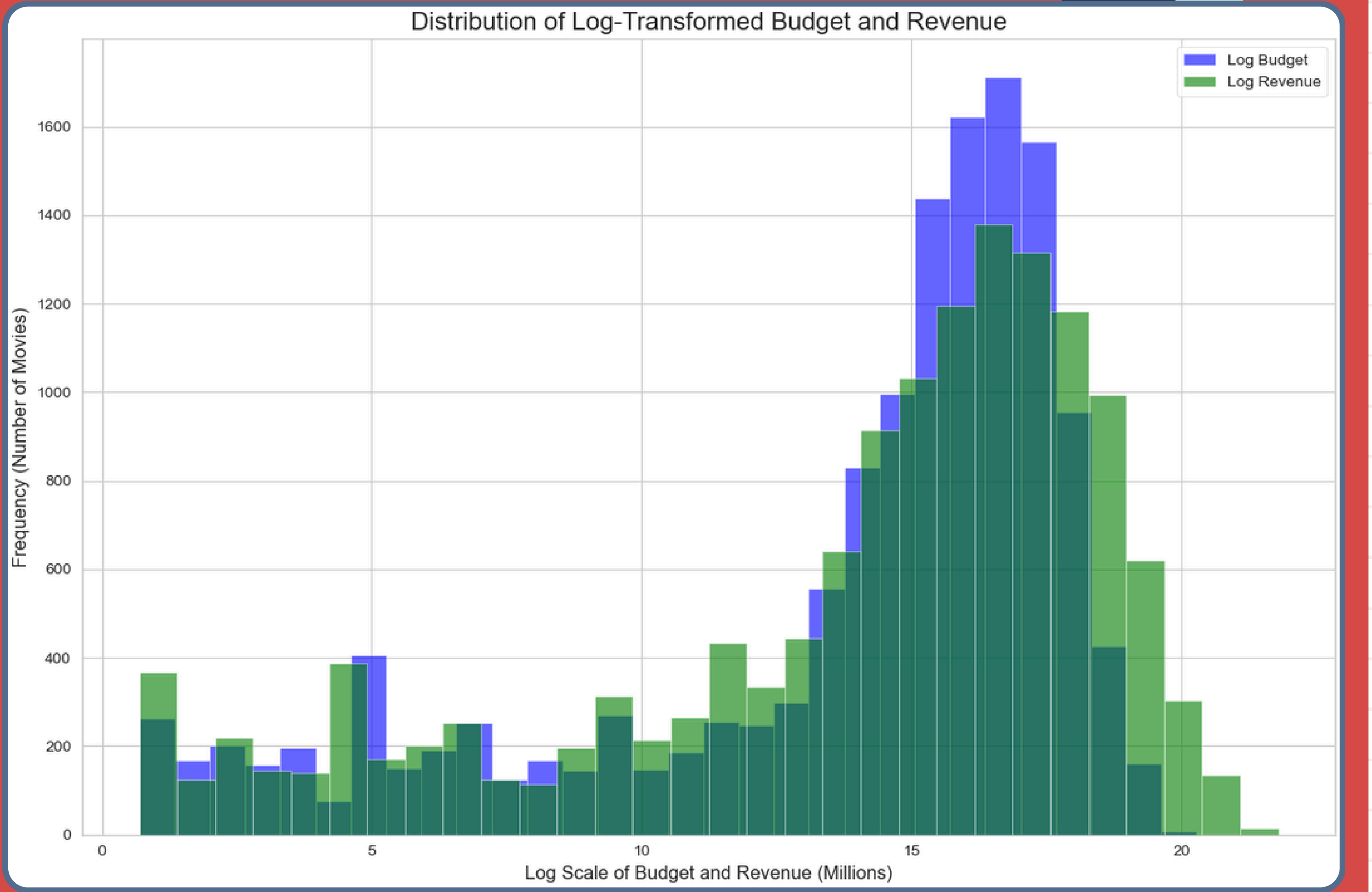
Predicting Movie Revenue

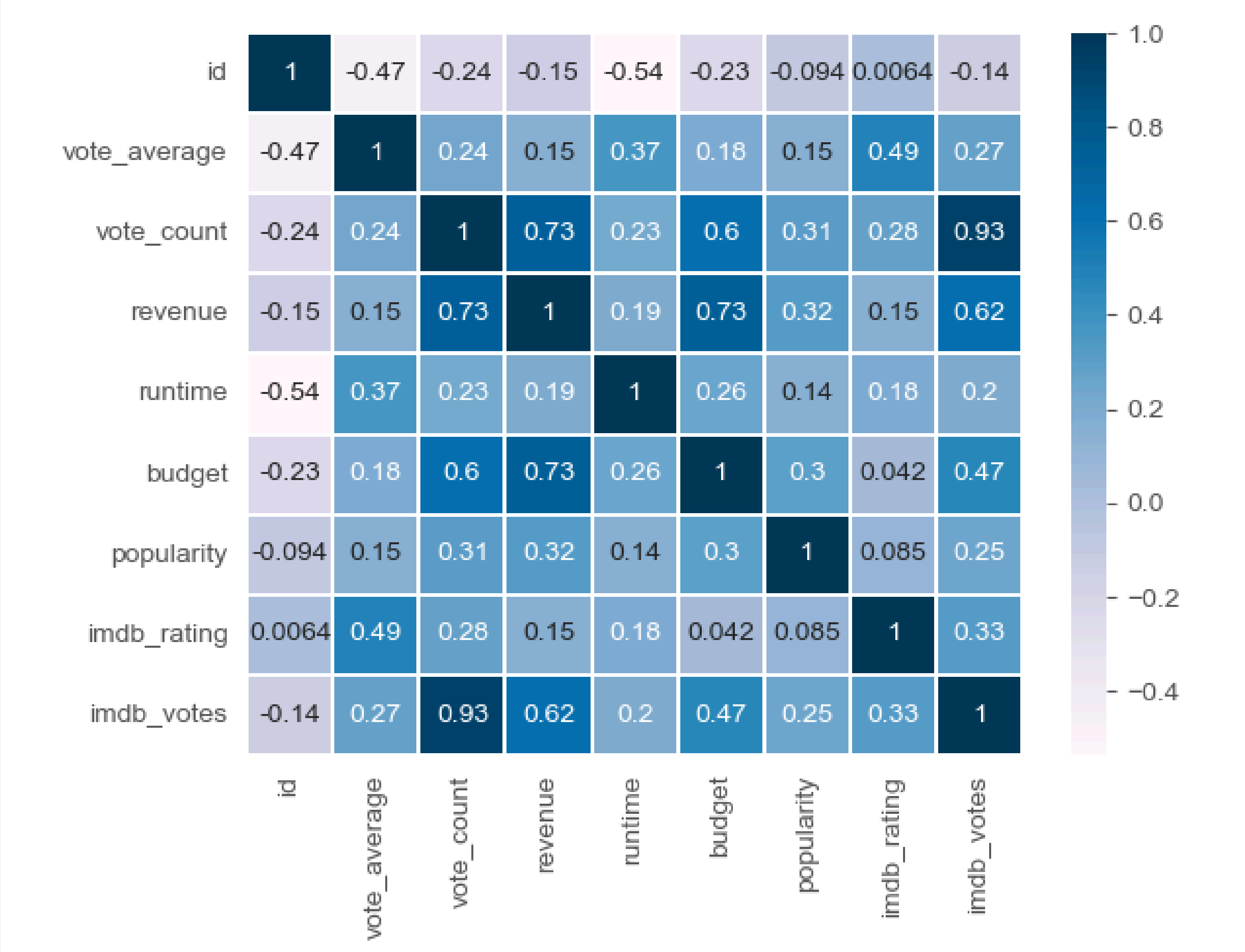
Ava Bender

November 25, 2024

Introduction

Predicting movie revenue before its release is a complex yet important task, as it plays a crucial role in budgeting and marketing strategies. Accurate revenue predictions empower production companies to distribute resources more effectively, enhance profitability, and reduce financial risks. In an industry where budgets continue to rise, understanding the factors that drive revenue is important for informed decision-making and strategic planning.





Data for Evaluation

Source: The Ultimate 1Million Movies Dataset (TMDB + IMDb) from Kaggle.

Size of Dataset:

```
df.shape
(1017605, 28)
```

```
data_all.shape
(14561, 54)
```

Key Features: Budget, revenue, genres, cast, directors, popularity.

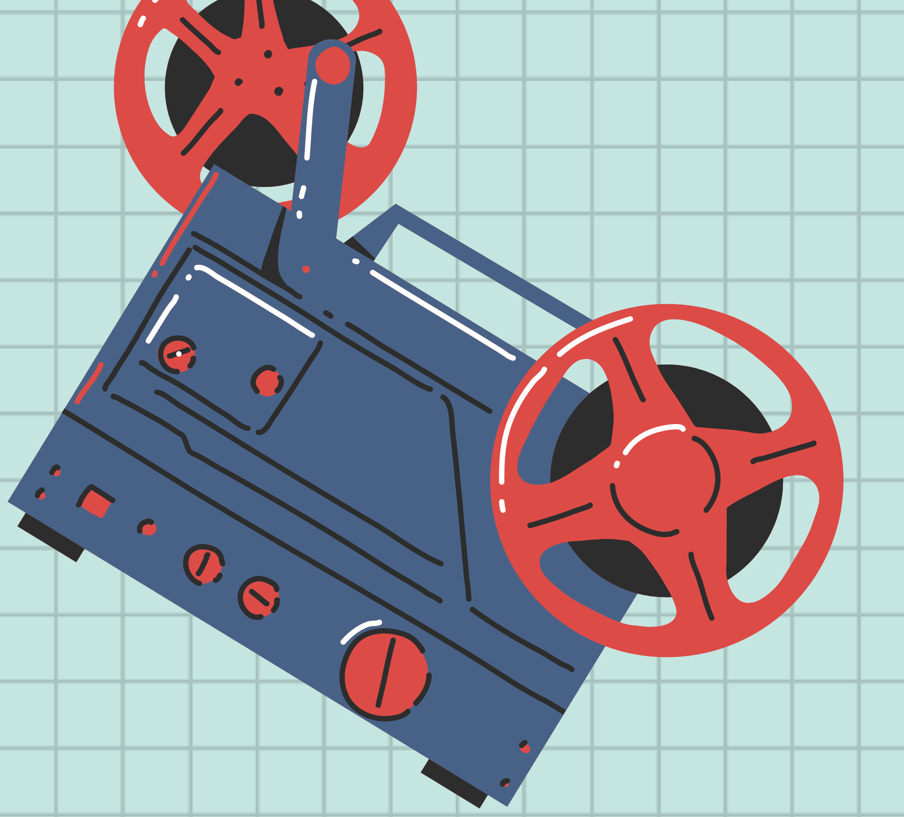
Data Columns Before and After

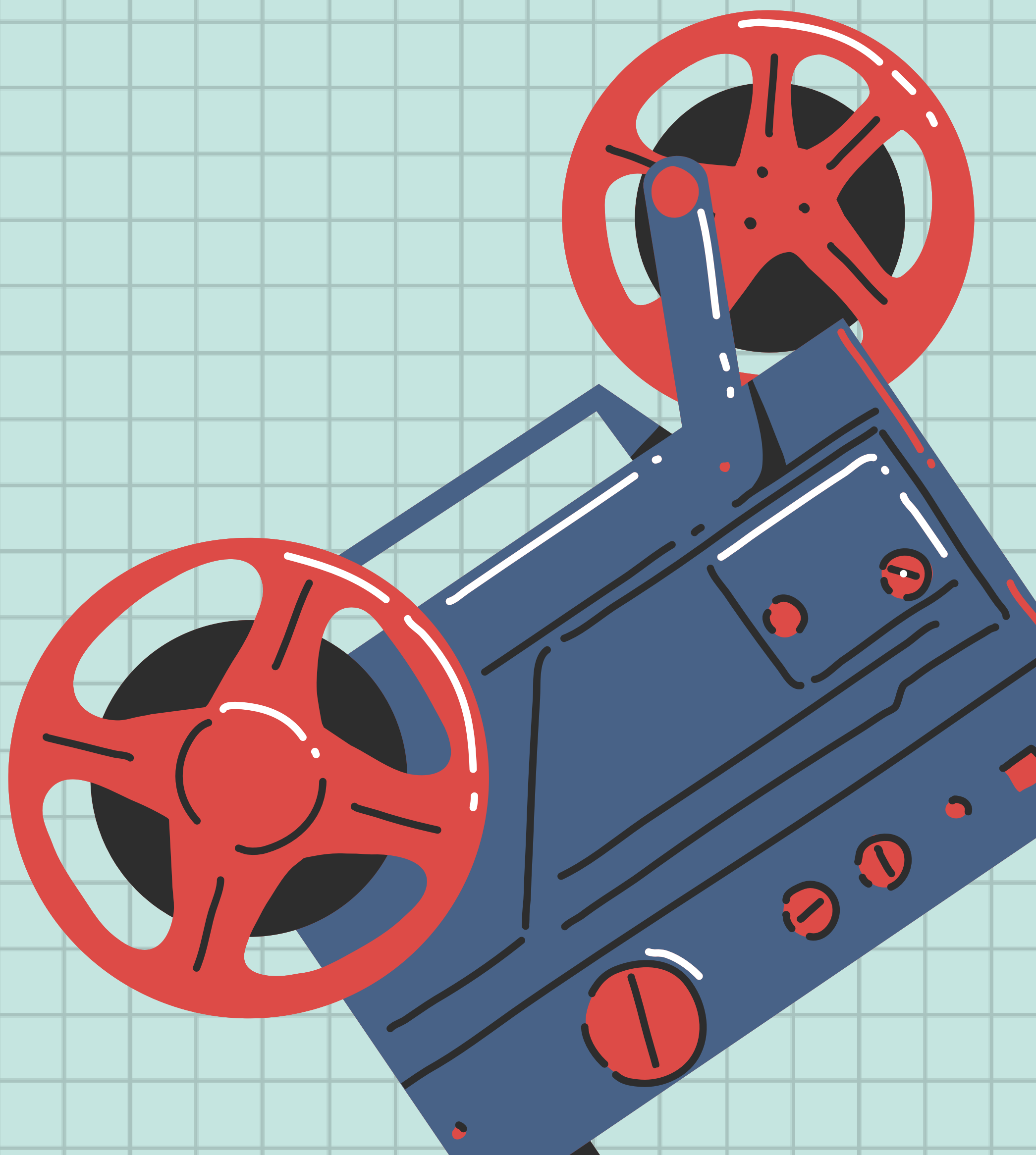
```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017605 entries, 0 to 1017604
Data columns (total 28 columns):
 #   Column                   Non-Null Count    Dtype
---  ------                   --------------    -----
 0   id                       1017605 non-null  int64
 1   title                    1017594 non-null  object
 2   vote_average             1017603 non-null  float64
 3   vote_count               1017603 non-null  float64
 4   status                   1017603 non-null  object
 5   release_date             904724 non-null   object
 6   revenue                  1017603 non-null  float64
 7   runtime                  1017603 non-null  float64
 8   budget                   1017603 non-null  float64
 9   imdb_id                  590676 non-null   object
 10  original_language        1017603 non-null  object
 11  original_title           1017594 non-null  object
 12  overview                 838363 non-null   object
 13  popularity               1017603 non-null  float64
 14  tagline                  151010 non-null   object
 15  genres                   722259 non-null   object
 16  production_companies     469581 non-null   object
 17  production_countries     615653 non-null   object
 18  spoken_languages         626897 non-null   object
 19  cast                     679181 non-null   object
 20  director                 833948 non-null   object
 21  director_of_photography  244148 non-null   object
 22  writers                  492821 non-null   object
 23  producers                323656 non-null   object
 24  music_composer           99252 non-null    object
 25  imdb_rating              429368 non-null   float64
 26  imdb_votes               429368 non-null   float64
 27  poster_path              719086 non-null   object
dtypes: float64(8), int64(1), object(19)
memory usage: 217.4+ MB
```

```
data_all.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 14561 entries, 2 to 1016135
Data columns (total 54 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   id                          14171 non-null  float64
 1   title                       14171 non-null  object
 2   vote_average                14171 non-null  float64
 3   vote_count                  14171 non-null  float64
 4   status                      14171 non-null  object
 5   release_date                14171 non-null  datetime64[ns]
 6   revenue                     14171 non-null  float64
 7   runtime                     14171 non-null  float64
 8   budget                      14171 non-null  float64
 9   imdb_id                     12210 non-null  object
 10  original_language           14171 non-null  object
 11  original_title              14171 non-null  object
 12  overview                    13396 non-null  object
 13  popularity                  14171 non-null  float64
 14  tagline                     10549 non-null  object
 15  genres                      14171 non-null  object
 16  production_companies        12835 non-null  object
 17  production_countries        12949 non-null  object
 18  spoken_languages            13093 non-null  object
 19  cast                        13836 non-null  object
 20  director                    13657 non-null  object
 21  writers                     12768 non-null  object
 22  producers                   11490 non-null  object
 23  imdb_rating                 11549 non-null  float64
 24  imdb_votes                  11549 non-null  float64
 25  log_revenue                 14171 non-null  float64
 26  log_budget                  14171 non-null  float64
 27  production_companies_list   14171 non-null  object
 28  genre_pairs                 14171 non-null  object
 29  release_date_year           14171 non-null  float64
 30  release_date_weekday        14171 non-null  float64
 31  release_date_month          14171 non-null  float64
 32  release_date_day            14171 non-null  float64
 33  release_date_quarter        14171 non-null  float64
 34  release_date_weekofyear     14171 non-null  UInt32
 35  Action                      14236 non-null  float64
 36  Adventure                   14236 non-null  float64
 37  Animation                   14236 non-null  float64
 38  Comedy                      14236 non-null  float64
 39  Crime                       14236 non-null  float64
 40  Documentary                 14236 non-null  float64
 41  Drama                       14236 non-null  float64
 42  Family                      14236 non-null  float64
 43  Fantasy                     14236 non-null  float64
 44  History                     14236 non-null  float64
 45  Horror                      14236 non-null  float64
 46  Music                       14236 non-null  float64
 47  Mystery                     14236 non-null  float64
 48  Romance                     14236 non-null  float64
 49  Science Fiction             14236 non-null  float64
 50  TV Movie                    14236 non-null  float64
 51  Thriller                    14236 non-null  float64
 52  War                         14236 non-null  float64
 53  Western                     14236 non-null  float64
dtypes: UInt32(1), datetime64[ns](1), float64(35), object(17)
memory usage: 6.1+ MB
```





Learning Tasks

Input

- Budget
- Popularity
- Release date features (year, month, weekday)
- Genre
- Runtime
- Cast and directors

Output

The goal is to predict the revenue a movie is likely to generate based on these input features.

	budget	popularity	release_date_year	release_date_month	revenue	predicted_revenue
0	10000000.0	16.561	1983.0	6.0	29450920.0	2.710456e+07
1	200000.0	0.600	2018.0	10.0	1000.0	-1.973619e+05
2	1560000.0	4.381	1957.0	5.0	1520000.0	3.103496e+06
3	120000000.0	43.873	2008.0	5.0	93900000.0	9.789097e+07
4	4520000.0	9.767	1983.0	3.0	7175592.0	8.883155e+06

Proposed Solution

Tools and Frameworks:

- Programming Language: Python on Jupyter Notebook
- Libraries: Scikit-learn, XGBoost, LightGBM, pandas, matplotlib, seaborn, etc

Algorithms Explored:

- Gradient Boosting
- XGBoost
- Random Forest
- Linear Regression
- Cross Validation

Solution Pipeline:

- Data Preprocessing: Handling missing data, outlier removal, and feature engineering.
- Model Training: Hyperparameter tuning with GridSearchCV.
- Evaluation: Cross-validation and model comparison.

Preproccesing → EDA → Feature Engineering → Modeling → Prediction

Challenges

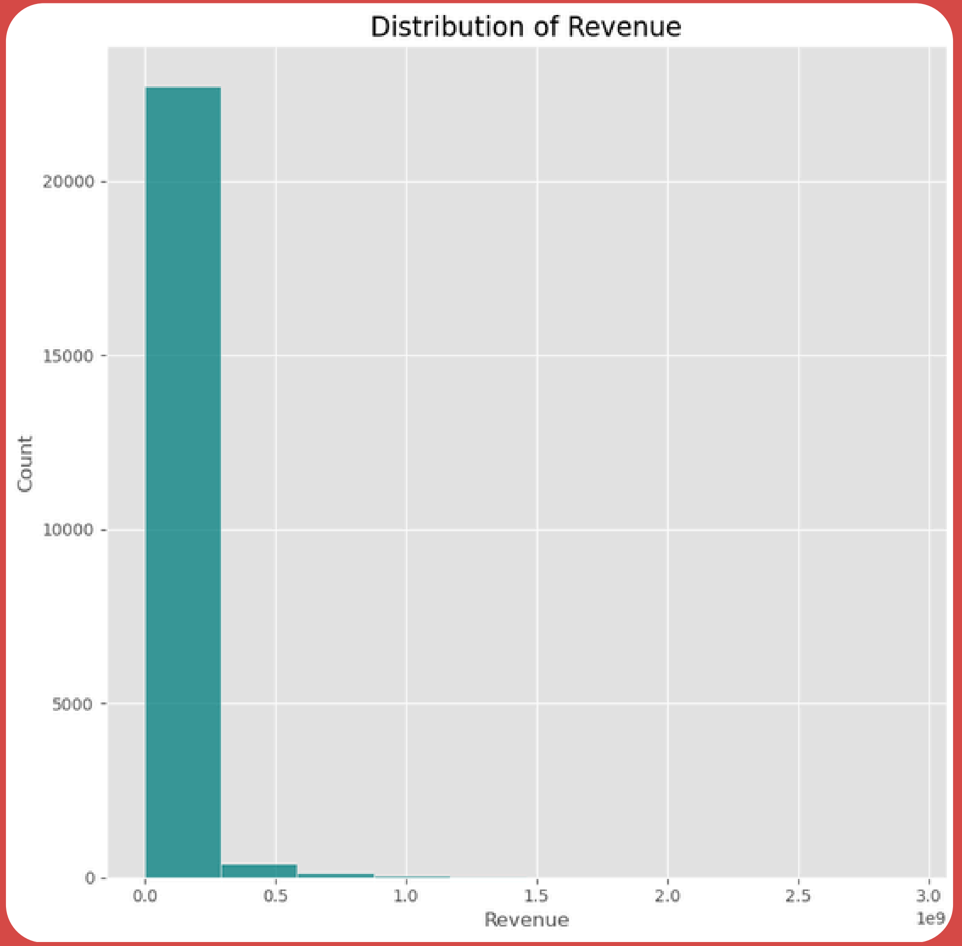
Challenge:

Revenue and budget data was heavily skewed, Most movies had relatively low budgets and revenues, while a few had exceptionally high values. This skewness can negatively affect model performance and predictions.

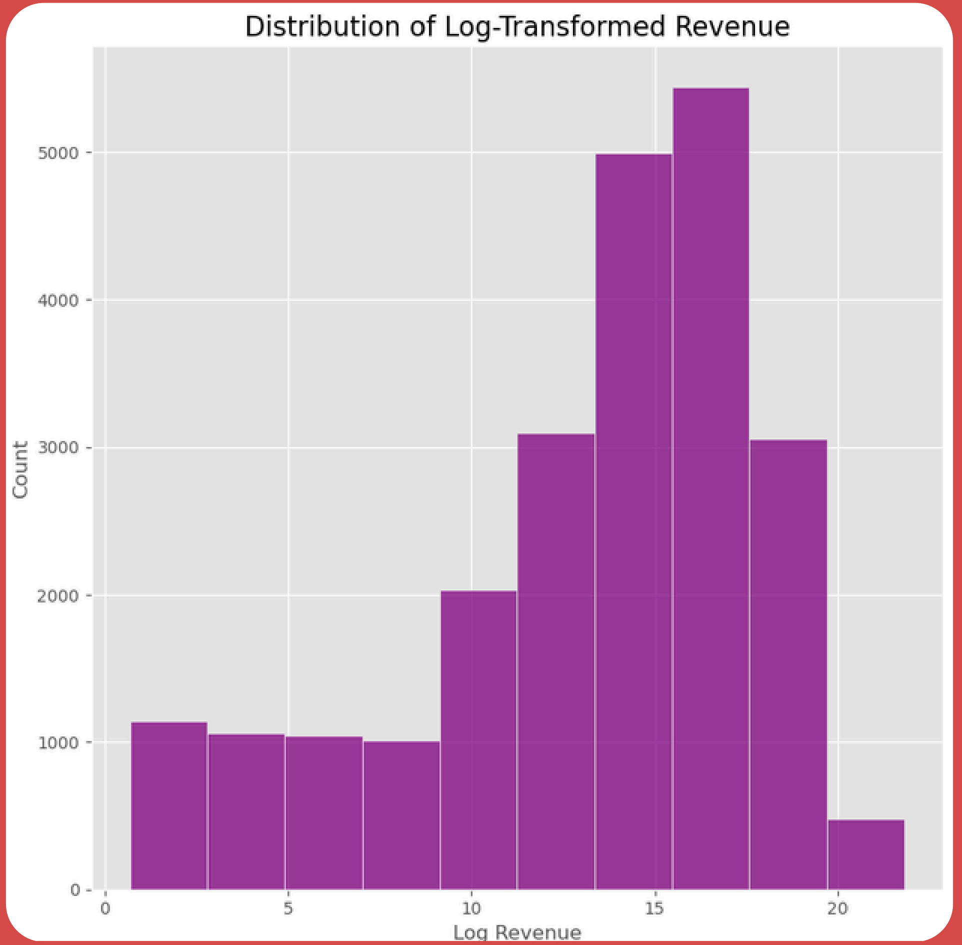
Solution:

A log transformation was applied to the revenue and budget data to compress large values and expand smaller ones. This transformation reduces skewness, making the distribution more normal and suitable for modeling.

Before:

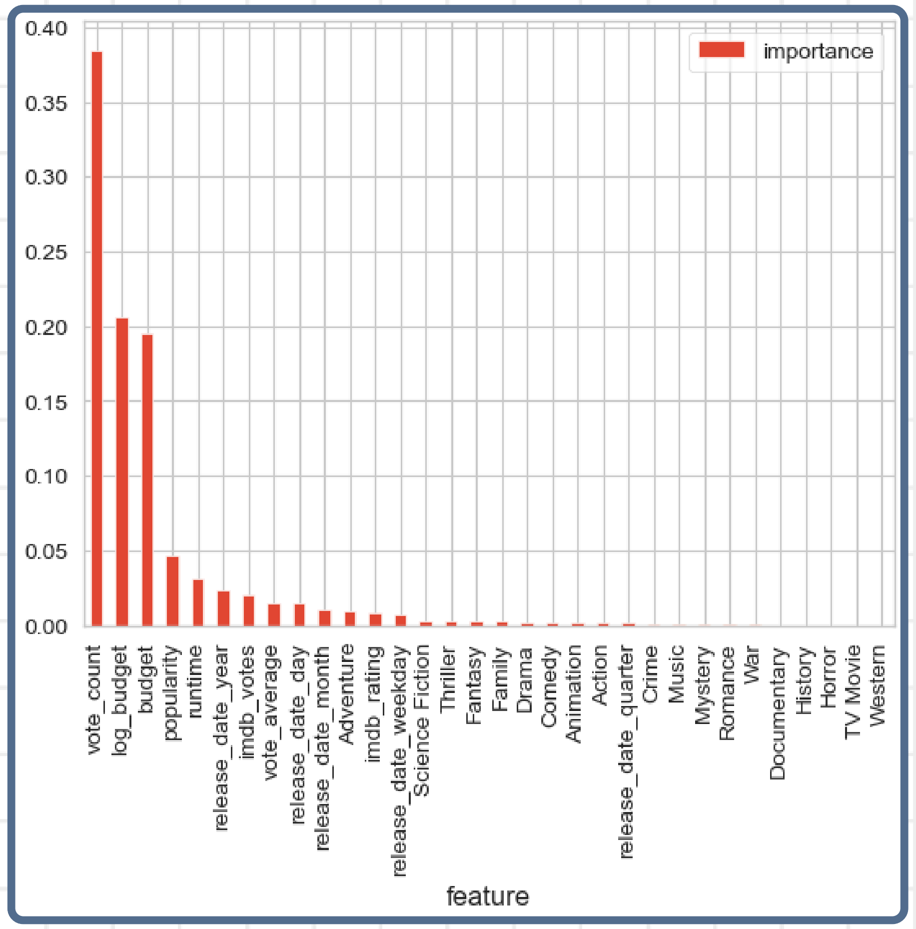


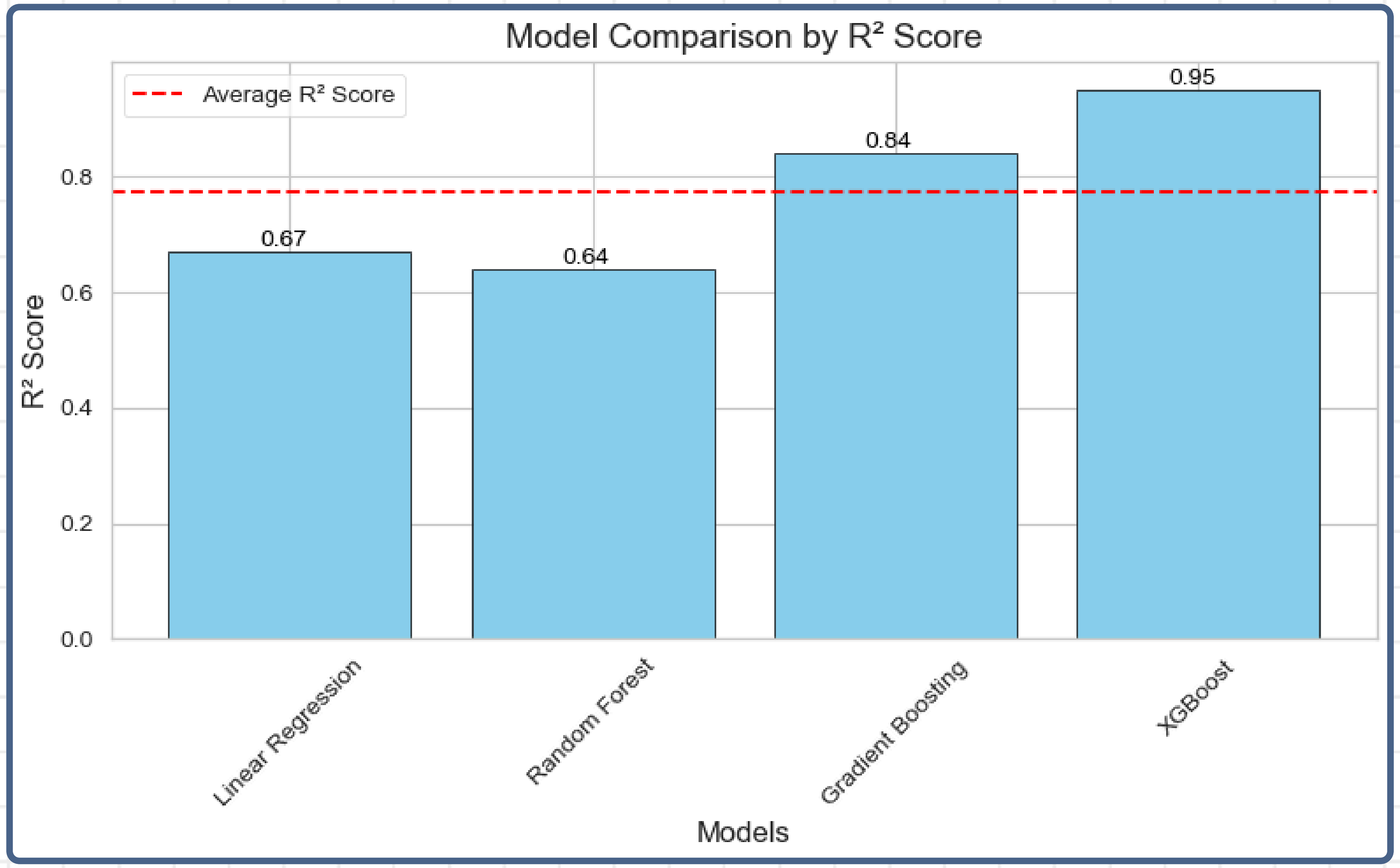
After:

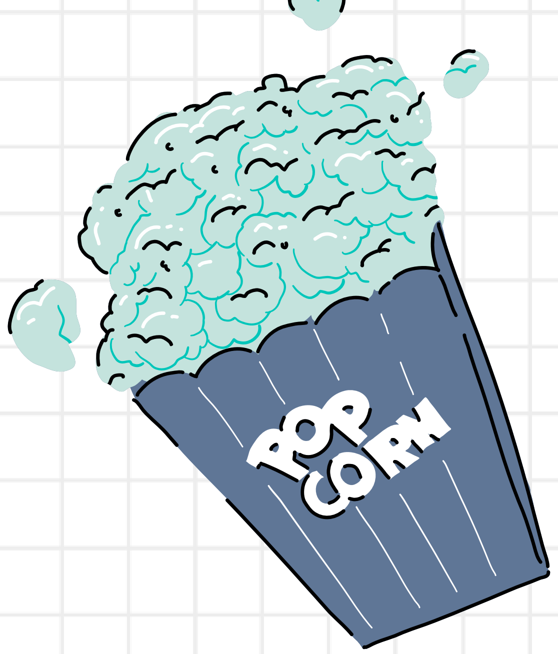


Current Results

Ensemble methods, such as XGBoost and Gradient Boosting, performed much better at predicting movie revenues compared to simpler models like Linear Regression and Random Forest. These methods work by combining multiple decision trees, allowing them to capture more complex patterns in the data. XGBoost was the best model, achieving an R^2 score of 0.95, meaning it could explain 95% of the variation in revenue. This shows that advanced machine learning techniques are highly effective for this type of analysis.







Current Results

Predicted Revenue for 'How to Train Your Dragon': 491,014,656.00
Log Predicted Revenue for 'How to Train Your Dragon': 20.01
Actual Revenue for 'How to Train Your Dragon': 494,879,471.00
Log Actual Revenue for 'How to Train Your Dragon': 20.02
Prediction Error: 3,864,815.00
Log Prediction Error: 0.01
Relative Prediction Error: 0.78%

Predicted Revenue for 'Inception': 852,675,637.72
Log Predicted Revenue for 'Inception': 20.56
Actual Revenue for 'Inception': 825,532,764.00
Log Actual Revenue for 'Inception': 20.53
Prediction Error: 27,142,873.72
Log Prediction Error: 0.03
Relative Prediction Error: 3.29%

Future Works

Enhancements to Explore:

- Add more features like franchise data or social media sentiment analysis.
- Incorporate actor and director popularity as predictors.

Improving the Model:

- Perform hyperparameter tuning for improved model accuracy.
- Explore advanced methods like deep learning.

Real-World Application:

- Develop a tool for production companies to estimate revenue based on metadata.

Conclusion

In conclusion, this project demonstrates how machine learning models, particularly XGBoost, can effectively predict movie revenue. Preprocessing and EDA played a crucial role in enhancing the dataset's quality, enabling strong model performance. While the results are promising, there is room for refinement to further improve prediction accuracy.