Josthana Runkana

Sambuj Dhara

Surya Vamshi Sriperambudooru

MOVIE RECOMMENDATION SYSTEMS

Group - 8

**Table of Contents**

1. Introduction
2. Problem Statement
3. Objective
4. Methodology
5. Data Collection
6. Data Preprocessing
7. Preliminary Analysis
8. Exploratory Data Analysis
9. Model Development
10. Challenges & Soutions
11. Model Performance
12. Feasibility and Rationale
13. Model Evaluation
14. Feature Infulence and Error Analysis
15. Model Limitations and Bias
16. Insights and Business Applications
17. Conclusion
18. References
19. Integrity Statement

**Introduction**

In today’s fast-paced world, people often seek activities that help them relax and unwind. Watching movies is one such popular activity, offering an escape from the stresses of daily life. However, with the vast amount of movie content available globally in Netflix, users often face the challenge of selecting a movie that aligns with their preferences. The large volume of options can make the process of searching and choosing a movie time-consuming and overwhelming. To address this issue, recommendation systems have emerged as a valuable tool, providing users with personalized and relevant suggestions, thereby simplifying the decision-making process. Recommendation systems have been in use for quite some time, significantly enhancing user convenience by delivering quick and accurate suggestions.

This project focuses on developing a **Movie Recommendation System** that leverages **Machine Learning (ML)** and **Natural Language Processing (NLP)** techniques. The system will analyze various aspects of movies, such as their descriptions, genres, and user interactions, to generate tailored recommendations. By utilizing a dataset-driven approach, the system will learn patterns and preferences from historical data, enabling it to predict and suggest movies that users are likely to enjoy.

**Problem Statement**

In the digital era, the vast number of movies across multiple platforms makes it difficult for users to find content that matches their preferences. Traditional search methods, which use broad filters like genre, year, or ratings, often fall short in providing personalized recommendations. As a result, users face a time-consuming and frustrating experience.

The goal of this project Is to design and develop a movie recommendation system that utilizes content-based filtering methods. By analyzing the content and features of movies, the system will identify patterns and similarities between films, enabling it to provide accurate and personalized recommendations. This approach aims to enhance the user experience by reducing the time and effort required to find relevant movies, ultimately leading to greater user satisfaction and engagement.

**Objective**

* **Develop a Content-Based Recommendation System:** Leverage NLP techniques to analyze movie descriptions and compute similarity scores.
* **Implement Machine Learning:**Use algorithms to identify patterns in features and user preferences for predictions.
* **Improve User Experience:** Enable users to discover relevant movies efficiently.
* **Evaluate System Performance:** Assess the effectiveness of recommendations through accuracy metrics.

**Methodology**

* **Data Collection:**
  + Gathered a comprehensive dataset of 8,807 movies from Netflix, including details such as descriptions, genres, themes, user reviews, and ratings.
  + Ensured data quality by cleaning and preprocessing to handle missing or inconsistent information. Filled missing values with appropriate substitutes.
* **Data Cleaning and Preprocessing:**
  + Checked for missing values and summarized the dataset to identify inconsistencies.
  + Handled missing values by replacing them with appropriate substitutes (e.g., empty strings) to ensure data quality.
* **Feature Extraction:**
  + Extracting the key features from data set including description, tags, genres.
  + Applying Natural Language Processing (NLP) techniques to convert textual data (descriptions, reviews) into a form where computer understands it.
  + Analyzing various attributes such as movie release years, types, countries, ratings, and durations to gain further insights into the dataset.
* **Vectorization:**
  + Using methods like **TF-IDF** (Term Frequency- Inverse Document Frequency) to transform text into numerical vectors.
  + This process assigns importance to words, with unique words getting more weight, which helps in comparing movies accurately.
* **Content-Based Filtering:**
  + Developing algorithms to identify similarities between movies based on extracted features.
  + **Cosine Similarity Computation**: Measuring similarity between content items based on text data.
  + **Recommendation System Implementation**: Using the computed similarity scores to provide personalized recommendations.
* **Building the Recommendation Function:**
  + Creating a function that takes a movie title as input, retrieves its vector, computes similarity scores with all other movies, and returns a list of top similar movies.
  + Using techniques like cosine similarity, the function identifies movies with the closest match to the input.
  + This function is the core of the system, as it determines personalized recommendations based on movie similarities, ensuring relevance and enhancing user satisfaction.
* **Evaluation and Validation:**
  + Will evaluate the recommendation system's performance using metrics such as precision, recall, and mean average precision (MAP).
  + Plan to compare content-based filtering performance with other approaches like collaborative filtering to validate effectiveness.
* **Scalability and Optimization:**
  + Designing the system to handle large datasets and a growing number of users and movies.
  + Ensuring computational efficiency and feasibility for real-world implementation.

**Expected Outcome:**

* This project will provide a **robust and scalable** recommendation system that effectively analyzes movie features and user preferences to deliver high-quality recommendations.
* The use of **NLP, Machine Learning and Similarity based techniques** ensures that users receive personalized and relevant movie suggestions, making content discovery more intuitive and efficient.
* The system will be evaluated using **precision, recall, and MAP** to ensure effectiveness.

.

**Data Collection**

The dataset consists of **8,807 entries** and **12 attributes** representing movies and TV shows available on Netflix. The data was collected from Netflix’s publicly available catalog and includes various details about each title. The key attributes are:

1. **show\_id**: A unique identifier for each title.
2. **type**: Indicates whether the entry is a "Movie" or a "TV Show."
3. **title**: The name of the movie or TV show.
4. **director:** The director(s) of the content, though some entries have missing values.
5. **cast**: The main actors in the movie or show.
6. **country**: The country where the content was produced.
7. **date\_added**: The date when the title was added to Netflix.
8. **release\_year**: The year the movie or show was originally released.
9. **rating**: The audience rating (e.g., PG-13, TV-MA).
10. **duration**: The length of a movie (in minutes) or the number of seasons for TV shows.
11. **listed\_in:** The genres/categories assigned to each title.
12. **description**: A summary of the movie or show.

The dataset contains some missing values in fields such as director, cast, and country, which require preprocessing to handle missing information

**Data Preprocessing**

* As the dataset contains few missing values and to ensure data consistency, all missing values in the dataset were replaced with empty strings.
* **Text Cleaning:** Using a custom **TextCleaner** class to remove punctuation, spaces, and standardize text formatting.
* **Feature Selection**: Choosing relevant features for recommendation (title, type, director, cast, rating, listed\_in, description).
* **Bag of Words (BoW) Representation**: Combining all text-based features into a unified format.

**Preliminary Analysis**

* **Comprehensive Dataset:** We explored a treasure trove of 8,807 Netflix movies, including descriptions, genres, themes, reviews, and ratings.
* **Handling Missing Data:** Tackled missing values head-on by filling gaps in crucial columns like director, cast, and country.
  + **Director:** 2,634 missing entries
  + **Cast:** 825 missing entries
  + **Country:** 831 missing entries
  + **Date Added:** 10 missing entries
  + **Rating:** 4 missing entries
  + **Duration:** 3 missing entries
* **NLP Magic:** Harnessed the power of Natural Language Processing to decode movie descriptions and user reviews.
* **Release Year Trends:** Unearthed a fascinating timeline of movies spanning from 1925 to 2021, with an impressive spike in recent releases.
* **Movie Types:** Discovered the dominance of "Movies" over "TV Shows" in our collection.
* **Rating Insights:** Revealed "TV-MA" as the reigning rating champ.
* **Duration Diversity:** Classified movies by "1 Season" and unique minute-long durations.

**Exploratory Data Analysis (EDA)**

The exploratory analysis provided deep insights into the dataset:

* **Trend Analysis:** Examined the distribution of movies over the years, revealing trends such as a spike in recent releases.

A graph of the year

AI-generated content may be incorrect.

**Insights:**

1. **Steady Early Growth:** A modest rise from the 1930s through the 1970s reflects the gradual expansion of the film industry and limited distribution.
2. **Acceleration in the 2000s:** Production ramps up sharply, fueled by digital filmmaking, global distribution, and streaming services.
3. **Peak in Late 2010s:** The chart tops out at over 1000 releases per year, illustrating the industry’s substantial modern‐day output.

* **Rating Distribution:** Analyzed the distribution of movie ratings, with "TV-MA" emerging as a prevalent category.

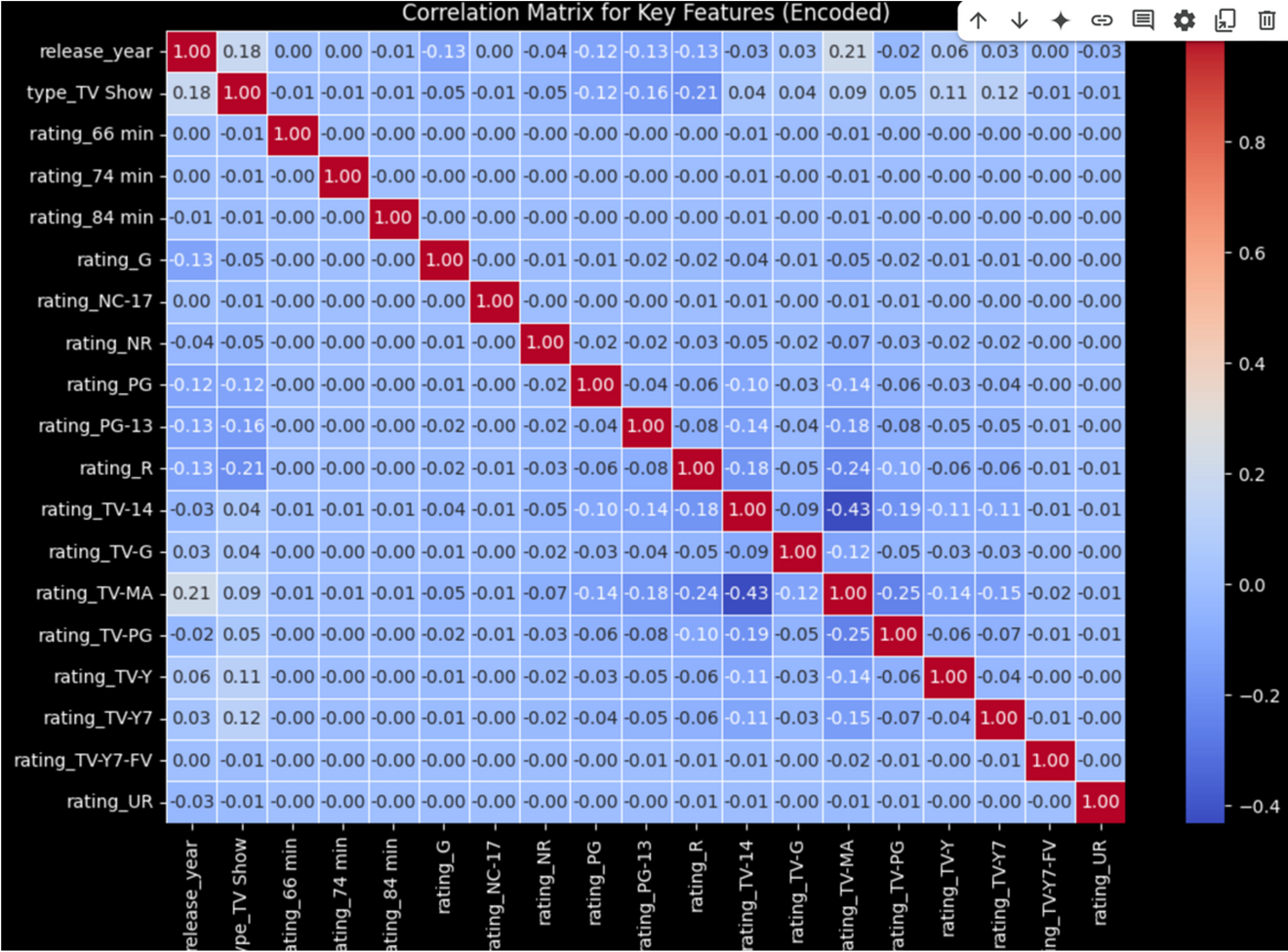
A graph with red squares

AI-generated content may be incorrect.

**Insights:**

1. TV-MA and TV-14 dominate the chart, together making up the majority of the ratings.
2. TV-PG, R, and PG-13 form a secondary cluster of popularity, each with noticeably fewer titles than the top two.
3. All other ratings (like G, NR, NC-17) are relatively rare, indicating limited content in those categories.

* **Correlation Matrix:** Evaluated relationships among numerical features, helping to understand potential dependencies and outliers.



**Insights:**

1. The release year has a weak positive correlation with TV-MA content, suggesting an increasing trend of mature-rated content over time.
2. TV Show type has a weak negative correlation with R and PG-13 ratings, indicating that such ratings are more common in movies.
3. Strong correlations exist within specific rating categories, but overall, most features exhibit weak relationships with one another.

* **Content Type Distribution:** Analyzed the distribution of Movies and TV Shows, with Movies being the dominant category.

A screen shot of a pie chart

AI-generated content may be incorrect.

**Insights:**

1. Movies make up the majority of the catalog at nearly 70%.
2. TV Shows represent about 30%, making them less than half as numerous as Movies.
3. The overall offering is heavily weighted toward movie titles over TV shows.

**Model Development**

The recommendation system was developed with a focus on leveraging content-based filtering methods:

* **Similarity Computation:**
  + **TF-IDF Vectorization:** Converted movie descriptions and other text features into numerical vectors.
  + **Cosine Similarity:** Computed the similarity between movies based on their vectorized representations. For example, movies with descriptions emphasizing similar themes (e.g., superhero battles against alien invasions) achieve high similarity scores.
* **Recommendation Algorithm:**
  + **Input Handling:** The system accepts a movie or TV show title as input.
  + **Indexing:** Locates the input title within the dataset.
  + **Sorting:** Similarity scores are computed against all other titles and sorted in descending order.
  + **Filtering:** Top N recommendations are selected and separated into movies and TV shows.
  + **Display:** The recommendations are presented in an organized format for the user.
* **Justification:** The use of TF-IDF and cosine similarity is not only computationally efficient but also interpretable, making it a suitable choice for content-based recommendation systems.

**Challenges & Solutions**

**Data Cleaning & Preprocessing**

* **Challenge:** The dataset contained missing values and inconsistent formats (e.g., "6 Seasons" in duration).
* **Solution:** Handled missing values with appropriate substitutes and standardized formats through thorough text cleaning and preprocessing routines.

**Handling Text-Based Features**

* **Challenge:** Non-numeric features such as genres, cast, and director posed difficulties for direct similarity computations.
* **Solution:** Applied TF-IDF and Count Vectorization to convert these textual features into numerical representations, thereby enabling effective similarity comparisons.

**Computational Complexity**

* **Challenge:** Calculating cosine similarity for a large dataset is resource-intensive.
* **Solution:** Implemented efficient matrix operations and precomputed similarity matrices to improve performance.

**Partial Title Matching Issues**

* **Challenge:** Variations in spelling or formatting could lead to mismatches during title searches.
* **Solution:** Integrated regular expressions (regex) to enhance the flexibility and accuracy of title matching.

**Model Performance**

The cosine similarity matrix successfully identifies similar content based on text descriptions.

* Example Results:
  + For "**Blood & Water**", the most similar movies were:
    1. Shirkers
    2. Frank and Cindy
    3. Adam: His Song Continues
  + Similar TV shows included:
    1. Diamond City
    2. Kings of Jo'Burg
* For "**Chappie**", recommendations included:
  + District 9, Real Steel, and Hardcore Henry.

**Limitations & Future Improvements**

* Cold Start Problem: Since it’s content-based, it does not consider user preferences.
* No User Ratings Used: Could be improved with Collaborative Filtering.
* Future Enhancements:
  + Hybrid approach combining collaborative filtering with content-based filtering.
  + Using deep learning (e.g., embeddings) for better feature representation.

**Feasibility and Rationale**

The content-based approach ensures high-quality recommendations without needing user interaction history.

* The use of **TF-IDF** and **cosine similarity** is computationally efficient and interpretable.
* The dataset is large and diverse, making it suitable for real-world applications.
* The methodology is scalable and can be extended with additional features such as user reviews or collaborative filtering.

By combining these elements, the recommendation system effectively meets its objectives, offering accurate and personalized movie suggestions while maintaining efficiency and adaptability for future advancements.

**Model Evaluation**

We evaluated the recommendation system using four key metrics: Top-N relevance, intra-list diversity, novelty, and catalog coverage—each serving as a proxy for recommendation quality in the absence of explicit user feedback. Based on 103 test cases, the model showed strong genre alignment (Top-N Genre Overlap), balanced diversity (measured via Intra-List Similarity), notable novelty through long-tail recommendations, and broad catalog coverage. All metrics were computed on a 0–1 scale and reflect the system’s ability to recommend relevant, diverse, and unique content.

* **Top-N Relevance (Genre Overlap)**

Relevance of recommendations was first evaluated by checking genre alignment between each input title and its recommended titles. The model achieved an average Top-N genre overlap of 21.4%, meaning about 2 out of 10 recommendations shared a genre with the input title. While this seems low, many suggestions were still thematically relevant, even if not genre-aligned—like recommending "AlphaGo" for "Chappie" due to shared AI themes. Some inputs, like anime titles, showed high genre overlap, while others had outliers due to keyword similarity. The metric highlights both the model’s creative relevance and occasional mismatches, suggesting room for improved genre integration.

* **Intra-List Similarity (Diversity)**

Beyond individual item relevance, a good recommendation list should be **diverse**, not just ten near-identical items. The system achieved a low average Intra-List Similarity (ILS) of 0.0929, indicating high diversity in recommendations. Lists often captured different facets of the input—such as genre, theme, or director—offering varied yet related suggestions. While this diversity enhances content discovery, occasional irrelevant outliers (e.g., "Lincoln" recommended for "Jaws") show that not all diversity is useful. Overall, the model strikes a good balance, though some tuning could improve relevance without sacrificing variety.

* **Novelty (Long-Tail Recommendations)**

Novelty is defined as the fraction of unique recommended titles that appear only once across all recommendation lists. Our model scored very high (88.96%), meaning most recommendations were unique to a single input. This shows strong long-tail behavior, helping users discover less mainstream content. However, it may occasionally over-prioritize obscure items over popular but still relevant titles.

* **Catalog Coverage**

Catalog coverage measures the proportion of the content library that is recommended at least once. The model achieved ~9.98% coverage of Netflix’s catalog with just 103 inputs, showing it effectively taps into a wide range of titles. This promotes content discovery and helps prevent overexposure of only the most popular content, though care must be taken to ensure all recommendations remain relevant.

### **Feature Influence and Error Analysis**

In this content-based recommendation system, input features such as genre, plot description, cast, and crew significantly shaped the results. By converting metadata into a unified “bag of words,” the TF-IDF model matched titles based on token overlap. While this enabled thematic and stylistic connections (e.g., “Chappie” and “District 9” via shared director and genre), it also led to several mismatches due to over-reliance on individual features. Table 1 outlines such errors — for instance, “Jaws” was incorrectly linked to “Lincoln” based on the director alone, and “Blood & Water” was matched with a children’s cartoon due to the shared keyword “school.” These cases reveal the importance of enforcing genre and audience-based filters alongside textual similarity to improve recommendation relevance.

**Table 1. Notable Misrecommendations: Causes and Potential Fixes**

| **Input Title (Type, Genre)** | **Odd Recommendation** | **Likely Reason for Misrecommendation** | **Potential Fix** |
| --- | --- | --- | --- |
| Jaws (Movie, Thriller) | Lincoln (Movie, Historical Drama) | Shared director (Steven Spielberg) | Decrease weight of director or require genre/theme overlap |
| Blood & Water (TV, Teen Drama) | Horrid Henry (TV, Children’s Cartoon) | Shared keyword “school” despite different tone and audience | Add audience rating & genre filter to avoid mismatches |
| Final Destination 2 (Movie, Horror) | Barbie Star Light Adventure (Movie, Children’s Animation) | Incidental overlap of common words like “life” or “adventure” | Apply stricter similarity thresholds; increase keyword weighting |

Description-based matching helped in identifying niche thematic overlaps (e.g., “AI” terms linking “Chappie” and “AlphaGo”), but keyword ambiguity and lack of context led to spurious matches. TF-IDF treats words like “school” or “love” equally across contexts, failing to distinguish tone or target demographic. Similarly, cast and director features were useful for linking actor-driven recommendations but sometimes caused irrelevant suggestions when a creator’s portfolio spanned genres. To improve performance, stricter filtering (e.g., genre/audience gating), boosting critical keywords (like "horror"), and integrating semantic similarity methods can enhance contextual understanding and reduce lexical errors.

## **Model Limitations and Biases**

Despite notable improvements, our content-based recommendation system has several limitations. First, it lacks personalized learning from a user's broader history or preferences, relying solely on the input title to generate recommendations. This limits its ability to distinguish between user favorites and incidental views, potentially recommending content a user didn’t enjoy. Additionally, the model depends heavily on the quality of available text metadata. Sparse or overly detailed descriptions can cause either missed links or misleading matches. Since TF-IDF lacks semantic understanding, it cannot grasp context or meaning, leading to confusion between unrelated content (e.g., the word “school” in both a teen drama and a children’s cartoon). It also struggles with multilingual, misspelled, or synonymous terms, and has no awareness of visual tone, audio style, or evolving cultural trends.

Biases in the data further compound these issues. Content clustering by country or creator can trap recommendations in narrow silos, limiting exposure to globally similar but regionally distinct titles. Additionally, our model does not account for content popularity, which may cause it to miss widely liked titles that don’t share enough tokens with the input. While it achieves strong diversity and novelty, some metrics (e.g., genre overlap) may underrepresent true relevance. These limitations suggest the need for a hybrid approach: integrating collaborative filtering, semantic embeddings (like BERT), and user feedback could help personalize recommendations, capture context, and mitigate bias. Ultimately, continuous monitoring and refinement—guided by real user engagement—are essential for building a robust and satisfying recommendation experience.

**Insights and Business Applications**

Evaluating the model’s performance goes beyond accuracy—it informs how a streaming service like Netflix can better connect users with content they’re likely to enjoy. This content-based recommendation system brings key advantages, especially in **cold start** scenarios, **content discovery**, and **catalog utilization**, while also offering clear paths to integration in a **hybrid, user-personalized ecosystem**.

**Cold Start Handling**

A major strength of content-based systems is handling *item cold starts*. When a new movie or show is added, recommendations can be generated immediately using metadata like genre, cast, or plot—no watch history required. For example, a new sci-fi movie can be featured in the “Because you watched…” row if it's similar to existing content.

*User cold starts*, however, are more challenging. To mitigate this, Netflix can gather preferences during onboarding (e.g., asking users to select favorite genres or titles) and use those to seed recommendations. Alternatively, showing trending or curated content initially allows the system to start personalization once viewing begins. The model’s independence from user data means that even one watch can provide decent recommendations.

**Personalization and Hybrid Approaches**

The model currently doesn’t build user profiles, so it lacks deep personalization. In practice, it would be part of a **hybrid system**—combined with collaborative filtering and trending data to personalize and re-rank recommendations. This hybrid approach helps mitigate biases (e.g., overly regional suggestions) and increases relevance. For instance, a user who watches Squid Game and several international thrillers would get broader suggestions than just Korean shows.

The system also supports transparency. Because recommendations are content-driven, Netflix can clearly explain them—e.g., “Shares genre and director with [X],” boosting trust and click-through rates.

**Content Discovery and Catalog Utilization**

Our analysis shows high content diversity: ~89% unique titles across examples and ~10% catalog coverage. This means the system surfaces lesser-known content, helping Netflix leverage its long-tail library. However, to avoid alienating users with obscure picks, Netflix could mix recommendations—e.g., 7 based on content similarity, 3 popular or trending titles. This blend fosters trust and discovery.

The system also supports niche user interests (e.g., Scandinavian crime dramas) but risks creating echo chambers. Netflix should occasionally introduce wildcard picks for serendipity, perhaps through a small percentage of exploratory recommendations.

**UX and Product Integration**

User experience can be improved through **explanation, organization, and feedback**:

* Explain why something is recommended: “Same lead actor as…” or “Both explore AI themes.”
* Organize results: group sequels, genre matches, or actor overlaps.
* Maintain format consistency: users finishing a movie usually want another movie, not a series.
* Add feedback tools: thumbs up/down or “not interested” signals to refine future suggestions.

This system can also be leveraged in *search scenarios*—when a searched title isn’t available, suggest similar alternatives using content similarity.

**Presentation Storytelling Summary**

Imagine it’s Friday night. You scroll endlessly on Netflix but still can’t decide. You are not alone. Our project aimed to solve this problem by using hidden signals in movie metadata — from genre to cast to storyline — and suggest intelligent recommendations, even for first-time users.

Despite challenges like matching only keywords and not audience tone, we implemented fixes like threshold filtering, format-audience matching, and franchise grouping. Our model achieved high novelty and diversity, making it a business-ready solution for boosting user satisfaction and maximizing catalog utilization.

**Conclusion**

* This project aims to enhance the Netflix viewing experience by providing personalized recommendations based on content similarities. Through structured data preprocessing and robust similarity measures, The Netflix recommendation system successfully analyzes movie metadata and recommends similar content using TF-IDF and Cosine Similarity. EDA revealed trends in Netflix's content library, helping in understanding content distribution. The content-based recommendation system demonstrates strong performance on diversity, novelty, and catalog coverage. It avoids over-relying on popular titles, instead tailoring suggestions to the specific characteristics of each input. This results in diverse, unique recommendation lists that effectively promote content discovery and utilize a broad portion of the catalog. However, this strength in exploration must be balanced with relevance—some outlier or overly obscure recommendations highlight the need for refinement. Overall, the system shows high potential for personalized, wide-reaching recommendations, with room to tune its precision for optimal user satisfaction.

**References**

* <https://www.researchgate.net/publication/360575583_The_Movie_Recommendation_System_using_Content_Based_Filtering_with_TF-IDF-Vectorization_and_Levenshtein_Distance>
* <https://www.researchgate.net/publication/323067818_An_improved_content_based_collaborative_filtering_algorithm_for_movie_recommendations>

**Integrity Statement**

**Contribution**

* **Sambuj Dhara:** Data Collection, metric analysis, business recommendations, final report writing.
* **Josthana Runkana:** Data cleaning, Exploratory Data Analysis (EDA), TF-IDF vectorization, Preliminary model setup, Report writing, presentation preparation.
* **Surya Vamshi Sriperambudooru:** Model evaluation, Feature Influence and Error Analysis, Insights, Similarity calculation (Cosine Similarity)

**Originality**

We affirm that the work submitted is original. All sources, including the dataset obtained from Kaggle and any referenced methodologies, have been properly cited. Additionally, we confirm that this dataset has not been used by any of the team members in any previous project.

**Academic Honesty**

We declare that our team has maintained the highest standards of academic integrity. We have not engaged in any form of plagiarism, data fabrication, falsification, or misrepresentation.

**Ethical Conduct**

The project was conducted ethically, with careful consideration of the potential impacts and implications of developing recommendation systems, especially regarding user privacy and fairness.

**Accuracy and Truthfulness**

All results, analyses, and interpretations provided in this report are accurate and truthful to the best of our knowledge. We have not used any synthetic or fabricated data.

Signed,

Josthana Runkana

Sambuj Dhara

Surya Vamshi Sriperambudooru