Content Recommendation System Using Sentiment Analysis and Spoiler Detection

Prathamesh Renghe[1], Vaibhav Udamale[2], Tejas Vaidya[3], Aparna Halbe[4]

[1,2,3]Student, Department of Information Technology, Sardar Patel Institute of Technology
[4]Professor, Department of Computer Science and Engineering, Sardar Patel Institute of Technology

Abstract

This web application is a user-friendly platform for movie and TV show enthusiasts. It provides detailed information about each title, including plot summaries, genres, durations, ratings, and the cast. Users can watch trailers to get a preview of the content. Leveraging a massive movie database, the app curates custom suggestions tailored to each user's unique taste, utilizing cosine similarity. The app takes recommendations a step further by utilizing a content-based system that analyzes user reviews. The sentiment analysis helps identify positive and negative feedback. To preserve the viewing experience, the system includes a spoiler detection mechanism that hides potential spoilers from user reviews. Users can create their own watchlists to keep track of desired content. With machine learning algorithms, the application continuously improves its recommendations, ensuring users find engaging content tailored to their tastes. This web app delivers a seamless and user-centric approach to discovering and enjoying movies and TV shows, emphasizing the joy of surprise while exploring reviews and suggestions.

Keywords: Movie Recommender System, Content Based Filtering, Sentiment Analysis, Spoiler Detection

1. Introduction

The landscape of entertainment has undergone a dramatic shift in the digital era. With tons of movies and tv shows available across various streaming platforms, finding content that truly resonates with individual tastes and preferences can be a daunting task. Movie and TV show enthusiasts often struggle with information overload, uncertain recommendations, and the risk of stumbling upon spoilers in user reviews. To address these challenges and provide a seamless, user-centric approach to discovering and enjoying content, we present a comprehensive web application.

The primary challenge for movie and TV show enthusiasts is the overwhelming array of options and the lack of a personalized, reliable, and user-friendly platform to navigate this vast landscape. They often face the following issues:

- Information Overload: The sheer volume of available content makes it difficult for users to access detailed information about movies and TV shows quickly.
- Uncertain Recommendations: Relying solely on general recommendations or trending lists often leads to dissatisfaction, as they may not align with individual preferences.
- Spoilers: The fear of encountering spoilers in user reviews can diminish the excitement and surprise factor of watching a movie or TV show.

- Lack of Organization: Keeping track of desired content for future viewing can be challenging without a convenient watchlist feature.

Our web application aims to address these challenges by offering a user-friendly platform tailored to the needs of movie and TV show enthusiasts. Leveraging the TMDB database, the app offers comprehensive details about each movie and show, including plot summaries, genres, runtimes, ratings, and cast information. Users can also watch trailers to gain a preview of the content.

2. Proposed System

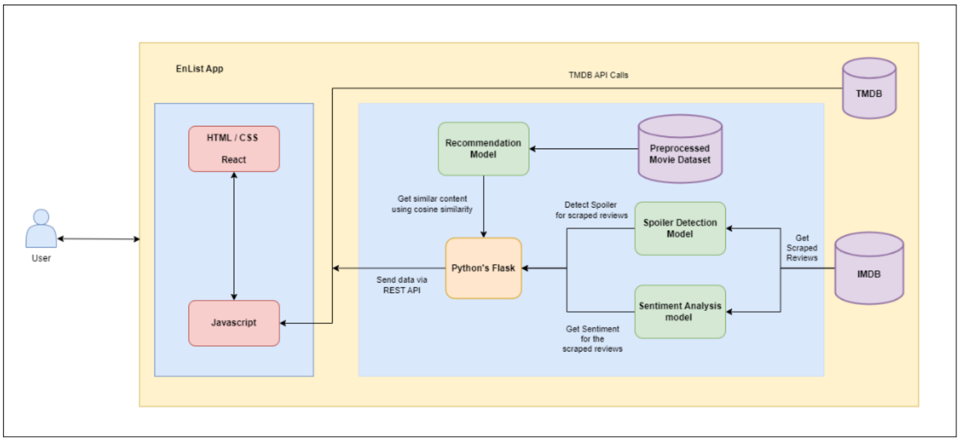


Figure 1: Architecture Diagram of the Proposed System

The application's architecture is bifurcated into two key components: the frontend and the backend. The user interface is crafted using HTML, CSS, and React, providing a seamless and engaging experience. Meanwhile, Flask serves as the robust backend, facilitating the management and integration of machine learning models and databases.

To enrich the user experience, the application leverages the TMDB API to fetch comprehensive movie metadata, including titles, cast, ratings, trailers, and more. This information is seamlessly presented on the frontend through a REST API. The recommendation model, employing cosine similarity on a preprocessed movie dataset, suggests personalized movie recommendations to the user.

For a deeper understanding of user sentiments, the application employs automated scripts to scrape IMDB reviews for movies and TV series. These reviews undergo analysis by both a Sentiment Analysis model and a Spoiler Detection model. The Sentiment Analysis model categorizes reviews as either 'positive' or 'negative,' while the Spoiler Detection model identifies spoilers, labeling reviews as 'True' or 'False.' The processed data is then communicated to the Flask backend, which, in turn, pushes it to the frontend through the developed REST API.

On the frontend, users are empowered with options. They can choose to hide reviews containing spoilers and have the flexibility to load additional reviews if the initially scraped ones are deemed insufficient. This dynamic interaction enhances the user's control over their experience, providing a comprehensive and user-friendly platform for exploring and engaging with movie content.

3. Implementation

A. Movie Recommendation System:

1. Data Collection:

We carefully collect data from two main sources - Kaggle and Movie Database (TMDB) to create quality content based on movie recommendations. Kaggle provided access to the TMDB 5000 Movies Dataset, enriching it with valuable information. The TMDB 5000 Movies Dataset offers essential details like title, category, duration, and rating, broadening our dataset's scope. Additionally, we integrated data from The Indian Movie Database on Kaggle to enhance the regional perspective, ensuring a comprehensive coverage of both Hollywood and Bollywood films.

Additionally, the TMDB API played a crucial role in obtaining detailed movie information. By utilizing this API, we accessed finer details such as title, category, duration, and rating. This not only expanded the breadth of our dataset but also allowed us to incorporate real-time updates, ensuring that our movie information remains relevant and up-to-date.

2. Data Preprocessing and Integration:

The transformation from raw datasets to a polished recommendation system involved a careful data preprocessing phase. Using Jupyter Notebooks, we refined the data, focusing on key attributes like cast, genres, movie IDs, titles, and plot summaries. The goal was to simplify the information, making it suitable for effective recommendation modeling.

The fusion of TMDB, Kaggle, and The Indian Movie Database was accomplished through careful integration. The resulting dataset, featuring 6000+ movies, seamlessly blends global and regional perspectives. This integration ensures a diverse coverage of Hollywood and Bollywood movies, providing a comprehensive snapshot of the film industry. Our integrated dataset now serves as the foundation of our recommendation system, ready to offer personalized movie suggestions by combining Kaggle's valuable contributions with the real-time updates from TMDB.

3. Data Exploration:

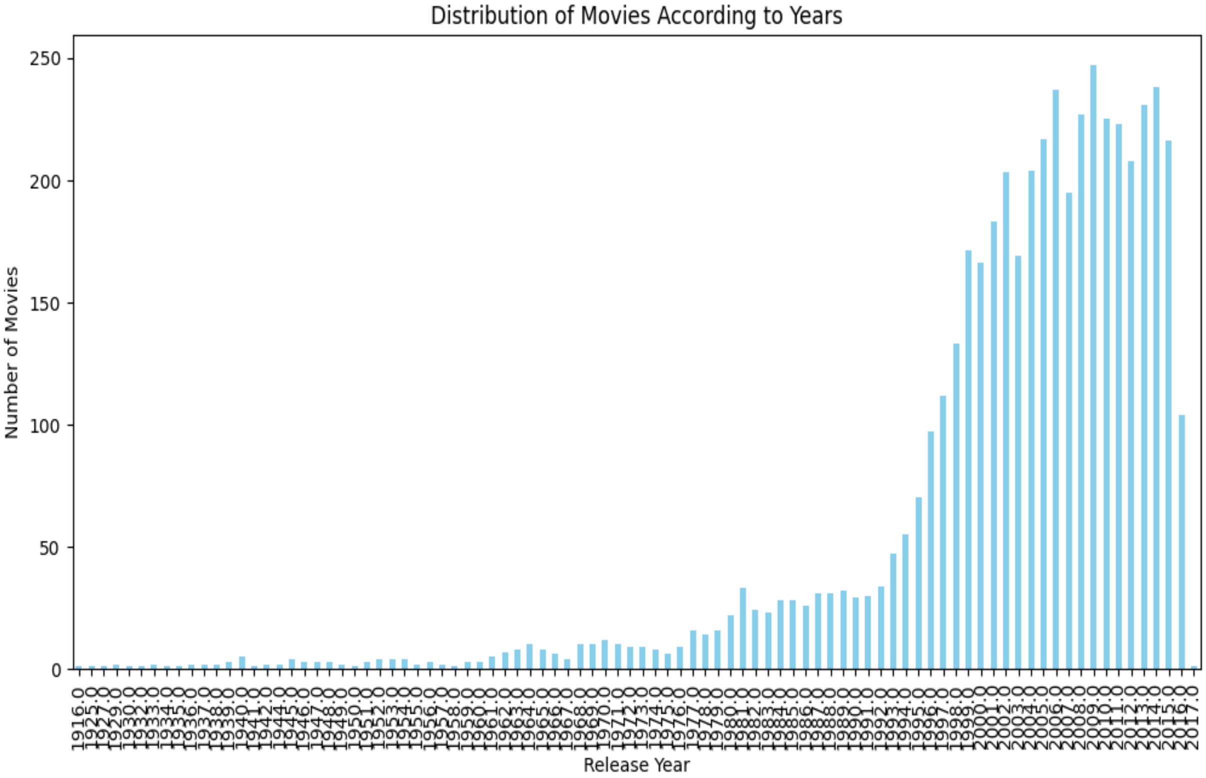


Figure 2: Yearly Distribution of Movies

The distribution of movies across different years, as illustrated in the bar chart, provides insights into the temporal patterns of film production. Peaks and troughs in the chart offer a glimpse into periods of increased or decreased cinematic activity, reflecting the evolving dynamics of the industry over time.

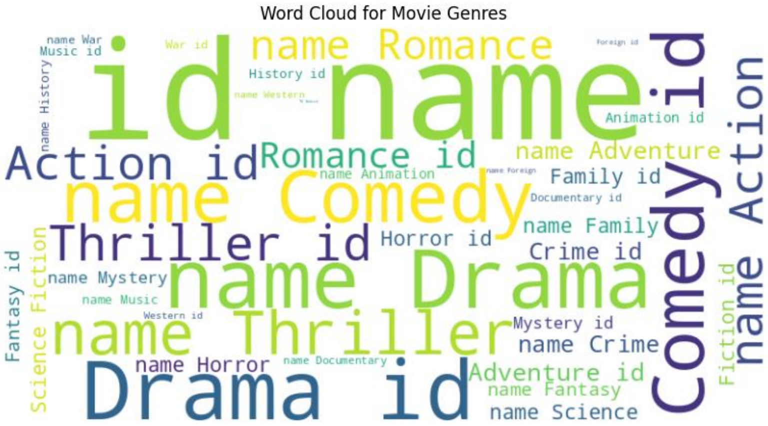


Figure 3: Word Cloud for Movie Genres

Additionally, the word cloud showcasing movie genres visually captures the prevalent themes within the dataset. Larger, bolder words in the cloud signify genres that occur more frequently, providing a quick overview of the thematic landscape in the movies. This visual summary contributes to a better understanding of the diversity of genres present in the dataset, offering a snapshot of the cinematic content's overarching themes. Collectively, these visual representations enhance our grasp of both the temporal and thematic dimensions within the dataset.

4. Content-Based Recommendation System:

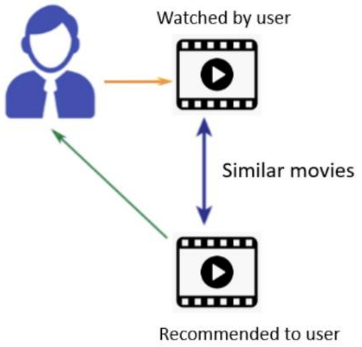


Figure 4: Content-Based Filtering

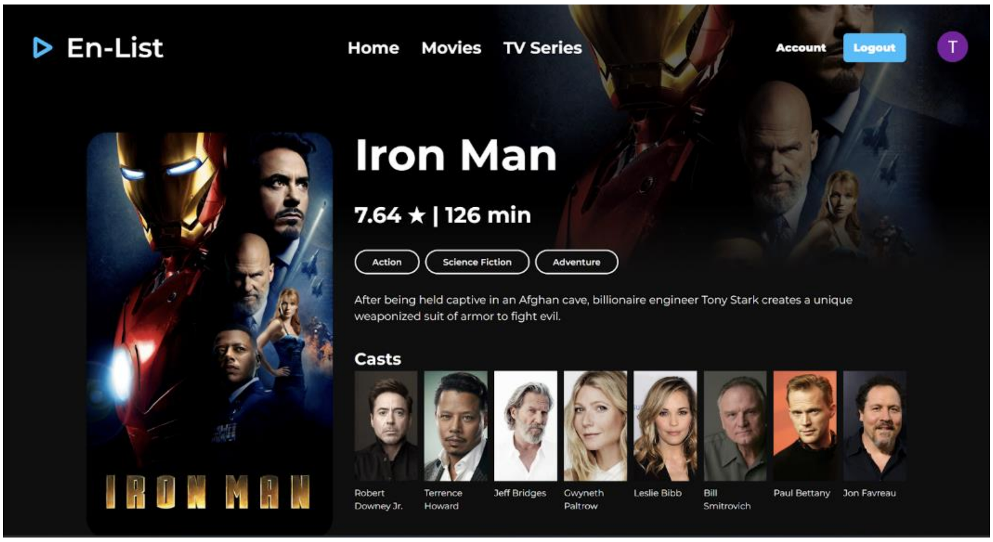
The Content-Based Recommendation System integrates datasets from Kaggle and The Movie Database (TMDB) to offer personalized movie suggestions. Utilizing the TMDB API, the system retrieves detailed movie information, including title, category, duration, and more. A meticulous data preprocessing step involves intelligently merging relevant columns like 'cast' and 'genres' to enhance the dataset and contribute to the accuracy of the recommendation engine.

In the transformation phase, advanced techniques such as CountVectorizer and TfidfVectorizer are applied. CountVectorizer creates a sparse matrix representing word frequency from the combined 'cast' and 'genres' features, while TfidfVectorizer analyzes the 'plot' column, assigning values based on term significance.

The system's innovation lies in the calculation of Cosine Similarity, assessing movie similarity based on extracted features. The Cosine Similarity matrix establishes relationships between movies, identifying those with comparable attributes like genre, cast, and plot elements. Upon user input of a movie title, the system dynamically retrieves the most similar movies by analyzing cosine values. The top 20 recommendations align closely with user preferences, providing an effective content-based movie suggestion experience. Understanding fundamental concepts such as Sparse Matrix, Bag of Words, Tf-Idf Vectorizer, and Cosine Similarity is recommended for a comprehensive grasp of the system's workings.

5. Integration of Recommendation System with React Web Application using REST API:

To enhance accessibility and adaptability, our recommendation system seamlessly integrates with a REST API using Flask. The Flask application, enriched with CORS functionality, serves as a robust conduit for cross-origin requests, facilitating interaction with the recommendation system. This API connects seamlessly with a React web application, establishing a symbiotic relationship between backend recommendation algorithms and frontend interfaces. Users can now enjoy personalized movie recommendations dynamically fetched through the API, providing an intuitive and visually enriched experience. This integration extends the reach of our Content-Based Recommendation System, ensuring effortless access to personalized movie suggestions with additional details and visual enhancements.



After being held captive in an Afghan cave, billionaire engineer Tony Stark creates a unique weaponized suit of armor to fight evil.

Figure 5: Movie Information Page

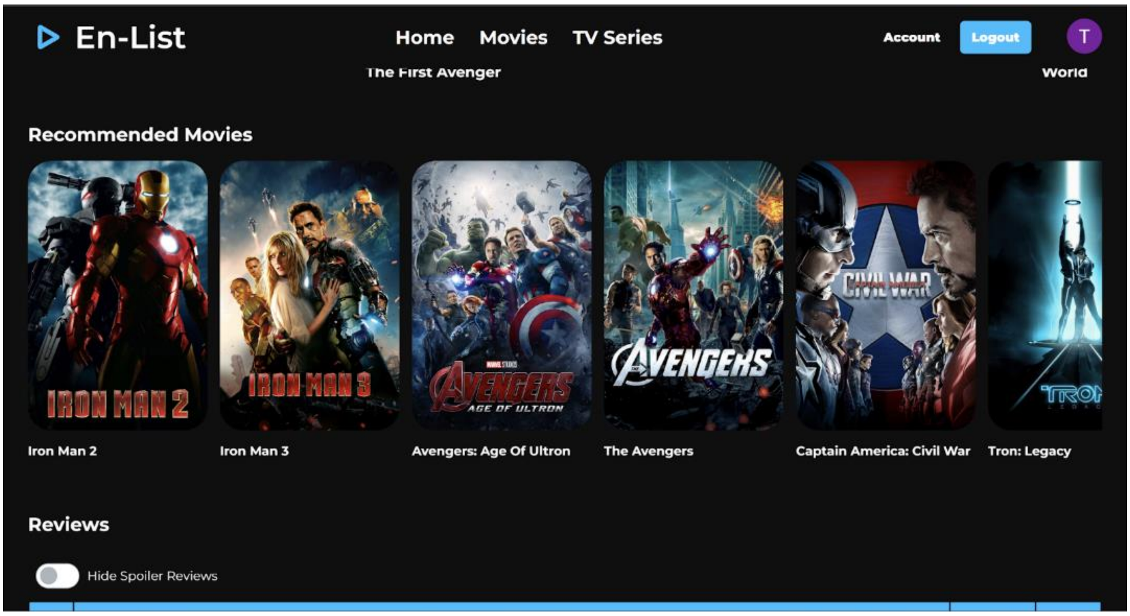


Figure 6: Recommended Movies Based on Selection

6. Cosine Similarity:

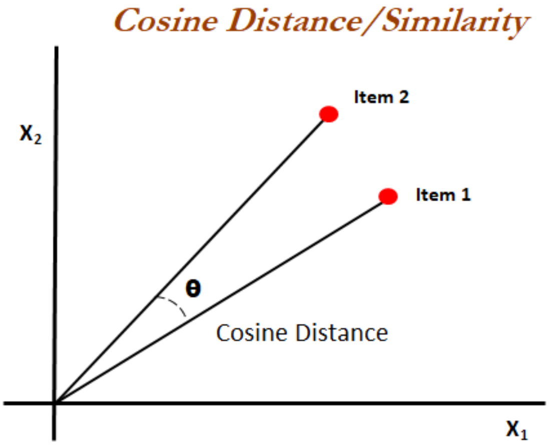


Figure 7: Cosine Distance/Similarity

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Figure 8: Cosine Distance/Similarity Formula

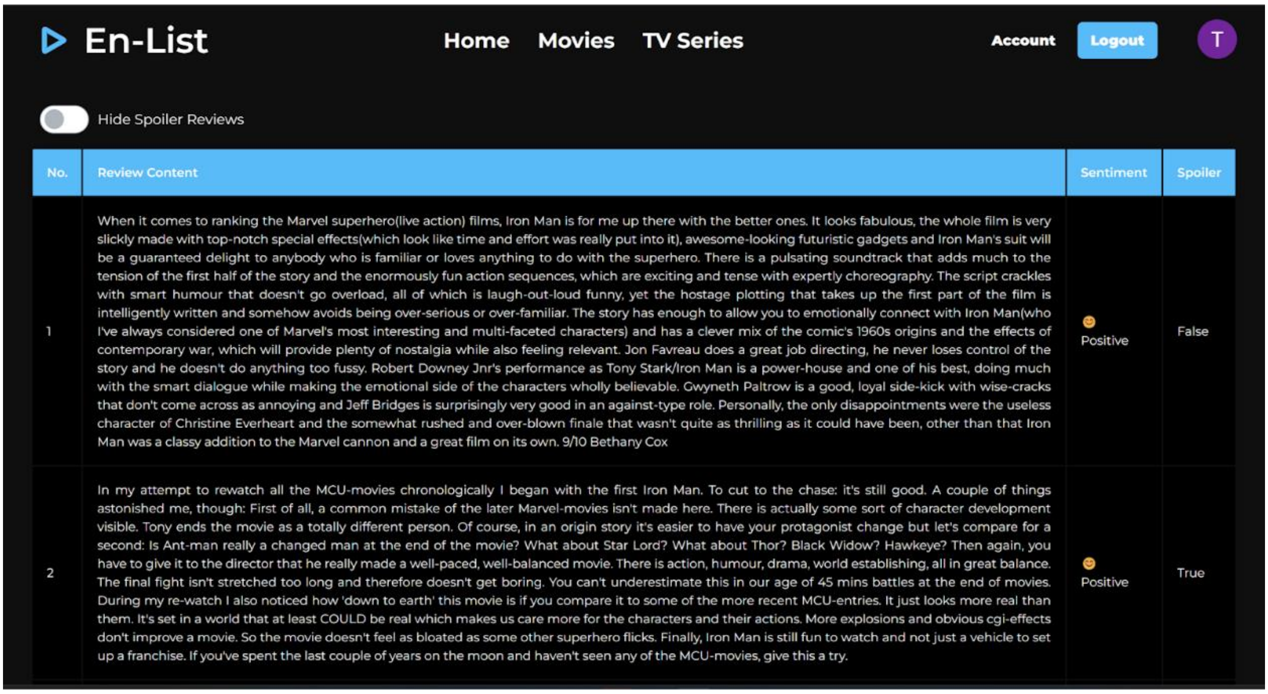
Cosine similarity, a pivotal component in our movie recommendation system, functions as a quantitative measure for assessing the likeness between movie vectors. This mathematical metric calculates the cosine of the angle between non-zero vectors, providing a straightforward approach to evaluating their similarity within an inner product space. In the context of our research, the vectors represent movies, and their cosine similarity serves as a crucial factor in identifying movies with comparable features. The formula involves the division of the dot product of vectors by the product of their Euclidean norms. As the cosine similarity score approaches 0, it signifies a higher degree of dissimilarity, while values closer to 1 indicate a stronger resemblance. This methodological choice finds application in various machine learning domains, offering an effective means of comparing movies and enhancing the accuracy of our recommendation system.

B. Sentiment Analysis:

The model for sentiment analysis was developed using the Transformers library, an interface that offers transformer models. This model leverages the DistilBERT architecture, a more efficient and quicker alternative to the BERT model, while preserving a substantial amount of BERT's capabilities. The DistilBERT tokenizer was utilized to convert the data into integer sequences, which were then padded to maintain a uniform length of 128 tokens across all sequences.

The model was assembled using the Adam optimizer, a common selection for deep learning tasks due to its effectiveness and minimal memory usage. Binary Crossentropy, a loss function suitable for binary classification tasks such as sentiment analysis, was used. The model's performance during training was assessed using the Binary Accuracy metric. The data labels were one-hot encoded to align with the model's output. One-hot encoding is used to transform categorical data to feed into machine learning algorithms to enhance prediction accuracy. Early stopping and progress bar logging were incorporated as callbacks during the training process.

The model's effectiveness was assessed over three training epochs. The validation loss, a measure of the model's prediction error on the validation data, was used to monitor the training process. The model exhibited high binary accuracy on both the training and validation datasets, indicating its proficiency in classifying the sentiment of the reviews. The model's high accuracy, along with its technical complexity, highlights the potential of transformer models in tasks such as sentiment analysis.

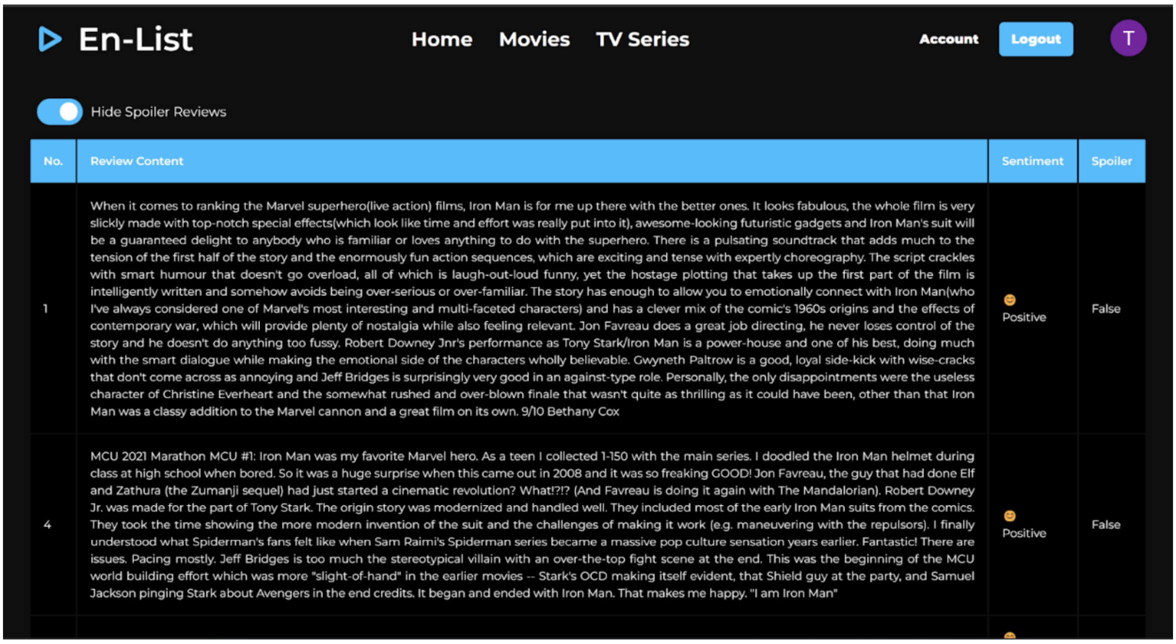


No.	Review Content	Sentiment	Spoiler
1	When it comes to ranking the Marvel superhero(live action) films, Iron Man is for me up there with the better ones. It looks fabulous, the whole film is very slickly made with top-notch special effects(which look like time and effort was really put into it), awesome-looking futuristic gadgets and Iron Man's suit will be a guaranteed delight to anybody who is familiar or loves anything to do with the superhero. There is a pulsating soundtrack that adds much to the tension of the first half of the story and the enormously fun action sequences, which are exciting and tense with expertly choreography. The script crackles with smart humour that doesn't go overload, all of which is laugh-out-loud funny, yet the hostage plotting that takes up the first part of the film is intelligently written and somehow avoids being over-serious or over-familiar. The story has enough to allow you to emotionally connect with Iron Man(who I've always considered one of Marvel's most interesting and multi-faceted characters) and has a clever mix of the comic's 1960s origins and the effects of contemporary war, which will provide plenty of nostalgia while also feeling relevant. Jon Favreau does a great job directing, he never loses control of the story and he doesn't do anything too fussy. Robert Downey Jnr's performance as Tony Stark/Iron Man is a power-house and one of his best, doing much with the smart dialogue while making the emotional side of the characters wholly believable. Gwyneth Paltrow is a good, loyal side-kick with wise-cracks that don't come across as annoying and Jeff Bridges is surprisingly very good in an against-type role. Personally, the only disappointments were the useless character of Christine Everheart and the somewhat rushed and over-blown finale that wasn't quite as thrilling as it could have been, other than that Iron Man was a classy addition to the Marvel cannon and a great film on its own. 9/10 Bethany Cox	😊 Positive	False
2	In my attempt to rewatch all the MCU-movies chronologically I began with the first Iron Man. To cut to the chase: it's still good. A couple of things astonished me, though: First of all, a common mistake of the later Marvel-movies isn't made here. There is actually some sort of character development visible. Tony ends the movie as a totally different person. Of course, in an origin story it's easier to have your protagonist change but let's compare for a second: Is Ant-man really a changed man at the end of the movie? What about Star Lord? What about Thor? Black Widow? Hawkeye? Then again, you have to give it to the director that he really made a well-paced, well-balanced movie. There is action, humour, drama, world establishing, all in great balance. The final fight isn't stretched too long and therefore doesn't get boring. You can't underestimate this in our age of 45 mins battles at the end of movies. During my re-watch I also noticed how 'down to earth' this movie is if you compare it to some of the more recent MCU-entries. It just looks more real than them. It's set in a world that at least COULD be real which makes us care more for the characters and their actions. More explosions and obvious cgi-effects don't improve a movie. So the movie doesn't feel as bloated as some other superhero flicks. Finally, Iron Man is still fun to watch and not just a vehicle to set up a franchise. If you've spent the last couple of years on the moon and haven't seen any of the MCU-movies, give this a try.	😊 Positive	True

Figure 9: Reviews with Their Sentiment

C. Spoiler Detection:

The spoiler detection model was architected using Keras, a high-level neural networks API. The model employs GloVe embeddings, which are pre-trained word vectors that encapsulate semantic relationships between words. The review texts were tokenized, transforming the corpus into sequences of integers. These sequences were then subjected to padding to ensure uniform length across all sequences. The model's architecture comprises an Embedding layer, two Bidirectional LSTM layers, combined with a Dense output layer. Among these, the Embedding layer is initialized with the GloVe embeddings, while the LSTM layers process the sequences bidirectionally, capturing both preceding and succeeding contextual information within the review texts. The Dense layer, with a sigmoid activation function, outputs the probability of a review containing a spoiler. While evaluation, the model threshold was set to be more sensitive towards spoilers. Consequently, this stringent threshold bolsters the model's capability to flag potential spoilers, thereby enhancing the reliability of the spoiler detection system.



Hide Spoiler Reviews

No.	Review Content	Sentiment	Spoiler
1	When it comes to ranking the Marvel superhero(live action) films, Iron Man is for me up there with the better ones. It looks fabulous, the whole film is very slickly made with top-notch special effects(which look like time and effort was really put into it), awesome-looking futuristic gadgets and Iron Man's suit will be a guaranteed delight to anybody who is familiar or loves anything to do with the superhero. There is a pulsating soundtrack that adds much to the tension of the first half of the story and the enormously fun action sequences, which are exciting and tense with expertly choreography. The script crackles with smart humour that doesn't go overload, all of which is laugh-out-loud funny, yet the hostage plotting that takes up the first part of the film is intelligently written and somehow avoids being over-serious or over-familiar. The story has enough to allow you to emotionally connect with Iron Man(who I've always considered one of Marvel's most interesting and multi-faceted characters) and has a clever mix of the comic's 1960s origins and the effects of contemporary war, which will provide plenty of nostalgia while also feeling relevant. Jon Favreau does a great job directing, he never loses control of the story and he doesn't do anything too fussy. Robert Downey Jnr's performance as Tony Stark/Iron Man is a power-house and one of his best, doing much with the smart dialogue while making the emotional side of the characters wholly believable. Gwyneth Paltrow is a good, loyal side-kick with wise-cracks that don't come across as annoying and Jeff Bridges is surprisingly very good in an against-type role. Personally, the only disappointments were the useless character of Christine Everheart and the somewhat rushed and over-blown finale that wasn't quite as thrilling as it could have been, other than that Iron Man was a classy addition to the Marvel cannon and a great film on its own. 9/10 Bethany Cox	😊 Positive	False
4	MCU 2021 Marathon MCU #1: Iron Man was my favorite Marvel hero. As a teen I collected 1-150 with the main series. I doodled the Iron Man helmet during class at high school when bored. So it was a huge surprise when this came out in 2008 and it was so freaking GOOD! Jon Favreau, the guy that had done Elf and Zathura (the Zumanji sequel) had just started a cinematic revolution? What!?!? (And Favreau is doing it again with The Mandalorian). Robert Downey Jr. was made for the part of Tony Stark. The origin story was modernized and handled well. They included most of the early Iron Man suits from the comics. They took the time showing the more modern invention of the suit and the challenges of making it work (e.g. maneuvering with the repulsors). I finally understood what Spiderman's fans felt like when Sam Raimi's Spiderman series became a massive pop culture sensation years earlier. Fantastic! There are issues. Pacing mostly. Jeff Bridges is too much the stereotypical villain with an over-the-top fight scene at the end. This was the beginning of the MCU world building effort which was more "slight-of-hand" in the earlier movies -- Stark's OCD making itself evident, that Shield guy at the party, and Samuel Jackson pinging Stark about Avengers in the end credits. It began and ended with Iron Man. That makes me happy. "I am Iron Man"	😊 Positive	False

Figure 10: Reviews with Detected Spoilers

4. Experimental & Result Analysis

The sentiment analysis model achieved a binary accuracy of approximately 95.77% on the training data and 87.05% on the validation data. The early stopping mechanism implemented ensured optimal stopping of the training process, preventing overfitting by halting training once the validation loss ceased to decrease. This mechanism enhances the model's generalizability, ensuring its performance extends to unseen data.

The spoiler detection model's performance was quantified using the accuracy metric, yielding a score of approximately 76.06%. The model was configured to exhibit high sensitivity towards potential spoilers. This was achieved by setting the classification threshold to 0.195, thereby lowering the barrier for classifying a review as a spoiler.

5. Conclusion & Future Scope

Through this application, we embarked on a multifaceted exploration of enhancing user experience and engagement in content consumption through the integration of content-based recommendation, sentiment analysis, and spoiler detection. Our objective was to develop a comprehensive system that not only recommends personalized content but also gauges user sentiment and protects against unintentional spoilers.

The content-based recommendation provided users with tailored recommendations, addressing the challenges associated with sparse user-item interaction data and improving the user experience. The sentiment analysis model allowed quick filtering of reviews to understand the context easily.

Furthermore, the integration of spoiler detection mechanisms was a crucial step in preserving the enjoyment of content for users. By identifying and mitigating spoilers, our system aimed to create a safer and more enjoyable environment for content enthusiasts, particularly in today's era of widespread online discussions.

In future, with the better tuning of hyperparameters and more sophisticated data, the accuracy of spoiler detection can be improved, eliminating the need for a lower threshold thus reducing the number of false positives. The recommendation system can be implemented with a Hybrid approach which includes both Content-based filtering and Collaborative filtering.

6. Acknowledgement

We extend our sincere gratitude to our Project Guide, Prof. Aparna Halbe, from the Department of Computer Science and Engineering at Sardar Patel Institute of Technology (SPIT), for her unwavering support, guidance, and encouragement throughout our project. Prof. Halbe actively engaged with us at every stage of our research, providing valuable insights and ensuring that our methodology was sound and our conclusions were well-supported by our findings.

7. References

1. P. Satish Chadokar, N. Jain, and A. Thakre, "Movie Recommendation Engine with Sentiment Analysis Using AJAX Request," International Journal of Innovative Research in Management, Psychology, and Social Sciences, accessed Nov. 23, 2023, https://www.ijirmps.org/papers/2023/1/230043.pdf
2. C.-S. M. Wu, D. Garg, and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," IEEE Xplore, Nov. 01, 2018, https://ieeexplore.ieee.org/abstract/document/8663822
3. "Movie Recommendation System Using Content Based Novel Feature Extraction Information," IEEE Conference Publication, IEEE Xplore, accessed Nov. 23, 2023, https://ieeexplore.ieee.org/document/10128249
4. S. Kumar, K. De, and P. P. Roy, "Movie Recommendation System Using Sentiment Analysis From Microblogging Data," IEEE Transactions on Computational Social Systems, pp. 1–9, 2020, doi: https://doi.org/10.1109/tcss.2020.2993585
5. S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," 2020 2nd International Conference on Computer and Information Sciences (ICCIS), October 2020, doi: https://doi.org/10.1109/iccis49240.2020.9257657
6. "Multi-task sentiment classification model based on DistilBert and multi-scale CNN," IEEE Conference Publication, IEEE Xplore, accessed Nov. 23, 2023, https://ieeexplore.ieee.org/document/9730378
7. "Analyzing the Performance of Sentiment Analysis using BERT, DistilBERT, and RoBERTa," IEEE Conference Publication, IEEE Xplore, accessed Nov. 23, 2023, https://ieeexplore.ieee.org/document/10059542
8. "Movie Recommendation System using Cosine Similarity with Sentiment Analysis," IEEE Conference Publication, IEEE Xplore, accessed Nov. 23, 2023, https://ieeexplore.ieee.org/document/9544794
9. "A Basic Study on Spoiler Detection from Review Comments Using Story Documents," IEEE Conference Publication, IEEE Xplore, Oct. 01, 2016, https://ieeexplore.ieee.org/document/7817115
10. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.
11. Misra, Rishabh. "IMDB Spoiler Dataset." DOI: 10.13140/RG.2.2.11584.15362 (2019).
12. F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 5, no. 4, pp. 19:1–19:19, 2015.

13. "TMDB Movie Metadata," Kaggle, 2023. [Online]. Available: [https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata]
14. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, June 2011, pp. 142-150.