

**Name of Faculty: Dr Sumit Kumar**

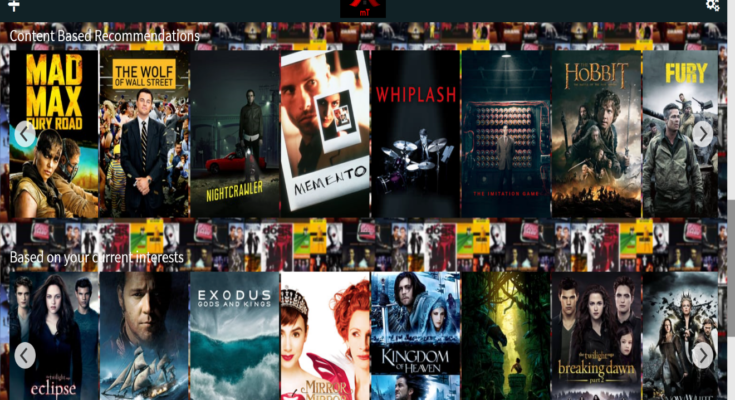
**Student Name: Survepalli Sreekruti**

**Roll No: A2305220266**

**AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY**

**AMITY UNIVERSITY UTTAR PRADESH**

**NOIDA**



**Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method**



**Declaration**

I, **Survepalli Sreekruti**, student of B.Tech (6CSE1-X) hereby declare that the research paper on **“*Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method*”**,which is submitted by me to **Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University, Noida, Uttar Pradesh**, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, & has not been previously submitted for the basis of the award in any other degree, diploma or any other recognition or title

DATE:

Survepalli Sreekruti

A2305220266

5CSE1X (2020-24)

**certificate**

This is to certify that **Ms. Survepalli Sreekruti**, student of B. Tech in Computer Science & Engineering, has carried out work on project entitled “***Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method***” as a part of her 6th year Program of Bachelor of Technology in Computer Science & Engineering from Amity University Noida,under my supervision.

I Dr. Sumit Kumar

iDepartment of Computer Science & iEngineering

iASET

iNoida, Uttar Pradesh

**ACHNOWLEDGEMENT**

The completion of this project would be incomplete without the mention of people whose constant guidance has to be awarded with my success.

I would like to express my deep and sincere gratitude to my research supervisor, **Dr Sumit Kumar,** for giving me the opportunity to do research on topic **“*Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method*”** and providing invaluable guidance throughout this research. It was a great privilege and honour to work and study under her guidance. I would like to thank her for solving my queries and also guide me through out.

**index**

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **CONTENT** | **PAGE NO.** |
| 1. | **INTRODUCTION**   * 1. Abstract   2. Introduction |  |
| 2. | **CONCEPTUAL FRAMEWORK**  2.1 Literature Survey/Related Work |  |
| 3. | **METHODOLOGY**  3.1 Methods Used / Proposed System  3.2 Architecture  3.3 Analysis Walkthrough |  |
| 4. | **RESULTS** 4.1 Results4.2 Limitations4.3 Future Work Conclusion , Recommendations and  Bibliography  Conclusion , Recommendations and  Bibliography  Conclusion , Recommendations and  Bibliography |  |
| 5. | **CONCLUSION & REFERENCES**  5.1 Conclusion  5.2 References |  |

**Introduction**

* 1. **ABSTRACT**

Engines that generate recommendations for users are trained to do so quickly and logically. The most well-known on-demand streaming service nowadays is Netflix. Its service, which offers films and TV episodes, is accessible in 190 countries. Data from Kaggle, a search engine that lists the Netflix content available, was used in our study to conduct an exploratory analysis.

The system's benefits include accurate suggestions even with a tiny data model and efficient recommendations. Future improvements will include user documentation and profiling, analytics reporting for users and producers, and web scraping for data collection.

For making ideas for items to buy or see, a recommendation system is being used. To direct customers to the items that can satisfy their needs, they search through a vast database of information. A data filtering system called a recommender system, sometimes referred to as a recommendation engine or platform, makes an effort to predict a user's "rating" or "preference" for a given item. They are generally used for commercial purposes. This project offers a technique for giving viewers general choices depending on a movie's popularity or theme.

The TF-IDF and Cosine similarity algorithms, which are popular models in Natural Language Processing (NLP), were also employed to construct a recommendation system. Interesting information about the present tendencies of Netflix content delivery was found by the exploratory analysis. Despite the recommendation system's limitations as it stands right now, it appears promising when further features are taken into account.

* 1. **INTRODUCTION**

A recommendation engine makes suggestions to consumers for products or objects found in a group of data based on their likes and interests. It is a highly common and important component of any business website for making product recommendations to the website's visitors. It enables the user to have adequate knowledge of a website's contents and enables them to make wise selections. Producers of any e-commerce website employ meta-data and data analytics in the recommendation engine to monitor user market performance, demands, and interests. A recommendation engine can be implemented in many different ways. The current systems' shortcomings include a lack of data, cold start issues, changing data, and user preferences.

The engines employed by global tech juggernauts like YouTube, Netflix, Amazon, and Pandora are a few instances of extremely effective recommendation systems. Massive volumes of data are processed by these systems at different phases, including training stages.

Since the beginning of civilization, people have relied on recommendations for all significant and minor decisions. When a skilled person and/or more than two or three other people propose the same thing, the person will be able to change their opinion (recommended). In the modern internet era, recommendation algorithms emerged and continued to support the original idea. Programmes known as recommendation systems provide end users with recommendations based on their preferences or the preferences of users with similar preferences. The two main categories of this recommendation system are: collaborative filtering recommendation systems and content-based filtering recommendation systems. Each of those categories will be examined in the following sections.

Similarity measurements support these classifications, but we have advanced to more sophisticated techniques like machine learning algorithms. As a result of the recommender system's positive results in e-commerce, film, music, books, and news suggestions, it has now expanded to other sectors like tourism and banking.The term "recommender system," sometimes known as "recommendation system," refers to a method for filtering data that aims to predict a user's "rating" or "preference" for a given item. The user receives recommendations or ideas following the creation of a forecast that are backed by the outcomes of the forecasts. There are many different types of recommender systems, and not every one of them is suitable for every issue and situation.

On the other hand, we have created a similarity analysis-based recommendation system utilising the algorithms Term Frequency - Inverse Document Frequency and Cosine Similarity. The title and description of the work make up the entirety of the initially investigated textual corpus. However, we intend to enhance our approach by taking into account additional determining factors, in this case, the user demographics. Our method is unique in that it is straightforward and only requires a small amount of data for training, allowing for quick implementation and use.

**CONCEPTUAL FRAMEWORK**

* 1. **LITERATURE SURVEY/RELATED WORK**

1. The QoS attribute value-based collaborative filtering service suggestion by **Xiaokun Wu et al**. contains two critical processes. One is the computation of similarity, and the other is the prediction of the value of the QoS feature, which the user has not yet encountered.
2. In their paper, **Xin Guan et al** provided a summary of the collaborative algorithms used to anticipate which additional products current users could find appealing based on their past preferences for particular products.
3. **Songtao Shanget al** discussed about the slope 1 algorithm to recommend the items to users by comparing the items.
4. A recommendation system proposed by **Gilda Moradi Dakhel** and **Mehregan Mahdavi** groups users based on their interests and likes and suggests films that are similar to those groups of users. This system uses the K-means clustering algorithm.
5. In their study "A generic hybrid recommender system based on neural networks," **Anant Gupta** and **Dr. B.K**. Tripathy outline a cognitive approach to recommendation engines.
6. Instead of the conventional hybrid-based technique, this system was developed by **Abhishek Singh, Samyak Jain, J Shanmukh Rao, Uppalpati Yogendra Reddy**, and **Abhishek Rawat** using technologies like matrix factorization and recall algorithms.

To train an emotional model that can convert a review that is in the form of text into a vector file and ascertain whether the feedback published was favourable or unfavourable, they also used a number of programmes, including TfidVectorizer, nltk, and others. When a film's title is included in the final result, related films are recommended. For this, Javascript was used. By geometrically displaying the vectors on a multidimensional space, the cosine similarity measure is utilised to determine how similar two documents are.

1. The Term-frequency analysis was utilised by **N. Muthurasu, Kavitha Coonjeevaram**, and **Nandhini Rengaraj**. This study uses an inverse document frequency technique to vectorize a hybrid audiovisual recommendation engine. The cosine similarity technique is used to calculate the similarity. The system is displayed to the user through a web-based user interface. The method offers accurate suggestions and effective predictions even with a little data model. Future plans include user categorization and recordkeeping, analytics reports for makers and consumers, and data collection through web scraping. Users can save time, and in the future, there will be updates like a data analytics site that will let filmmakers examine and monitor viewer behaviour and preferences for a particular genre or video.
2. A technique was proposed by **J. Aswin** and **P. Sabari Ramkumar** to address the cold start issue and recommend films to its audience. A hybrid approach that combines content and collaborative-based approaches is presented. It includes a neighbor-based approach for item-based collaborative filtering, a similarity-based approach for user-based collaborative filtering, and a model-based approach for user-based collaborative filtering. The network's overall performance is improved by integrating several filtering strategies.They used two cooperative strategies—user-based and item-based—to improve the accuracy of both recommendation systems. The consumer collaborative filter is based on the Pearson product technique correlation coefficient algorithm, whereas the object collaborative filter is based on Bayesian customised ranking.
3. In this study, **Yu Zhu, Shibi He, Ziyu Guan, Jinhao Lin, Beidou Wang, Haifeng Liu**, and **Deng Cai** primarily focused on the item cold-start problem. It's helpful to record user feedback on new products, along with details (such item qualities) and early user ratings. The research's proposed system is a ground-breaking item cold-start recommendation method that makes use of both enhanced learning and item attribute data. Then they incorporated the data in an optimisation approach for user selection, creating consumer selection relevant to item attributes and user rating history. Using the feedback ratings, previous ratings by users, and item characteristics, we then create accurate rating projections for the remaining unselected individuals.

**Methodology**

**3.1 methods used/proposed system:**

In order to recommend films, our system uses a cosine similarity algorithm, term frequency-inverse document frequency, and a content-based movie recommendation system (MRS). This system is trained by understanding customers' actions and activities. The primary benefit of this method is that the algorithm is built to function effectively with only a minimal collection of data. The recommendations are based on the user's preferences, history, user profile information, and knowledge of the movie's plot. The users can save time, and in the future, there will be improvements like a data analytics portal that will let the movie creators analyse and track how well each user responds to a particular genre or video. Additionally, more effective recommendation systems expand their market reach.

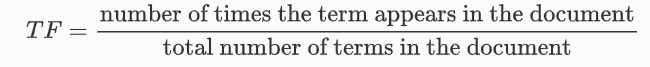
1. **Term Frequency-Inverse Document Frequency (TF-IDF)-**

The 2 fundamental ideas used in this recommendation system are cosine similarity and term frequency-inverse document frequency (TF-IDF). The data is vectorized using TF-IDF, and the similarity between the vectors is calculated using cosine similarity.

This technique is frequently applied in conjunction with content-based recommendation systems. There are two terms in it. They are Inverse Document Frequency (IDF) and Term Frequency (TF). The frequency of interests and favourites in a user profile is the subject of the term frequency. As opposed to this, inverse document frequency examines the phrase frequency throughout the entire set of information provided by the user profile. These two ideas are coupled to give a user a recommendation based on the information provided by their user profile. This idea is mostly used to calculate the value of a suggested video by weighing the influence of often occurring interests.

The TF-IDF method works by determining the relative frequency of words in a particular document in comparison to the frequency of words throughout the full set of documents. This ultimately affects how frequently a word is used in a specific document. Common terms like articles and prepositions tend to have lower TF-IDF scores than words that are common in a single or limited set of publications . Common terms like prepositions, articles, and pronouns that don't have the right meaning for an inquiry are an exceptional situation. As a result, these words receive a very low TF-IDF score, which makes searchers less likely to use them. It is kept in n-dimensional space as a vector representing its qualities, and the angle between the vectors is determined to ascertain how similar they are.

The term frequency can be represented as-



The inverse document frequency for word is commonly represented as-

A black text on a white background

Description automatically generated

Hence, the TF-IDF weight for keyword in document can be written as-

A math equation with a equal sign

Description automatically generated

It has been noted that the TF-IDF method does not take the relationship between words that is present in the document into account when determining synonyms. When searching for materials that might be related to a question, TF-IDF would use the word reverend rather than the word "priest" if the user wished to learn more about the word. The TF-IDF algorithm occasionally fails to recognise a word in its plural form , for instance, if the searched term is work, the algorithm fails to take the word works' plural into account.

The benefit of this recommendation system is that it analyses all the information that the user has provided in their user profile and then recommends the video based on their interests ,i.e., user independence. There is also no cold start for new items that don't have enough reviews or descriptions, and there is transparency that makes the recommender system's operation clear by listing features or descriptions.

The shortcomings of the currently used recommendation system include limited content analysis, which results in less accuracy of the recommendation system, very poor user profile analysis, the serendipity problem (mind cages for a particular group of users based on their interests), also known as over-specialization, and new users who lack the necessary number of ratings before a content-based recommender system can recognise them.

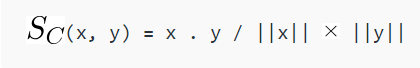
1. **Cosine Similarity-**

A difficult idea that has been extensively addressed in information retrieval is cosine similarity. A text document is transformed into a vector of phrases via this algorithm. By calculating the cosine value between two vectors, this model allows one to determine how similar two datasets are. This approach can be applied to any two texts, including sentences, paragraphs, and documents. The results of the similarity assessment between the vectors can occasionally be unstable. In the case of search engines, the degree of similarity between user queries and documents is assessed, and the results are then ranked from highest to lowest.

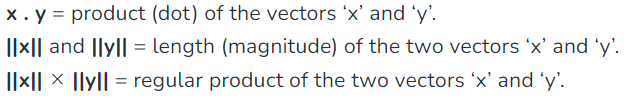
More relevancy between the user's query and the document is indicated by a higher similarity score between the two vectors.The term's definition should be examined when comparing the user query with the document. On the other hand, cosine similarity still struggles to handle the query's semantic meaning. The discrepancy in syntax matching does not meet the semantic meaning challenge. Since information retrieval systems may give inconsistent results, they may not function to their full potential.

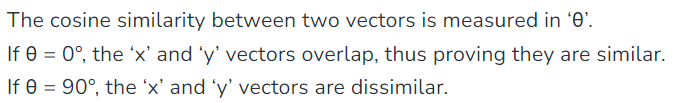
WordNet has been discovered to be used in research on issues of a similar nature. WordNet is the most popular technique since it uses a lexical dictionary as a semantic network. This enhances the ability to compare the cosine similarity of the two vectors along with their semantic analysis. The goal of this application is to more precisely calculate the similarity between two vectors. A document can be represented as a term vector in document-query scenarios, with the vector's dimensions denoting the terms present in the document. The value of a dimension is the frequency of a term within a document.

The formula to find the cosine similarity between two vectors is –



Where,

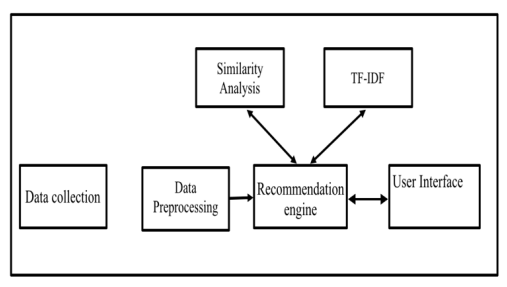




As a result, the null distribution's cosine similarity equals the distribution of the dot product of two different vectors. This distribution has a mean of 0 and a variance of 1/n, where n is the number of dimensions. Although the distribution is bounded between -1 and +1, as n increases, the normal distribution becomes more and more accurate at describing it. The null distribution will have a different form and may have a nonzero mean for other forms of data, such bit streams (which only accept values of 0 or 1).

**3.2 architecture:**

All the modules and implied process flow are explained fully in a system's architectural diagram. It aids in allocating each module among the group and gives an overview of what has to be done in order to finish the procedure. The architecture diagram contains the organization's information. The developer's architecture diagram can be used to forecast how the process will behave. This section provides a quick explanation of the modules used in our suggested system.



The technique described in this work is put into practise in three stages. The first stage involves data collection and analysis. For this, we used a dataset with about 5000 videos, and we cleaned the data to prepare it for further analysis. The vectorization process and the determination of the dataset's similarity score were the main topics of the project's second phase. The scikit learn python library's Tfidf Vectorizer and cosine similarity functions were utilised for that. The Sci-Kit Learn library is the best resource for machine learning algorithms because it has almost all ML methods for Python, which speeds up and simplifies evaluations. In the last stage, the model is validated, and its recommendations are examined.

**3.3 analysis walkthrough:**

Data collection-

The dataset was procured from Kaggle. Kaggle is a hub for datasets where the datasets are published by a community of data scientists and machine learning practitioners. It contains roughly around 8808 movies and 12 attributes. The attributes of the dataset are: Index, Show\_ID,Type,Title,Director,Cast,Country,Date\_Added,Release\_Year,Ratings,Duration,Listed\_In and Description.

The dataset was procured from Kaggle and can be found here. In the data processing stage, this data is given as a csv file, which is then parsed.

Data processing-

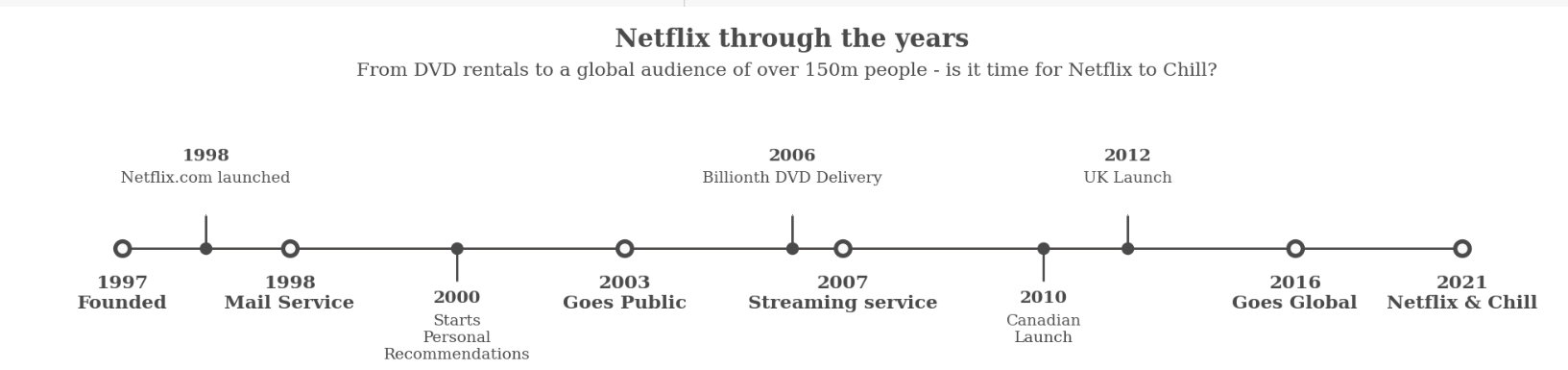
The first phase in the recommendation system is data processing. The gathered information is imported as a record, divided, and flattened using keys into lists of categories including movie titles, movie genres, and narrative summaries. An individual identification number, or Show\_Id, is used to identify each film. The user-provided ID or name is verified before being submitted to the engine to produce recommendations. The validation procedure guarantees that the input provided by the user is a valid alphanumeric character and examines the input length, input range, and input length.

1. **Removing Missing values**- 5 columns had missing values.

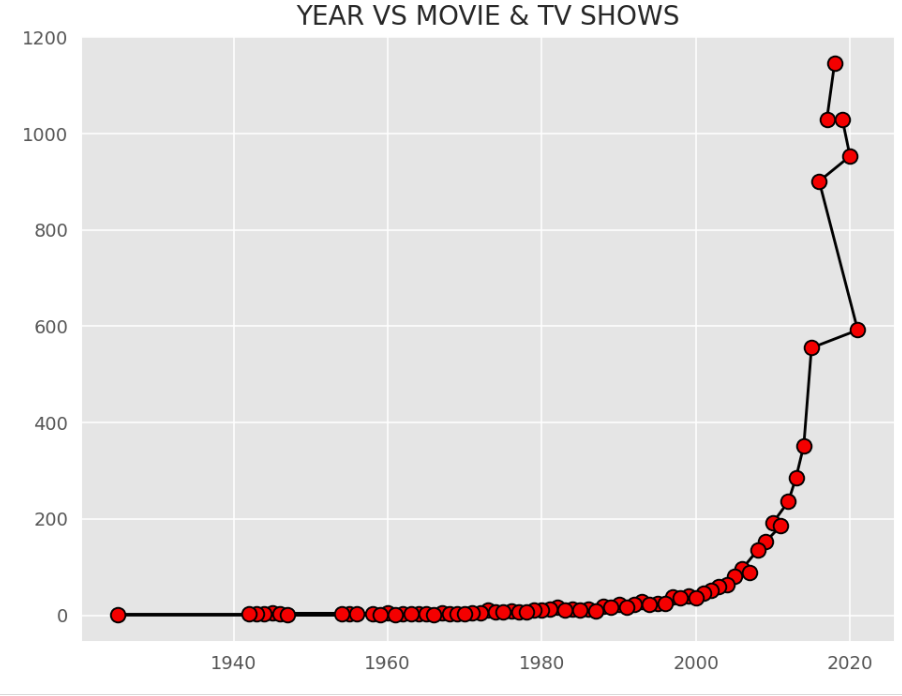
The Country missing attributes were filled with the mode, ie the most frequently occurring country.

For Director and Cast columns we just replace missing values with ‘No Data’.

1. **Dropping Duplicates**
2. **Dividing Dataset For Movies and TV Shows**
3. **Exploratory Data Analysis On Dataset-**
4. **Netflix Through The Years**

****

1. **Year Wise Trend Of Movies and TV Shows**

****

1. **Distribution of Movies and TV Shows**

**A red and black rectangular box with white text

Description automatically generatedA screenshot of a graph

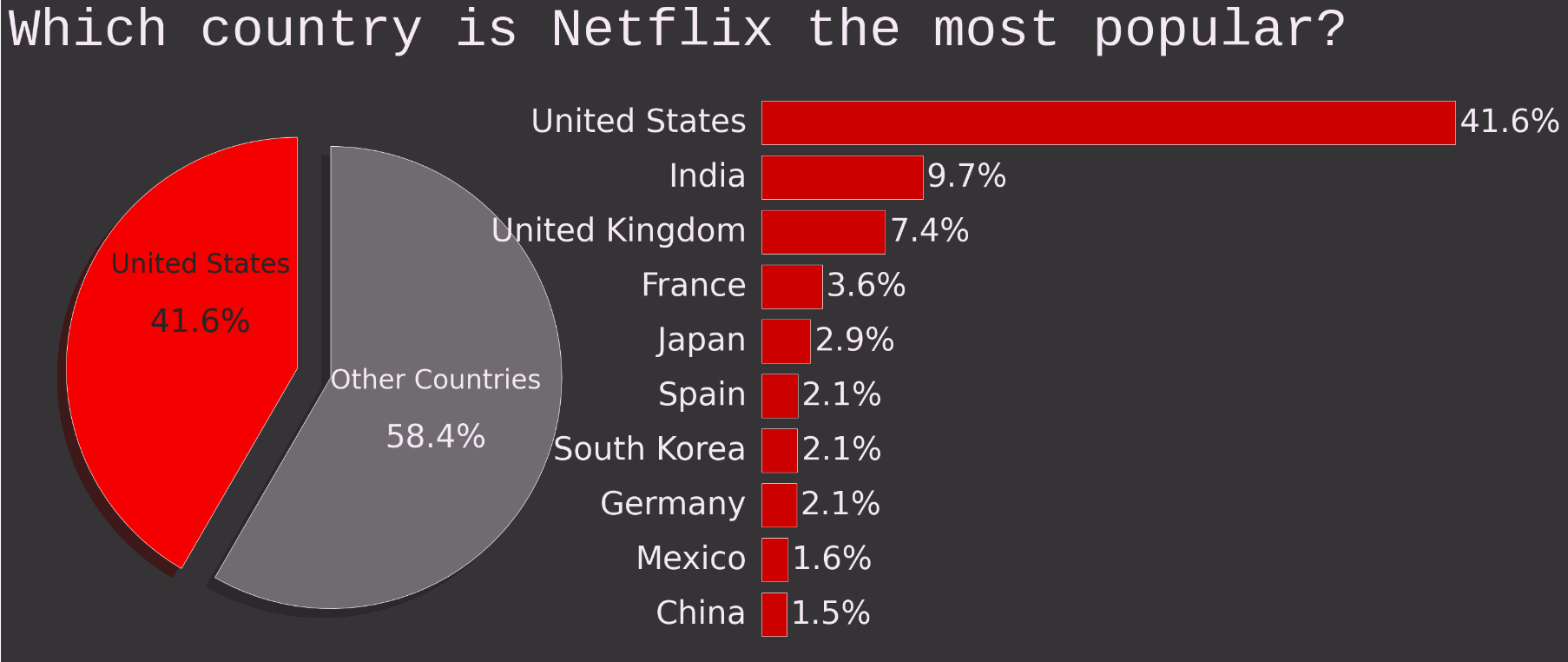
Description automatically generated**

1. **Top Countries That Give Content**

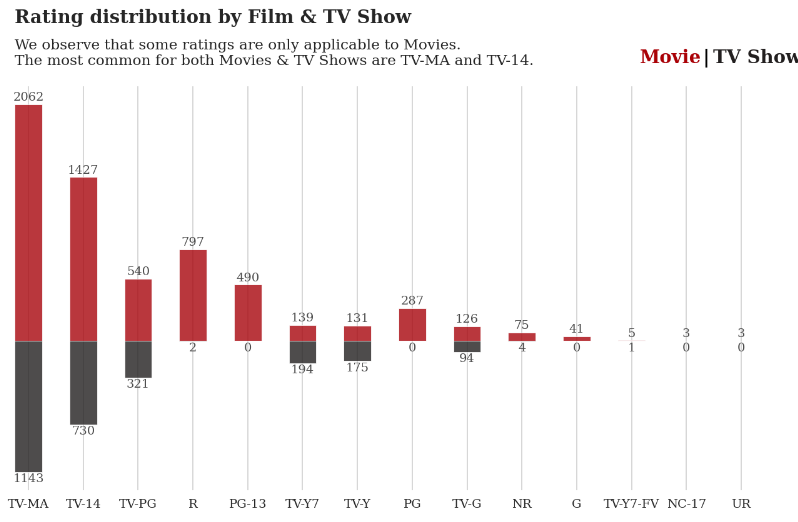
**A graph of different colored squares

Description automatically generated**

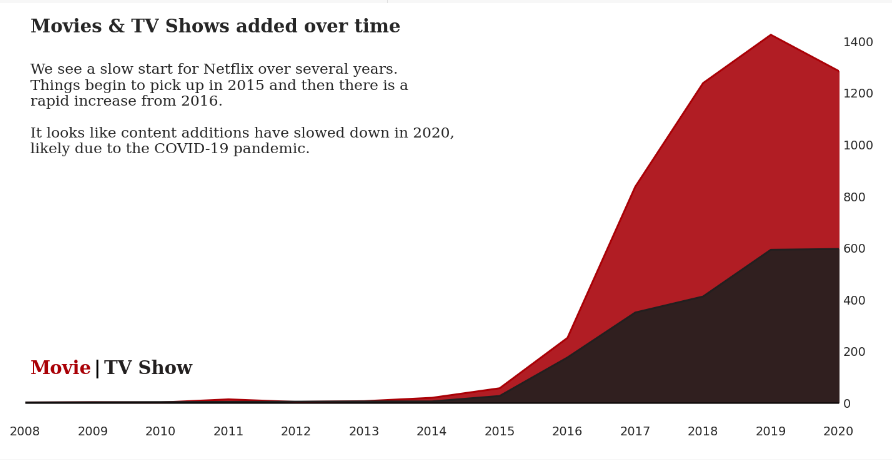
1. **Most Popular Countries**

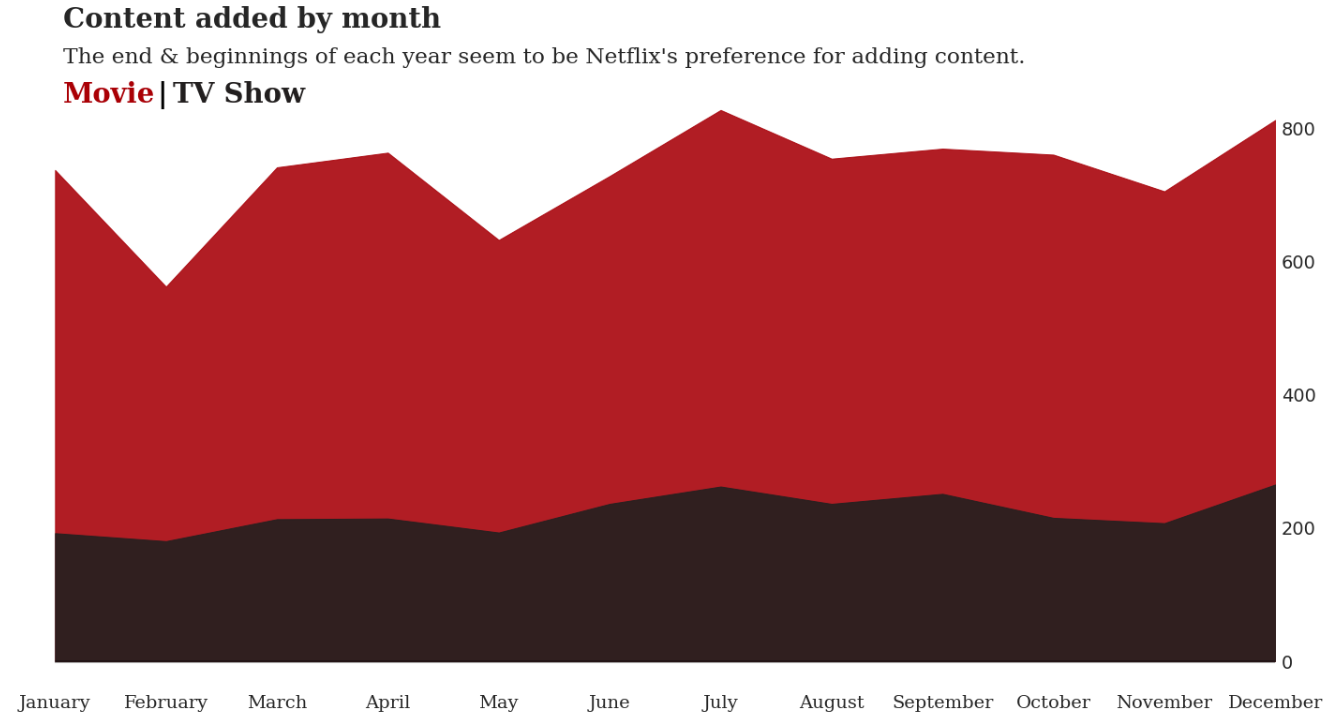
****

1. **Rating Distribution**

****

1. **Movies And TV Shows Added Over Time**

****

****

* January and December are the best months for new content

1. **Netflix Stock Price**

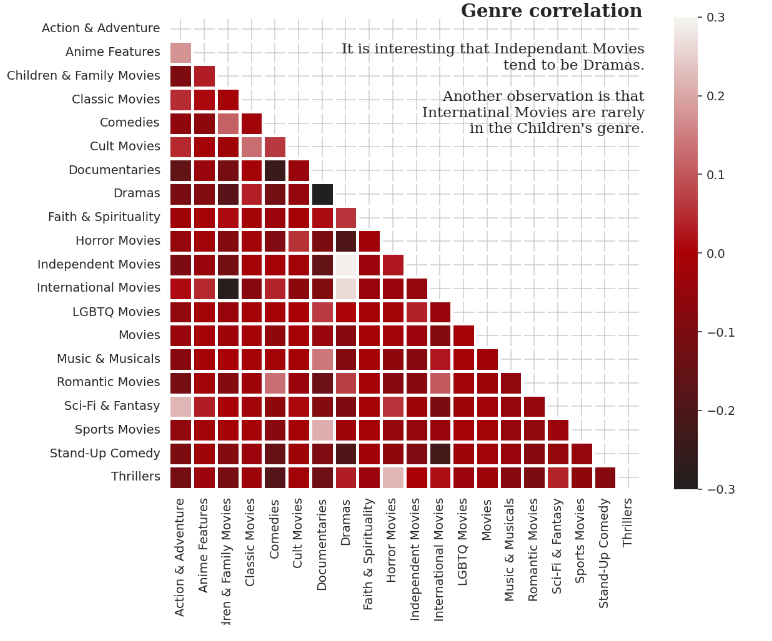
****

**A graph with a line going up

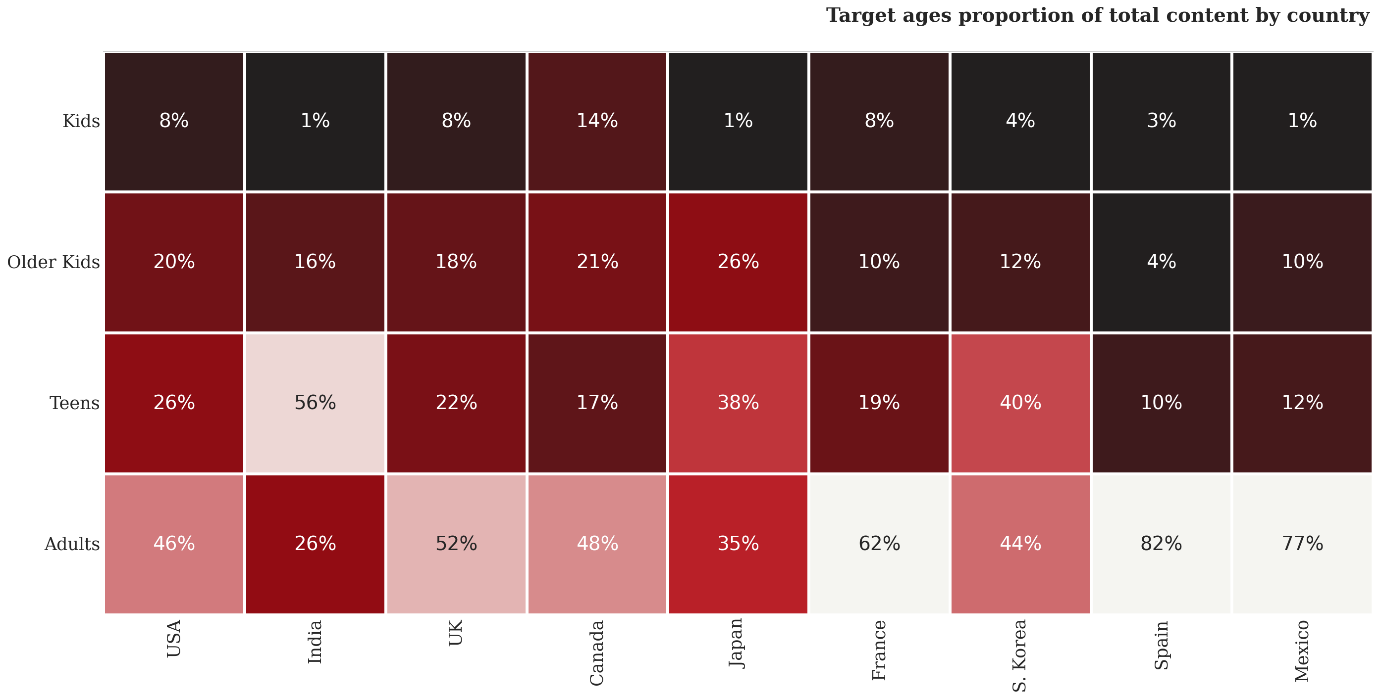
Description automatically generated**

* I prepared two chart to show Netflix's growth.First is Netflix's stock price.
* Surely, Stock price doesn't explain that It's growth all.
* Now second chart.
* This is Netflix's movie releasement increase.
* It shows that Netflix made an aggressive investment.
* Although, at 2020 discreased, because of Covid-19But, I think it has increased again after Covid-19 and start of the release of the movie.
* We can notice that, both stock price, and movie releasement increase have exploded since 2014.

1. **Visualisations On Generes**

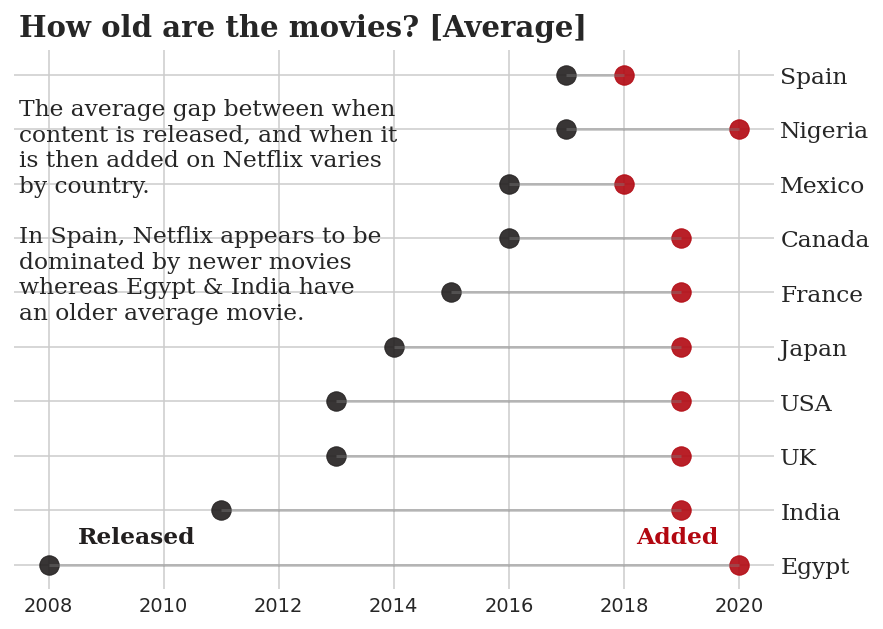
****

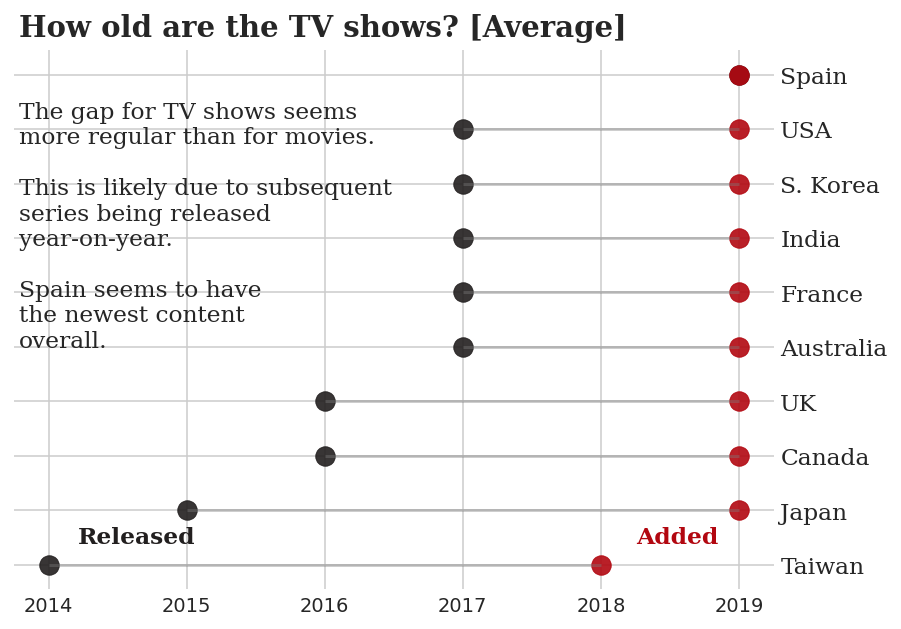
1. **Visualisations On Target Ages**

****

* Here we see interesting differences between countries. Most shows in India are targeted to teens, for instance.
* It is also interesting to note similarities between culturally similar countries
* the US & UK are closely aligned with their Netflix target ages, yet vastly different to say, India or Japan.

1. **Lag Between The Releasement Of Content And When It Is Added On Netflix**

****

****

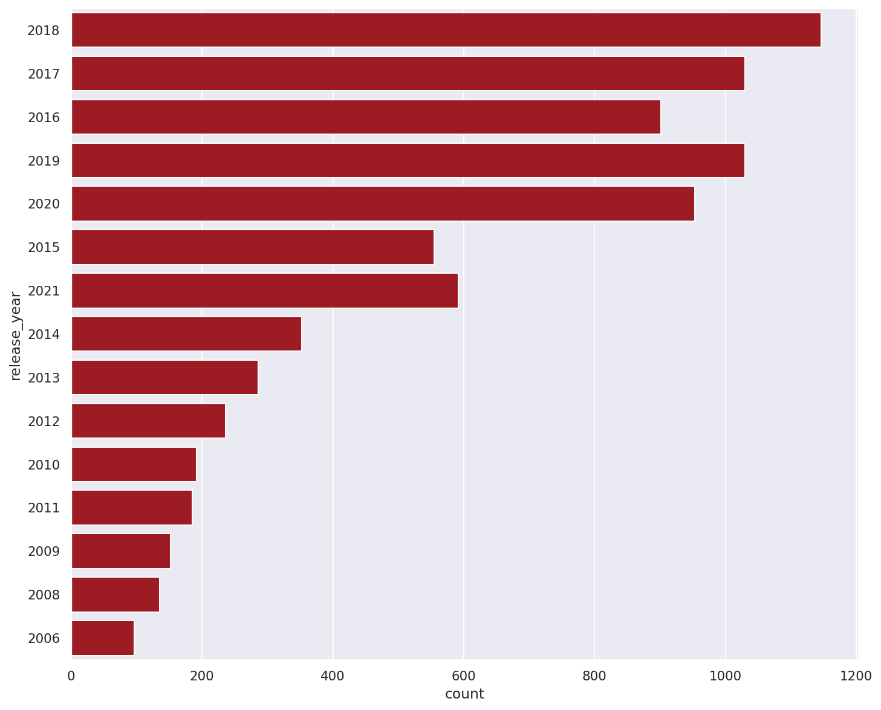
1. **Netflix Global Popularity**

**A map of the world

Description automatically generated**

* Netflix is enjoyed in most countries, without some of Africa. Especially, America and Ease Asia, Australia, India stand out.

1. **Year Wise Analysis**

****

* 2018 was the year when most movies were released.

1. **Analysis Of Duration Of Movies**

**A graph with a blue line

Description automatically generated**

* So, a good amount of movies on Netflix are among the duration of 75-120 mins.
* It is acceptable considering the fact that a fair amount of the audience cannot watch a 3 hour movie in one sitting.

A graph showing a line of a graph

Description automatically generated

* In the years of 1960 to 1965, Movies durations were over 200 minutes, after 1965 the durations became comparatively shorter.
* From the year 1980, we can see consistent trend of movie durations, of which duration time is around in between 100-150 minutes.

1. **Top rated 10 movies on Netflix**

**A graph with red lines

Description automatically generated**

1. **Genres vs their count on Netflix**

**A graph with red dots and numbers

Description automatically generated**

1. **TV shows with largest number of seasons**

**A graph with red lines

Description automatically generated**

**A pie chart with numbers and a few black text

Description automatically generated**

1. **Recommendation System**

Term-Frequency Inverse Recommendations are made based on document frequency and the preferences of the user. The previously mentioned TF-IDF vectorization algorithm is used to vectorize each data record. The cosine similarity method is used to calculate a similarity measure for each vector. The correlation coefficients for the movies with regard to a given movie are created when a user asks a specific number of recommendations for that movie. Each chosen similar film will receive a score indicating how similar it is to the referenced film, which will be sorted in descending order to list the films with high to low similarity.

1. **Data Preparation**:

We selected the relevant features required for an accurate recommendation. Content based filtering on the following factors:

**Title, Cast, Director, Listed in and Description**

1. **Data Cleaning:**

We make all the words lower case.

1. **Bag Of Words:**

Creating a "bag of words" for all rows.



1. **Vectorization Of Data**:

Using the TfidVectorizer function, we transformed the text input into feature vectors in this stage. The sklearn library contains a method called Tfidvectorizer.

A screen shot of a computer code

Description automatically generated

1. **Calculating Cosine Similarity:**

Here we calculate the cosine similarity using the Cosine\_similarity function found in the sklearn library. Below seen is the similarity score matrix of the dataset.

**A computer code with black text

Description automatically generated**

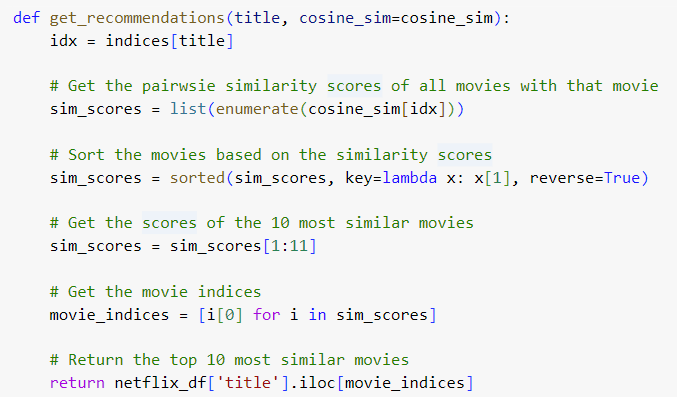
For getting better or more accurate results we can also use CountVectonizer which is based onContent based filtering on multiple metrics.

**A screen shot of a computer code

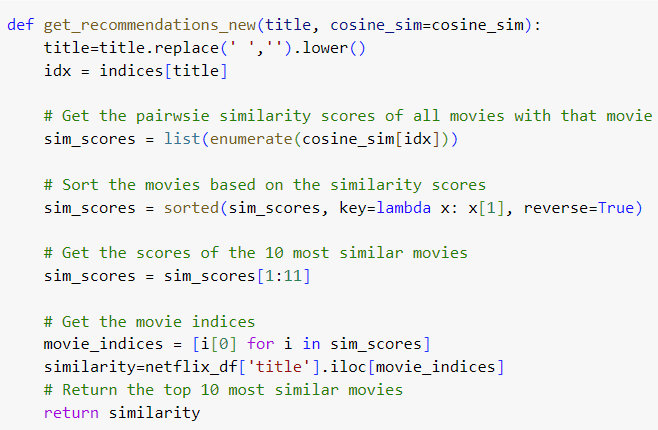
Description automatically generated**

1. **Creating a Function for containing the Recommended Movies:**

We create a function called ‘get\_recommendations’ and pass the movie title and cosine similarities.

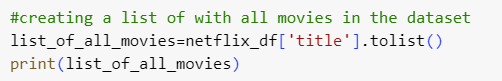


For multiple metrics-

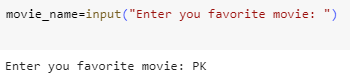


1. **Model Validation and Suggestions:**

When a user provides an input, a list of all the movies in the dataset is generated, and the algorithm then searches for the movie that most closely matches the user's input. Using the similarity score, a list of films that are similar to the closest match is produced. The films are arranged according to how similar they are. After that, a list of films that are comparable to the input is printed.



Input By the User:



**RESULTS**

**4.1 RESULTS**

* **Top 10 Vectorized data-**

****

****

* **Top 10 recommended movies-**

**A screenshot of a computer

Description automatically generated**

* The programme successfully vectorized the data derived from the photos using TF-IDF vectorization, and it also determined the similarity scores using cosine similarity. Additionally, we can see that when the algorithm is given the name of ‘PK’ as an input, a list of films with a high similarity score is presented as the result.
* Many small-scale commercial websites lack the necessary data for the movie recommendations engine to operate effectively. However, our method also solves this issue because it performs well with little data, which is advantageous for modestly sized business websites. The benefit of this recommendation system is that it examines all the information that the user has provided in their user profile and then recommends the video based on their interests (i.e., user independence). There is also no cold start for new items with insufficient description or reviews, and there is transparency, which makes it clear how the recommender system operates by explicitly listing features or descriptions.

**4.2 limitations**

* The shortcomings of the currently used recommendation system include limited content analysis, which results in less accuracy of the recommendation system, very poor user profile analysis, the serendipity problem (mind cages for a particular group of users based on their interests), also known as over-specialization, and new users who lack the necessary number of ratings before a content-based recommender system can recognise them.
* Despite all the benefits our system offers, there are still some possible drawbacks that we will need to work to resolve in the future. One of these restrictions is that when our system is trained to recommend films to consumers, it doesn't take into account user or customer ratings or likes (likes comparable to those seen on Facebook). Due to issues that need to be addressed, such as sarcasm and homophones, this has not yet been included in our system. This needs a lot of information to operate effectively.
* Another drawback is that it only performs well with modest amounts of data. This is due to the fact that handling a big volume of data demands a very complex system, one that should also analyse video data. Our system is also constrained by users' ability to change their preferences. The interests and preferences of a user can fluctuate. It can be challenging to interpret this conduct and suggest films that are tailored to the person's mood and tastes. One of our future initiatives that we had intended to conquer is this.
* However, this strategy is just the beginning of a more sophisticated recommendation system that takes into account additional factors like the show's duration, Netflix score, top players, etc. Additionally, as Netflix does not take into account the demographics of its customers, we decided to do the same in order to maximize the likelihood that the client will accept our recommendations.

**4.3 FUTURE WORK**

* User profiling can be used to make it simpler to use the website's user interface. It is possible to store the user's past preferences, likes, etc. After getting the necessary legal clearances, information regarding the films for which recommendations are sought can also be acquired from a number of websites utilising web scraping or web data extraction technologies.
* Small e-commerce site sellers can purchase and integrate this as a separate programme (engine), which can also be developed. The programme can be used with intelligent data analysis for processing vast amounts of data utilising big data approaches to deliver reliable analytics.
* Sentiment analysis can also be applied to “comments” information to identify the emotion behind the comments (positive, negative or neutral) to recommend movies appropriately.
* Additionally, we intend to create a hybrid movie recommendation system that is more accurate and effective.

**Conclusion and references**

* 1. **Conclusion**
* Currently, Netflix is the most widely used on-demand streaming service. It airs thousands of programmes to customers in 190 different nations. Since the beginning of online programming, films have dominated the field. The year 2020, however, and the beginning of the year 2021 both witness the first-ever dominance of TV shows.
* We performed an exploratory examination of Netflix data for our study. This analysis highlighted data on the most popular Netflix-accessible nations, including the United States, India, and the United Kingdom; the distribution of broadcast programmes by category (69% films and 31% TV shows); and the semantic analysis of the words used in the works broadcast descriptions.
* On the other hand, we felt it would be fascinating to do a more in-depth study on this subject to understand the causes, especially given the rise in the number of TV shows offered at the expense of films in 2020. In fact, the COVID-19 epidemic, which has compelled billions of people to limit themselves globally, could be listed as one of the possible reasons.
* Large data volumes are required to generate trustworthy recommendations, which opens the door to other applications like the usage of big data technology and efficient data processing techniques. Data and information are never static; they constantly change as a result of shifting user preferences and behaviours.
* The completion of this research might serve as a precursor for the creation of more effective content-based recommender systems. Using TF-IDF vectorization and Cosine similarity, we have successfully constructed a movie recommendation system in this project.
* Every system carries a balance of strengths and weaknesses. Finding the best algorithm for processing the data and balancing the results with a combination of similarity algorithms are the keys to creating the ideal recommendation system for the user's demands.
* Nevertheless, the simplicity and small amount of training data needed to create the recommendation system characterise our work. This has the benefit of making it simple to use and apply.
  1. **REFERENCES**

1. Richar Macmanus. Redwrite.com 5 Problems of Recommender Sys-tems. Web Source. Published on Jan 28, 2009 in WEB.
2. Xiaokun Wu, Bo Cheng, and Junliang Chen. Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method. IEEE Transactions on Services Computing( Volume: 10, Issue: 3, May-June 1 2017 ). 352 – 365
3. Spruill, Marcus C (2007). ”Asymptotic distribution of coordinates on high dimensional spheres”. Electronic communications in probability.
4. CrossValidated: Distribution of dot products between two random unit vectors in RD
5. Graham L. Giller (2012). ”The Statistical Properties of Random Bit-streams and the Sampling Distribution of Cosine Similarity”. Giller Investments Research Notes (20121024/1).
6. .https://en.wikipedia.org/. Flask Web Framework. Web Source. Wikipedia.
7. Nandhini Rengaraj. C.M.Kavitha. .Sabitha Loganathan. N.Muthurasu. A study of existing systems of recommendation engines. National Conference of Emerging Computing Technologies and Applications. Vel Tech University. Avadi, Chennai. 05-06 April 2018.
8. Rajarajeswari, S., et al. "Movie Recommendation System." Emerging Research in Computing, Information, Communication, and Applications. Springer, Singapore, 2019.
9. Bennett, J., Lanning, S., 2007. The Netflix prize. In Proceedings of KDD cup and workshop, Vol. 2007, p. 35). New York, NY, USA. Chiny, M., Bencharef, O., Hadi, M.-Y., Chihab, Y., 2021.
10. A Client-Centric Evaluation System to Evaluate Guest’s Satisfaction on Airbnb Using Machine Learning and NLP. Applied Computational Intelligence and Soft Computing. Colbjornsen, T., Talleras, K., Ofsti, M., 2020.
11. Contingent availability: a case-based approach to understanding availability in streaming services and cultural policy implications. INTERNATIONAL JOURNAL OF CULTURAL POLICY.
12. Ensaf, H.-M., Mohammed, E.-M., Mohamed, H.-H. H., 2020. An Enhanced Sentiment Analysis Framework Based on Pre-Trained Word Embedding. International Journal of Computational Intelligence and Applications. Flixable, 2021.
13. Guo. A. and Yang, T., 2016. Research and improvement of feature words weight based on TF-IDF algorithm. In 2016 IEEE Information Technology, Networking,
14. Bennett, J., Lanning, S., 2007. The Netflix prize. In
15. Proceedings of KDD cup and workshop, Vol. 2007, p.
16. 35). New York, NY, USA.
17. Chiny, M., Bencharef, O., Hadi, M.-Y., Chihab, Y., 2021.
18. A Client-Centric Evaluation System to Evaluate
19. Guest’s Satisfaction on Airbnb Using Machine
20. Learning and NLP. Applied Computational Intelligence
21. and Soft Computing.
22. Colbjornsen, T., Talleras, K., Ofsti, M., 2020. Contingent
23. availability: a case-based approach to understanding
24. availability in streaming services and cultural policy
25. implications. INTERNATIONAL JOURNAL OF
26. CULTURAL POLICY.