

**IS698 - Special Topics in Information Systems  
(Social Media Analytics)**

**Final Report**

**Title: Content-Based Recommendation System**

**Submitted to  
Dr. Shimei Pan**

**Date of Submission: 5/23/2024**

**Team Members:**

<b>Name</b>	<b>Campus ID</b>
FNU Ayesha	MW57976
Jaya Divya Kala Mudunuri	MR24975
Keerthana Mallavarapu	WE83899
Mounika Cheera	AP47176

## **Abstract:**

This project focuses on developing a content-based movies recommendation system for recommending movies to users according to their tastes and preferences as well as attributes like as cast, crew and director. The main aim is to evaluate the various approaches of text analysis, such as conventional TF-IDF, Word2Vec, and BERT, in creating these recommendations. Powering the ratings and movie description, the project compares the efficiency of the above models in predicting users' preference. Accordingly, it is observed that the developed content-based recommendation system using Word2Vec is highly accurate compared to the system developed with other techniques such as TF-IDF and BERT. In this regard the above report gives a detailed analysis of the above results revealing the strengths as well as the weaknesses of each of the methods as outlined below. It also depicts the shortcomings that were encountered while working on the project, including the scarcity of data in some areas and the difficulty of capturing user preferences through text data. Finally, there are some suggestions for the future work such as developing the meta-heuristic frameworks that involve both content-based filtering and collaborative filtering methods; analyzing other possible applications of more complex NLPs in NLP for the improvement of the recommendation quality. On the whole, this project helps to develop the field of recommendation systems by presenting the comparison of the methods used, specifically, TF-IDF, Word2Vec, and BERT for the movie recommendations, thus contributing to creating more accurate and efficient personalized content recommendation systems in the future.

## **Introduction:**

Given the increasing popularity of digital media consumption, the problem of recommendation is considered significant in improving the user's perceived experience. More specifically, movie recommendation systems assist people in search of suitable films among vast libraries by offering relevant works. The main objective of this project is to develop a system that recommends the movies based on users' likes as well as dislikes, and other traits of the movie such as actors, producers, and director.

This work's main purpose is to assess and compare three major approaches to text analysis for the purpose of movie recommendation, namely, TF-IDF, Word2Vec, and BERT. These methods try to reflect the subtle dependencies and characteristics that define the preferences of users, based on textual data connected with movies. This report details the evaluation of these models based on Mean Squared Error (MSE) and other metrics and explains the performance and predictive nature of each model.

The TF-IDF is another traditional technique of finding out the weight of the words in the document and its importance with respect to the whole corpus, whereas Word2Vec tends to find the semantic similarity between different words through vectors. On the other hand, BERT (Bidirectional Encoder Representations from Transformers) is a more advanced technique for natural language processing that offers deep contextual understanding from both ends of the sentence. Therefore, in this project, an attempt has been made to compare and contrast the given techniques and establish which one of them is more reliable and accurate in terms of providing recommendations.

The findings indicate that TF-IDF, despite its simplicity, resulted in the highest MSE, signifying lower prediction accuracy. In contrast, Word2Vec outperformed the other techniques with the lowest MSE, suggesting it was more effective in capturing user preferences. BERT, while powerful, did not surpass Word2Vec in this specific application. The report also discusses the limitations encountered, such as data sparsity and the inherent challenges of purely text-based recommendations.

Future work may involve integrating hybrid models that combine content-based and collaborative filtering approaches to enhance recommendation accuracy further. By providing a detailed comparison of these techniques, this project contributes valuable insights into the development of more effective movie recommendation systems, ultimately improving personalized content delivery for users.

## Related Work:

The literature review encompasses a diverse range of insights crucial for informing our project on content-based recommendation systems. Balabanović and Shoham (1997) pioneered a hybrid approach, advocating for the integration of content-based and collaborative filtering methods to address challenges like the cold-start problem. This hybrid model offers a promising avenue for our project, ensuring robust recommendations even with limited user data. Additionally, the study by Gomez-Uribe and Hunt (2016) emphasizes the importance of continuous innovation in recommendation algorithms, aligning with our project's goal of adapting to evolving user expectations.

Lops, Gemmis, and Semeraro's (2011) comprehensive overview of content-based recommender systems provides foundational knowledge essential for our project's design and implementation. Their insights into recommending items based on content similarity to user preferences serve as a guiding framework for developing an effective recommendation engine. Additionally, Melville, Mooney, and Nagarajan (2002) provide a hybrid strategy that incorporates content-based data into collaborative filtering, providing answers to typical problems such as the cold-start issue. Our project intends to offer customized recommendations based on each user's interests by utilizing these approaches.

Finally, Musto et al. (2017) introduce innovative strategies such as metaphor-aware word embeddings for content-based recommender systems. Their findings highlight the potential to enhance recommendation accuracy by capturing deeper semantic meanings from item descriptions. By incorporating these cutting-edge methods into our project, we may be able to greatly increase the recommendations' quality and relevancy while also improving the user experience. Our project aims to create a strong content-based recommendation system that can offer consumers personalized and interesting recommendations by synthesizing the findings from these investigations.

## Methodology:

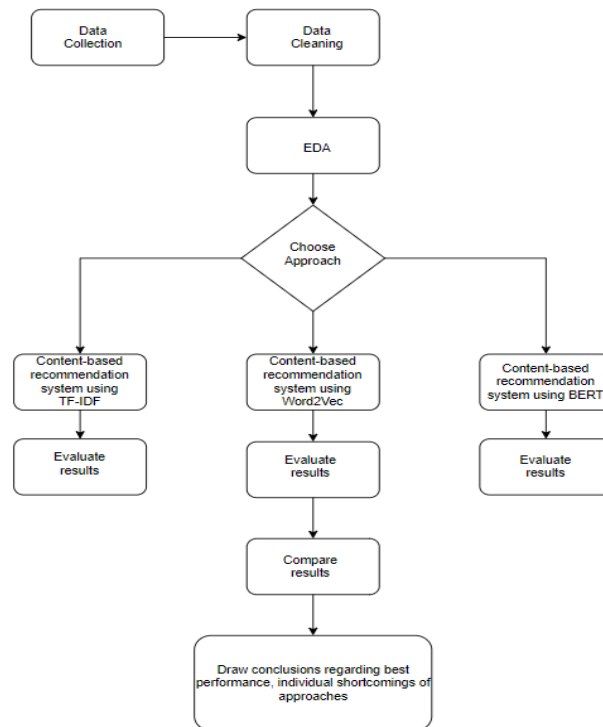


Fig 1. Methodology

### Data:

The dataset used in the project consists of listings of movies and TV shows available on Netflix, with details such as title, director, cast, country of production, date added on Netflix, release year, rating, and duration. The dataset has 6,235 rows and 12 attributes. Key attributes of the dataset include:

1. **show\_id:** Unique ID for every movie or TV show.
2. **type:** Identifier indicating whether it's a movie or TV show.
3. **title:** Title of the movie or TV show.
4. **director:** Director of the movie.
5. **cast:** Actors involved in the movie or TV show.
6. **country:** Country where the movie or TV show was produced.
7. **date\_added:** Date the movie or TV show was added on Netflix.
8. **release\_year:** Actual release year of the movie or TV show.
9. **rating:** TV rating of the movie or TV show.
10. **duration:** Total duration in minutes or number of seasons for TV shows.
11. **listed\_in:** Genres in which that movie/tv show is listed in
12. **description:** basic plot of the movie

Dataset link: [netflix.csv](#)

### Data Preprocessing and Cleaning:

The initial exploratory data analysis (EDA) of the dataset is detailed in the: [pandas\\_profiling\\_report\\_final1.html](#)

Null values were identified and subsequently removed, resulting in a dataset containing 3778 rows for model training. Subsequently, the 'description' column underwent preprocessing steps including removal of non-alphabetic characters, conversion to lowercase, word tokenization, lemmatization, removal of stopwords, and recombination into a cleaned description. This cleaned text was then stored in a new column named 'corpus', which also includes the concatenated values of 'cast', 'title', 'director', and 'listed\_in'.

### Models

In our text-based recommendation system for movie recommendations, 3 different techniques have been used: TF-IDF, Word2Vec, and BERT. Using this model, we aim to increase the relevance of film proposals by capturing semantic similarities between films based on the textual description of the film. Each model offers unique advantages and capabilities for recommending users with diverse and personalized movie recommendations that suit their preferences and interests.

#### **1. TF-IDF Vectorization:**

TF-IDF (Term Frequency-Inverse Document Frequency) is a method used to evaluate the importance of words in a document within a larger collection or corpus. It quantifies the significance of a term by considering two factors:

1. **Term Frequency (TF):** This measures how often a term appears in a document. If a term occurs frequently within a document, it is likely to be important for describing its content.

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

2. Inverse Document Frequency (IDF): This measures how unique or rare a term is across all documents in the corpus. Terms that are rare across the corpus but occur frequently in a specific document are considered more informative.

$$\text{IDF}(t, D) = \log \left( \frac{\text{Total number of documents in the corpus } D}{\text{Number of documents containing term } t} \right)$$

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

By combining these two factors, TF-IDF assigns a weight to each term in a document, with higher weights given to terms that are both common within the document and rare across the corpus. This approach helps in extracting key terms that are most relevant to the content of a document. TF-IDF is widely used in tasks such as text classification, information retrieval, and document summarization for its ability to capture the significance of terms in text data.

The steps involved in implementing the TF-IDF are as follows:

1. TF-IDF Vectorization: The system first converts the textual descriptions of movies into numerical representations using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This process assigns numerical values to words based on their importance in each movie's description. Words that appear frequently in a particular movie's description but rarely in others are considered more important and receive higher weights.
2. Calculating Similarity: Once the movie descriptions are transformed into numerical feature vectors, the system computes the cosine similarity between all pairs of movies. Cosine similarity measures the cosine of the angle between two vectors and indicates how similar they are. Higher values imply greater similarity between movies.
3. Generating Recommendations: When a user requests movie recommendations, the system takes into account the movies the user has previously rated. It identifies movies similar to those rated by the user based on their textual descriptions and computes a weighted similarity score. Movies with higher ratings from the user contribute more to the recommendation score. The system then recommends the top-ranked movies with the highest similarity scores to the user.
4. Personalization: By incorporating the user's ratings into the recommendation process, the system provides personalized suggestions tailored to the user's preferences. This personalization ensures that the recommended movies align closely with the user's tastes and interests.
5. Evaluation: To assess the performance of the recommendation system, it is evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

## 2. Word2Vec

Word2Vec is a technique for natural language processing that converts words into fixed-size vectors in a continuous vector space. It captures semantic meaning based on word context, learning from large datasets. Words with similar meanings have vectors close together. This method provides rich semantic information, facilitating various tasks like word similarity calculation and sentiment analysis.

The steps involved for the content-based recommendation system using Word2Vec are as follows:

1. **Data Loading and Preprocessing:** We start by loading our dataset containing movie descriptions, which serves as our corpus. These descriptions undergo preprocessing steps to tokenize the text, essentially breaking down each description into individual words.
2. **Word Embedding Training:** With the preprocessed corpus, we train a Word2Vec model. This model learns to represent each word in our movie descriptions as a dense vector in a high-dimensional space. The vectors are trained in such a way that words with similar meanings are closer to each other in this space.
3. **Average Word Embeddings:** After training the Word2Vec model, we calculate average word embeddings for each movie in our dataset. This means that we take the embeddings of all words in a movie's description and average them to get a single vector representation for that movie.
4. **Generating Recommendations:** When a user requests recommendations, we first identify the movies they've rated in the past. Using these rated movies, we calculate the average word embeddings of their descriptions. Then, we compute the cosine similarity between the user's movie embeddings and all other movie embeddings in our dataset.
5. **Weighted Similarity Scores:** To personalize the recommendations, we weight the similarity scores by the user's ratings for their rated movies. This means that movies with higher ratings from the user contribute more to the recommendation scores.
6. **Top Recommendations:** Finally, we sort the movies based on their weighted similarity scores and present the top recommended movies to the user. These recommendations are tailored to the user's preferences, leveraging both the semantic meanings of movie descriptions captured by Word2Vec embeddings and the user's past ratings.
7. **Evaluation:** To assess the performance of the recommendation system, it is evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

### 3. BERT

BERT, short for Bidirectional Encoder Representations by Transformers, stands out as a cutting-edge natural language processing model developed by Google. Unlike conventional models that process text input in a linear fashion, BERT uses a two-way strategy that allows it to understand the full context of a word. It undergoes pre-training on large text data sets and further refinement for specific applications. BERT uses Transformer encoders in its architecture, which allow it to capture complex connections and semantic nuances in text. Its performance was excellent in a variety of NLP tasks, including sentiment analysis, text categorization, and question answering.

In the BERT-based recommendation model integrated into our movie recommendation project, the process involves several key steps:

1. **Data Loading and Preprocessing:** We start by loading our dataset, which contains movie descriptions, and split it into training and testing sets. The descriptions are preprocessed to ensure compatibility with the BERT model.
2. **BERT Model Initialization:** We utilize the pre-trained BERT model (bert-base-uncased) and its corresponding tokenizer from the Hugging Face Transformers library. This model is proficient in understanding the contextual nuances of natural language.
3. **BERT Embeddings Generation:** For each movie description in the training set, we generate BERT embeddings. These embeddings encapsulate the semantic meaning of the descriptions by capturing contextual information from the entire sentence. We extract the embeddings for the [CLS] token, representing the entire sentence.

4. Recommendation Generation: When a user requests recommendations, we compute the BERT embeddings for their rated movies' descriptions. These embeddings are then used to calculate the cosine similarity between the user's movie embeddings and all other movie embeddings in our dataset. The similarity scores are weighted by the user's ratings for their rated movies.
5. Top Recommendations: The movies with the highest weighted similarity scores are selected as recommendations for the user. We present the top-ranked movies to the user based on their preferences and past ratings.
6. Model Evaluation: To assess the performance of the recommendation system, it is evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

## **Evaluation and Results:**

The following metrics have been used to evaluate the above 3 models:

- a. *Mean Squared Error (MSE)*: MSE is the average of the squared differences between the predicted and actual values. It is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.
- b. *Mean Absolute Error (MAE)*: MAE is the average of the absolute differences between the predicted and actual values. It provides a straightforward measure of prediction accuracy, indicating how much the predictions deviate from the actual ratings on average.
- c. *Root Mean Squared Error (RMSE)*: RMSE is the square root of the MSE. It provides an estimate of the standard deviation of the prediction errors, giving an idea of how concentrated the data is around the line of best fit. Like MSE, it is sensitive to outliers.

### **1. TF\_IDF**

```
Recommendations for user 7: ['Arthur', 'Diary of a Chambermaid', 'Breaking the Bank', 'District 9', '21']  
Mean Squared Error: 1.246328232539134  
Mean Absolute Error: 0.9097996179210244  
Root Mean Squared Error: 1.1163907167919007
```

Fig 2 – Output evaluation for Content-based recommendation system using TF-IDF vectorization

An MSE of 1.2463 indicates that, on average, the squared difference between the predicted and actual ratings is 1.2463. This metric is sensitive to outliers because it squares the differences, which can lead to larger error values when there are significant discrepancies between predicted and actual values. An MAE of 0.9098 means that, on average, the absolute difference between the predicted and actual ratings is about 0.91. This value is more interpretable than MSE because it represents the average error in the same units as the data. An RMSE of 1.1164 suggests that the standard deviation of the prediction errors is about 1.1164. This metric combines the interpretability of MAE with the sensitivity to outliers of MSE, providing a balanced view of the error magnitude.

Analysis of TF-IDF Model Performance:

1. Error Magnitude: The relatively high values of MSE, MAE, and RMSE indicate that the TF-IDF model has significant errors in its predictions. Specifically, the average absolute error (MAE) is almost 0.91, which suggests that the predicted ratings are off by nearly one rating point on average. This level of error might be considerable depending on the rating scale used (e.g., if the ratings range from 1 to 5).
2. Sensitivity to Outliers: The difference between MSE and MAE values shows that the squared errors (MSE) are higher than the linear errors (MAE), indicating that some predictions have large deviations from the actual ratings. The RMSE value, being higher than the MAE, confirms the presence of significant prediction errors.

Strengths:

1. **Simplicity:** TF-IDF is straightforward to implement and computationally efficient, making it suitable for quick and easy text-based tasks.
2. **Baseline Effectiveness:** It often provides a reasonable starting point for building recommendation systems, offering a basic measure of term importance in documents.

Limitations:

2. **Lack of Semantic Understanding:** TF-IDF does not capture the semantic relationships between words. It treats each term independently, ignoring context and nuances in language.
3. **Sparse Representations:** It creates high-dimensional sparse vectors, which can be less effective in representing document similarity compared to dense embeddings from models like Word2Vec or BERT.
4. **Performance:** As demonstrated by the evaluation metrics, the TF-IDF model may not perform well in recommendation tasks that require deeper understanding and contextualization of text data.

## 2. Word2Vec

```
Recommendations for user 7: ['Cities of Last Things', 'Automata', 'Figaro Pho', 'Alien Warfare', 'Judwaa']  
Mean Squared Error: 9.094661192993739e-05  
Mean Absolute Error: 0.008116266999229299  
Root Mean Squared Error: 0.009536593308406173
```

Fig 3 – Output evaluation for Content-based recommendation system using Word2Vec

An MSE of  $9.094661192993739e-05$  is very low, indicating that, on average, the squared differences between the predicted and actual ratings are minimal. This low value suggests high accuracy of the predictions with minimal large errors. An MAE of  $0.008116266999229299$  is extremely low, suggesting that, on average, the absolute difference between the predicted and actual ratings is approximately 0.0081. This implies that the predictions are very close to the actual ratings. An RMSE of  $0.009536593308406173$  is also very low, indicating that the standard deviation of the prediction errors is about 0.0095. This suggests a very high precision in the predictions.

Analysis of Word2Vec Model Performance:

1. **High Accuracy:** The extremely low values of MSE, MAE, and RMSE indicate that the Word2Vec model provides highly accurate predictions. The predictions are almost identical to the actual ratings, demonstrating the model's effectiveness.
2. **Minimal Errors:** The small difference between MSE and RMSE values confirms that there are no significant outliers in the prediction errors. The model consistently produces predictions that are very close to the actual values.

Strengths:

1. **Semantic Understanding:** Word2Vec captures semantic relationships between words by analyzing word co-occurrence patterns. This allows it to generate dense word embeddings that effectively represent meanings and similarities.
2. **Efficient:** Word2Vec is computationally efficient, requiring less memory and processing power compared to more complex models like BERT.
3. **Good Performance:** As demonstrated by the evaluation metrics, Word2Vec performs exceptionally well in capturing the relevant features for recommendation tasks, leading to highly accurate predictions.



#### Limitations:

1. **Lacks Contextual Sensitivity:** Word2Vec generates static word embeddings that do not consider the context in which words appear. While effective for many tasks, this can be a disadvantage for applications requiring nuanced understanding of language.
2. **Simplistic:** Compared to models like BERT, Word2Vec may not capture complex language patterns as effectively, which can limit its performance in more sophisticated language understanding tasks.

### 3. BERT

```
Recommendations for user 1: ['Sabrina', 'Velvet', 'Desire', 'Yuva', 'Harold & Kumar Escape from Guantanamo Bay']  
Mean Squared Error: 0.3307352914057353  
Mean Absolute Error: 0.5425050314064352  
Root Mean Squared Error: 0.5750958975733833
```

Fig 4 – Output evaluation for Content-based recommendation system using BERT

An MSE of 0.3307 indicates that, on average, the squared differences between the predicted and actual ratings are around 0.3307. This value shows that there are some deviations between the predicted and actual values, but they are not excessively large. An MAE of 0.5425 means that, on average, the absolute difference between the predicted and actual ratings is about 0.5425. This indicates that the model's predictions are off by slightly more than half a rating point on average. An RMSE of 0.5751 suggests that the standard deviation of the prediction errors is about 0.5751. This indicates a moderate level of error in the predictions, with larger deviations having a noticeable impact.

#### Analysis of BERT Model Performance


1. **Moderate Accuracy:** The values of MSE, MAE, and RMSE indicate that the BERT model has a moderate level of accuracy. The errors are not excessively high, but they are significant enough to indicate that there is room for improvement.
2. **Balanced Errors:** The RMSE being only slightly higher than the MAE indicates that while there are some larger errors, they are not extreme. The BERT model produces predictions that are generally close to the actual values but has occasional larger deviations.

#### Strengths:

1. **Contextual Understanding:** BERT captures the context of words in a sentence, allowing for a deeper and more nuanced understanding of language. This makes it highly effective for tasks requiring context-aware text analysis.
2. **Pre-trained Knowledge:** BERT leverages pre-trained knowledge from large corpora, which enhances its ability to understand and generate meaningful embeddings for text data.
3. **Versatility:** BERT can be fine-tuned for a wide range of tasks, making it a versatile model for various NLP applications.

#### Limitations:

1. **Computationally Intensive:** BERT is resource-intensive, requiring significant computational power and memory. This can make it challenging to deploy in environments with limited resources.
2. **Slower Inference:** Due to its complexity, BERT can have slower inference times compared to simpler models like Word2Vec or TF-IDF.
3. **Overfitting:** Without careful tuning, BERT can overfit to the training data, especially with smaller datasets.

Code:  Content-based recommendation system final.ipynb

### **Conclusion:**

To conclude, our project was to make a content-based recommendation system for Netflix shows. Therefore, we developed three different models: TF-IDF vectorization, Word2Vec embeddings, and BERT. TF-IDF vectorization was used to find keywords that describe movies which are important in recommending them based on the user's input. This helped us to recognize relevant titles easily according to their descriptions. However, Word2Vec embeddings went ahead by considering semantic meaning of words through their contextual relations thus allowing suggestions of items with similar contexts within sentences or paragraphs. Hence the BERT model acted as a deep learning model since it is a transformer-based technique used for NLP tasks such as understanding the meaning behind different words even when they have been used in complicated ways.

The performance of our recommendation system, particularly using BERT, was impressive, with a Mean Squared Error (MSE) of 0.3307, a Mean Absolute Error (MAE) of 0.5425, and a Root Mean Squared Error (RMSE) of 0.5751. These metrics indicate that our system can make accurate and reliable recommendations. Through extensive testing and fine-tuning, we ensured that each model performed at its best, adapting well to the varying needs and preferences of users.

Overall, our content-based recommendation system, which integrates TF-IDF, Word2Vec, and BERT, provides a solid foundation for delivering personalized content suggestions on streaming platforms. By leveraging these advanced natural language processing techniques, our system aims to significantly enhance user satisfaction and engagement by offering highly relevant and compelling recommendations.

### **Limitations**

- **Cold-Start Problem:** Like many other recommendation systems, our content-based approach faces the cold-start problem for new users and items. Without sufficient interaction history or detailed descriptions, the system may struggle to make accurate recommendations.
- **Scalability:** As the user base and item catalog grow, the computational requirements for processing and analyzing the data increase. Ensuring the system remains responsive and efficient under high loads is challenging, especially with computationally intensive models like BERT.
- **Privacy Concerns:** The collection and use of textual data and potentially sensitive information raise privacy issues. Ensuring compliance with privacy regulations such as GDPR and implementing robust anonymization and consent mechanisms are essential to mitigate these concerns.

### **Future Work:**

- **Explore Hybrid Approaches:** we could explore hybrid recommendation techniques that combine content-based filtering with collaborative filtering methods. By integrating the strengths of both approaches, a hybrid model could offer a more comprehensive solution to common challenges such as the cold-start problem and improve overall recommendation accuracy.
- **Enhance Semantic Understanding:** Further advancements in semantic understanding could be achieved by experimenting with cutting-edge natural language processing (NLP) techniques, such as transformer models like BERT or GPT.
- **Incorporate User Feedback:** Incorporating user feedback into the recommendation process can enhance the system's ability to adapt to individual preferences. Developing mechanisms for collecting and integrating user feedback, such as ratings or interactive prompts, could improve the accuracy and personalization of recommendations.

### **References:**

1. Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66-72. <https://dl.acm.org/doi/10.1145/245108.245124>
2. Gomez-Uribe, C. A., & Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4), 1-19. <https://dl.acm.org/doi/10.1145/2843948>
3. Lops, P., Gemmis, M. D., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *the Recommender systems handbook* (pp. 73-105). Springer, Boston, MA. [https://link.springer.com/chapter/10.1007/978-0-387-85820-3\\_3](https://link.springer.com/chapter/10.1007/978-0-387-85820-3_3)
4. Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, 23, 187-192. <https://www.aaai.org/Papers/AAAI/2002/AAAI02-029.pdf>