Prediction of Movie Revenue

Arufa Khanom, Zachary Formes, Ana Ramirez

Machine learning offers movie producers insights into profitability, allowing for informed financial adjustments.

The problem:

- The Movie industry generates over $100 billion in revenue every year.
- However, over 80% of films produced do not make a profit.

⇩

Understanding the factors that drive movie profitability can lead to more enjoyable and financially successful films.

Why use a model?

- Movie producers who are gearing up to release new movies could use our model to better understand how well their movie would perform, and make adjustments accordingly.
- For example, a producer could decide to shift their release date to a new time of year because it appears like it will be more profitable.

1. Our data...

kaggle **Movies Daily Update Dataset**

Description of initial dataset: 700,000 movies listed in the TMDB Dataset
Included title, genres, overview, release date, cast, crew, plot keywords, budget, revenue, posters, companies, vote counties, reviews...

2. Data cleaning

1. Dropped unnecessary columns that would not function as predictors (as they were information gathered from AFTER a movie is released, not prior):
   a. Poster, popularity, overviews, reviews, status, recommendations, backdrops, votes, keywords
2. Dropped invalid movies (no revenue, no budget (lost most data here))
3. Split genre column into columns for each genre (dummy variables for each)
4. Dropped movies with null pertinent values (language)
5. Created a “main_actor” column, getting the name from the “credits” column
6. Separated date column into day, month and year columns. Also made a “season” predictor
7. Created predictor column of actor previous success: grouped movies with same actor and calculated the mean ratio. If no other movies are found for the same actor, a 0 is put in place.

2. Data cleaning

8. Created a number of competitors released within the same month predictor

Data fetching:

- We wanted to calculate the director's previous success as a predictor, however, the dataset did not have director information..

9. Installed tmdbsimple, a python wrapper for data fetching for the TMDB dataset API
10. Fetched directors from the API based on the movie ID the dataset provided for each movie

- We also wanted to use age rating as a predictor, but ran into same issue of database not having such information...

11. Fetched age rating information from API based on id for each movie
12. If age rating value was not available or API call did not work, "unrated" was added

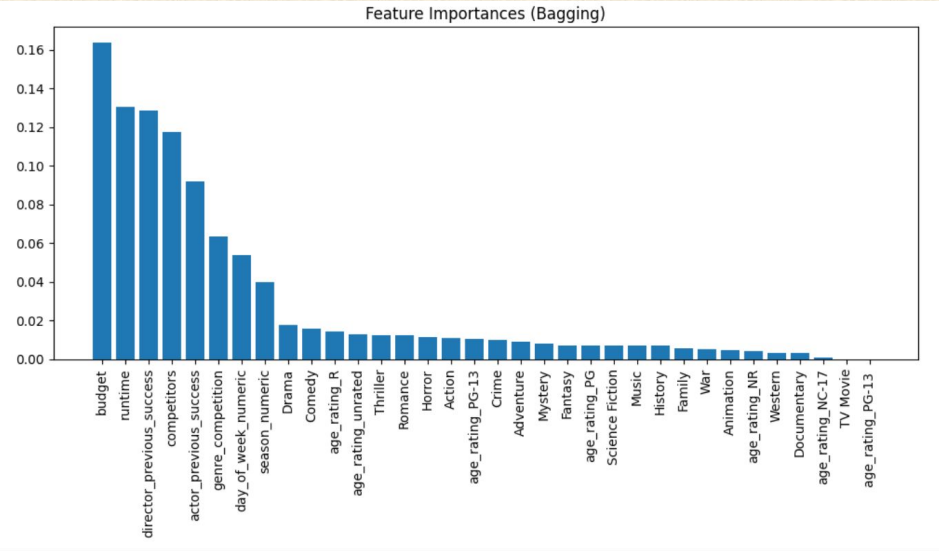
13. Calculated director previous success (similar as main actor's previous success) and made a predictor column
14. Dummy variables for age_rating
15. Dropped unnecessary columns: credits, genre, director, firstActor, companies

Now we ensured we have only numeric predictors, size of our data frame ended up being (9489,41)

2. Data summary- Features added + Exploratory data analysis

Features added:

- Season
- Competitors by season
- Competitors by genre
- Main Actor's previous success
- Director's previous success
- Age rating
- Genres

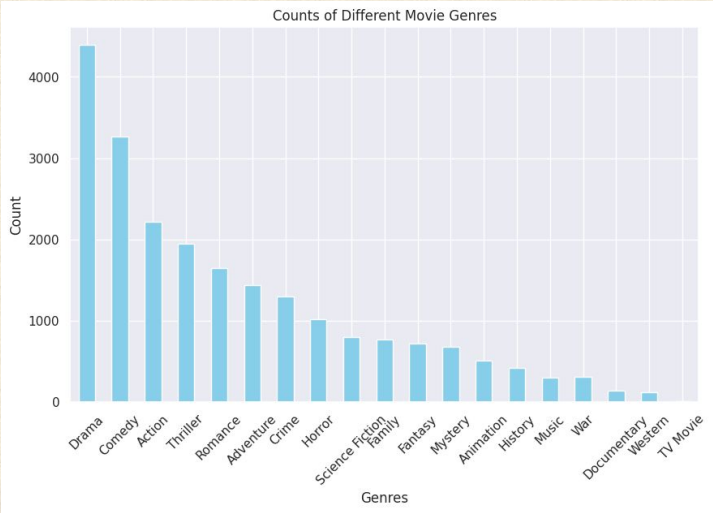


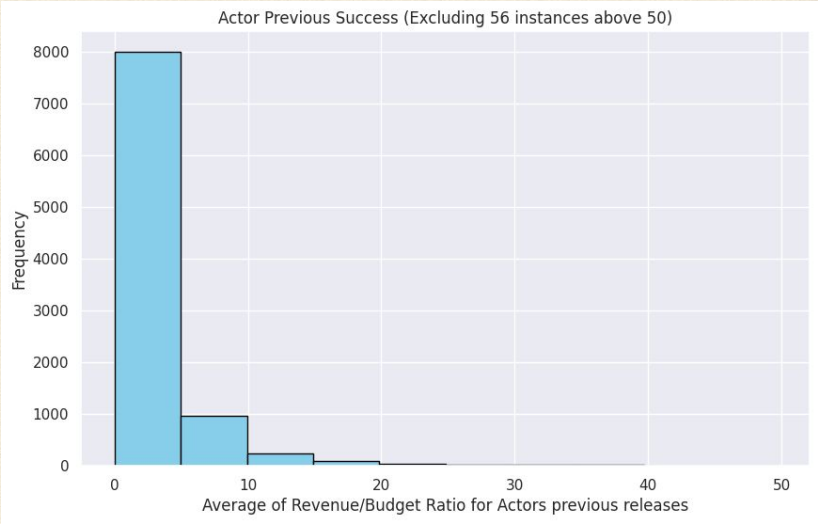
Our goal:
Predict revenue from our variety of predictors.
Data space:

```
Highest revenue: 2,920,357,254.0
Lowest revenue: 1.0
```

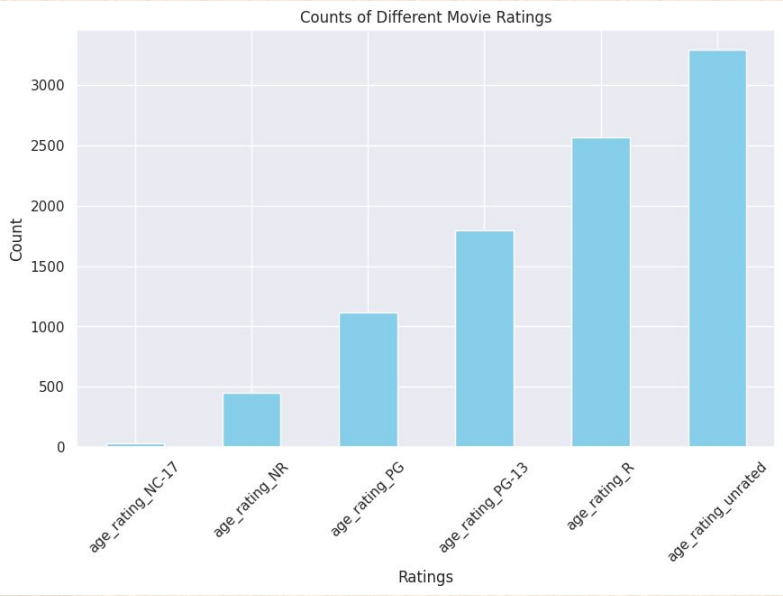
- Avatar
- Soap (Short film)

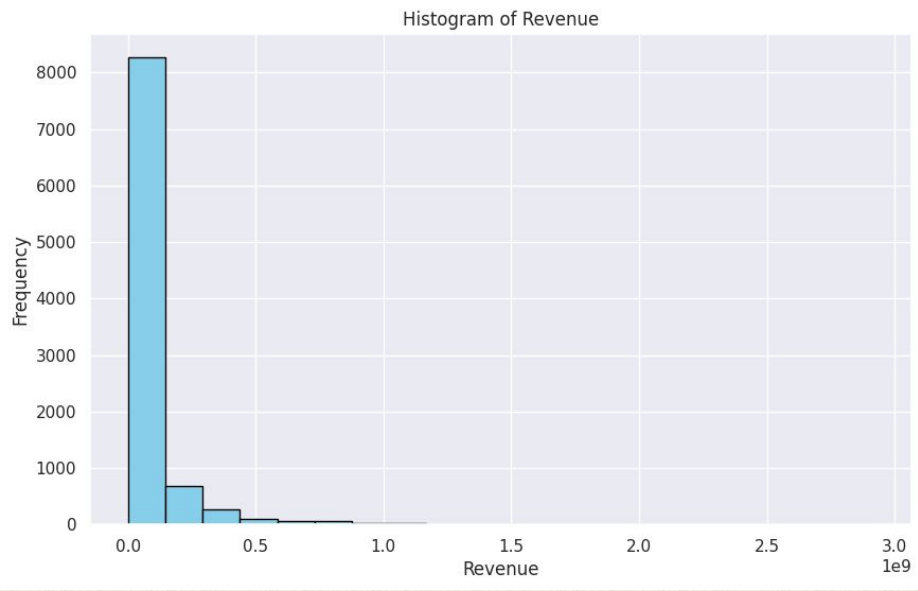
4. Presentation of descriptive analysis (plots and tables)





4. Presentation of descriptive analysis (plots and tables)





Modeling

Simple baseline model: Predicting the mean of the target variable for all samples.

```
Simple baseline model predictions
Train Baseline Mean MSE: 2.538207943782218e+16
Test Baseline Mean MSE: 2.5490427484196852e+16
```

1. Initial approach: Linear, LASSO and Ridge

***common theme: signs of overfitting from the model, as alpha gets tuned, MSE decreases, but overfitting increases. Bias/variance tradeoff.

a. Simple linear regression

```
inear Regression model - Train MSE: 1.1206112764208054e+16
inear Regression model - Test MSE: 1.2803847258601242e+16
```

b. LASSO

1. Initially tuned alphas and range tested with trial and error:

```
Best alpha: 625055.1925273977
Train MSE with best alpha: 1.1228009505142858e+16
Test MSE with best alpha: 1.2795609645781912e+16
```

2. Used GridSearchCV and best alpha calculated before as a range, to find best alpha for LASSO model

Decreased overfitting from LASSO 1. Did not improve MSE.

```
Best alpha with cross-validation: 694505.7694748862
Train MSE with best alpha found using cross-validation: 1.1231060232875704e+16
Test MSE with best alpha found using cross-validation: 1.2795634180933116e+16
```

```
Train RMSE with best alpha found using cross-validation: 105976696.65013957
Test RMSE with best alpha found using cross-validation: 113117788.96766466
```

Modeling

```
Simple baseline model predictions
Train Baseline Mean MSE: 2.538207943782218e+16
Test Baseline Mean MSE: 2.5490427484196852e+16
```

1. Initial approach: Linear, LASSO and Ridge

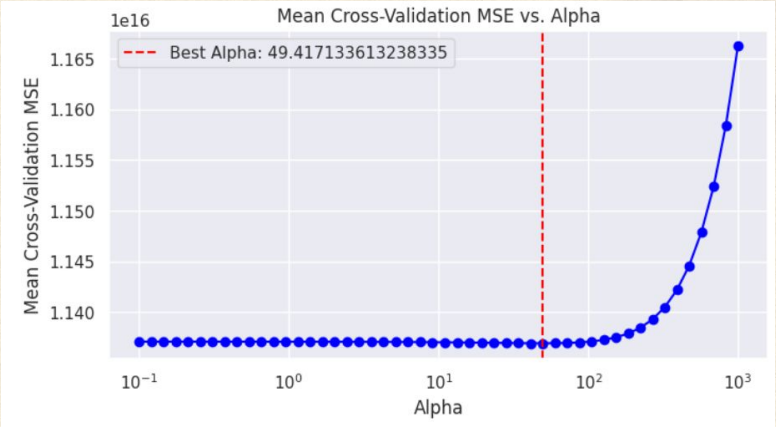
c. Ridge

Tuning:

GridSearchCV object :

```
param_grid = {
    'alpha': alphas,
    'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga'],
    'max_iter': [1,5,10,50,100]
}
```

Best hyperparameters:
{'alpha': 49.417133613238335, 'max_iter': 1, 'solver': 'auto'}

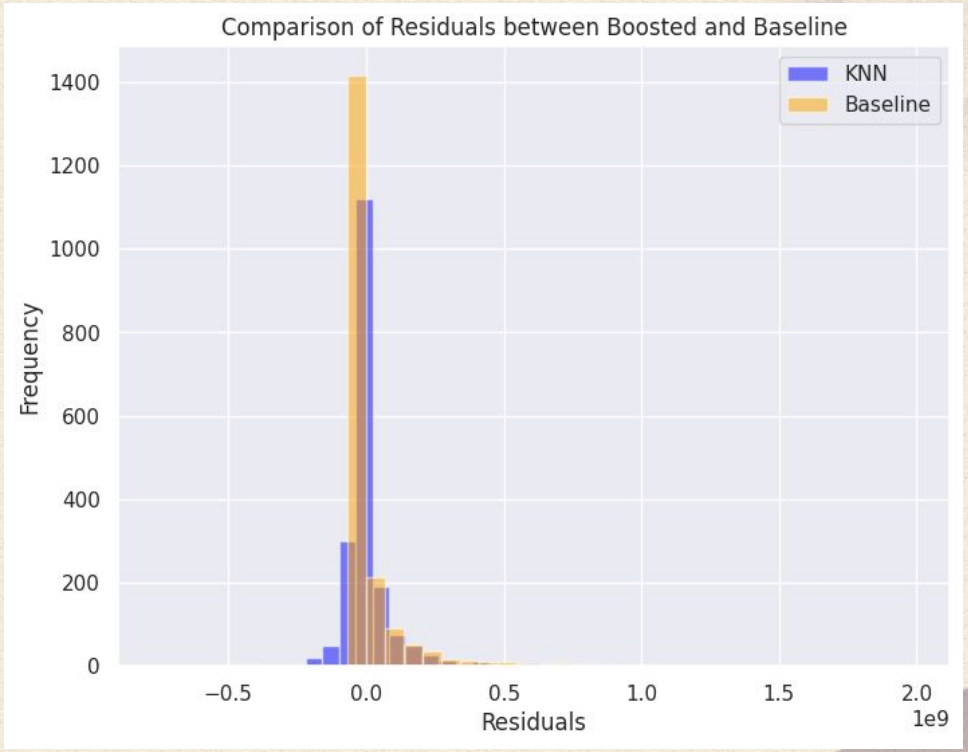



```
Train MSE with best hyperparameters: 1.1207206706721056e+16
Test MSE with best hyperparameters: 1.2785545761414382e+16
```

Modeling

KNN Model :

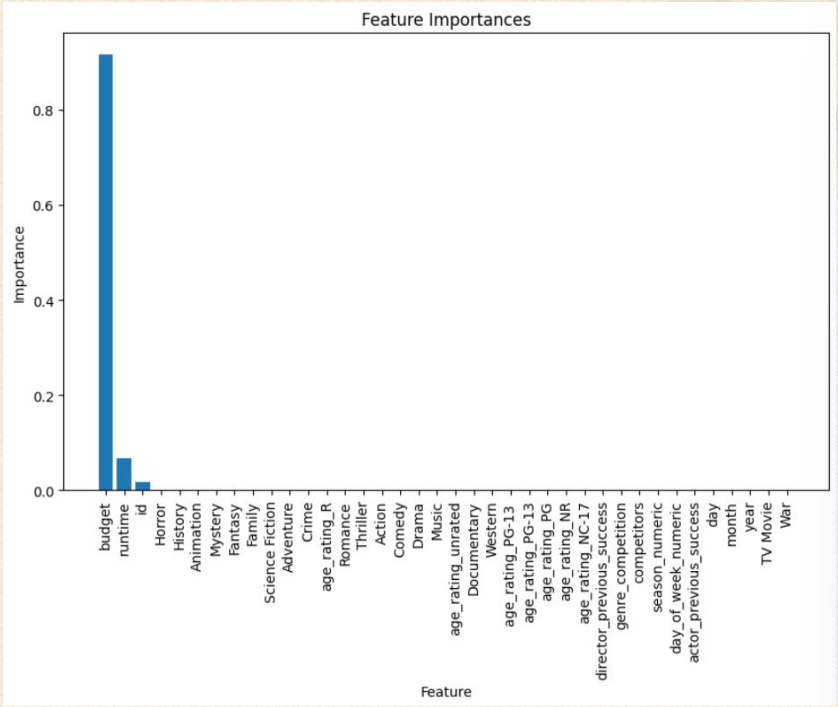
- ***Tuned using 5 fold CV, with number of neighbors between 1 to 100.***
- ***Best Number of Neighbors: 18***
- ***Test MSE: 1.31 e+16***
- ***RMSE: 116,750,459***



Modeling

Decision Tree :

- *Used input parameters of*
  - *MaxDepth: 3*
- *Train MSE: 1.06e+16*
- *Test MSE: 1.34e+16*

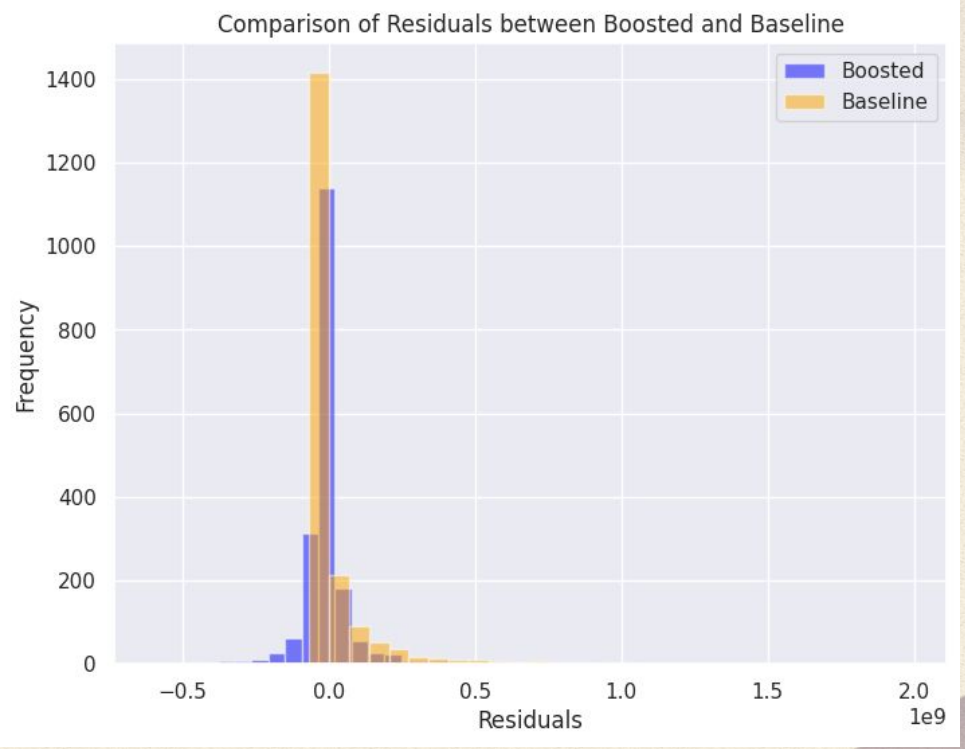


**Chosen model

Modeling

Boosted Decision Tree :

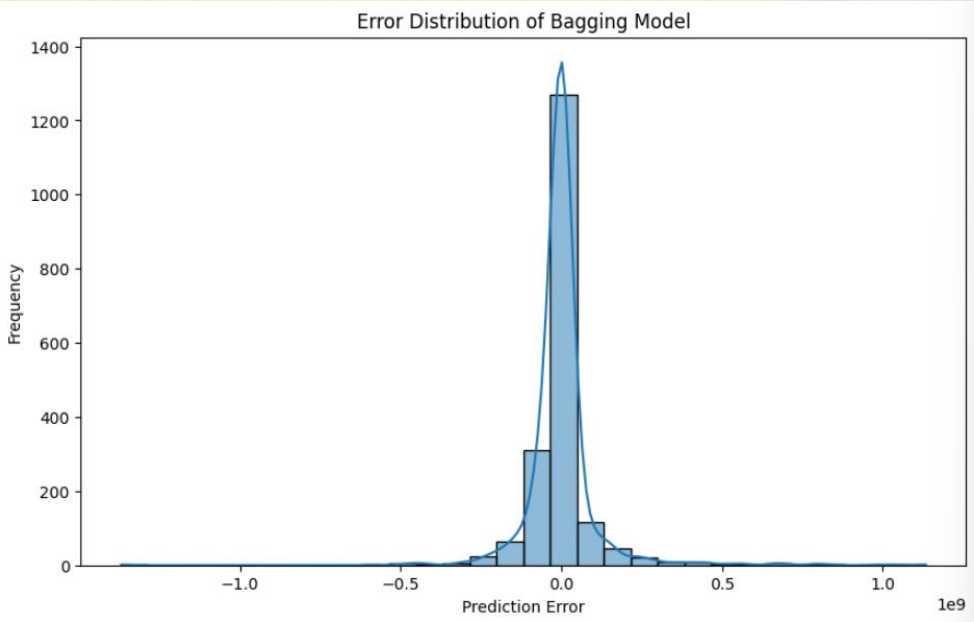
- *Tuned using GridSearch*
- *Used input parameters of*
  - *Max Depth: 1, 3, 5, 10*
  - *Num Trees: 25, 50, 100, 300*
  - *Learning Rate: .001, 0.01, 0.1, 1*
- *Train MSE: .66 e+16*
- *MSE: 1.15 e+16*
- *RMSE: 107,354,359*



Modeling

Bagging:

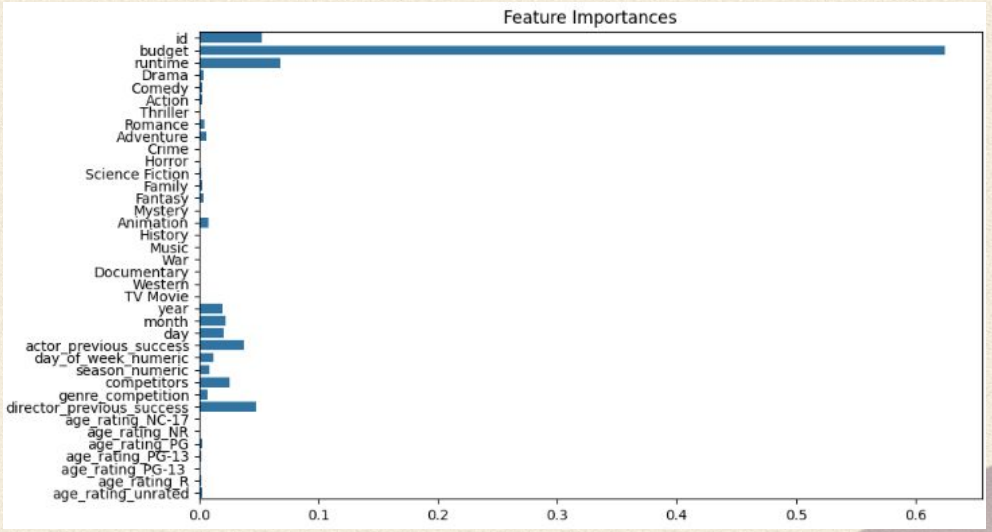
- *Linear Regression Base*
- *Used input parameters of*
  - *N_estimators=50*
- *Train MSE: 1.12e+16*
- *Test MSE: 1.28e+16*



Modeling

Random Forest:

- *GridSearchCV*
- *Used input parameters of*
  - *N_estimators=100, 200*
  - *max_depth= none, 10*
  - *min_samples_split=2, 5*
- *Train MSE: 2.67e+15*
- *Test MSE: 1.18e+16*



Modeling

Also Attempted Neural Networks...

Tanh Activation Model :

- *Input Layer with 64 Neurons*
- *Hidden layer with 32 Neurons*
- *Output layer with 1 Neuron*
- *20 Epochs*
- *MSE as Loss Function*
- *MSE: 3.08 e+16*

Relu Activation Model :

- *Input Layer with 64 Neurons*
- *Hidden layer with 32 Neurons*
- *Output layer with 1 Neuron*
- *20 Epochs*
- *MSE as Loss Function*
- *MSE: 3.06 e+16*

Did not fully tune either model, so there is potential for better results.

Challenges

- Selecting Y Variable → Shift from Rev/budget ratio to revenue
- Computational Limits → Used models that required less tuning
- Lack of predictors → Engineered features that could be helpful
- Extreme Outliers → Limited our Dataset to usable instances

Conclusion

Limitations:

- Budget Variable (skewed the result)
- Other important movie revenue aspects not included in the dataset (marketing $, distribution company, streaming platform etc.)
- Dataset included short movies (including youtube videos)

Potential impact of the model:

Movie directors can focus on key factors that are associated with higher revenues, such as budget and runtime, while also considering additional elements like the past successes of directors and actors.

Takeaway:

Given our current data and predictors, we are unable to make effective or accurate revenue predictions for movies. More comprehensive information would be required to improve the accuracy of our predictions.

Thank You