Netflix Viewership Predictor (4741)

Andrea Siby as3246

Kayla Runkel kmr227

Adelaida Dominguez ad667

Introduction

In the fast-paced world of entertainment, accurately predicting the success of original content, especially on platforms like Netflix, is a pressing challenge due to the unpredictable nature of success in the entertainment industry. With substantial upfront investments required and a highly competitive market, understanding the trajectory of movies and TV shows is crucial for mitigating financial risks and maximizing returns. This project seeks to address this challenge by leveraging machine learning algorithms to predict the success of Netflix, using weekly viewership data (in hours) as a means of success.

The overarching question we wanted to answer was: what factors contribute to the success of movies and television shows on Netflix? Furthermore, could we develop an accurate predictive model using features related to the content itself (e.g. genre, runtime, type) as well as viewing data to forecast the likelihood of success for Netflix's content offerings?

To answer our question, we leveraged multiple datasets on Netflix's content and performance. The core dataset came from Netflix's weekly "Global List" of popular titles since January 2021, containing nearly 6,000 entries [1]. We supplemented this with data from Netflix's Engagement Report and Kaggle datasets providing details like global availability and genres [2][3][4].

Through data aggregation and engineering across these sources, we combined quantitative metrics and engagement metrics. Our final dataset contains the features such as runtime, weekly views, weeks in top charts, type of title, language, and global availability. In total, this dataset comprises nearly 2,000 titles for analysis in our model.

Our approach involves applying regression methods to identify relationships between these features and the success metric (viewership). While the available data sets offer valuable insights, we acknowledge potential limitations and challenges, such as the data being constrained to releases within the past 5 years, and the assumption that viewership equates to success for a show or movie. Incorporating financial data like budgets and revenue figures could further enhance the analysis, but finding this comprehensive data has been a challenge.

Through this project, we aim to develop an accurate predictive model that can guide Netflix's multibillion-dollar content investment decisions. With the insights from this project, Netflix can optimize their content strategy, better cater to audience demand, and maximize returns.

Methodology

Feature Engineering

In order to answer our research question and uncover any trends and patterns underneath Netflix's success, we utilized some feature engineering techniques to resolve any areas in our datasets before fitting different models, enabling us to uncover meaningful insights into the factors contributing to Netflix's success. With our collection of data relying on various sources, process aggregation led to some empty values and categorical values. Since our output space is real valued data, we had to employ various tools to ensure we would be able to do the analysis.

To begin, we utilized one-hot encoding to transform categorical values, such as title type (TV show or movie), language (English or non english), and global availability, into binary representations. We leveraged the functionality of the pd.get_dummies() function to make this conversion seamless.

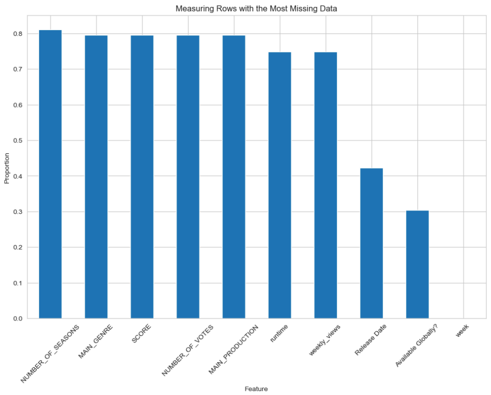

Figure 1: Plot showing the features with the most missing data

Addressing missing data was paramount, as Figure 1 illustrates a substantial proportion of absent values. For example, in instances where the number of seasons was missing for movie titles (predominant in our dataset), we resolved the issue by imputing a value of 0 and having a boolean variable indicating the type of title.

Of particular significance was the consideration of title runtimes. To resolve this, we decided to use imputation methods. By imputing missing runtimes with the mean value specific to the content type (TV shows or films), we ensured the derived averages were representative approximations of the actual values, accounting for the inherent differences between these content categories.

For the remaining features with missing values, we employed a matrix completion process. This technique was necessary to enable us to utilize the features we deemed important in our analysis. Even though a significant portion of our data was missing from some columns, we recognized that matrix completion is still effective in approximating the true values of the regime. We utilized Principal Component Analysis via SVD decomposition to complete the matrix, employing three principal components. We chose to use SVD to reconstruct the data seeing as the only missing data that we had was real value data (categorical data was not missing). This choice was critical as it allowed us to leverage the information in our data while addressing the issue of missing values. By employing PCA, we were able to capture the most significant patterns and relationships in our data, even with missing values.

Model Exploration

Looking at the features, we suspected that linear regression would be a good model for this problem. However, we realized that certain features, “weekly views” and “weekly hours viewed” for example, had values much greater than those of other features. Hence, obtaining a mean square error (MSE) using non standardized values might skew the results obtained. Hence, before training any model, we proceeded to standardize our feature values using StandardScaler. We then used our data to get a 80:20, training-testing split.

Linear Regression

With us having obtained the test and training datasets, we proceeded to train a simple linear regression model, using the LinearRegression library in sklearn, on the training dataset. We utilized the training data to find the learning accuracy of the model. Figure 2 shows a scatter plot comparing the predicted values to actual views for the Netflix content while Figure 3 shows the residual graph showing $||y - \hat{y}||$ where $\hat{y}$ represents the predicted number of views and y represents the actual number of views.

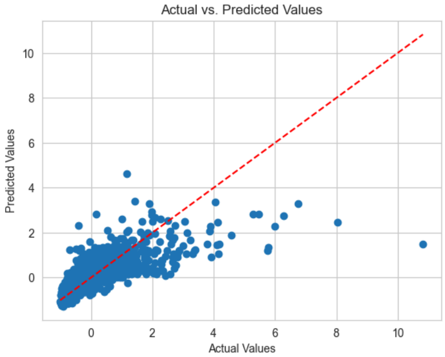

Figure 2: Scatter plot comparing the predicted number of view compared to the actual number of views on learned data by a simple linear regression model

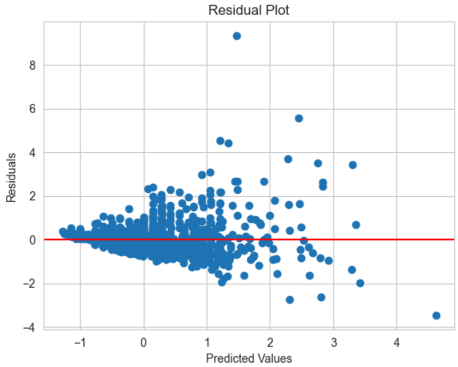

Figure 3: Residual plot of regression model on learned data

We obtained a cross validation score of 0.56 accuracy with a standard deviation of 0.07. The R^2 score of the learned values was 0.56 and the MSE was 0.45, as shown in Table 1. We then proceeded to find the scatter plot comparing the predicted number of views compared to the actual number of views on the test data as well the test data's residual plot, as shown in figure 4 and 5 respectively.

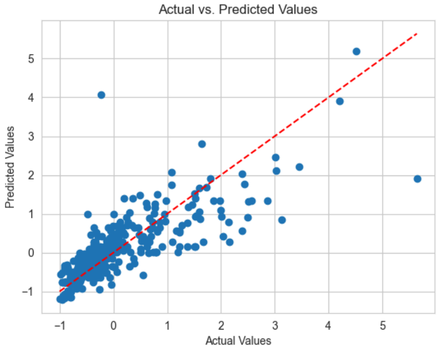

Figure 4: Scatter plot comparing the predicted number of view compared to the actual number of views on test data by a simple linear regression model

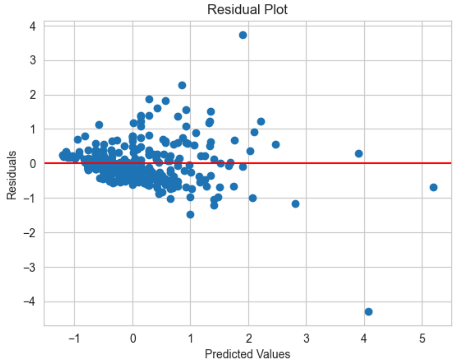

Figure 5: Residual plot of regression model on test data

The R^2 score and MSE on test data was 0.63 and 0.32 respectively.

We will now compare the scores between the training and test data shown in Table 1. We know that the closer to 1 the R^2 score is, the better the fit of the data to the curve plotted. With the R^2 score higher for the test data compared to the training data, we can conclude that the linear model is indeed performing better on the test data than the training data. Hence, it might be possible that the model is not too overfitted to the training data. Similarly, we know that the closer the MSE to 0, the better the fit of the model to the data. Hence, since the MSE of the test data is lower than the training data, we can say that the model performed better on the test data, corroborating the results obtained through the R^2 score. The better performance of the linear regression model is why the scatter plot shown in Figure 4 deviates less from the actual y values than the plot in Figure 2, observed using the red dotted line on both figures. Similarly, the lower mean square errors of the test data is why the residual graphs of the test data, shown by 5, is closer to the base-line along the x-axis when compared with Figure 3.

Now, we wanted to understand the impact generalization would have on both training and test datasets. We hence performed both ridge and lasso regression on the dataset for multiple alpha values. This enabled us to identify the best regression model and the best performing alpha value, as shown in Table 1.

Alpha Value	**Ridge Regression MSE**	**Lasso Regression MSE**
0	0.3296427853141802	0.3296427853141805
0.0001	0.3296427853428225	0.329633023945662
0.001	0.32964278560071514	0.32957204102169235
0.003	0.32964278617453713	0.32949833457868405
0.0035	0.3296427863181492	0.3294927453482972
0.004	0.3296427864618241	0.32949229107005623
0.005	0.3296427867493619	0.32950678737001265
0.1	0.3296428152083248	0.3588413308952503

Table 1: Comparison of MSE values of ridge and lasso linear regression on training and test data for certain alpha values

The lowest MSE of 0.3294 (to 4 d.p) is obtained with lasso regression and an alpha parameter of 0.004. When we compare this with the MSE of the simple linear regression, we recognize that the lasso regression does better by 0.046%. Upon conducting the cross validation test for lasso regression with the optimal alpha value of 0.004, we obtain a cross validation score of 0.5518792875878954.

Polynomial Regression

Next, we analyzed a polynomial model to ensure we captured the complexity of the relationship between the features in our dataset. We utilized the PolynomialFeatures function from sklearn to fit polynomial regression to the training dataset. The polynomial model was trained at degree 1 (linear) to degree 7. The model yielded the best training and test MSE of 0.4502 and 0.3296, respectively, at degree 1 which proved that the linear model is the better fit for our data compared to the polynomial model at a degree greater than 1. However, at degree 7, the train MSE for the polynomial model is 0.1244. Since higher order polynomial models have high variance, we must consider the training set MSE where in this case, the test MSE is 4.436e10. The low training set error and high testing set error displays that a polynomial model at degree 7 overfits our data. A similar disparity between training set MSE and test set MSE can be found for all polynomial models we analyzed. This extreme variance can also be concluded from the 10-fold cross-validation score of -1.906e20.

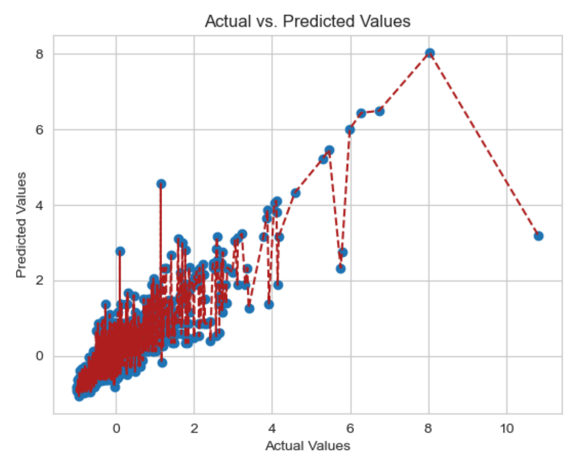

Figure 6: Scatter plot comparing the predicted views to the actual views using a linear regression model at degree 4.

Decision Trees

In addition to linear and polynomial fits for the data, we explored tree-based models capable of capturing non-linear patterns and providing interpretable feature importances.

For decision tree regression, we utilized the DecisionTreeRegressor function from sklearn. To mitigate overfitting, we tuned the maximum depth hyperparameter through grid search, ultimately setting it to 10. This decision tree model yielded a training MSE of 0.1186 and test MSE of 0.5486, with a 10-fold classification score of 0.50414. This indicates that the model, on average, captures around 50.414% of the variance in the data, demonstrating its capability in generalizing to unseen samples. While these results demonstrate the model's ability to generalize to unseen data, the relatively high error suggests room for improvement.

To enhance the predictive performance, we employed gradient boosting, an ensemble technique that combines multiple weak learners into a strong predictive model. We chose to utilize this ensemble technique as a tool to help the bias in our model. Using sklearn's Gradient Boosting Regressor, we optimized hyperparameters such as learning rate (0.1), maximum depth (3), and the number of estimators (50) through grid search. The boosted tree model exhibited a train MSE of 0.21102 and test MSE of 0.31706, reflecting the bias-variance tradeoff. Although the training error increased, indicative of higher variance, the test error decreased compared to the single decision tree, suggesting a reduction in bias.

Notably, the boosted tree model achieved a more promising 10-fold cross-validation score of 0.6347, indicating that it captures, on average, 63.74% of the variance in the data. This significant improvement over the single decision tree model demonstrates the boosted regression superior ability to generalize and learn complex patterns in the dataset.

Results

Feature Importance

To gain insights into the key drivers of success for Netflix content, we analyzed future importance from the decision tree and linear regression models built on our dataset. The decision tree model provides an interpretable structure that can highlight which features are most deterministic in predicting high viewership levels. By examining the feature importances derived from the decision tree, we identified the top factors that the model used in making accurate predictions.

Similarly, for the linear model, we evaluated the coefficient magnitudes and statistical significance to understand which continuous and categorical values exhibited the strongest relationship with the viewership metric. Interpreting these feature importances from both tree-based and linear models allow us to decipher the influential features that should be prioritized when considering potential content investments.

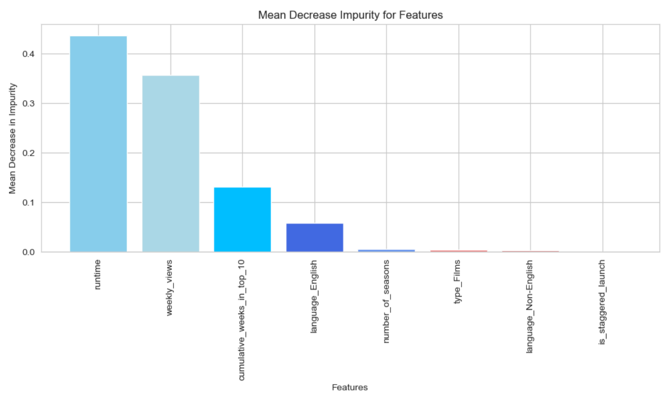

Figure 7: Feature importance of Boosted Tree Model

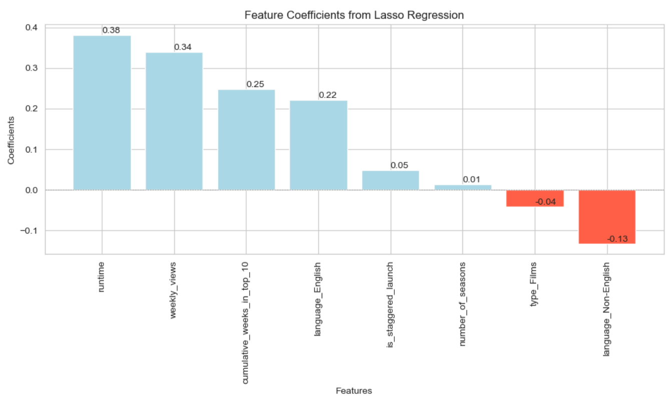

Figure 8: Parameter coefficients of Lasso Regression Model

Based on the Mean Decrease Impurity for features, we can identify that the most important feature by far is runtime (0.437), indicating that the length of a movie or TV show episode is a major determinant of viewership success on Netflix. Longer runtimes seem to be favored by the model for predicting high viewership. This is followed by weekly_views, which makes intuitive sense and currently weekly viewership would be highly predictive of overall success and popularity of a title. Interestingly, cumulative_weeks_on_top (0.132) also carries substantial importance suggesting that longevity and sustained presence in Netflix's top 10 charts is a key indicator used by the tree model. The model places little importance on binary features, implying these factors have minimal impact in predicting viewership according to the decision tree.

Similar to the insights from the decision trees, the features, runtime, weekly_views and cumulative_weeks_in_top, have the largest positive coefficients in the linear model indicating that they are strong positive predictors of viewership success. Notably, language_English (0.226) has a much higher positive weight compared to its low importance in the decision tree model. language_Non-English (-0.132) has a negative coefficient, suggesting non-English language content tends to have lower viewership.

Overall, the top predictive variables are consistent, but there are some deviations in how much relative importance each model assigns to the language and release strategy features. The decision tree appears to be more agnostic to these factors.

Comparing Models

Through our analysis, we gained valuable insights into the predictive capabilities of different machine learning models and key drivers influencing Netflix viewership success. While each technique had strengths and weaknesses, the gradient boosting tree model emerged as the best fitting model, demonstrating strong generalization performance and capturing complex patterns in the data.

Our resulting models had the following results. The linear regression model provided a decent baseline and was a useful starting point for our analysis. Using regularization, we were able to identify Lasso Linear Regression as having the best output, which indicates that our data may be sparse. It had a decent cross-validation score, however, the residual plots revealed its limitation in fully capturing the relationship in the data. This aligns with our initial hypothesis that linear models may oversimplify the problem. Nonetheless, linear regression offered interpretable parameter coefficients, highlighting runtime, weekly views and language as influential factors. Polynomial regression models of higher degrees exhibited a concerning tendency to overfit the training data severely. The extreme variance and poor generalization, evident from the disparity between training and test errors, rendered these models unreliable for our case. This reinforces the notion that increased model complexity does not necessarily translate to better performance, especially when dealing with limited and noisy data. The decision tree model, with tuned hyperparameters, demonstrated moderate predictive capabilities, achieving a cross validation score of 0.50. Notably, the feature importance metrics from the tree aligned with the linear model's insights, further solidifying the dominance of runtime and viewership-based features. However, the relatively high MSE indicated room for improvement.

The gradient boosted tree ensemble emerged as the clear winner, leveraging the strengths of multiple weak learnings to create a more powerful predictive model. With a cross-validation score of 0.64 and lowest MSE at 0.32, this ensemble approach exhibited superior generalization performance compared to all other techniques. While the training MSE did increase, indicating a higher variance, this is an expected trade-off when using ensemble methods. The increase in training error was relatively modest while the reduction in test error was more substantial compared to the other models. This suggests that the ensemble effectively reduced the bias of this model. As we know, the goal of a predictive modeling task is to achieve high accuracy on unseen data, not just the training set. The cross-validation score and low test MSE demonstrate that the boosted tree ensemble strikes a balance between bias and variance, learning the underlying patterns in the data while avoiding overfitting. By allowing the trees to learn iteratively and correcting errors from previous iterations, boosting effectively captures complex, non-linear relationships in the data that are challenging for individual models to represent accurately.

Model	**Train MSE**	**Test MSE**	**Cross Validation (k=10)**
Linear Regression	0.37496	0.32964	0.56059
Linear Regression with lasso regression	0.45028	0.32949	0.55187
Polynomial Regression	0.12436	4435902445228.489	-1.906470
Decision Tree	**0.11865**	0.54866	0.50414
Boosted Decision Tree	0.21102	**0.31706**	**0.637402**

Table 2: Comparison of trained models train and test MSE and cross validation score

While no model is perfect, using the current features at hand, the strong cross-validation score and low test error of the boosted tree ensemble compared to the models instill confidence in its ability to guide decision-making at Netflix.

Applications

As we manipulated our data and trained the model, we kept the ultimate goal in mind. What insights can we provide to Netflix (investors, producers, etc.) to best optimize their success? Based on the feature importance of the boosted tree model and the parameter coefficients of the ridge regression model, the top four features were consistent: runtime, weekly views, the number of weeks in the top ten, and the language of the show/movie. It's interesting to note that essentially only one of those categories is completely decided by the content creators; only the language of the media can be decided by the writers and producers. The runtime, weekly views, and number of weeks in the top ten are affected by the choices and talents of the production team, but it ultimately comes down to the viewers and their evaluation of the show or movie.

Ethical and Fairness Considerations

There are a few potentials for creating a Weapon of Math Destruction that is introduced from our dataset and the conclusions. First, it is important to remember that our dataset consists of data starting in 2021 and ending mid 2024. As a result, the model we trained already reflects biases in production and viewership data; it is likely that past societal and cultural inequalities have caused specific content to be favored. This could lead to a feedback loop where the same type of media is constantly favored, while others are marginalized. In addition, due to the constraints of this project, we considered only weekly hours of viewership data as a means of success for Netflix shows and movies. This narrow definition of success overlooks the cultural or artistic innovation of some content that may not receive the standard of views to be considered "successful". This hence ties in with notions about fairness in our model. At the moment, our model indeed shows biases towards shows with English as the main language. If Netflix were to utilize the model, it would not be fair to non-English shows which may not have been given the same opportunity to succeed as without the presence of the model when language may not have had much implications on the notions of success. Additionally, we should realize that this model purely uses viewership data to assess the success of a show when in reality, other factors such as profitability of shows and societal impact, may also be as strong an indicator of success. Hence, defining success via viewership might put us at risk of having a microscoping view on the situation, in turn impeding diversity in perspectives as mentioned previously.

Conclusion

Our analysis demonstrates the effectiveness of machine learning techniques, particularly gradient boosted trees, in predicting Netflix viewership success. However, we must exercise caution in blindly following the models recommendation as they may inadvertently reinforce existing biases and limit the diversity of content offered.

As noted previously, our model was more significantly impacted by features that were, more or less, in the control of the public. Runtime, weekly views and weeks in the top charts were among the most influential factors, which are ultimately determined by audience perception. While this provides valuable signals, it limits the direct insights for guiding production decisions.

As an extension to this project, we would analyze the more granular features under the control of creators that could yield actionable recommendations. For example, training models that incorporate features like genre, actors, and reputation will uncover some of the specificities of high viewership. It would be useful to Netflix content creators to understand the impact of all of the features in which they have control of, so they can tailor their media to the appropriate audience. Additionally, our current analysis solely utilized viewership metrics as a measure of success. However defining success through this narrow commercial lens risks overlooking content that pushes creative boundaries, represents marginalized voices, or prioritized artistic expression over mass appeal. Future work could explore incorporating multidimensional success criteria that could account for cultural impact, social relevance and critical acclaim alongside viewership data.

Ultimately, while data driven models offer valuable insights, Netflix should leverage these models judiciously, using them to identify audience preference while leaving ample room for other forms of success and feedback. With a holistic lens, the platform can truly maximize its potential as a force of artistic innovation and cultural enrichment.

References

[1]: https://www.netflix.com/tudum/top10/most-popular?week=2023-12-03
[2]:https://about.netflix.com/en/news/what-we-watched-a-netflix-engagement-report
[3]:https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies?select=titles.csv
[4]:https://www.kaggle.com/datasets/shivamb/netflix-shows

GitHub Link:

https://github.com/adelaidad/ORIE4741FinalProject.git

Contributions:

Andrea Siby:

- Chipped in ideation and data collection
- Created a web-scraper using selenium to obtain more data from the internet (we did not use this in the end due to time constraints with respect to data collection but the code is present in directory labeled "scraping")
- Helped with feature engineering slightly
- Helped write part of the introduction
- Linear Regression (code and report)
- Regularization (code and report)
- Fairness considerations

Kayla Runkel:

- Chipped in ideation and data collection
- Feature engineering, specifically the genre data (which we did not end up using due to complications with web-scraping)
- Polynomial Regression (code and report)
- Introduction, application and ethical considerations

Adelaida Dominguez:

- Chipped in ideation and data collection
- Feature Engineering, specifically matrix completion using PCA to resolve the issue of missing values (code and report)
- Decision Tree and Boosted Decision Tree (code and report)
- Dataset explanation, results and conclusion (report)