CSCI 6443

Data Mining

Project Paper

Title: **Predicting TV Series Ratings Using Ensemble Learning**

Submitted by:

Harshita Chadha (GWID – G40737617)

Abdul Irfan Mohammed (GWID - G33655938)

Abstract

TV series are arguably one of the most lucrative sectors within the broader entertainment media industry. In the past decade, due to the advent of online streaming services, the number of such series available for audience consumption has been exponentially increasing. Although, it is a consumer-based industry, economic profitability lies at the center of this sector - for media houses expansion of revenue means more resources for improved production quality. Thus, optimizing the performance of a series and by extension the incidental monetary gain is a prime consideration. While various non-computational techniques have been used in the past to estimate series performance, these suffer from various bias problems thereby creating a need for computational/quantitative models to achieve the same functionality. Various statistical and predictive techniques are thus increasingly being applied in the entertainment industry to predict success to better orient the high investment needed for their preparation. This article introduces an approach that utilizes text analytics, natural language processing, and data mining techniques to extract quantitative features from the script corpus of the popular television series "Friends." These features are then fed into ensemble learning models, including stacked architectures, Gradient Boost, AdaBoost, among others, to predict IMDb ratings for the scripts. Positioned at the intersection of the technological evolution of the media production industry, this project combines diverse data sources and employs advanced data mining methods to provide a robust and objective tool for estimating episode reception amongst consumers.

Table of contents

1. Introduction

Ever since its invention in the mid-20th century, television has transformed from a rarity to an integral part of the modern household as well as that of global culture, acting as a soft power and profoundly impacting societies by shaping collective popular narratives [1,2]. While the television media industry is a source of different types of information, arguably its most popular niche is the sitcom industry. Born in the post-World War II era, this sub domain of the industry has transformed into a multi-billion-dollar powerhouse [3], providing employment for millions worldwide [4]. As a people driven industry, its success hinges on its capability to capture the attention and engagement of audiences. With millions invested in content creation, it becomes imperative to understand how well the produced shows resonate with viewers [5]. This emphasis on audience reception is not just a facet but the very heart of the industry's operations.

Unfortunately, despite precautions and human analysis-based fidelity checks, not every project achieves the desired success. In fact, approximately 92% of all produced shows in a year fail to perform to expectations leading to varied degree of financial loss [6,7]. Year after year, a substantial number of television series face the harsh reality of low ratings, leading to project cancellations.

Nevertheless, despite the setbacks, the industry persists in its efforts, investing increasing amounts of money to create compelling content, shelling huge amounts of money to ensure utmost quality. For instance, the final season of "Game of Thrones" reportedly had a budget of around $90 million, emphasizing the significant financial stakes involved [8]. This commitment to spending is reflective of the industry's belief in the potential returns, as successful series can become lucrative assets, generating revenue through syndication, streaming rights, and merchandise sales.

Inarguably, the scripts are the very heart of series, providing the narrative foundation and creative essence that captivates audiences. However, despite understanding tis importance and employing human script analysts to gauge audience reception, the inherent subjectivity of this approach poses unique challenges. Human analysis is limited by biases and may not capture the diverse nuances of audience preferences. Thus, as the industry grows, embracing technological advancements becomes imperative. This is where statistical modeling and techniques like data mining take the center stage. Statistical models offer a more objective and data-driven approach, leveraging the vast amounts of information available to predict the potential success of a TV series. This shift towards quantitative analysis is evident in the industry's adoption of predictive modeling to inform decision-making processes. An example of this is the recent development of deep learning models at 20th Century Fox studios that utilize granular customer data and movie scripts to identify patterns in audience preferences [9]. Additionally, they have partnered with Google to create Merlin Video, a computer vision tool that analyzes movie trailers to predict audience response reinforcing the practical importance of using statistical means to quantify media reception [10].

The project presented in the current article is one such computational methodology that leverages various text analytics and data mining techniques to assist in prediction of user reception of a television series based on various factors. To test the hypothesis, a case study was conducted on the script corpus of the television series “Friends”. The presented project sits at the intersection of the technological evolution of the media production industry and by amalgamating diverse data

sources and employing advanced feature engineering methods, aims to provide a robust and objective tool for understanding the factors influencing audience reception.

In the sections that follow, we present the past and current industry landscape, outline the data collection and feature engineering methodology, present the created data mining regression models and evaluate their performance on the engineered dataset. The results obtained showcase a robust and objective tool for understanding the factors influencing audience reception and reinforce the potential of data-driven methodologies to redefine how television series are conceptualized and produced in the 21st century.

2. Related Work

The concept of predicting the success of a TV show based on its script involves the intersection of various study areas, including natural language understanding, sentiment analysis, machine learning, and the broader examination of media such as TV and movies in the cultural context. This article introduces a relatively novel approach to predict ratings from TV series' scripts using data mining. While this specific methodology is not extensively explored, with only one very similar approach existing, it is important to note that the broader field is not entirely an uncharted territory. There exist plenty of previous studies that have delved into predictive and statistical analysis in an effort to gain valuable insights into performance within the realm of TV and media.

The work of Jehoshua Eliashberg [11] of the Wharton School of Business at the University of Pennsylvania has since the early 1990's focused on predicting reception of movies amongst audiences and identifying key success factors. In a study published in the journal of the Institute for Operations Research and the Management Sciences in the year 1994 Eliashberg et al [12], proposed a conceptual framework for understanding the enjoyment of a hedonic experience, emphasizing the dynamic interaction between stable individual difference factors, temporary moods, and the emotional content of the movie going experience. The authors took pre-movie measurements to predict individual differences in the post-movie enjoyment and conduct an empirical test of the proposed movie enjoyment model (ENJMOD) with encouraging results at the individual and segment level. Following this, Eliashberg and his group went on to propose MOVIEMOD [13] - a prerelease market evaluation model for the motion picture industry. The model was designed to forecast box-office sales and support marketing decisions for new movies before release without requiring actual sales data. It is based on a behavioral representation of the consumer adoption process for movies, using an interactive Markov chain model to account for word-of-mouth interactions.

In their article, Jayashree et al [14] attempt to predict the MPAA (Movie Picture Association) certification ratings of movies based on their scripts. They treat movie scripts as text corpus and train a bidirectional LSTM (Long Short Term Memory) model with attention to classify the scripts into 5 rating categories. Their model's final reported accuracy was 56%. Along similar lines, Shafei et al [15] proposed an RNN-based model with attention that jointly models the genre and emotions in the script to predict the MPAA rating and achieved an 81% weighted F1-score for the classification task. The authors also analyzed the impact of genre, emotion, and similar movies on model performance in their study. Additionally, the article discusses the effect of emotion and

attention mechanisms on the model, as well as an analysis of the impact of bad words on the MPAA rating of movies.

Further, a group of researchers at the Signal Analysis and Interpretation Lab (SAIL) at USC Viterbi have recently worked on creating a neural network based artificial intelligence tool that utilizes linguistic cues to classify words and phrases that constitute the script into three categories: violence, drug abuse and sexual content [16-19]. Their group identified strong correlation between the levels of these three categories in movie scripts and the user reception of the movies as indicated by the ratings. Through this tool, it becomes possible to assess the content of a movie swiftly by analyzing the script, even before any scenes are filmed. This method offers movie executives the opportunity to pre-determine and tailor the movie rating according to their preferences by making necessary adjustments to the script prior to the commencement of shooting.

Frangidis et al [20], went a step ahead and looked not only at the numerical ratings, but extracted sentiment out of user reviews. Their study focused on utilizing techniques such as VADAR, multinomial naive bayes, etc. to study the correlation between the unison or dissonance of the sentiment of the reviews and scripts of movies themselves. There also have been multiple non-research-based attempts that utilize just IMDb metadata to predict movie ratings [21-23]. Further, Hunter et al [24], attempted to develop a statistical model to predict the audience size of new TV series and tests it on a sample of 116 hour-long scripted series from major US television networks during the 2009-2014 seasons. They created a text network out of pilot episode scripts to test the correlation between concept of originality and viewership.

Another interesting study conducted into predicting performance of a TV series through textual and network analysis was undertaken by Colladon et al [25]. They conducted a case study into the popular American sitcom “Big Bang Theory” and not unlike the present study, used text analytics to engineer features such as the number of words and character dialog interactions from the script. Unlike the present study though, their goal was to identify and map the correlation between the various engineered attributes and the episode ratings. They utilized correlation and regression analysis to identify the most relevant predictors of popularity and appreciation for the episodes. Going one step further, Benjamin S created the project titled Simpsonian [26] that not only engineered features from the scripts of the animate sitcom, the Simpsons but also used various predictive models to obtain the IMDb ratings for each script. The Simpsonian too utilizes a stacked model to make rating predictions. However, the television show studied, and the engineered features differ from the proposed approach in this article. The final RMSE score of their stacked model was 0.351(approx.) and the results were used to build a custom recommendation system.

As can be seen from the studies listed above, new statistical methods for predicting ratings and user reception of movies and television series are gaining momentum. The arena, however, is far from unfamiliar. Human script analysts have, for years, scrutinized the factors influencing the success of movies. While scripts remain a crucial element, additional factors like marketing strategies, release timelines, and audience sentiment can significantly contribute [27]. Despite this complexity, our project involves mimicking these influential factors by engineering metadata and manipulating scripts to construct a statistical model. In the subsequent sections, we outline our methodology and provide a comprehensive examination of the results and performance of the employed models.

3. Proposed Approach

In this project, an attempt was made to predict the ratings of episodes of the popular TV series Friends, given engineered features from its script. The Figure X below depicts a diagram illustrating the systematic approach to achieve this. The first step of the proposed approach involved obtaining data from suitable sources and engineering relevant features from it for prediction. To achieve this episode metadata was collected using APIs such as the TMDb API [28], which provides extensive information such as episode ratings, director and actor names, and summaries. Parallelly, scripts of episodes were scraped from the web, leveraging resources like the "Lives in a Box - Crazy for Friends" blog [29]. This data scraping step involved the use of Python libraries such as Requests and Beautiful Soup [30,31] to navigate and extract HTML content.

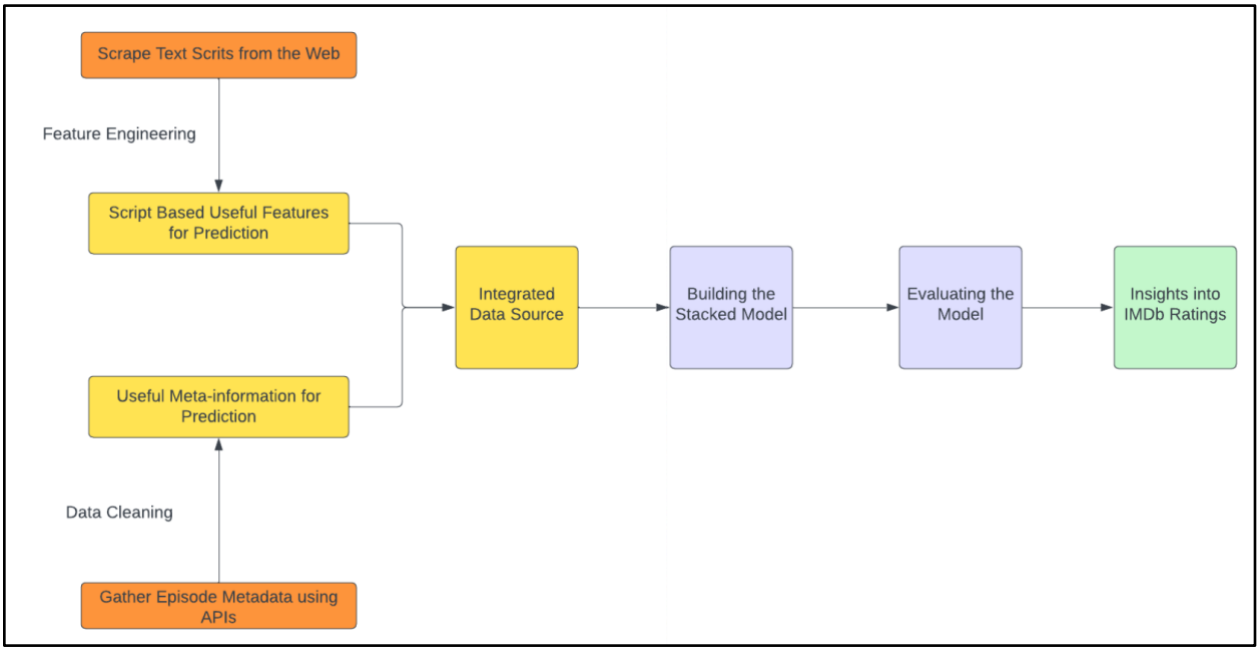

Figure 1: Overview of the Approach Used to Address the Ratings Prediction Problem

Next, various data mining, natural language processing, and text analytics techniques were used to engineer several relevant features from the script corpus. For script-based data, feature engineering involved analyzing the text to extract features such as the frequency of character interactions, sentiment analysis, sarcasm detection, and identifying the emotional archetypes of the scripts. Metadata features were also engineered to derive valuable predictors such as director popularity, writer influence, airing dates, etc. The script-based features and meta-information were then integrated into a consolidated dataset. This integrated data source became the foundation for the predictive model, providing a multifaceted view of the factors that may influence episode ratings.

Using the obtained integrated dataset, experimentation was conducted with various tree based regressor models to arrive at accurate ratings predictions. To this effect, five different models were worked with namely - Random Forest Regressor, AdaBoost, Gradient Boost, and Stacked model

with and without random parameter search. To compare and contrast the performance of these different models, they were evaluated using appropriate metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) to determine the superior choice from amongst the ones tested. As illustrated in the section 5 below, it was discovered that a stacked model with random hyperparameter search shows optimal output. The sections that follow outline the approach followed in detail and provide a comprehensive understanding of the methodology employed in predicting the ratings of episodes from the popular TV series Friends based on the approach outlined in this section.

4. Dataset and Feature Engineering

For the purpose of the present project, various feature engineering techniques were utilized in order to obtain the final attribute set that was used for the rating prediction task. There were two main sources used for the base information base creation, the first was the TMDb API [28] which is a rich source of metadata such as ratings, director and actor names, overview, etc. for movies and TV series. Using the TMDb API, the target variable or episode ratings for Friends and other episode metadata such as brief episode summary, episode director, episode writer, special guest appearances, etc. was obtained. Figure 2 below shows an initial snapshot of this metadata source (some rows omitted for clarity of representation).

id	air_date	episode_number	name	overview	runtime	season_number	vote_average	list_of_characters	average_character_popularity
85987	1994-09-22	1	Pilot	An introduction to the gang. After Rachel leav...	23	1	7.081	[Paul, Franny, Jasmine]	7.465333
86012	1994-09-29	2	The One with the Sonogram at the End	Ross's lesbian ex-wife, Carol, is pregnant wit...	23	1	7.333	[Carol Willick, Dr. Oberman, Robbie, Marsha, B...	10.371889
85981	1994-10-06	3	The One with the Thumb	Monica finds it difficult to break up with her...	23	1	8.130	[Lizzy, Paula, Alan, Gunther]	9.014250
85983	1994-10-13	4	The One with George Stephanopoulos	While the men attend a hockey game and end up ...	23	1	7.840	[Nurse Sizemore, Joanne, Leslie, Pizza Guy, Br...	6.773571
86034	1994-10-20	5	The One with the East German Laundry Detergent	While Chandler and Phoebe decide to break up w...	23	1	8.041	[Horrible Woman, Angela, Bob, Janice, Gunther]	8.251000

Figure 2: Metadata obtained using the TMDb API

The second major source of attributes was the actual episode scripts that were sourced from an online blog from the early 2000's called "Lives in a Box - Crazy for Friends" [29]. The scripts were housed as separate HTML pages with an index linking to these. Python Programming language and its accompanying web scraping packages such as Requests and Beautiful Soup were used to first scrape all episode HTML page links and then the script data itself. The HTML code of each page had to be carefully studied to understand which tags housed the relevant data to suitably extract it. After extraction, each script along with the title of the episode, season number and episode number were stored in the form of a data frame. Figure 3 below shows a snapshot of the initial scraped script data.

text script	Title	Episode	Season
[Scene: The Subway, Phoebe is singing for\ncha...	The Pilot-The Uncut Version	1.0	1.0
[Scene Central Perk, everyone's there.]\nMonic...	The One With The Sonogram At the End	2.0	1.0
[Scene: Chandler and Joey's, Chandler is helpi...	The One With The Thumb	3.0	1.0
[Scene: Central Perk, Ross and Monica are watc...	The One With George Stephanopoulos	4.0	1.0
[Scene: Central Perk, all six are there.]\nMon...	The One With The East German Laundry Detergant	5.0	1.0

Figure 3: Script data obtained through web scraping

Using the scripts data, as illustrated in the figure above, the main set of attributes used for ratings prediction task was obtained through various feature engineering techniques. First, the text scripts themselves were cleaned and the way in which information was structured within the scripts was studied. It was discovered that special characters and parentheses patterns were used to stre distinct information. For instance, all scene/ set change information was contained within square braces '[]'. These patterns were exploited, and regular expressions and other techniques were used to extract the most useful information.

Firstly, as stated above, regular expressions and pattern matching were used to obtain the list of all the speaking characters in each episode. Further, for each of these characters, the number of spoken words and consequently the total number of spoken words in a script were also obtained. Next, the same pattern-matching technique was also used to obtain the count of the total number of scene changes in an episode. To get the list of locations themselves, initially Named Entity Recognition or NER was used. For this, the Stanford NLP Group's pretrained model Stanford NER [32] was used. The Stanford NER is also known as CRF Classifier. The software provides a general implementation of (arbitrary order) linear chain Conditional Random Field (CRF) sequence models. Unfortunately, the NER technique using the above model or the packages from Python Programming Language's Spacy Module did not adapt very well for our dataset leading to at the very least about 48-50% null values for the location attribute. To resolve this, an alternative approach was identified.

Studying the production format of the television series Friends [33-35], it was discovered that most of the episodes were shot with a live audience in attendance, thus the number of sets used was few and limited to a set of mostly seven locations. Therefore, fuzzy string and pattern matching in the sequence information were used to obtain this information from the episode scripts. Using this technique only about 10% of all episodes had empty values for this attribute. This could be ascribed to "out of the norm shoots" such as Episodes 23 and 24 of Season 4 [36] which was based in the United Kingdom and did not include any of the usual sets. The absence itself could lend important information about the user popularity for the episodes, thus the parameter was retained.

Next, a parameter called the emotional archetype of the script was extracted. The main idea behind this attribute comes from the almost a century-old technique of graphing the plot lines for stories by the acclaimed American writer Kurt Vonnegut [37]. After being initially rejected as a thesis by the University of Chicago when it first debuted, his work recently gained recognition when it was picked up and implemented using computational methods by a group of University of Vermont researchers [38]. According to Vonnegut, stories can be categorized into one of the six main storytelling arcs including "Rags to Riches" (rise), "Riches to Rags" (fall), "Man in a Hole" (fall

then rise), “Icarus” (rise then fall), “Cinderella” (rise then fall then rise) and “Oedipus” (fall then rise then fall). For the present project, a seventh arch, uncharted was also added to identify those episodes for which, a predefined arch was not a fitting label.

In order, to map each script to an emotional archetype, a moving window approach of 200 words at once was used to estimate the sentiment of that part of the script. The sentiment analysis was done using a pre-trained BERT-based transformer model from huggingface [39]. This model returns after analysis, one of the following seven labels for a given text - [anger, disgust, fear, joy, neutral, sadness, surprise]. After a list of sentiments for each script was obtained, this script was mapped to each of the previously mentioned seven archetypes using a logical rule-based approach. For instance, if the first half of the list of detected sentiments was composed of positive emotions and the second half had mostly negative emotions, then the script was labeled as “Icarus” (rise then fall) and so on.

Since sarcasm was a major part of the humorous appeal of the TV series Friends, Sarcasm detection was also performed on the dialogues of the episode scripts and the resultant Sarcasm Index was used as an attribute for ratings prediction. To generate this attribute, a pre-trained text analytics sarcasm detection model was used as obtained from huggingface [40]. Depending on the number of words, each episode was divided into 10 parts, and the overall normalized confidence level of sarcasm was taken as the sarcasm index for the given script.

Building on the episodes scripts the project harnessed the Apriori algorithm [41] for analyzing character interactions within each "Friends" episode. We focused on extracting the top four frequent character interaction sets for each episode, a method that helped identify significant patterns in character dynamics. These interaction sets were believed to have a potential impact on the episode ratings.

Following the identification of these frequent interaction sets, the next step involved pinpointing the top 10 most repeated interaction sets across the series. This was crucial in understanding the recurring character dynamics pivotal in audience engagement. Subsequently, this qualitative data was transformed into a quantifiable format through one-hot encoding. This conversion into a binary vector format was essential to make the data compatible for use in machine learning models aimed at predicting episode ratings.

For each episode script, Text Blob [42] processed the text to evaluate the overall sentiment. This involved breaking down the script into sentences and words, assessing the sentiment of each component, and then aggregating these to form an overall sentiment score for the entire script. This score was used as a feature in predicting episode ratings.

The TMDb API, helped extract values such as the writer’s name and popularity, director’s name and popularity, date of airing etc. which were used to create attributes writer_popularity, director_popularity, holiday_episode (0/1 binary attribute) for each episode.

Total Spoken Words	number of scene changes	Sarcasm Index	vote_average	holiday_episode	director_popularity	writer_popularity	Joey_Rachel_Monica_Chandler_itemset
2290	13	0.666667	7.081	0	3.7445	4.7685	1
2520	11	0.714286	7.333	0	3.4485	4.7685	0
2336	14	0.625000	8.130	0	3.4485	1.7880	0
2721	15	0.333333	7.840	0	3.4485	1.3880	0
2765	16	0.625000	8.041	0	2.1950	1.0455	0

Figure 4: Final Consolidated Dataset

The figure above shows the final consolidated dataset containing the attributes engineered as explained in the paragraphs above. Some rows have been omitted for clarity.

```
['Total Spoken Words', 'number of scene changes', 'Sarcasm Index',
'vote_average', 'holiday_episode', 'director_popularity',
'writer_popularity', 'Joey_Rachel_Monica_Chandler_itemset',
'Rachel_Ross_Monica_Chandler_itemset',
'Joey_Phoebe_Monica_Chandler_itemset', 'Joey_Ross_Monica_Chandler_itemset',
'Rachel_Phoebe_Monica_Chandler_itemset',
'Joey_Rachel_Phoebe_Chandler_itemset', 'Joey_Rachel_Ross_Chandler_itemset',
'Joey_Ross_Phoebe_Chandler_itemset', 'Ross_Phoebe_Monica_Chandler_itemset',
'Joey_Rachel_Phoebe_Monica_itemset', 'sentiment', 'Ross_spoken_words',
'Rachel_spoken_words', 'Chandler_spoken_words', 'Monica_spoken_words',
'Phoebe_spoken_words', 'Joey_spoken_words', 'Central Perk_occurrences',
'Monica's Apartment_occurrences', 'Ross's Apartment_occurrences',
'Chandler's Apartment_occurrences', 'Ralph Lauren_occurrences',
'Bloomingdales_occurrences', 'Phoebe's Apartment_occurrences',
'Emotional_Archetype_Cinderella', 'Emotional_Archetype_Icarus',
'Emotional_Archetype_Man in a Hole', 'Emotional_Archetype_Oedipus',
'Emotional_Archetype_Rags to Riches', 'Emotional_Archetype_Uncharted']
```

Figure 5: Final Attribute Set

The figure above illustrates the final set of attributes used to train the rating prediction model. The categorical attributes have been one-hot encoded and the target variable, vote_average, containing the rating has been highlighted in yellow. In the section that follows, the models trained using the data and the consequent results are presented in detail.

5. Predicting Ratings Using Different Models

Predicting the ratings of scripts using the engineered attributes as explained in the previous section using various data mining techniques presented a unique challenge. The attributes were a combination of categorical and continuous values, limiting the range of models that could be used. Nevertheless, for the present study, five distinct models were used namely Gradient Boost, AdaBoost, Random Forest Regressor, Stacked/Ensemble model without Random Search, and Stacked/Ensemble model with Random Search. The performance of each of these models was evaluated using the RMSE and MSE values and suitably contrasted. The details of the constructed models along with the performance comparison are presented in the sections that follow.

5.1. Random Forest Regressor Model

The Random Forest Regressor Model was created using Python Programming Language's sklearn module. The Random Forest Regressor is an ensemble technique that combines the predictive power of several weak learners or trees and aggregates their predictions to arrive at the most accurate predictions possible [43]. It was selected as one of the potential models, here, because of its ability to handle both continuous and categorical variables effectively. An 80-20 train-test split was done on the episode script dataset for training and evaluation purposes. The model was composed of 100 weak learners/decision trees. After training, the mean squared error (MSE) value for the model was obtained to be 0.1161 (approx.) and the root mean squared error value was found to be 0.341 (approx.).

5.2. AdaBoost Model

AdaBoost is another class of ensemble learning technique that combines the power of several weak learners to create a highly accurate overall predictor model. However, unlike the decision tree regressor as in the previous case, the weak learners are trained in series where the weights of misclassifications of a previous weak learner is increased in the subsequent one to minimize the overall prediction error [44]. For the present application, an AdaBoost model was created using the AdaBoost Regressor class of the sklearn.ensemble module withing python programming language. The model was composed of 100 weak learners. After training, the mean squared error (MSE) value for the model was obtained to be 0.098 (approx.) and the root mean squared error value was found to be 0.314 (approx.).

5.3. Gradient Boost Model

Gradient Boost, just like AdaBoost is an ensemble method that combines the performance of multiple weak learners in series to create a strong overall predictor model. However, the way in which the model is built differs. Starting by taking the average of the values for a regression-based mode, the Gradient Boost algorithm then continually adds trees based on residuals or the difference between the observed and the predicted values [45]. At each stage, the tree is scaled using a learning rate specified in this case as 0.1. The training continues until the maximum number of trees are built or no performance improvement is observed across subsequent weak learners. In this case, the number was set as 100. After training, the mean squared error (MSE) value for the model was obtained to be 0.1082 (approx.) and the root mean squared error value was found to be 0.329 (approx.).

5.4. Stacked Models

Stacked models fall under the umbrella of ensemble learning techniques, some of which have already been utilized for the ratings prediction task as demonstrated in the preceding sections [46, 47]. However, these models differ from traditional ensemble techniques in one fundamental way and that is that instead of aggregating or incrementally improving the performance of weak learners, base models are trained and then a meta-model is used to best select the optimal output from the individual base model predictions. This technique leverages the diversity of base models to create a more robust final model. For the present task, two base models were chosen, namely Gradient Boost and AdaBoost and the meta model was a linear regression model. The number of estimators for the base models was chosen to be 50. The stacked model was trained in two different

ways - once without using random search for optimal parameters and the second time by utilizing this technique.

When a stacked models without random search was created and trained on an 80-20 train-test split dataset, the mean squared error (MSE) value for the model was obtained to be 0.0716 (approx.) and the root mean squared error value was found to be 0.267 (approx.). In contrast, when the same model was trained with random parameter search, the mean squared error (MSE) value for the model was obtained to be 0.054 (approx.) and the root mean squared error value was found to be 0.232 (approx.).

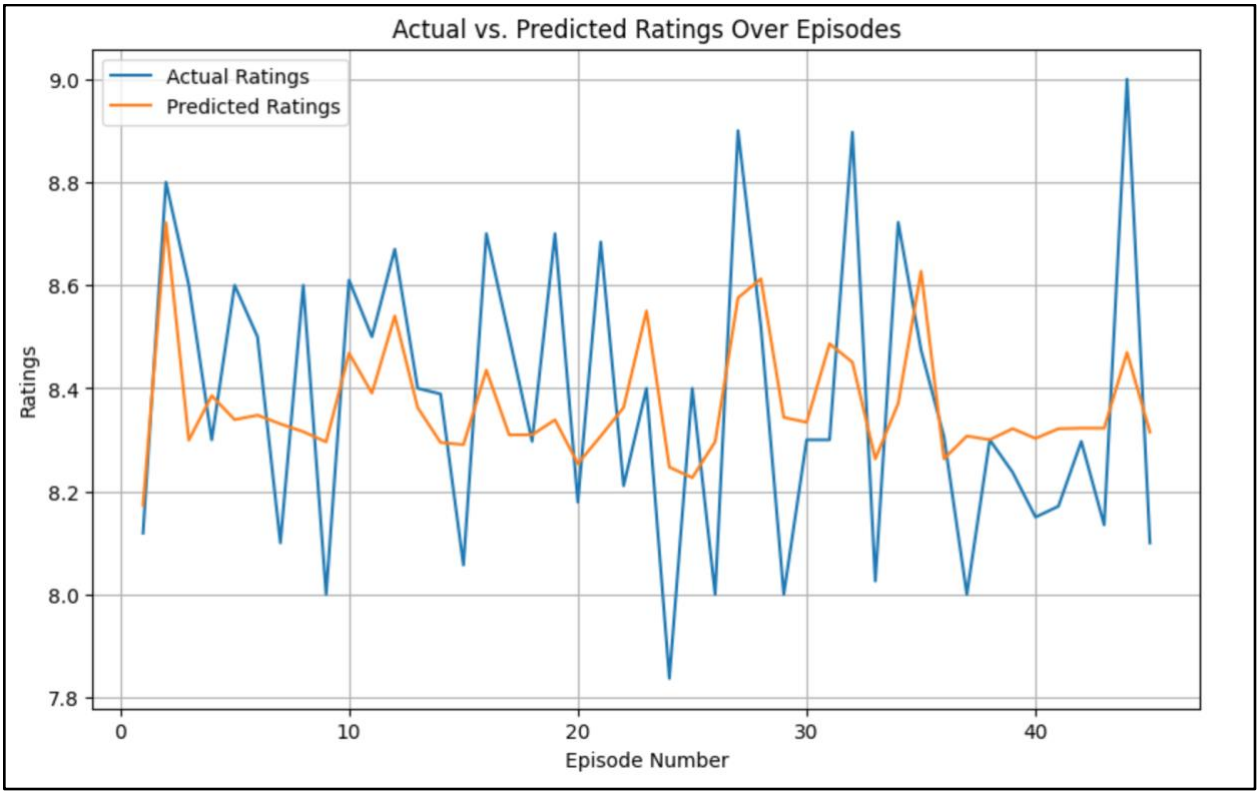

Figure 6: A Plot of The Actual Vs Predict Episode Ratings For Stacked Model

The plot in figure 6 above shows a plot of the difference in the predictions of the stacked model with random search with the actual ratings. As can be seen from the plot, the model is well generalized and looks not to be overfitted. The RMSE value for the model is also shown to be better than those of all the other created models, indicating that out of all the demonstrated approaches, the stacked model with random parameter search shows superior performance.

5.4. Performance Comparison Across Models

In the preceding sections, the details of the different models used for ratings prediction task were provided. In this section, the performance measures obtained by evaluating the models are used to compare these with each other. The bar plot presented in figure 7 below serves to present a comparison of the MSE and RMSE values across the different models.

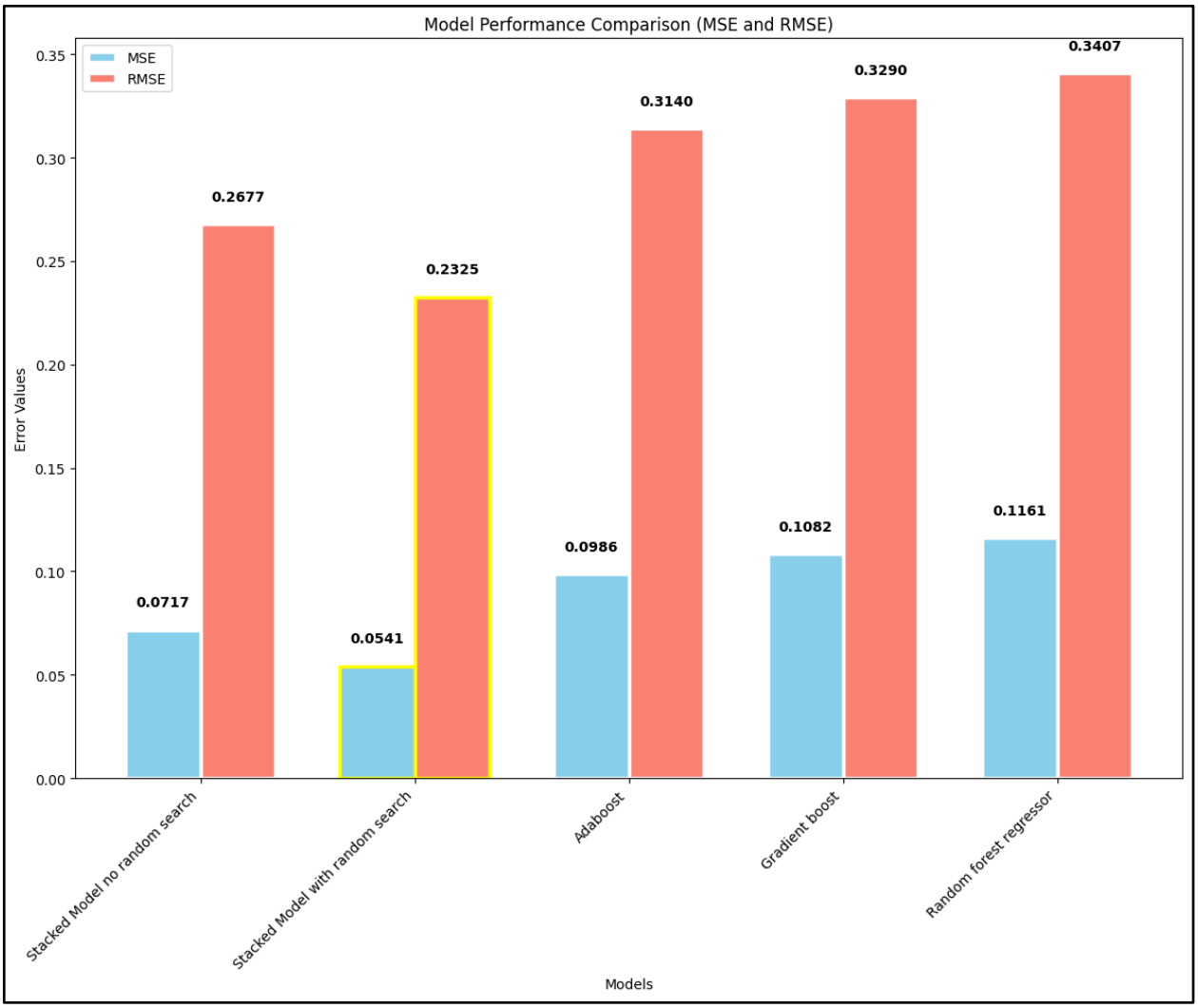

Figure 7: Bar plot of Performance Comparison Across Models

As can be seen from the plot above, the lowest value of both MSE and consequently RMSE is obtained in the case of the stacked model with random search for hyperparameter optimization. This outcome underscores the effectiveness of stacking in combining the strengths of different models when fine-tuned with optimal hyperparameters. The performance of the stacked model without hyperparameter optimization comes a second close at an RMSE value of 0.0717. his observation suggests that the inherent model diversity and complementary nature of the base models in the stacking ensemble contribute significantly to predictive accuracy, even without an exhaustive hyperparameter search.

The AdaBoost and Gradient Boost models have similar performances. Notably, the Random Forest Regressor, which is typically robust and versatile, shows the least favorable performance among the models considered. This could be attributed to suboptimal hyperparameter settings or a mismatch between the characteristics of the Random Forest algorithm and the inherent patterns in the script rating data. Comparing the performance of our stacked model with that of the Simpsonian project as presented in section 2, we see that our random search stacked model outperforms their approach. However, due to the nature of the problem, a direct comparison might not be the best

strategy since the dataset used in the present study as well as the feature set on which the model is trained are fundamentally different from each other.

6. Conclusion and Future Work

In summary, the project introduces a sophisticated approach to predicting TV series ratings through the application of ensemble learning techniques. By integrating diverse data sources and employing advanced feature engineering methods, we have worked to developed a robust model that provides insights into the factors influencing audience reception. The work demonstrates that a combination of metadata from the TMDb API, script analysis, and sentiment analysis yields a comprehensive understanding of a TV series episode. The utilization of character interaction analysis via the Apriori algorithm, sentiment analysis using Text Blob, and the incorporation of innovative methods such as sarcasm detection and emotional archetype mapping have collectively contributed to the predictive prowess of the model.

While the presented model, shows promise, there are various ways in which the performance can be improved and or additionally capabilities can be extracted. For instance, including more TV series in the dataset could provide a broader understanding of the variables affecting ratings across different genres and styles. Additionally, utilizing more sophisticated natural language processing methods to analyze scripts could yield deeper insights, especially in understanding narrative structures and dialogues. Further, as indicated by some previous studies presented in section 2, integrating real-time data, such as social media sentiment about episodes, could enhance the model's accuracy and make it more dynamic.

Nevertheless, this project opens the door to a more data-driven approach in the entertainment industry, leveraging data mining to understand and predict viewer preferences and ratings. The potential applications of this work extend beyond TV series ratings, offering a template for similar analyses in other forms of media and entertainment.

7. References

[1] Otmazgin, N., & Ben-Ari, E. (2013). Popular culture and the state in East and Southeast Asia. https://doi.org/10.4324/9780203801536

[2] Flew, T. (2016). Entertainment media, cultural power, and post-globalization: The case of China’s international media expansion and the discourse of soft power. Global Media and China, 1(4), 278–294. https://doi.org/10.1177/2059436416662037

[3] Focus, W. (2023, December 6). Exploring the field of entertainment: How it became a Multi-Billion dollar industry. The Week. https://www.theweek.in/focus/economy/2023/12/06/exploring-the-field-of-entertainment-how-it-became-a-multi-billion-dollar-industry.html

[4] Statista. (2023, October 24). Employment in the motion picture & sound recording industries in the U.S. 2001-2023. https://www.statista.com/statistics/184412/employment-in-us-motion-picture-and-recording-industries-since-2001/

[5] Ryan, M. (2018, March 15). Variety. Variety. https://variety.com/2017/tv/news/tv-series-budgets-costs-rising-peak-tv-1202570158/

[6] Hawley, N. (2013, March 28). The Hollywood Reporter. The Hollywood Reporter. http://www.hollywoodreporter.com/news/my-generation-creator-pilot-season-430848

[7] Bukszpan, D. (2010, December 10). 16 major TV show failures. CNBC. https://www.cnbc.com/2010/12/10/16-Major-TV-Show-Failures.html

[8] Canal, A. (2022, May 6). The most expensive TV shows of all time: "Stranger Things" and "Lord of the Rings" enter pantheon. Yahoo! Finance. https://finance.yahoo.com/news/the-most-expensive-tv-shows-of-all-time-stranger-things-and-lord-of-the-rings-enter-pantheon-152120290.html

[9] Campo-Rembado, M., & Oakley, S. (2018, October 29). How 20th Century Fox uses ML to predict a movie audience. Google Cloud Blog. https://cloud.google.com/blog/products/ai-machine-learning/how-20th-century-fox-uses-ml-to-predict-a-movie-audience

[10] Hsieh, C. (2018, October 18). Convolutional Collaborative Filter network for video based recommendation systems. arXiv.org. https://arxiv.org/abs/1810.08189

[11] Can Computers Pick Better Movie Scripts? (n.d.). Forbes. https://www.forbes.com/2006/12/03/hollywood-dvd-writers-guild-ent-sales-cx_kw_1201wharton.html

[12] Jehoshua Eliashberg, Mohanbir S. Sawhney, (1994) Modeling Goes to Hollywood: Predicting Individual Differences in Movie Enjoyment. Management Science 40(9):1151-1173.

[13] Eliashberg, J., Jonker, J., Sawhney, M., & Wierenga, B. (2000). MOVIEMOD: An Implementable Decision-Support system for prerelease market evaluation of motion Pictures. Marketing Science, 19(3), 226–243. https://doi.org/10.1287/mksc.19.3.226.11796

[14] R. Jayashree and A. Nayan Varma, "MPAA Rating Prediction Using Script Analysis for Movies," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/I2CT54291.2022.9825434.

[15] Shafaei, M., Samghabadi, N. S., Kar, S., & Solorio, T. (2020). Age Suitability Rating: Predicting the MPAA Rating Based on Movie Dialogues. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 1327–1335). Marseille, France: European Language Resources Association.

[16] USC Viterbi, School of Engineering. (2023, May 16). AI tool may predict movies' future ratings-USC Viterbi|School of Engineering.

[17] Signal Analysis and Interpretation Laboratory (SAIL) – Ming Hsieh Department of Electrical Engineering and Computer Engineering; Department of Computer Science – USC Viterbi School of Engineering. (n.d.). https://sail.usc.edu/

[18] Rosso, C. (2020, November 18). New AI Predicts Movie Ratings Before Filming. Psychology Today. https://www.psychologytoday.com/us/blog/the-future-brain/202011/new-ai-predicts-movie-ratings-filming

[19] Hiathman, D. (2020, December 8). Can Artificial Intelligence Help Predict Movie Ratings? Aol. https://www.aol.com/news/artificial-intelligence-help-predict-movie-140027266.html

[20] Frangidis, P., Georgiou, K., & Papadopoulos, S. (n.d.). Sentiment Analysis on Movie Scripts and Reviews. IFIP Advances in Information and Communication Technology. https://doi.org/10.1007/978-3-030-49161-1_36

[21] Chen, K. (2022, January 6). Predicting IMDB ratings of new Movies - Web Mining [IS688, Spring 2021] - medium. Medium.

[22] Toshkov, D. (2014, March 2). Predicting movie ratings with IMDb data and R. Rules of Reason.

[23] Personalized movie recommendations based on script text. (2016, November 11).

[24] Hunter, S. D., Smith, S., & Chinta, R. (2016b). Predicting New TV Series Ratings from their Pilot Episode Scripts. International Journal of English Linguistics, 6(5), 1. https://doi.org/10.5539/ijel.v6n5p1

[25] Colladon, A. F., & Naldi, M. (2019). Predicting the performance of TV series through textual and network analysis: The case of Big Bang Theory. PLOS ONE, 14(11), e0225306. https://doi.org/10.1371/journal.pone.0225306

[26] Be-Ns. (n.d.). GitHub - be-ns/simpsons_analysis: Predicting IMDB ratings from the scripts alone; A ratio-based approach to Recommenders. GitHub. https://github.com/be-ns/simpsons_analysis

[27] Can You Really Predict How Well a Movie is Going to Do Based on the Script? (2023, January 11). Scripts Shadow. https://scriptshadow.net/can-you-really-predict-how-well-a-movie-is-going-to-do-based-on-the-script/

[28] Steinman, A. (2021, December 16). Navigating my first API: the TMDb Database - Adina Steinman - Medium. Medium. https://adinasteinman.medium.com/navigating-my-first-api-the-tmdb-database-d8d2975b0df4

[29] Nikki. (n.d.). Crazy for Friends fan site - for NBC's sitcom "Friends."

[30] Requests: HTTP for Humans — Requests 2.31.0 documentation. (n.d.). https://requests.readthedocs.io/en/latest/

[31] Richardson, L. (2007). Beautiful soup documentation. April.

[32] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[33] Graceling-Moore, R. (2021, March 22). Friends: Ranking the main Sets/Locations from least to Most important. ScreenRant. https://screenrant.com/friends-sets-locations-most-least-important-ranked/

[34] Blake, L. (2021, May 26). Revisit These Iconic Locations Made Famous by Jennifer Aniston and 'Friends' on the Hit Show. The Hollywood Reporter. https://www.hollywoodreporter.com/lifestyle/real-estate/friends-tv-series-filming-locations-1234959673/

[35] How many different sets/stages were used to film the TV Series Friends? (n.d.). Quora. https://www.quora.com/How-many-different-sets-stages-were-used-to-film-the-TV-Series-Friends

[36] Wikipedia contributors. (2023, May 17). The One with Ross's Wedding. Wikipedia. https://en.wikipedia.org/wiki/The_One_with_Ross%27s_Wedding

[37] David Comberg. (2010, October 30). Kurt Vonnegut on the shapes of Stories [Video]. YouTube. https://www.youtube.com/watch?v=oP3c1h8v2ZQ

[38] Reagan, A. J., Mitchell, L., Kiley, D., et al. (2016). The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science, 5*, 31. https://doi.org/10.1140/epjds/s13688-016-0093-1

[39] michellejieli/emotion_text_classifier · Hugging Face. (n.d.). https://huggingface.co/michellejieli/emotion_text_classifier

[40] helinivan/english-sarcasm-detector · Hugging Face. (n.d.). https://huggingface.co/helinivan/english-sarcasm-detector

[41] Apriori Algorithm in Data Mining: Implementation with examples. (2023, June 27). Software Testing Help. https://www.softwaretestinghelp.com/apriori-algorithm/

[42] API Reference — TextBlob 0.16.0 documentation. (n.d.). https://textblob.readthedocs.io/en/dev/api_reference.html

[43] Beheshti, N. (2022, March 5). Random Forest regression - towards data science. Medium. https://towardsdatascience.com/random-forest-regression-5f605132d19d

[44] Saini, A. (2023, September 20). AdaBoost Algorithm: Understand, implement and Master AdaBoost. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/

[45] Saini, A. (2023a, August 2). Gradient Boosting Algorithm: A complete guide for beginners. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

[46] Sundar, S., V. (2022, August 30). Improve your Predictive Model's Score using a Stacking Regressor. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/12/improve-predictive-model-score-stacking-regressor/

[47] Wadkins, J. (2022, March 30). Simple model stacking, explained and automated - towards data science. Medium. https://towardsdatascience.com/simple-model-stacking-explained-and-automated-1b54e4357916