Analyzing The Effect Of Features Derived From Critic Reviews On IMDb Movie Score Prediction

Project In A.I And Machine Learning

236502

Faculty Of Computer Science – Technion

Students:

Salah Kryem kryem.salah@campus.technion.ac.il **212264667**

Zaid Mreed zaid.mreed@campus.technion.ac.il **212019699**

Introduction

The film industry is a rapidly evolving field, where successful movie performance relies heavily on both critical and public reception. IMDb (Internet Movie Database) ratings are particularly influential in this context, often serving as a key indicator of a film's quality and popularity. These scores, a blend of viewer and critic ratings, play a significant role in shaping a movie's legacy, audience reach, and even its commercial success. Consequently, predicting IMDb scores accurately has become a valuable pursuit in film analytics, marketing, and recommendation systems. However, this task remains challenging, as traditional predictors, such as budget, genre, or casting choices, often miss the nuanced, subjective insights that critiques provide.

In this project, we focus on enhancing IMDb score prediction by investigating the potential of features derived directly from critic reviews. Using a dataset of 5,000 films, sourced from IMDb and similar platforms, we have access to a rich array of existing attributes—including titles, genres, main actors, and crew details—that have been used in numerous prediction models. While these attributes provide useful foundational data, they often lack the depth necessary to fully capture the impact of narrative, theme, or stylistic factors conveyed in critic reviews. Our aim is to bridge this gap by developing a set of new, review-based features.

The additional features we derive will include linguistic indicators such as average word count per review, frequency of punctuation, capital letter usage, and more. These textual markers are not typically integrated into IMDb score predictors, but they hold potential in capturing the underlying sentiment and emphasis that critics bring to their reviews. For instance, a review with higher punctuation frequency or emphatic language might indicate stronger opinions, which could correspond to more extreme ratings. By incorporating such features, we aim to create a more comprehensive dataset that can capture elements of reviewer sentiment and style, thus potentially enhancing prediction accuracy.

Motivation

The motivation behind this project stems from the growing realization that traditional, non-textual movie attributes provide an incomplete view of what influences ratings. While factors like genre and budget give insights into a movie's structural or demographic appeal, critic reviews capture subjective nuances that are essential to understanding a film's broader reception. For example, the language critics use—whether it's positive, neutral, or critical—often reflects the movie's impact, artistic qualities, and overall appeal, factors that may not be evident through numeric data alone.

Moreover, the methods we explore have relevance beyond the film industry, offering insights for any domain where subjective analysis impacts outcomes. By examining linguistic patterns as predictive features, we aim to uncover associations between critic sentiment and IMDb scores, expanding our understanding of how language-driven factors influence audience perception. This analysis could benefit not only movie prediction models but also recommendation systems, marketing strategies, and content analysis across various media platforms.

In summary, this project sets out to evaluate the impact of review-based features on IMDb score prediction accuracy. By developing and testing a model enriched with linguistically derived data, we hope to reveal the added value that critic sentiment and style bring to traditional prediction metrics. Ultimately, our goal is to contribute to a broader understanding of movie reception dynamics, laying the groundwork for future studies in film analytics and media sentiment analysis.

Description of the suggested solution:

Our objective is to compare the prediction accuracy of movie success by using two different sets of features: the original features alone, and a combination of the original and the newly added features. By conducting this comparison, we hope to determine how much the additional features improve the prediction accuracy.

System Description:

The system will be divided into the following sections:

1. Acquiring the critic reviews texts as strings (scrapping from IMDb).

2. Processing the texts to generate new desired features.

3. Adding the new features to the existing dataset.

4. Cleaning up the datasets and running the experiments.

Techniques and algorithms that we used:

This analysis will involve leveraging machine learning techniques to predict movie success, evaluating the performance of the models with both sets of features, and assessing the value added by the newly introduced features from critic reviews. Through this process, we aim to provide insights into the effectiveness of incorporating critic reviews into movie success prediction models.

Generating the new features based on the critic reviews invloved the use of a deep learning model, which we will talk about in detail later in this report.

Technical Details

- Language: Python

- Libraries:

    Data Scrapping: BeautifulSoup (bs4)

    Machine Learning: Scikit-learn, transformers, torch, evaluate, joblib

    Plotting: matplotlib, seaborn

    User Interface: tk, pillow, tmdbv3api

    Other: pandas, numpy, imdb

Detailed description:

The goal of this project is to analyze the effect of additional features derived from critic reviews on predicting the IMDb score of movies. We start by acquiring critic reviews for each movie, from the IMDb website, and storing them in a dataset.

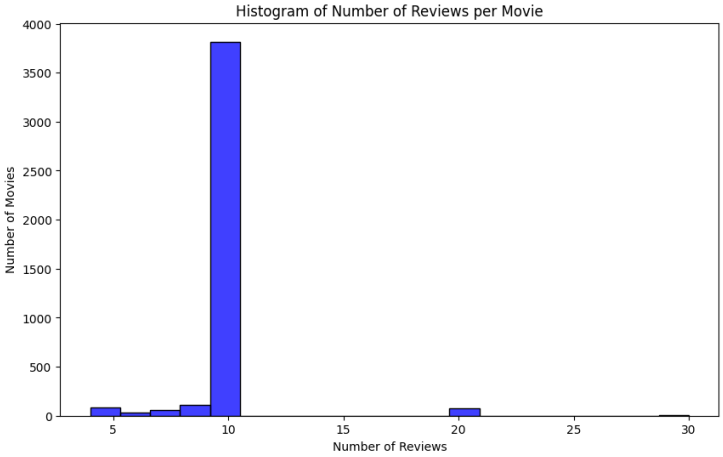
Then, we supplement this data by generating new features for each movie related to the critic reviews. The process involved the following steps:

Data Scraping: Extracting Critic Reviews

The first step in our project involves gathering the critic reviews from the IMDb page of the movie. We accomplish this using by using the bs4 library to scrape the required information from the website.

For most of the movies, we acquried 10 reviews. However, some movies had a very small number of reviews which meant that the data we acquried was limited.

Here is a histogram showing the number of reviews acquried for each movie:



Feature Generation: Deriving Insights from Critic Reviews

After gathering the raw data through the scraping process, the next step in our project involves generating features, from the collected critic reviews.

The set of derived features consists of features that we thought would be useful for the prediction task.

We calculate the following features for each review. In order to calculate the feature value for a movie, we take the average of the feature values of reviews that belong to that movie. For example, to calculate the **Letters per Word** for a particular movie, we take the average **Letters per Word** of all reviews for that movie.

The sentiment of a movie is calculated by the following weighted aggregation function:

$$Sentiment(movie) = \frac{N_{positive} - N_{negative}}{N_{total}},$$

$$N_X = number\ of\ reviews\ with\ X\ sentiment. N_{total} = N_{positive} + N_{negative} + N_{neutral}$$

Derived Features:

1. **Mean Words per Review**: This feature captures the average number of words in the review. Longer reviews may indicate more detailed feedback, which could provide richer information and potentially correlate with the film's success.
2. **Letters per Word**: This feature measures the average word length in the review. A higher ratio might suggest a more formal or sophisticated review style, which could align with certain types of films or specific critique patterns.

3. **Capital Letter Frequency**: This feature evaluates the proportion of capital letters used in the review. Capital letters are often associated with emphasis or strong sentiment, and they can also be used in names, dates, or important references. Reviews with a higher frequency of capital letters might indicate heightened emotional intensity or contain specific details that influence our understanding of the film.
4. **Punctuation Frequency**: This feature tracks the occurrence of punctuation marks, offering insight into the writing style of the reviewer. Punctuation can influence the tone and structure of the review, which may reflect the critical or descriptive nature of the review.
5. **Numbers Frequency**: This feature captures the frequency of numerical references in the review, such as statistics, comparisons, or ratings. Reviews with more numbers may be more analytical or fact-driven, providing a different perspective on the film.
6. **Sentiment Analysis**: This feature determines the overall sentiment of the review (movie)—whether it is positive, negative, or neutral. Sentiment analysis helps to quantify the emotional tone of the review and provides insights into the critic's opinion of the movie, which could directly affect our prediction of the movie's success. We employ a fine-tuned deep learning model for this task (details in the next section).

Sentiment Analysis Using Pretrained BERT

For the sentiment analysis part of the project, we used a pretrained BERT model, which we fine-tuned for the specific task of predicting the sentiment of movie reviews. BERT is a transformer-based language model that was pre-trained on vast amounts of text data. BERT is highly effective for various natural language processing (NLP) tasks.

Fine-tuning allows us to adapt the general language understanding of the pre-trained BERT model to our specific task. In this case, we fine-tuned BERT on a labeled dataset of movie reviews, adjusting the model's parameters to improve its performance in predicting sentiment categories (e.g., positive, neutral, negative). By retaining the rich language representations learned during pre-training and tailoring the model with our specific data, we achieved a more accurate and task-specific sentiment analysis model.

Fine-tuning Dataset Preparation

- Dataset Acquisition and Preparation:

We downloaded a pre-existing dataset containing movie reviews and made several adjustments to prepare it specifically for training a model capable of Sentiment Prediction.

Initially, we cleaned up the dataset by removing unnecessary columns such as "title", "user", and "spoilers". We also ensured that only English-language reviews from critics were retained, dropping other types of reviews.

After that, we converted review scores to sentiment labels. The original dataset included critic reviews with ratings on a scale of 0-100. In order to utilize this dataset for sentiment analysis, we had to transform the raw review scores into a categorical sentiment feature.

Finally, we removed reviews that appeared both in our original dataset and the fine-tuning dataset (overlapping reviews).

- Converting Review Scores to Sentiment Labels:

The critic review scores were mapped to three distinct sentiment categories: negative, neutral, and positive. We achieved this conversion by defining specific thresholds that map the review score to one of the three categories. Mathematically, the mapping is described as follows:

$$Sentiment(review) = \begin{cases} negative, & review\_score \leq 35 \\ neutral, & 35 < review\_score \leq 65 \\ positive, & review\_score > 65 \end{cases}$$

These thresholds ensure that the continuous review scores are transformed into discrete sentiment values, enabling us to train the model using these categorical labels.

Removal of Overlapping Reviews:

A critical part of the preparation involved removing overlapping reviews. This step was essential to prevent data leakage during the fine-tuning process. Data leakage can artificially inflate performance metrics and lead to poor generalization on new, unseen data. By eliminating these overlapping reviews, we ensured that the fine-tuning dataset consisted solely of unique reviews, providing a more reliable and unbiased training process for sentiment prediction.

- Fine-Tuning the Sentiment Prediction Model:

We fine-tuned a pre-trained deep learning model (based on BERT architecture) to perform sentiment prediction on movie reviews. During the fine-tuning process, we trained the model for 3 epochs and achieved an accuracy of approximately 75.6%.

The fine-tuning process required a considerable amount of time due to the model's complexity and the dataset's size, prompting us to save the fine-tuned model for use “as is” in subsequent runs.

Saving Fine-Tuned Model Results:

Given the time-intensive nature of the fine-tuning process, we carried it out only once and saved the resulting weights and model configuration. This enables us to reuse the fine-tuned model for future sentiment predictions without needing to retrain it from the beginning, streamlining our workflow and ensuring faster inference times.

Creating The Final Dataset

we focused on preparing the dataset by combining the newly derived features with the original IMDb movie dataset, ensuring a unified and structured dataset for further analysis. The process included merging data collected from critic reviews with the original dataset, handling any inconsistencies such as missing values or duplicates, and transforming the data into a consistent format.

Initial Data Cleaning:

We began by loading the original movie dataset and performing basic cleaning operations. This includes removing duplicates, dropping irrelevant columns (such as movie links, movie titles, etc..), and handling missing values. Some columns were cleaned by dropping rows with missing values, while others are filled with either the most frequent values or median values.

Merging Derived Features:

After the initial cleaning, we merge our derived feature sets (like average sentiment, review statistics) into the main dataset.

Final Cleanup:

Finally we handle missing values in the newly added features. We offer two options: either filling in missing values using the median, or removing rows with missing data.

Based on previous performance tests, the default is to remove rows with missing reviews, as this yields better predictive performance.

For most prediction experiments, the following step was also applied:

Handling Categorial Data:

In this step we process categorical variables (like language, content rating, etc..) in the dataset to make them suitable for machine learning models. We apply **one-hot encoding** to transform the columns into numerical form, allowing these features to be used in the model. This step is crucial for converting categorical data into a format that machine learning algorithms can understand.

Interactive Movie Score Prediction Program with User Interface

This program offers an interactive experience for users to input a movie title, retrieve detailed movie information via web scraping, and predict the movie's IMDb score using pre-trained machine learning models. The latest version includes a user-friendly interface, which now displays movie details, poster images, and suggestions for similar movies, enhancing the overall experience.

IMDb Score Predictor

Enter Movie Name:

Avatar

Movie: Avatar | Year: 2009 | Genres: Action, Adventure, Fantasy, Sci-Fi

Predicted IMDb scores for 'Avatar':
SVM: 7.05
RandomForest: 6.58
MLP: 5.92
Actual IMDb Score: 7.9

Similar Movies:
Clear and Present Danger
Nacho Libre
Police Story 4: First Strike

Predict Score

IMDb Score Predictor

Enter Movie Name:

Harry Potter

No exact match for 'Harry Potter'. Did you mean:
Harry Potter and the Sorcerer's Stone
Harry Potter and the Goblet of Fire
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Chamber of Secrets
Harry Potter and the Order of the Phoenix

Predict Score

1. Program Overview

The program takes a user-provided movie name and automates the process of gathering necessary features for IMDb score prediction. It collects data from online sources, including both original features (e.g., genre, budget, release date) and derived features (e.g., sentiment metrics), with missing values filled by median values from the training dataset. After feature collection, a sample is formatted and processed through our trained models to generate an IMDb score prediction.

2. User Interface Enhancements

The new user interface allows for a more interactive and visually appealing experience:

- **Movie Details**: Displays key movie information including title, release year, and genres.
- **Movie Poster**: Shows the movie's poster if available, adding a visual aspect to the prediction experience.

3. Model Training and Datasets

- **Models Used**: The program uses three machine learning models for IMDb score prediction:
    - Support Vector Machine (SVM)
    - Random Forest
    - Multi-Layer Perceptron (MLP)
- **Training Dataset**: The Expanded Filtered Dataset was used for training, including a range of features such as genre, budget, and derived features like sentiment metrics.
- **Training Process**: Each model was fine-tuned on this dataset to optimize prediction accuracy.

4. Model Integration in the Interactive Program

The pre-trained models are saved in a dedicated directory (`./saved_trained_models/interactive_prog`) and loaded at runtime, making the prediction process fast and resource-efficient by avoiding re-training.

5. Prediction Output

After gathering the features of the specified movie, the program passes the sample through each model to generate predicted IMDb scores. The user sees:

- **Predicted IMDb Scores**: An estimated IMDb score is displayed, showing the output of each prediction model.
- **Ground Truth IMDb Score**: The program retrieves the actual IMDb score (if available) during web scraping, allowing users to compare predictions with real-world data.

Overall, the Interactive Movie Score Prediction Program combines a rich UI experience with robust data analysis, providing users with both predicted and actual IMDb scores, and enabling exploration of similar movie titles.

6. Similar Movies and Suggestions Features The program offers additional interactive features:

- Similar Movies: After each prediction, the program fetches a list of similar movies using an external API, displaying their titles as selectable text. This enriches the user experience by offering alternative titles for exploration.
- Suggestions for Close Matches: If an exact movie match isn't found, the program suggests closely matching titles based on partial name matches. This feature helps guide the user to the correct movie or discover other titles related to their query.

Experimental Methodology

Objective

The primary goal of our experiments is to evaluate the performance of various machine learning models in predicting IMDb scores by utilizing features extracted from both movie metadata and critic reviews. The focus is to determine how these features contribute to the predictive power of the models. We aim to compare the models' performance on two datasets: one containing only basic metadata, and another that includes both metadata and additional features derived from critic reviews. This comparison will help us understand the impact of incorporating textual data and sentiment analysis on the accuracy of the predictions.

Motivation

The motivation behind this study stems from the hypothesis that while traditional metadata such as cast, genre, and release date provide a solid foundation for prediction, they may not capture the full spectrum of factors influencing a movie's success. Critic reviews, on the other hand, offer rich qualitative data that can provide deeper insights into a film's reception. By integrating features like sentiment analysis and linguistic patterns from these reviews, we aim to enrich the dataset and potentially improve the model's ability to predict IMDb scores more accurately. These experiments will also allow us to explore how machine learning models can leverage both numerical and textual data to enhance performance.

In addition to evaluating model performance, we conduct further experiments to gain deeper insights into the relationships between different features and their contribution to the prediction process. This will help us identify the most influential features and understand their role in predicting movie success.

Datasets

We use two datasets in most of the experiments:

1. **Original Dataset**: Contains the basic movie metadata and critic review features.
2. **Expanded Dataset**: Contains additional derived features (e.g., sentiment metrics, word frequencies) to improve predictive performance.

Models Evaluated

- **Support Vector Regression (SVR)**
- **Random Forest Regressor**
- **MLP**

Model Training and Hyperparameter Tuning

- **Training/Validation Split**: After splitting the data into train and test sets, we further divide the training set into train and validation sets. The validation set is used for hyperparameter tuning.
- **Hyperparameter Tuning**.
- **Standardization**: The features are standardized to ensure proper feature scaling.

Evaluation Metrics

- **Mean Squared Error (MSE)**
- **Mean Absolute Error (MAE)**
- **R-squared (R^2)**
- **Root Mean Squared Error (RMSE)**

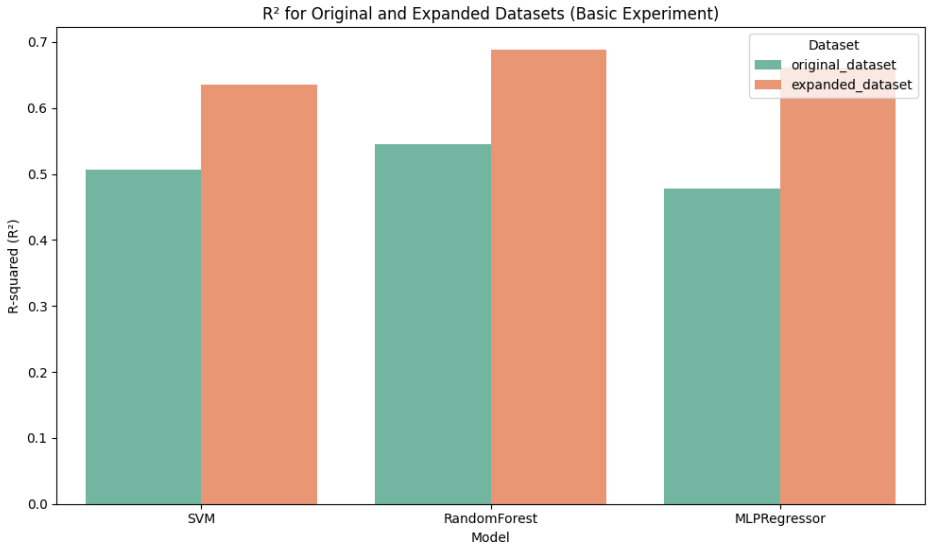
Each model's performance is logged after testing, and hyperparameters for SVR and Random Forest are saved for future experiment runs.

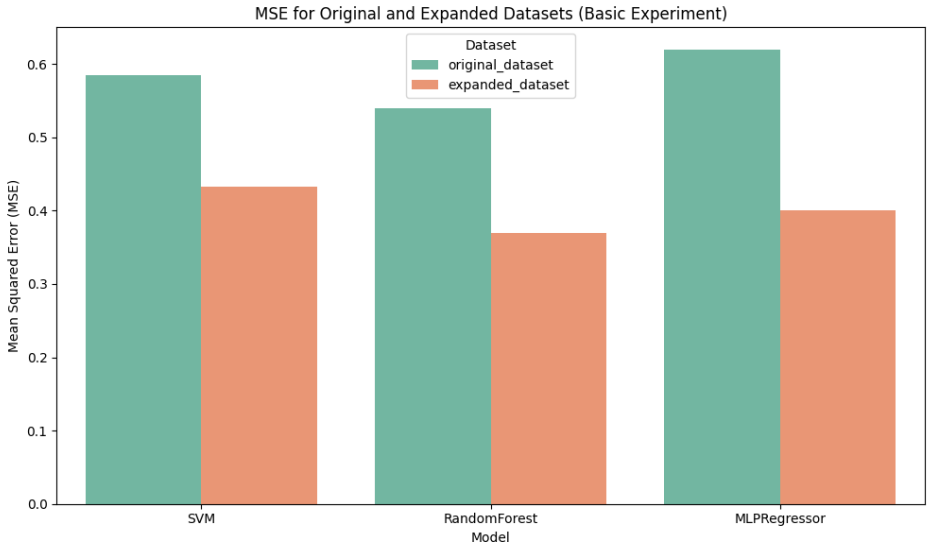
Experiment Details:

1- Basic Experiment:

In this experiment we compare the performance of the prediction models on both the original dataset (without our added features) and an expanded dataset that contains our features.

Results:





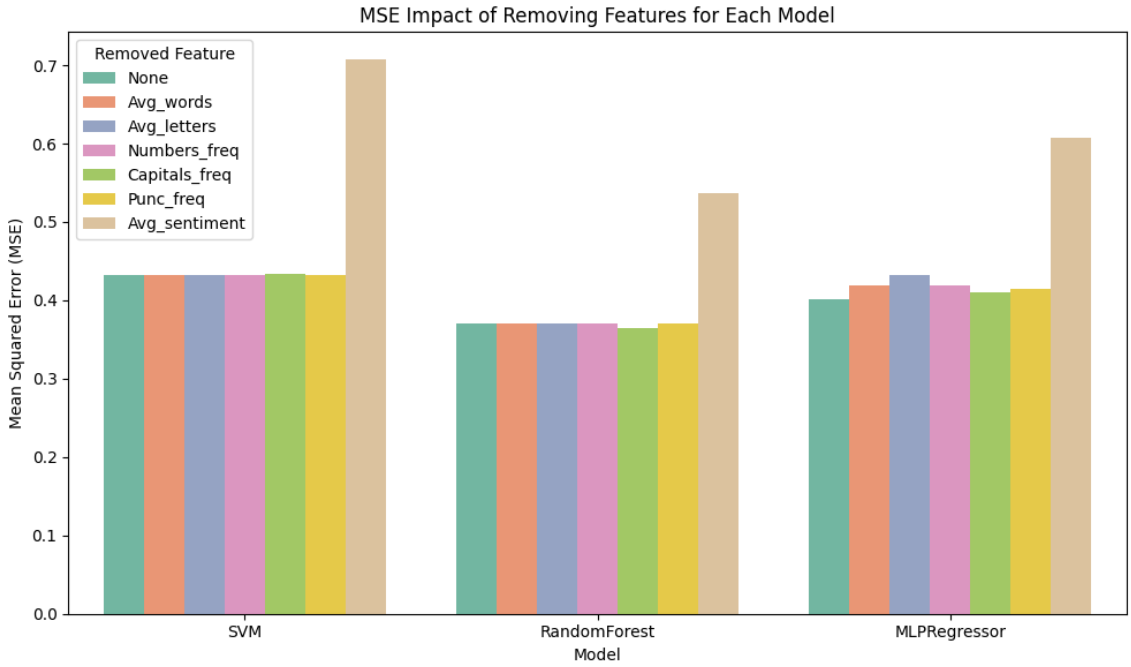
Conclusions:

The experiment results show the advantage of using the **expanded dataset** over the **original dataset**, as evidenced by improved performance metrics across all models.

2- Individual Feature Impact Experiment:

In this experiment we go through the added features and compare the performance of the prediction models on the expanded dataset that contains all the features, and a dataset without a specific added feature, to measure the impact of the removed feature.

Results:



Conclusion:

As we can see, the performance of all models stays nearly the same when removing the following features :{Avg_words, Avg_letters, Numbers_freq, Capitals_freq, Punc_freq}.

This indicates that these features were not very helpful to for the prediction process.

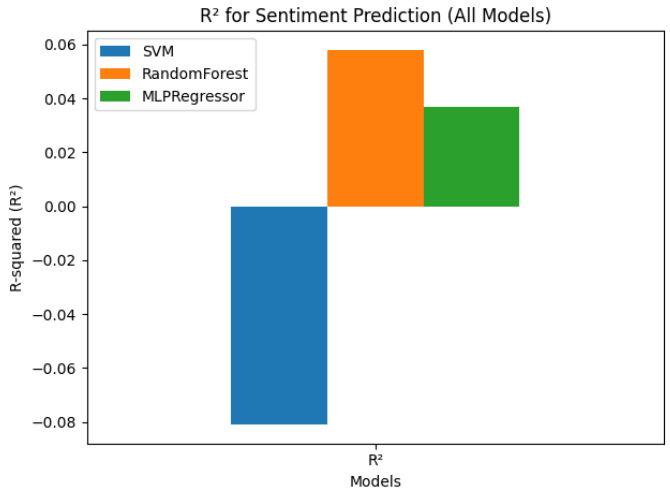
However, we see a huge drop in performance when removing the {Avg_sentiment} feature (MSE is higher), which indicates that this feature is significant for the prediction process and has a high impact on the result.

3- Average Sentiment Prediction Using Other Derived Features:

In this experiment we try to build a model that predicts the Avg_sentiment feature using the other derived features (Avg_words, Avg_letters, Numbers_freq, Capitals_freq, Punc_freq).

If the results are good enough, we can use this method to calculate the Avg_sentiment instead of using a fine-tuned BERT model, which saves time and resources.

Results:



Conclusion:

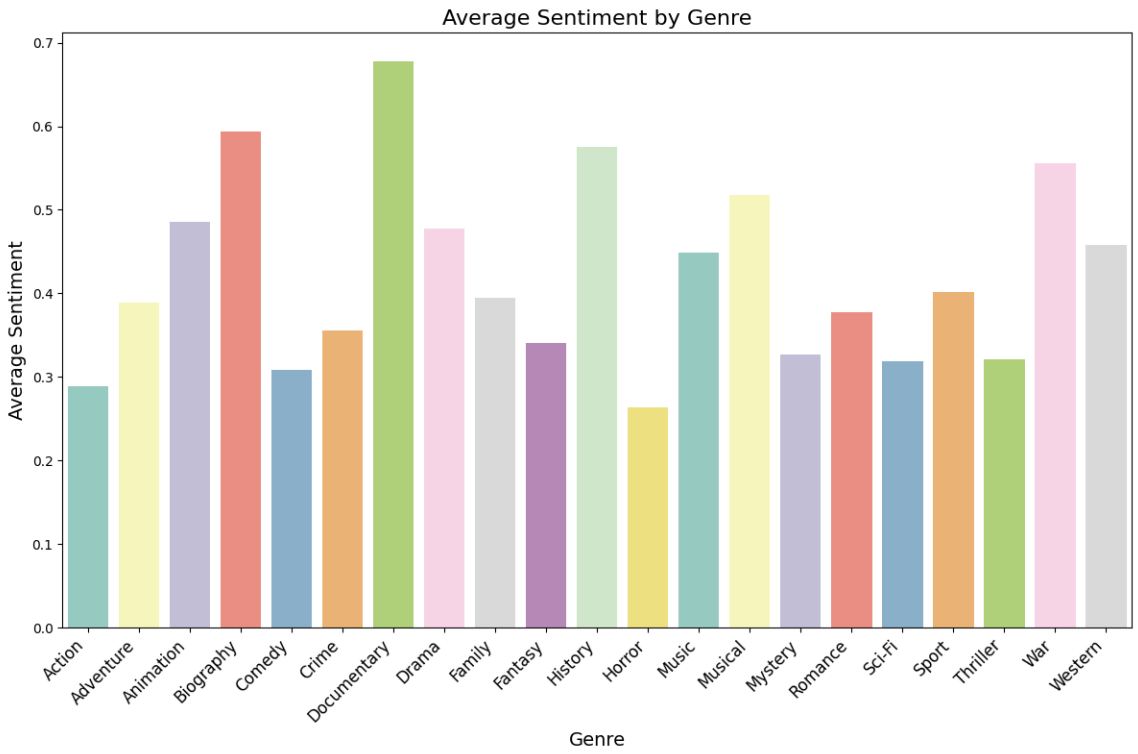
As we can see, the R^2 score is very low for all models, which means that the models failed to capture the Avg_sentiment feature using the other features.

Further work could be done in this experiment to improve performance, like using different models, applying feature engineering, and other Machine Learning techniques.

4- Calculating The Average Sentiment For All Genres:

In this experiment, we calculate the Avg_sentiment for all movies in a single genre, and calculate the average. We consider only genres that have more than 10 movies, as to not get biased results.

Results:



Conclusion:

As the results show, the average sentiment differs from genre to genre.

Genres that tend to focus more on real-life content, such as **Documentary** and **Biography**, had a higher sentiment average. This suggests that critics may respond more positively to films with educational or emotional narratives grounded in reality.

Horror, **Action**, **Sci-Fi**, and **Thriller** have the lowest average sentiments. These genres often deal with darker, more intense themes that may divide audiences, or they may appeal to niche audiences that are harder to please across the board.

5- Correlation Between Average Sentiment and Other Features:

In this experiment we calculated the correlation between Avg_sentiment and the following features: {imdb_score, gross, budget}.

Results:

Correlation between **Avg_sentiment** and **imdb_score**: 0.695.

Correlation between **Avg_sentiment** and **gross**: 0.129.

Correlation between **Avg_sentiment** and **budget**: 0.017.

Conclusion:

1- A correlation of **0.695** suggests a **strong positive relationship** between Avg_sentiment and IMDb scores. This implies that movies with higher average sentiments (based on reviews) tend to receive higher IMDb scores as well.
This also supports the results of Experiement 2, that indicate a high impact of the Avg_sentiment feature on predicting the imdb_score target.

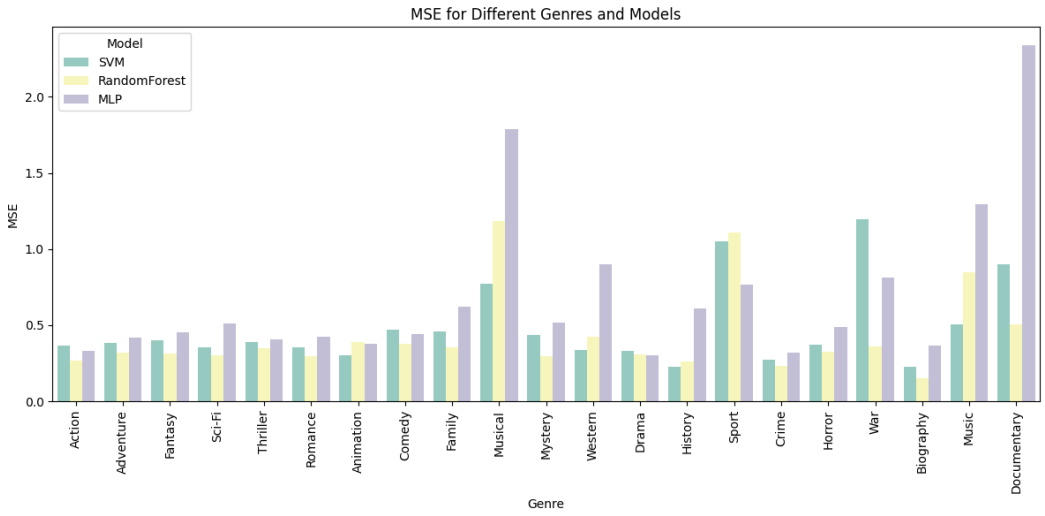
2- A correlation of **0.129** suggests a **weak positive relationship** between sentiment and gross revenue. While there is a slight positive connection, it's not a strong indicator.
This means that the financial success (gross) of a movie doesn't always match the sentiment of its reviews. Big-budget films can make a lot of money even with mixed reviews, while movies that get good reviews may not always do well at the box office.

3- A correlation of **0.017** shows an **extremely weak, almost negligible relationship** between sentiment and budget.
The amount of money spent on making a movie (budget) doesn't seem to influence how positively or negatively it's received by critics in terms of sentiment.

6- Prediction Accuracy For A Single Genre:

In this experiment we aim to determine if IMDb score predictions improve when the models are trained and tested within a specific genre. We focused on individual genres, training the models on movies exclusive to each genre and evaluated the accuracy through MSE and R^2. We consider only genres that have more than 10 movies, as to not get biased results

Results:



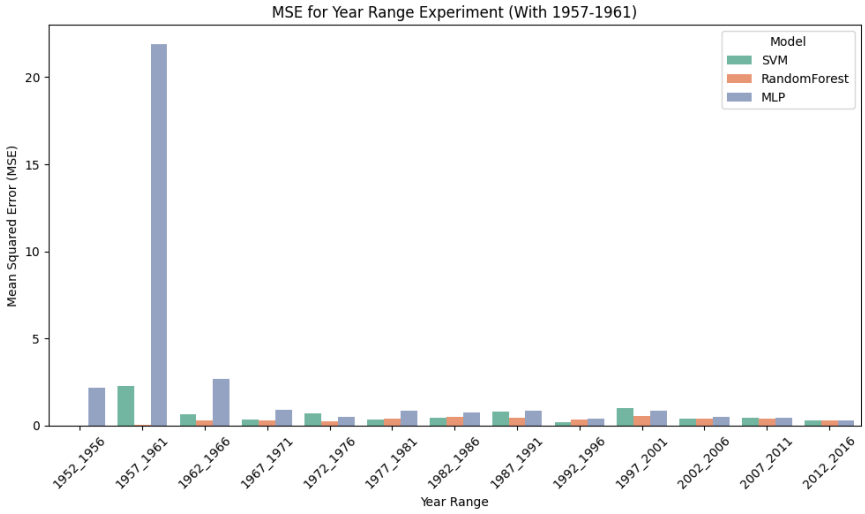
Conclusion:

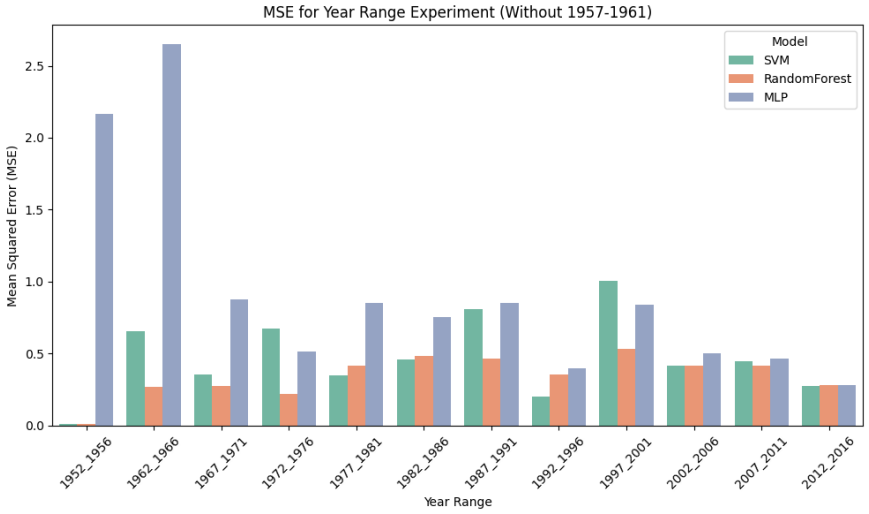
The experiment reveals that prediction accuracy varies considerably by genre, likely due to the inherent differences in storytelling and audience reception within each genre. The lower MSE values in Action, Comedy, and Drama genres suggest that these genres may have more standardized characteristics or appeal, allowing models to perform better. In contrast, genres with higher MSE scores, like Musical and Western, may have more diverse or niche characteristics, making them harder to predict.

7- Prediction Accuracy For A Single Time Range:

In this experiment we aimed to assess the model accuracy in predicting IMDb scores across different 5-year movie release intervals. The dataset was split into groups based on movie release years (e.g., 1960-1964, 1965-1969, etc.), and each group was evaluated separately using the three models.We consider only ranges that have more than 10 movies, as to not get biased results

Results:





The results indicate that the MLP model showed a significantly higher error for the 1957-1961 range, making it an outlier. When excluding this range, the MSEs across the remaining intervals were generally lower and more consistent across all models. Overall, the SVM and Random Forest models tended to perform more consistently across different year ranges compared to the MLP.

Conclusion:

This analysis reveals that movie release periods might impact model accuracy, especially for older intervals. Removing outliers like 1957-1961 improves the readability of performance trends across models, suggesting that specific historical periods could pose unique prediction challenges. contrast, genres with higher MSE scores, like Musical and Western, may have more diverse or niche characteristics, making them harder to predict.

Summary:

We found out that using the information of the “Average Sentiment” feature enhances the performance of prediction models in predicting the “IMDb score” of movies.

Questions for further study:

1- Using more critic reviews: In our project, we used 10 reviews for most of the movies to derive new features. This number can be changed and the results can be measured accordingly (less reviews for faster runs, more reviews for better performance).

2- Deriving other features from the reviews: From our set of derived features, only the “Average Sentiment” was impactful for the prediction task. Further study could be done to come up with other impactful features.

3- Sentiment Prediction and calculation: We used a pretrained BERT model and fine-tuned it to predict the sentiment of reviews based on their text, and we chose customized thresholds to translate from 0-100 score to a sentiment label, and a customized weighted aggregation function to calculate the average sentiment of a movie. These (the DL model, thresholds and weighted aggregation function) can be expiremented with to try and improve the performance.

4- Predicting the Average Sentiment based on Derived Features: In this experiment, we used the derived features to try and predict the average sentiment. Further work can include using Machine Learning techniques like feature engineering to try and improve the prediction accuracy.

A.I Usage:

ChatGPT was used for some parts of the code, as well as refining this report.