Introduction and Background

In the film industry, word of mouth, defined as “informal communication among consumers about products and services,” plays a significant role in consumers’ decision-making process. Because the power of word of mouth is often cited as a key factor in the success of certain movies, movie studios and distributors have a clear incentive to understand what consumers are saying [1]. Fortunately, user-generated online text or “electronic word of mouth” has proliferated in recent decades, representing a rich dataset for understanding consumer sentiment [2, 3].

Analyses of user-generated content for movies appears in the literature, such as a Naïve Bayes sentiment analysis conducted by Novendri and colleagues [4] or box office success predictions based on movie reviews [2]. Our project seeks to build on these analyses by analyzing comments on trailers posted to the r/movies subreddit to predict movie success, defined as money made during a movie’s opening weekend. To do so, we will use a custom dataset from the APIs of Reddit, TMDb, and OMDb containing movie metadata and user comments on trailers.

Problem Definition

For both movie creators and viewers, differentiating one film from the massive amount released each year presents a challenge. From a studio’s perspective, determining which word-of-mouth metrics predict financial success and / or attention could lead to a more optimized or profitable movie promotion strategy. *Meanwhile, from a moviegoer’s perspective, a model that sorts through public comments on movies and clusters them into topic could give an idea of what the notable aspects of a movie’s trailer might be.*

Methods

BERTopic

To gain insight into the general landscape of conversations that occur on r/movies, we decided to implement an unsupervised clustering method known as topic modeling. By understanding what conversation topics are occuring on social media, as well as which of them are correlated with high performing movies, we may be able to extract useful features that could help in our predictions. Text already presents unique challenges in modeling, and social media likely adds even more complexity. As a medium, social media text is almost exclusively short and unstructured. In addition, since so many different people are contributing text, there is naturally more variation in the vocabulary than if there was a single writer.

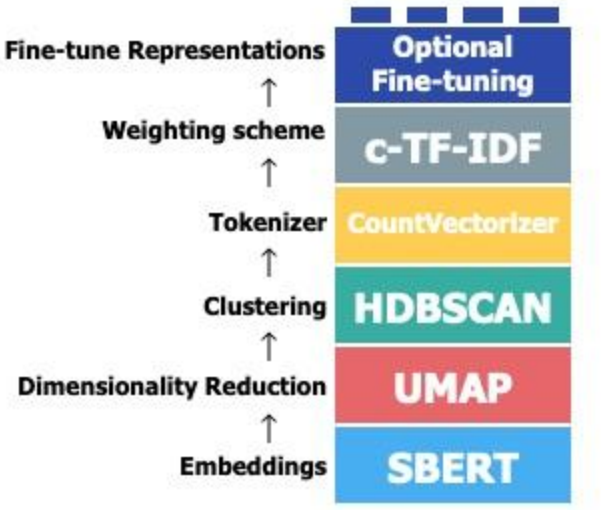
A key resource for this step was a study [5] by Egger and Yu in which they evaluated and compared four topic modeling techniques: latent Dirichlet allocation (LDA), non-negative matrix factorization (NMF), Top2Vec, and BERTopic. Their research was applied to Twitter posts, but we felt that this should still be applicable to another text based social media site like Reddit. Based on the suggestions from the paper and our own experimentation, we decided to implement BERTopic.

BERTopic is a topic modeling technique that leverages 🤗 transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions. Documentation

Text Preprocessing Methods

Since BERTopic relies on text embeddings, it is best to keep the structure of the comments as close to their original form as possible. This allowed us to skip many of the usual NLP preprocessing methods and minimize the time spent on text cleaning. However, we did need to remove duplicate comments, remove deleted comments, remove links and remove any other non-word text. After that, the only preprocessing needed is to create the embeddings.

The Algorithm



1. Embed documents using sentence-transformers
2. Dimensionality reduction using UMAP
3. Clustering comments using HDBScan
4. CountVectorizer to determine word frequencies within clusters
5. Class-based TF-IDF to determine words most important to topic

Each of these steps has its own benefits which are worth mentioning. The big benefit of embeddings is the ability to identify semantic similarity. For example, our topic model would be able to determine that “song” and “music” have more similar meanings than “song” and “graphics”. These representations are far too large and require dimensionality reduction. Using UMAP can preserve some local and global structure of the comments dataset. HDBScan doesn’t require you to predetermine the number of clusters, and is able to allocate comments to clusters without requiring that every comment be part of a group. This allows for outliers and will improve overall coherence of the topics. Finally, by defining our clusters in terms of the most important words using TF-IDF, we can immediately extract topics.

Multiple Linear Regression

Preprocessing Methods for Supervised Models:

1. **Reddit Comments Dataset Preprocessing:**
    a. Pre-processed a dataset of Reddit comments by filtering out comments posted after the movie release dates and cleaning the text. This involved

removing special characters, URL links, and deleted comments to ensure data quality and relevance.

b. utilized a pre-trained RoBERTa-based Transformer from hugging face to generate sentiment scores for each comment, categorizing them into positive, negative, and neutral probabilities.
c. Summarized the Reddit comments data at the post ID level to obtain the audience sentiment for a movie based on the comments.

2. **Sentiment Score Weighting:**
    a. Calculated weighted sentiment scores for each post by integrating upvotes with sentiment scores. This generated 'weighted positive scores,' 'weighted negative scores,' and 'weighted neutral scores' for a comprehensive sentiment analysis at the movie level.

[Weighted sentiment formula](/weighted_sentiment.png)

3. **Feature Engineering on Sentiment Data:**
    a. Created metrics such as the number of positive comments, number of negative comments, % positive comments, and % negative comments.

Integration with Movie Dataset:

1. **Merging and Cleaning:**
    - Merged the summarized Reddit comments data with a movie dataset, followed by filtering out irrelevant features to focus on key variables.
2. **Feature Engineering:**
    - Introduced new features such as the year and quarter of the movie release date.
    - Flagged movies with missing sentiment data to handle incomplete records accurately.
    - Applied Multiple Correspondence Analysis (MCA) on genre features(binary) using the 'prince' module to reduce dimensions from 19 to 6 components. This step was based on the hypothesis that certain genre combinations occur more frequently (e.g., action/adventure/sci-fi or romance/comedy).

      [Cumulative Explained Variance](/MCA_cumulative_var.png)

3. **Categorization of Continuous Variables:**
    - Segmented movies based on runtime into four categories (<100min, 100-120 min, 121-150 min, 150+ min) and grouped movie ratings into major categories (General audience, Teen+, Adult, Others, and Missing).
4. **Encoding Categorical Variables:**

- Applied one-hot encoding to the categorical variables to prepare the dataset for machine learning applications.

5. **Exploratory Data Analysis:**
   - We analyzed the distribution of Reddit post engagement over time, alongside the correlation between domestic openings and trailer posted dates. This analysis helps identify potential biases in the data, such as reduced engagement during earlier periods and a noticeable decline in both online engagement and box office performance during the pandemic period.
     [comments with time](/comments_with_time.png)
     [BO_v_time](/BO_v_time.png)

   - Distribution of input and target features such as budget, Post scores, num_comments and Domestic opening on log scale.
     [Hist of BO](/hist_log_domestic_opening.png)
     [Hist of Budget](/hist_log_budget.png)
     [Hist of post score](/hist_post_score.png)
     [Hist of number of comments](/hist_num_comments.png)

   - Quarterly analysis of movie release trends shows that the first quarter (Q1) generally sees the fewest number of releases.
     [Qtrly trend of movies](/qtrly_trends.png)

   - Checking for correlation among predictors and the target variable.
     [correlation matrix](/correlation_chart.png)

MLR Model Development and Refinement**:**

1. Baseline Model:
   - Built a baseline Multiple Linear Regression (MLR) model to set a preliminary standard for performance evaluation.
   - Model performance:
     [baseline_MLR_results](/baseline_MLR_results.png)

   - Model diagnostics - Heteroscedasticity & Normality violations.
     [baseline_diagnostics](/baseline_diagnostics.png)

- ***Observations***: The residual plots clearly indicated non-linear relationships between the input and target variables and skewed distribution among numerical features. Additionally, the significant discrepancy between the RMSE and Median Absolute Error suggests that outliers significantly impact the model's performance. To address these issues, we plan to implement feature transformations and apply outlier treatment strategies in our next steps.

2. **New MLR model with Feature Transformation:**
    - Based on the distribution and some trial and error the following transformations were applied - log transformation on 'target variable' and 'post score', Square root on 'Budget' & Box Cox on 'comments'.
    - Linear regression model is fitted on the transformed dataset which resulted in the below performance:
      [MLR_trans_results](/MLR_trans_results.png)

    - Model diagnostics - Improvement in Heteroscedasticity & Normality assumptions but still not ideal with non-constant variance and left skewness.
      [MLR_trans_diag1](/MLR_trans_diag1.png)
      [MLR_trans_diag2](/MLR_trans_diag2.png)

    - ***Observations***: The updated residual plots show marked improvement over the baseline model, with a notable increase in the randomness of residual variance. However, a linear trend in the residuals versus log(y) suggests a violation of the constant variance assumption. Despite this, both the adjusted R-squared and Median Absolute Error have shown significant improvements compared to the baseline models. Conversely, the RMSE has deteriorated. Our next step will involve identifying and possibly eliminating influential points to further enhance the model's performance.

3. **MLR model without the influential points:**
    - Using cook's distance, identified influential points where cook's dist > 4/#datapoints and removed them from the training dataset.
    - A new regression model is fitted using the training dataset with the following performance metrics
      [MLR_final_results](/MLR_final_results.png)

    - Model diagnostics - The Q-Q plot suggests the normality assumption is valid but heteroscedasticity is still an issue.

[MLR_final_diag](/MLR_final_diag.png)

- ***Observations***: The updated residual plots show marked improvement over the baseline & Transformed models, with a notable increase in the normal distribution of residuals. However, a linear trend in the residuals versus log(y) suggests a violation of the constant variance assumption. Based on the performance metrics the model seems to be a good middle ground between Baseline & Transformed model. Our next step will involve exploring non-parametric models such as XG Boost & Random forest due to nonlinear relationships observed from the regression model results.

Random Forest and XGBoost Tree-Based Models

Since we wanted the models to be comparable, the tree-based models started with the dataset that existed after the first 10 steps of the MLR preprocessing explained above. Our first step was to get a baseline for both random forest (RF) and XGBoost (XGB) models. To do so, we trained each model on 80% of the full dataset using the default parameters in scikitlearn's RandomForestRegressor and xgboost's XGBRegressor. We then tested the trained models on the remaining 20% and compared the RMSE and R^2 values between the training and testing sets (reported in the table below).

[Base_Trees](/base_trees.png)

At this point it was clear that both default models were severely overfitting to the training data, a common weakness of tree-based models. To solve this, we generated new variations of the dataset, each aimed at addressing a different potential cause of overfitting. We split each dataset into a training set (80%) and a testing set (20%), using a consistent random seed to ensure the same rows of datasets of the same size would end up in the same set. We then took each dataset through a grid-search cross-validation process for both RF and XGB.

First, we know from the data preprocessing steps that we were missing reliable production budget data for 420 out of the 1,359 movies in our dataset, so we tried training both models on the subset of data that excluded any movies for which we were missing budget data. As another workaround for missing budgets, we also tried training the models after removing the budget column altogether.

Our next approach was feature selection. To do this, we calculated the permutation importance of each feature in the baseline RGB and XGB models,

then created two new subsets of data, each containing only the features with a permutation score above 0.005.

Finally, we know tree-based models are particularly susceptible to outlier influence. With that in mind, we looked for outliers in the numeric columns of our data using boxplots. Although multiple features did show data points that could be considered outliers, we narrowed our focus to the features for movies budget and positive comments on reddit posts, since those were the two most important features for both RF and XGB in the baseline models.

[Budget_Boxplot](/budget_boxplot.png)

[Pos_Comments_Boxplot](/positive_comments_boxplot.png)

Based on these results, we constructed 4 slightly different datasets. The first removed the largest budget outlier, and the second removed the 4 largest budget outliers. We then duplicated each of those datasets and removed the positive comments outlier from each as well.

That left us with 10 permutations of our original dataset on which to train and test models. After going through the entire process, a few trends became clear.

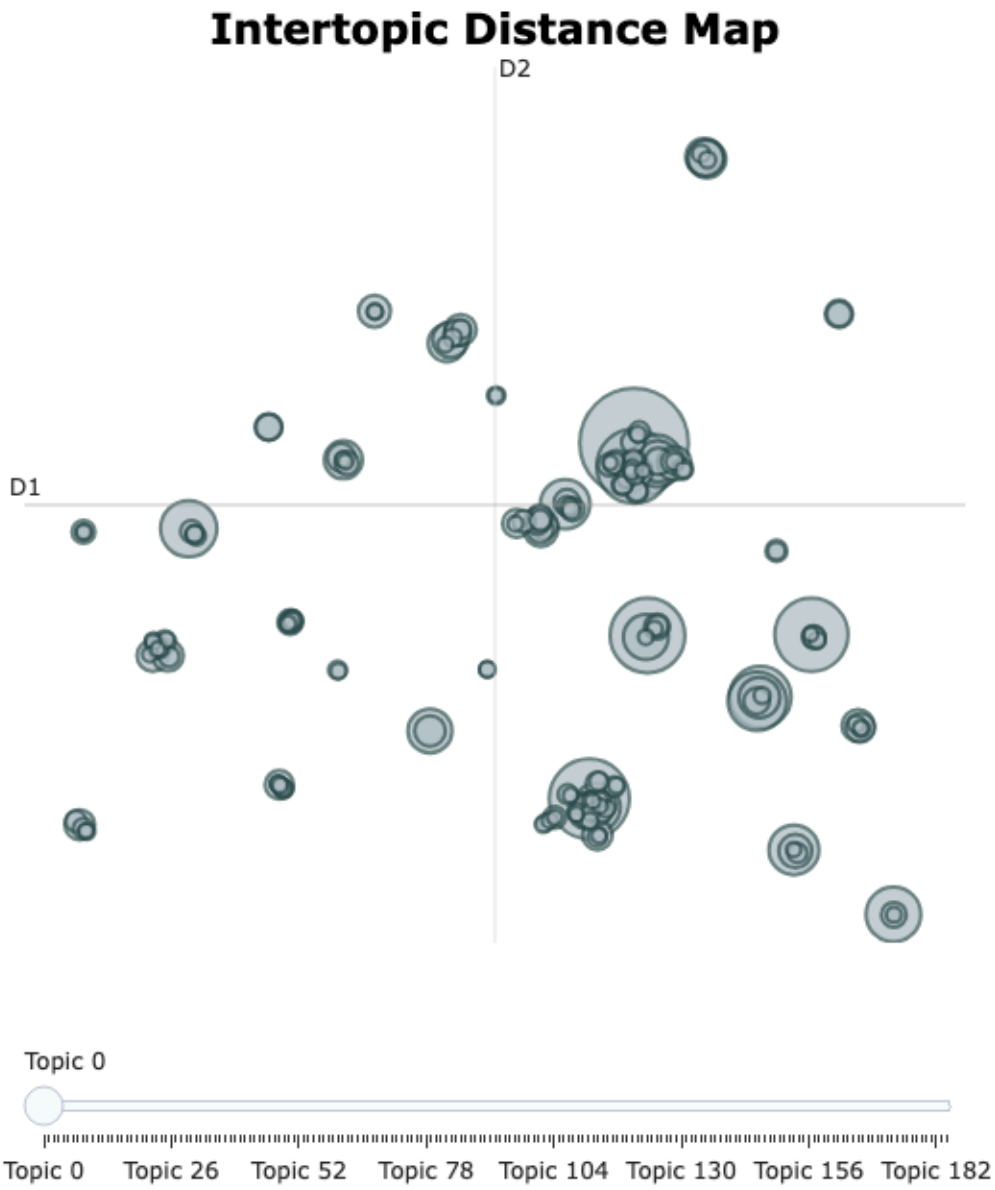
1) XGBoost models tended to heavily overfit the training data, with an average increase in RMSE of approximately $16 million and an average decrease of 0.45 in adjusted R^2 between training and test sets.
2) In addition to overfitting, when comparing RF to XGB on the same dataset, XGB was often both overfit and performed worse than RF on the test set.
3) Focusing only on the RF models, the various permutations performed similarly, with test adjusted R^2 values ranging from 0.59 to 0.66 and test RMSE values ranging from approximately $15 million to $21 million.
4) Most interestingly, the full dataset with no features or data points removed had the 3rd highest adjusted R^2 value and the lowest RMSE value on the test set, while also not appearing to be particularly overfit. For these reasons, we will focus our results analysis on the random forest model fit on the full dataset.

Results and Discussion

BERTopic

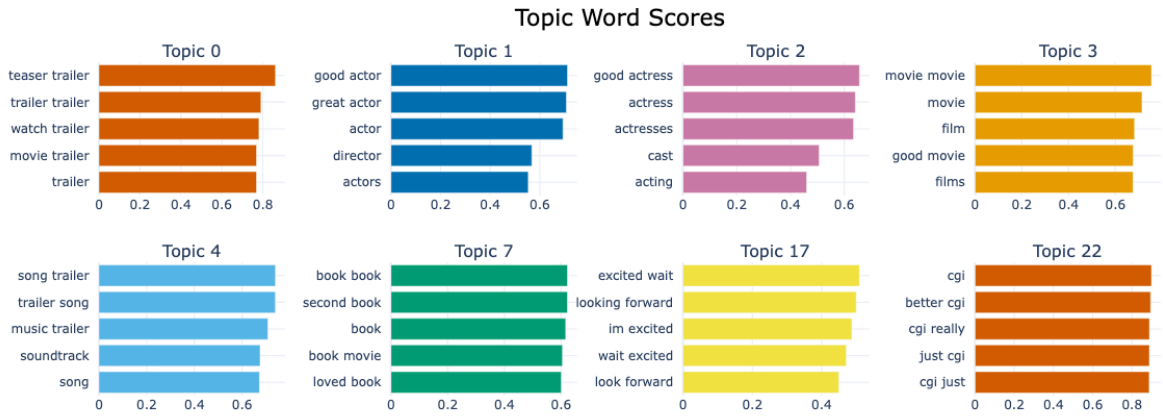
Overall, we were happy with the results of the model. While there is certainly more fine-tuning that could be done, the results were helpful for the model. The model itself was able to achieve a silhouette score of 0.56 which indicates reasonable clustering. In addition, there were

[topic effects](/topic_coefficients.png)

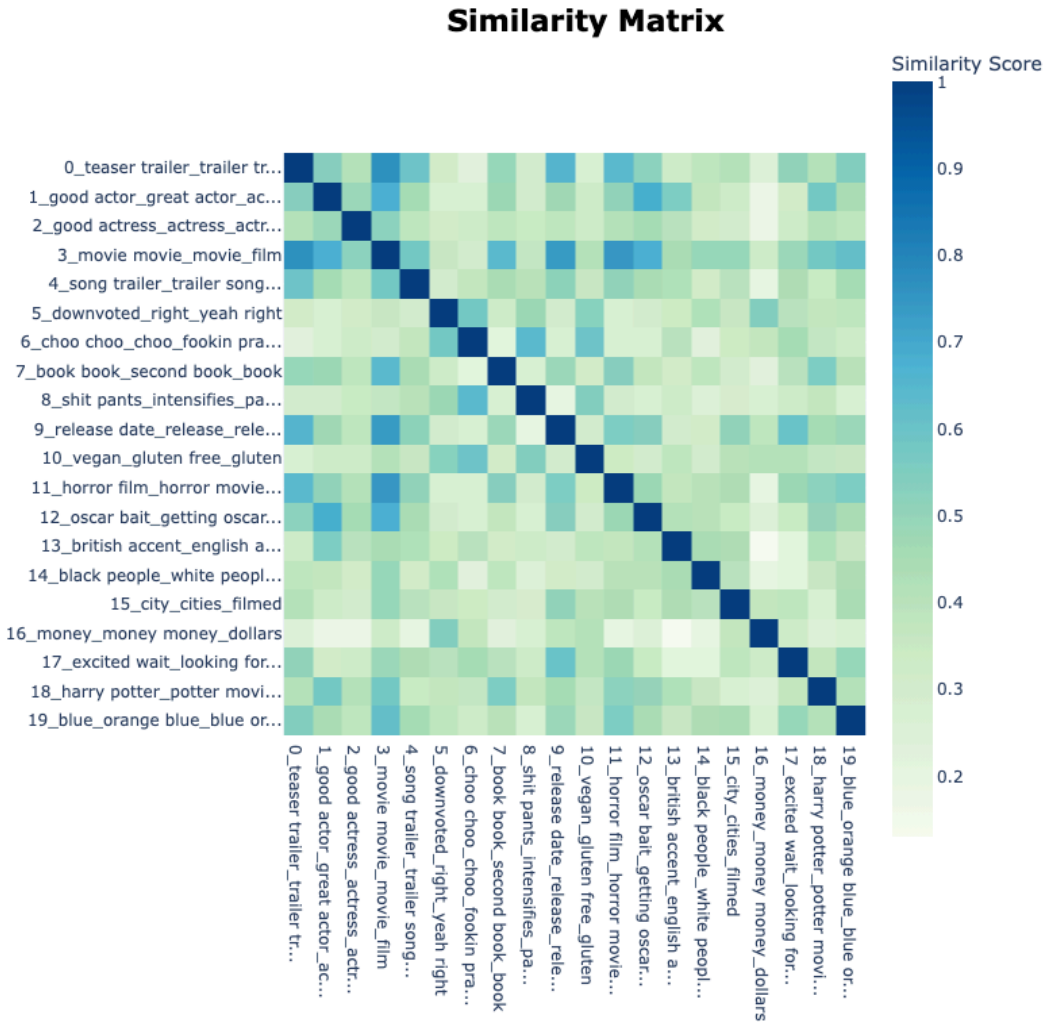


A visualization of topics. Note that we reduced the clustering to 2 dimensions for visualization purposes, so clusters may not be as on top of each other as they

appear here. Overall, we are happy with the separation.



A sampling of words in some of our clusters. As you can see some topics are similar (i.e. actor and actress), and some are quite general (i.e. movie, movies), but others could be indicative of topics that are important to audiences.



It appears none of the topics are overly similar to each other.

Multiple Linear Regression

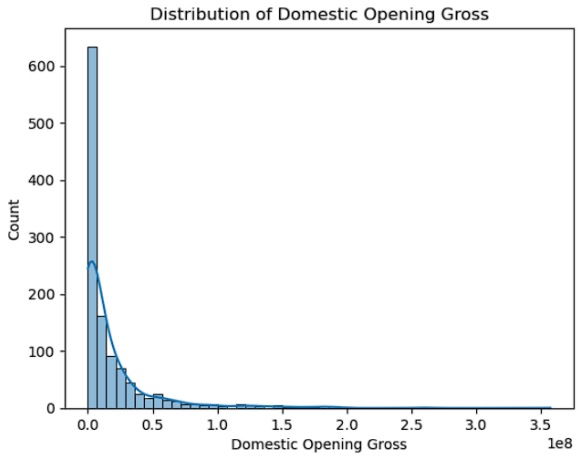
Based on model diagnostics and performance metrics we conclude that linear regression is not the best fit for the data as we identified complex non linear relationships between ‘domestic openings’ and input features. Exploring polynomial interactions and other transformations improved some of our model metrics but we couldn’t fix the heteroscedasticity issue using linear regression.

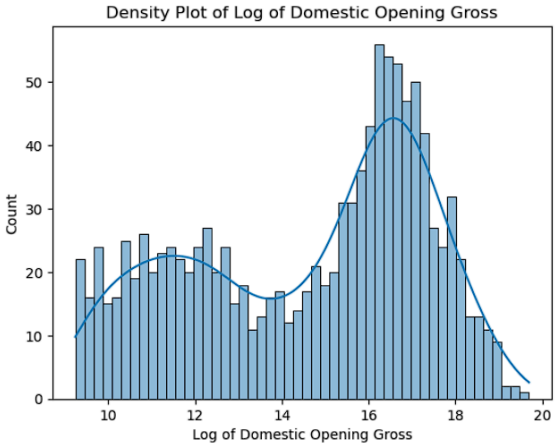
[MLR_results_table](/MLR_results_table.png)

The results indicate that our predictions have high margin of error

Elastic Net Regularization Model

During EDA, we observed that the distribution of domestic_opening was skewed, so we also explored fitting several models using feature engineering to log transform the target variable. Below are the charts showing the distribution of domestic opening before and after the transformation.





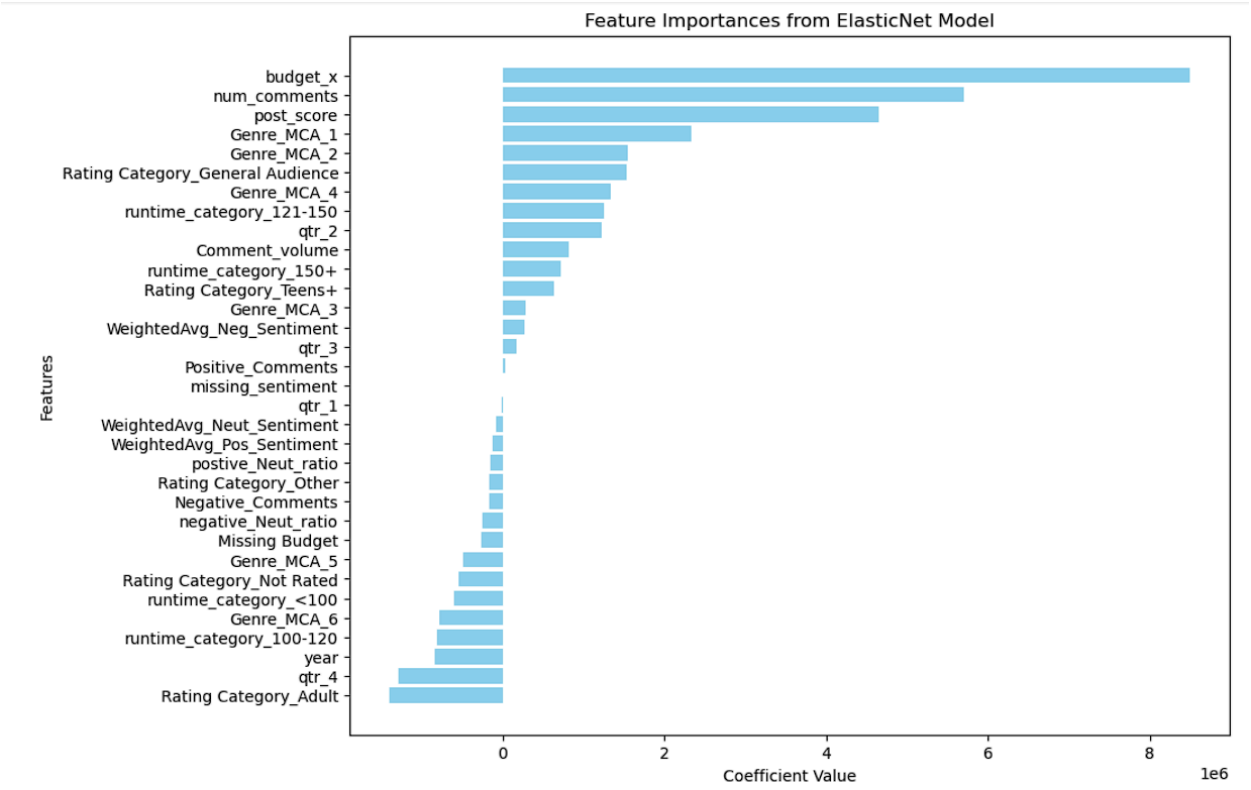
The transformed variable has a much tighter distribution than the untransformed variable. However, it appears that the distribution is bimodal, which could negatively impact the model. In addition to the log transformation, we also decided to train several models using a polynomial degree of two so that we could explore any possible interactions between our variables.

Starting from step 11 of the MLR preprocessing, we fit four elastic net (EN) models using the sklearn ElasticNet regressor. Using the same method as the tree-based models, we trained each model on 80% of the full dataset. We then tested the trained models on the remaining 20% and compared the RMSE and R^2 values between the training and testing sets (reported in the table below).

Model	Train R^2	Test R^2	Train RMSE	Test RMSE
Polynomial degree 1	0.6276	0.5250	$ 18,792,101.07	$ 27,141,985.69
Polynomial degree 2	0.7600	0.5644	$ 15,085,103.46	$ 25,992,206.15

Log transformed target variable and degree 1	0.1025	0.1680	$ 29,174,314.11	$ 35,921,981.02
Log transformed target variable and degree 2	0.0282	0.0103	$ 13,122,4012.16	$ 30,357,385.08
Polynomial degree 2 with feature selection	0.6907	0.5951	$ 17,124,420.99	$ 25,056,635.46

The results indicated that our models were having some problem with overfitting, particularly the models having polynomial degree two. The log transformed models were significantly underperforming the standard models, so we decided to further explore the second model (untransformed target with degree two) using feature selection to potentially reduce the effects of overfitting. Below is a chart displaying the importance of each feature, which indicates that a selection of important features should capture most of the same information as the full model. To select the most important features we began by calculating the permutation importance of each feature then creating a subset of data containing only the features with a permutation score above 0.005. We then fit another Elastic Net model using this subset. The evaluation results for this model are shown in the last row above. The feature selection does appear to have reduced the overfitting somewhat, with the R^2 falling to 0.69 and the RMSE increasing slightly. The model utilizing feature selection slightly outperformed the model including all features, with a slight increase in R^2 and decrease in RMSE.



Random Forest

Based on the combination of factors described above, we determined that the random forest model trained and fit without removing any data points or features is the most appropriate model. The results for that model are below - the full table of RF and XGB performance can be found in the appendix.
[Xfull](/xfull.png)
The best parameters reported for this model are as follows:

- Number of trees - 500
- Maximum depth of each tree - 10
- Minimum number of samples required to split an internal node - 2
- Minimum number of samples for a leaf node - 4

Once we had our results, we looked deeper into the model by examining important features, residuals, as well as what types of movies for which the model was a good or bad predictor. First, we examined feature importance, calculated using permutation importance with 30 shuffling trials. We also scaled the importances to more easily compare their effects. The results are displayed below.

[Permutation_Importances_RF](/importance.png)

Clearly, budget was the far more important feature for predicting opening weekend performance, which makes logical sense. After all, if a movie's budget didn't correlate with its success, movie studios would have much smaller budgets. After budget, the relative importance drops dramatically. It is interesting to note that while the baseline models with default parameters ranked number of positive comments as the second most important feature, the hypertuned model found number of negative comments to be more important to predicting opening weekend box office performance. The third most important feature is the Genre_MCA_4 feature created in our preprocessing phase. Defining exactly how that genre correlates to more traditional conceptions of movie genres is difficult, but the 10 movies with the best opening weekend performance of that genre indicates that it is some combination of films focusing on music, sequels, and comedy.

[Genre_4](/genre_4.png)

Beyond the 3rd most important feature, the importance scores continue to shrink to the point that we didn't find analysis of those features meaningful. Next, we looked at the plot of residuals for our model, as well as the top 5 movies in terms of underprediction, overprediction, and smallest absolute error.

[RF_Resids](/rf_resids.png)

5 Largest Overestimations (Dollars)

[Overs](/overestimations.png)

5 Largest Underestimations (Dollars)

[Unders](/unders.png)

5 Most Accurate Estimations (Percent)

[Accurate](/accurate.png)

Note that the over and underestimation graphs are in absolute dollar amounts while the most accurate estimations are in percents. This is because plotting the missed estimations as a

percentage showed only movies with exceedingly small domestic openings, and the opposite occurred for plotting the accurate estimations.

In examining the residuals plot, it appears as though the models predictions are essentially being pulled down by the large amount of movies that perform poorly.

Conclusions and Next Steps

Based on our analysis in totality, it appears that cash is indeed king when it comes to which movies perform the best in their respective opening weekend. However, we were still able to craft a sentiment analysis that had some feature importance in both regression models and a random forest. Though it had limited predictive power when compared to budget, that also makes sense - commenters on the r/movies subreddit make up only a small (probably particularly passionate) subset of moviegoers. However, the limited predictive power of their comments does suggest that if there were a way to gather sentiment from other popular sites such as YouTube or Twitter, the predictive power might increase. Additionally, the model could be improved by adding some features that captured data to which we did not have access or that we judged too labor-intensive for this project. For instance, a categorical feature for whether a film was produced by a major studio such as Disney, whether the movie belonged to an already successful film series, popularity of actors in the movie or if a movie was based on a bestselling novel all might have an impact on performance. As it stands, we feel that we have made a reasonable first step at using internet sentiment to predict movie performance and are excited by future possibilities.

Appendix

RF and XGB Full Results

[Full](/full_results.png)S