A picture containing text, electronics

Description automatically generated

**A COMBINED MACHINE LEARNING ALGORITHM FOR NETFLIX MOVIE SELECTION FOR VIEWERS**

A dissertation submitted in partial fulfilment of the

requirements for the degree of Master of Science

by

**MANIDEEP REDDY ANANTHA**

**ST20201450**

Master's Degree in Data Science

Cardiff School of Technology

Cardiff Metropolitan University

September 2022

**Table of Contents**

Abstract……………………………………………………………………………………….6

Chapter 1: Introduction……………………………………………………………………….7

1.1 About Netflix……………………………………………………………………………..8

1.2 Movie Recommendation system…………………………………………………8

1.3 Research Aim…………………………………………………………………….8

1.4 Research Objective………….….………………………………………………..8

1.5 Research Question………………………………………………………………..8

Chapter 2: Literature Review………………………………………………………………...9

2.1 Movie Recommendation System………………………………………………...9

2.2 Machine learning algorithms……………………………………………………10

Chapter 3: Methodology…………………………………………………………………….14

3.1 About dataset…………………………………………………………………....14

3.1.1 Context………………………………………………………………...14

3.1.2 Content………………………………………………………………...14

3.1.3 Ethical and Social issues………………………………………………15

3.2 Rationale steps in the proposed methodology…………………………………..15

3.2.1 Data collection………………………………………………………...16

3.2.2 Data extraction and pre-processing…………………………………...18

3.2.3 Data Analysis and Data Visualisation………………………………...18

3.2.4 Machine Learning Models…………………………………………….19

3.2.5 Movie Recommendation based on filtering techniques………………19

3.2.5.1 Demographic Filtering………………………………………19

3.2.5.2 Content-Based Filtering……………………………………..19

3.3 Proposed Flow…………………………………………………………..19

Chapter 4: Implementation………………………………………………………………….21

4.1 Support Vector Machine………………………………………………………...25

4.2 Random Forest Classifier……………………………………………………….26

4.3 Naive Bayes……………………………………………………………………..27

4.4 Logistic Regression……………………………………………………………..27

Chapter 5: Software and Tools……………………………………………………………...29

5.1 Python…………………………………………………………………………...29

5.2 Anaconda………………………………………………………………………..30

5.2.1 Anaconda Navigator and IDEs………………………………………..30

5.2.2 Spyder…………………………………………………………………31

5.3 The Jupyter Notebook as well as the Jupyter Lab………………………………31

5.4 Numpy ………………………………………………………………………….32

5.5 Pandas…………………………………………………………………………...33

5.6 Matplot…………………………………………………………………………..34

5.7 Scikit-learn………………………………………………………………………35

5.8 Seaborn………………………………………………………………………….35

Chapter 6: Implementation………………………………………………………………….36

Conclusion…………………………………………………………………………………..44

References…………………………………………………………………………………..46

**List of Figures**

Figure.1.1 Netflix logo………………………………………………………………………7

Figure.1.2 Movie Recommendation system…………………………………………………8

Figure.3.1 Proposed Block diagram……………………………………………………….18

Figure.3.2 Data pre-processing…………………………………………………………….19

Figure.4.1 Importing Library Packages and Export Dataset……………………………….23

Figure.4.2 Importing Library Packages and Export Dataset……………………………….23

Figure.4.3 Finding average rating score……………………………………………………23

Figure.4.4 Average user rating size………………………………………………………...24

Figure.4.5 Copying data frame based on the user rating size………………………………24

Figure.4.6 Weighted Rating…………………………………………….……………….….24

Figure.4.7 Sorting movies based on attributes…………………………………….………..25

Figure.4.8 TV shows based on rating level………………………………………………...25

Figure.4.9 Vectorizer…………………………………………………………………….....26

Figure.4.10 Cosine Similarity Index……………………………………………………….26

Figure.4.11 Movies Indices based on movies title…………………………………………26

Figure.4.12 Get movie recommendations………………………………………………….27

Figure.4.13 Label Encoder & Train and Test Split………………………………………...27

Figure.4.14 Support Vector Machine………………………………………………………28

Figure.4.15 Random Forest Classifier……………………………………………………...28

Figure.4.16 Naïve Bayes…………………………………………………………………...29

Figure.4.17 Logistic Regression……………………………………………………………29

Figure.6.1 Distribution of Netflix Shows by Year by Rating……………………………....40

Figure.6.2 Distribution of Netflix Shows by Year by Single Rating……………………….40

Figure.6.3 Netflix shows by Rating…………………………………………………………41

Figure.6.4 Netflix Shows Release Date by Year…………………………………...……….41

Figure.6.5 Sample Netflix dataset…………………………...……………………………...42

Figure.6.6 Data frame after adding class attributes.……………………….………………..42

Figure.6.7 Movie shows based on Ratings………………………………………………….42

Figure.6.8 Movies Indices……………………………………………………………...…...43

Figure.6.9 Recommended movies……………………………………...…………………...43

Figure.6.10 Confusion matrix for Support vector machine………………………………...40

Figure.6.11 Confusion matrix Random Forest………………………………………….….40

Figure.6.12 Confusion matrix for Naïve bayes…………………………………………….41

Figure.6.13 Confusion matrix for Logistic Regression…………………………………….41

**Chapter 1: Introduction**

**1.1 About Netflix**

The happiness of Netflix subscribers can be thought of as the company's fundamental goal, and it applies to all parts of the world. They allow users access to the best-in-class television shows, documentaries, feature films, and mobile games, regardless of the user's preferences and where the user resides. This applies to both the content and the fun. This holds even though the user may be interested in watching different content types or playing other content types. Our paying customers have complete control over the content shown to them and the timing of when they watch it, and their individual subscription does not include any types of advertising. They think that great stories can originate from any location and are loved in all parts of the world; for this reason, they broadcast in more than 30 distinct languages and 190 different nations worldwide. They are the most devoted consumers of entertainment in the entire globe, and they are continually looking for novel approaches to aid clients in locating the narrative that will become their new favourite (netflix.com).



Figure.1 Netflix Logo

Customers and the users can be forced to search for stories and movies depending on their tastes, which can include both positive and negative ratings. To achieve this goal, they need to search for reviews and uncover stories or movies pertinent to the topic. We can create a recommendation system for movies on other websites by using the same techniques we use to create a recommendation system for Netflix movies by analyzing user reviews and suggestions based on user interests. The machine learning algorithm plays a crucial role in the review and analysis of movies; here, we can train ourselves using the dataset made available by Netflix.

**1.2 Movie Recommendation system**

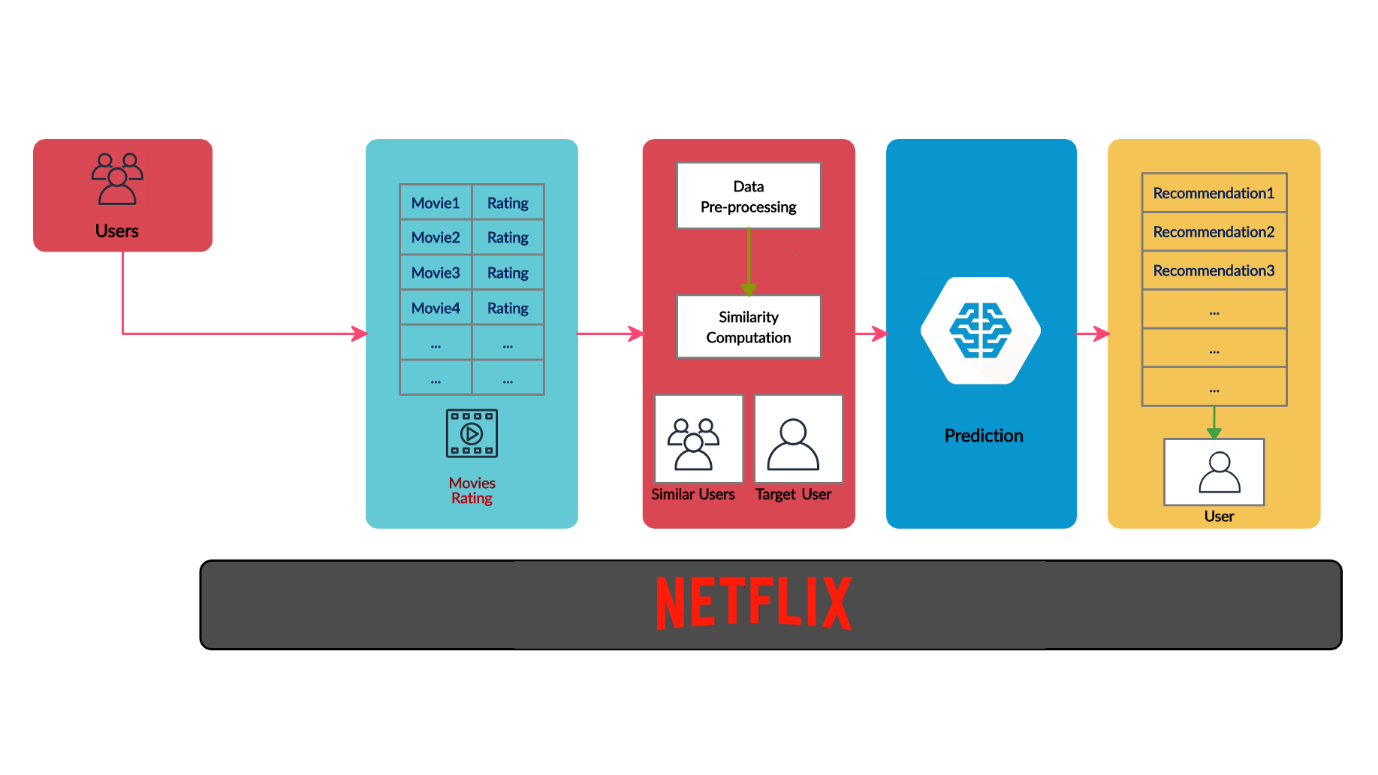


Figure.1.2 Movie Recommendation system

Because of the vast amount of knowledge accumulated by the 21st century and the rapid spread of information online, you will always get many relevant results if you are seeking for a particular video on YouTube. YouTube's video library is very hug, because of which you may have trouble locating a movie unrelated to your study topic. If the results are appropriately ranked, there should not be much of a problem. If not, you would spend a lot of time finding a movie that meets the needs and to discover the perfect movie based on the requirements. In order to achieve this, you could count on us to find the perfect film for us. we would put in lot of work to choose the best movie among the lists based on the requirement of the user. After a website search, the results will be show in a ranked manner based on the user requirements. Next time you visit a website, the system may offer stuff it thinks you will like without requiring an investigation.

Movielens, TMDB, and Netflix are well-known data collections. Netflix and Amazon Prime employs "movie recommendations" to earn revenues by improving the user experience while using their services. This is done by suggesting movies to clients based on their watch history and preferences. They do this by giving clients additional movie suggestions. Netflix sponsored a $1M competition in 2009 for a system that was at least 10% better than the existing one. The competition required a 10% improvement over the current system. The competition sought a 10% more effective method than the present one. The winner was the competitor who devised the most effective approach. We have access to a lot of data, but before we can utilize it, we must analyze it using filters since we are not interested in every piece of information. We must process data to use it. First, the data must be filtered using a variety of methods. Several filtering procedures and movie recommendation algorithms may be used to construct a recommendation system.

**1.3 Research Aim**

This project aims to give recommendations on movies and stories on Netflix using machine learning algorithms.

* To analyze the dataset of Netflix.
* To perform a critical analysis on the category of the movies and stories.
* To find the best machine learning algorithms for this Netflix recommendation system.

**1.4 Research Objective**

The project Objective is to construct a recommendation using multiple machine learning algorithms to provide utmost accuracy in projecting movie suggestions to the user

* To learn about the existing models and overcome the drawbacks in those models
* To analyse the recommender system that uses graphs, content-based filtering, hybrid recommender systems and collaborative frameworks.
* Analyse the results obtained and based on that make predictions.
* To train the processed data using multiple machine learning algorithms.

**1.5 Research Question**

* Where can we get Netflix dataset? And is that reliable?
* What kind of machine learning techniques will be used for a recommendation?
* Which tool or programming language will produce better results on the Netflix
* recommendation system?
* How do we evaluate different kinds of recommendation systems using various techniques?
* How will we implement a combined machine learning algorithm to improve accuracy?

**Chapter 2: Literature Review**

**2.1 Movie Recommendation System**

In their study, Choi et al. looked at the drawbacks of collaborative filtering (2012). The sparsity bound and the cold start bound are two examples of such constraints. The authors suggest utilising category information as an alternative because of more reliability. The authors have created a system for making movie suggestions that considers intergenre relationships. According to the authors, categorization of data is readily available for use in revising the work. Based on the genre pairings in each movie, the system initially determines genre associations. Following that, the system uses the average rating of each film and user-preferred genres to combine the genres of all movies based on genre correlations (choi et al, 2012). The drawback they are referring to is regarding the latest additions to the site. Even though it does not yet have enough ratings or views, freshly released movies may still be featured in the recommendations list, provided the information about its category or genre is utilised. The suggested strategy does not provide preference to either popular older content or less-seen newer stuff. It does not favour either type of material. Content with a high rating and a significant audience is more likely to be seen by people. This allows the recommendation system to suggest a film that just debuted but has not yet been distributed to the public.

The companions of the study effort (Lekakos et al. 2008) proposed a hybrid strategy to the process of suggesting movies. This was created using the MERP framework. Everyone calls this system "MoRe," even though that is not its official name. Specifically, the system asks a new user to offer a number of ratings as soon as they register with it so that it may start the prediction process. A content-based approach, a pure collaborative filtering method, and the proposed hybrid technique that has been put into practise in two iterations (called switching and substitute). The parameter that determines when collaborative filtering and content-based filtering are switched on and off distinguishes the two iterations of the hybrid algorithm. This system utilises a web clawer to gather information on each film's genre, actors, director, writers, producers, and story. Both the collaborative and content-based engines are used in hybrid methodologies. Although the system can produce recommendations using multiple methods, only one way is used at a time as decided by the system administrator (Lekakos et al. 2008).

Das et al. (2017) provides a comprehensive overview of several recommendation frameworks. The study's is made up of reports from both individualised and generic recommendation services. Using a superb model, we were able to conduct in-depth research into both user-based and item-based collaborative filtering. The authors discussed the pros and cons of categorising recommendation systems.

Using collaborative filtering, Zhang et al. (2019) created movie suggestions. Weighted KM-Slope-VU was one of the presented choices since the authors used k-means clustering to group users with similar characteristics together. At that point, we may start thinking of them as a single entity. Each cluster has a designated "virtual opinion leader," a person who is seen as an authority figure by the other members of the cluster. This individual was picked because of their extensive experience working in cluster environments. Instead of employing the enormous user-item rating matrix, the authors of this study opted to utilise the virtual opinion leader-item matrix in their investigation. A greater amount of work was completed by the writers. Due to this, analysis time was reduced in half. Next, the first algorithm operates on the smaller matrix. When the authors use the compressed matrix, they do this.

The current study's authors (Rajarajeswari et al., 2019) looked at the cosine and SVD similarity assessments. Their algorithm suggests 30 films most similar to one another based on their cosine scores. They dispatched the films to SVD and asked for feedback from customers. Because the authors only supplied a method for one movie, the system only considers the most recent film that was watched because the procedure can only produce a single movie. The protocol is only capable of supporting a single video. Because the machine can only process one film at a time, this is the only choice available. Clustering by K-means is one approach to the problem (Ahmed et al., 2018).

**2.2 Machine learning algorithms**

The writers have classified users into categories according to their shared traits. In order to provide specific suggestions for each group, the researchers used neural networks. You may think of them as the system's features. The number of items bought was factored in, as were user ratings and preferences. The authors utilised a neural network to predict how uninitiated moviegoers would evaluate the films after categorising them into categories. Having this done was crucial to succeeding in their mission. Authors' objectives were met. Ideas benefit from the use of projected success rates. The importance of them is not lessened in any way by this. Subramaniyaswamy et al. created a similarity-based movie recommendation system (2017).

The study report is lacking important information on the company's inner workings. No research was done by the authors. Therefore, it is unable to answer the company's inquiries. The authors claim that the recommendation system is based on a hybrid technique that combines context-based and collaborative filtering; however, they do not describe the parameters of the system or how it works (despite mentioning "City Block Distance" and "Euclidean Distance" in the section titled "Methodology"). The researchers claim that the system's suggestion methodology is a mix of collaborative and context-based filtering. Collaborative filtering is proposed as a method to better tailor movie recommendations to each individual user. If this works, it might simplify the process of recommending movies to individuals.

Use the Euclidean distance to pinpoint the user who is most like yourself. The winner is determined by the person with the least possible Euclidean distance. While the user's top-rated films are considered, they aren't the only ones taken into account when making recommendations. The creators of the feature maintain that the suggestions may be tailored to the individual's tastes. This allowed the system to adjust to the individual's tastes. This modification was made so that the programme could respond to users' evolving tastes. The investigation and analysis of the MovieLens Dataset was conducted in Harper et al. (2015). In certain cases, this information is used to suggest films to users. The MovieLens 100K, 1M, 10M, 20M, and 1B Dataset is now available. User access to fresh data sets has been expanded.

Information such as user ID, item ID/movie ID, rating, time stamp, movie title, IMDb URL, and release date are all included in the set. Recommendation systems may help with the problem of too much data, say Lavanya et al. (2021). The authors discussed issues such as data sparsity, the difficulty of getting started from cold, and scalability. Fifteen prior research projects on movie recommendation systems were analysed in this study. Analysis of the literature led them to the conclusion that most writers made use of collaborative filtering. This result was reached after a thorough analysis of the available data. This conclusion was reached as a result of their research, which started following the first finding. The authors state that many contemporary authors use a "hybrid approach." This fact was noted by the writers. While much research has been done on recommendation systems, more is needed to address the pressing problems of today. And this is true despite a mountain of studies. The presentation (Immaneni et al., 2021) covered the topic of hybrid recommendation methods. To solve the problem of filtering hierarchical information, this proposal combines content-based filtering with collaborative filtering. Customers tastes and interests are taken into account to choose which films to suggest. In its whole, this collection's creative growth is reflective of the film's evolving narrative framework. The level of innovation and imagination shown in this part of the project is unparalleled. The result is a higher quality photograph.

The writer also discussed a collaborative filtering structure, a hybrid recommender system, content-based approaches, and a graph-based recommendation system. There are four recommendations made. Users of social networking sites like Facebook help in learning more about them and their interests over time so that it may better cater to them. Examining the film's audience reaction and making changes is the next phase.

An improved narrative is required to enhance the visuals. Akter Hossain and his colleagues worked together to launch NERS initially. The authors were able to successfully identify the interaction between the two data sets (Hossain et al., 2018). The system's developers say their results are better because they include both conventional statistics and behavioural data. Makers of the system brag that their products are tailor-made for their rivals. Their system was built using both structured and unstructured data. The authors use three distinct estimators to evaluate their strategy in relation to those already on the market.

Collaborative filtering is a collection of methods, all of which are responsible for the two primary functions of rating prediction and Rating. Rating encompasses both processes. In contrast, ranking styles employ the user's implicit input, such as clicks, to generate a ranked list of goods that the user is then encouraged to explore. As a result of the user's actions, entities are ranked in an organisation style (Davidson et al., 2017). More and more researchers are paying attention to collaborative filtering that safeguards user privacy as a means of hiding statistical information while still offering useful leads. This is because collaborative filtering offers suggestions as well. The need to share leads while keeping data private is growing, and that is why it was speculated that many methods may be used to estimate indices without severely infringing on people's privacy. This was done to provide forecasts and make the lives of statistics owners more pleasant. Such solutions erase or greatly reduce the concerns of the statistic owners about the financial implications, legal consequences, and invasion of personal privacy by using remarkable privacy-retaining strategies. This ensures the statisticians have complete authority over their data (Bilge et al., 2013). One of the most pressing problems to be solved in this age of abundant data is how a person may quickly and easily find their favourite movie amid a large library's worth of films. Multiple keywords are one of the most efficient ways to do this. A customised recommendation system may be particularly useful when a user does not already have a specific movie in mind that they want to see (Calandrino et al., 2018).

As part of our investigation, we devised and constructed a mock-up of a movie-recommendation machine. Together, the KNN algorithm and the collaborative filtering algorithm form a knowledge-acquisition system that is applied to the machine's real needs for movie selection (Okkalioglu et al., 2013). In this study, we focus on the randomised response technique, an approach to collaborative filtering for binary facts. Its full title is the randomised response technique. This method ensures that users' anonymity is protected. Using publicly available and publicly available supplementary information, we develop a method based on the second component of privacy to calculate fake binary rankings. Several places online and off provide this data (Kaleli et al., 2015).

They may grow worried with the procedures for prediction creation even if they are provided with privacy safeguards, but this will depend on the specifics. We propose privacy-preserving methods that put an end to concerns held by online retailers over their customers personal information, hence accelerating the spread of predictions derived from the allocated data (Mario et al. 2010). An enormous increase in the recommendation machine's use level may be traced back to the recent developments in e-commerce and the Internet. Here, we compare and contrast the electronic commerce recommendation system, which uses a similar technique and focuses on the same collaborative filtering algorithm, with the successful personalised film recommendation system. The motivation for writing this research is the key role that the collaborative filtering algorithm plays in the individualised film recommendation system (Peng et al., 2015).

The author went on to talk about a recommendation system that uses graphs, as well as content-based methods, hybrid recommender systems, and collaborative filtering frameworks. There are suggested steps for four different situations. Early adopters of social media sites like Facebook are often willing to provide personal information that helps the service better meet their needs. Then, analyse the responses to the film and provide improvements.

**Chapter 3: Methodology**

As a direct consequence of the significant advancements that have been made in data collecting, a new age of information has dawned. The use of data in developing more efficient systems is now taking place, and recommendation systems play a significant role in this process.

**3.1 About Dataset**

**3.1.1 Context**

Over the last 5–10 years, Netflix has successfully attracted a sizable population of viewers. A more significant number of viewers almost always results in an expansion of the available programming options. However, do consumers comprehend how ratings are distributed throughout the many programs on Netflix?

**3.1.2 Content**

As part of this data set, I started by selecting four videos based on four ratings, resulting in 16 different shows. Next, I chose 53 recommended shows for each video. The ratings go from G to PG, then TV-14, and finally TV-MA. I decided not to draw from every Rating (e.g., TV-G, TV-Y, etc.).

The data set is downloaded from the website (HTTPS/data.world/chasewillden/Netflix-shows). This is downloaded from the secondary data source, and it has 1000 records with the following attributes:

* Title
* Rating
* Rating level
* Rating Description
* Release year
* User Rating Score
* User Rating Size

So totally, it has 1000 X 7 records which are equal to the 7000 records.

**3.1.3 Ethical and Social issues**

Within the scope of our proposal, we will adhere to the social and ethical norms, which stipulate that we must not compromise the privacy of any personal information about students, children, or patients. It is fully collected from the secondary resource on the Internet.

**3.2 Rationale steps in the methodology**

The rationale steps and block diagram is shown in the below figure. It has the following blocks in the movie recommendation system.

1. Data collection
2. Data extraction and pre-processing
3. Data analysis and visualization
4. Train and test split
5. Machine learning models
6. Performance Metrics
7. Selection of multiple models
8. Implementation of combined machine learning models

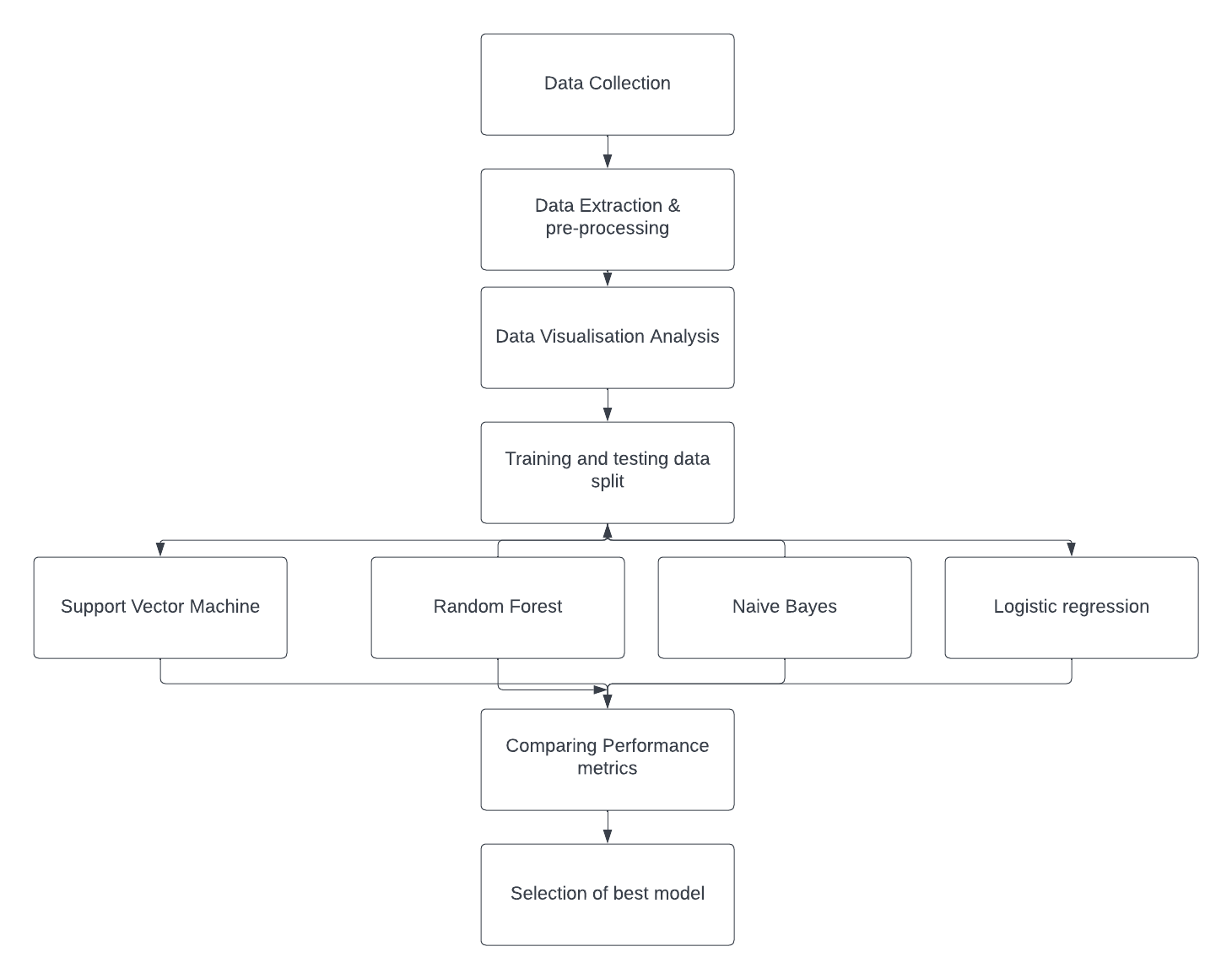


Figure.3.1 Proposed Block diagram

**3.2.1 Data collection**

Information required for business decisions, research, strategic planning, and other purposes may be gathered via data collection. It is essential for research and the analysis of data. The efficient collection of data makes it possible to find answers to problems, evaluate business performance, and anticipate future actions, circumstances, and trends. In organizations, there are often several levels at which data is collected. IT systems collect firm data, including information on customers, employees, and sales, when transactions are processed, and data is recorded. Clients are surveyed, and input is gathered through social media and other online sources. Researchers in data science and analytics and business users collect internal and external data for analysis. Obtaining data for business intelligence (BI) and analytics applications is the first step in the data preparation process, which is currently underway. Researchers in various fields, including medicine, public health, higher education, and others, develop and implement protocols to collect specific data sets. Accurate data is required to provide valuable analytics and research results when conducting research and conducting business.

**3.2.2 Data extraction and pre-processing**

The process of data pre-processing is a method of data mining used to change the raw data into a valuable and practical format.

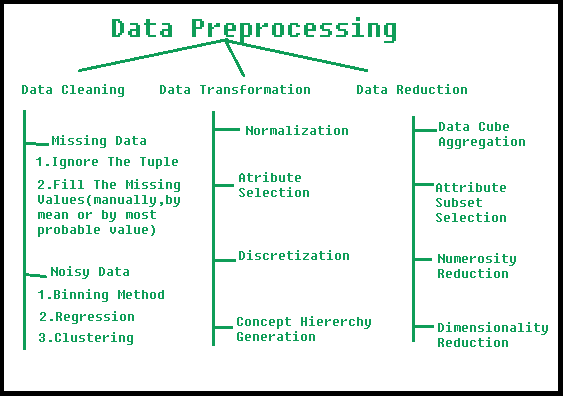


Figure.3.2 Data pre-processing

1. Data Cleaning: The data may lack essential components or have several portions that are irrelevant. To manage this aspect, the data is cleaned. It includes working missing, noisy, and other types of data.

(a). Missing Data: This problem occurs when there are gaps in the data that need to be filled in. It may be approached from several different angles. Among them are the following:

* Ignore the tuples: We should consider using this method in situations where we lack a sufficiently big dataset and a tuple contains many values that aren't complete.
* Fill the Missing values: There are various ways to do this task. You have the option of manually filling in the missing values, using either the attribute mean or the value that is the most likely.

(b). Noisy Data: Data that is noisy is nonsensical data that machines cannot comprehend. Errors in the data gathering process, mistakes made during data input, and other things might lead to its production. The following are some possible approaches to dealing with it:

* Binning Method: This technique works to smooth out the applied sorted data. The whole data is divided into segments of equal size, and then various methods are performed to complete the task. Separate consideration is given to each section. One option for completing the work is to replace all of the data in a segment with the segment's mean, while another option is to utilize boundary values.
* Regression: When this happens, the data may be smoothed out by fitting them to a regression function. The type of regression that is performed could be linear (meaning that there is only one independent variable), or it could be multiple (having multiple independent variables).
* Clustering: This technique groups the data together into a cluster according to the commonalities they share. It's conceivable that the outliers won't be spotted or that they'll be found outside of the groups altogether. Both of these outcomes are feasible.

2. Data Transformation: This phase changes the data into acceptable formats, so they are ready to be used in the mining process. This necessitates taking the appropriate steps.

* Normalization: This is done to size the data values to fit inside a specific range, which may be either -1.0 to 1.0 or 0.0 to 1.0.
* Attribute Selection: To assist the mining process, this tactic involves creating new characteristics by building on top of the existing collection of attributes.
* Discretization: To do this, the raw values of the numeric property are first converted into interval levels or conceptual levels.
* Concept Hierarchy Generation: In this process stage, qualities are elevated from lower to higher levels in the hierarchy. For example, the value of the property "city" may be changed to "country."

3. Data Reduction: Given data mining is a method that may be used for the management of massive amounts of data. When working with a large volume of data, analysis becomes more complex in situations like these. We use the data reduction approach to get rid of this. Its goal is to improve storage efficiency while simultaneously lowering the expenses of data storage and processing. The several stages of data reduction are as follows:

* Data Cube Aggregation: The data go through a procedure called aggregation before being used in the production of the data cube.
* Attribute Subset Selection: Only the very significant traits should be utilized; the rest of them may be ignored. The degree of importance and p-value associated with a characteristic are two tools that may be used throughout the attribute selection process. It is permissible to disregard the part if its p-value exceeds the significance threshold.
* Numerosity Reduction: This makes it possible to keep the data model rather than the actual data itself, as in the case of regression models.
* Dimensionality Reduction: Encoding procedures allow this to result in a size reduction of the data. The lossy or lossless operation, your choice. Such reductions are lossless reductions if the original data can be reconstructed from the compressed data after the compression process; otherwise, they are referred to as lossy reductions. Wavelet transformations and principal component analysis (PCA) are the two successful approaches to dimensionality reduction (Principal Component Analysis).

**3.2.3 Data Analysis and Data Visualisation**

Data visualisation involves expressing information and data visually (For example, charts, graphs, and maps). Data visualisation tools help discover anomalies, patterns, and trends. To examine large amounts of data and make decisions, data visualisation techniques and technologies are needed. Visuals have helped individuals understand things since ancient times. Data visualizations include charts, tables, graphs, maps, and dashboards.

Data analytics is evaluating data sets to make choices based on available information. This method uses more software and systems. Commercial industries employ data analytics to make informed business decisions. Data may help firms better understand their customers, improve advertising, provide tailored content, and increase profitability. Data analytics methods and procedures may now be performed on raw data and utilised by individuals. This opens many doors. This automation used robotic techniques. Data analysis may improve a company's performance.

**3.2.4 Machine Learning Models**

Machine learning models are mathematical representations of training outcomes. Machine learning studies algorithms that improve themselves via experience and historical data. A machine learning model is like computer software that identifies patterns based on prior knowledge or data. The learning algorithm explores the training data for recurrent patterns and builds a machine learning model that saves them and utilizes them to make predictions based on new data.

**3.2.5 Movie Recommendation based on filtering techniques**

**3.2.5.1 Demographic Filtering:** They provide every user generic movie selection based on popularity or category. These suggestions are based on movies popularity or genres. The algorithm recommends the same film to demographically-targeted consumers. This technique is insufficient since everyone has unique traits. This method assumes that the public would favourably appreciate movies praised by crowds and reviewers—this system's core premise.

**3.2.5.2 Content-Based Filtering:** They suggest similar items to yours. In the case of movies, item metadata comprises genre, director, description, actors, etc. These recommender systems assume that if a person enjoyed something, they would also like a similar item.

**3.3 Proposed Flow**

The dataset is collected from internet source, that is directly downloaded from Netflix website, this is a secondary data source because, we are using dataset from internet, which is available, but if we want to use a primary source data means, we need to perform interview or data collection with the users and we should follow the ethics and social legal issues. This case is not possible in our scenario so we are using secondary data source.

Collected data is uploaded to a current working folder, using Jupyter notebook we are exporting the dataset into the data frame and we are performing data pre-process where we need to change the datatypes and handling missing values or incorrect values. Then we perform data analysis on this pre-processed data, where we can be able to get visuals and analysis such as statistical analysis. For an example mean, median, standard deviation and ranges of each column attribute. We also convert the column attributer to find similarity between the TV shows, where we can able to convert text into vector, then the cosine similarity is used to find similarity between two vectors. Then we are using demographic and content-based filtering for the movie or shows recommendations where, we get recommendations based on the user option or liked movie. Then we are approaching machine learning techniques to find the rating and scores of the movies and shows. In the next step, we are performing train and test split for training and testing dataset, and then the four machine learning algorithms are modelled using SVM, Random Forest, Naïve Bayes and Logistics regression, then we are finding the performance metrics of the models.

**Chapter 4: Implementation**

In this section, we are going to implement combined machine learning algorithms where those are used to predict the Rating of movies and TV shows. And used for recommendations.

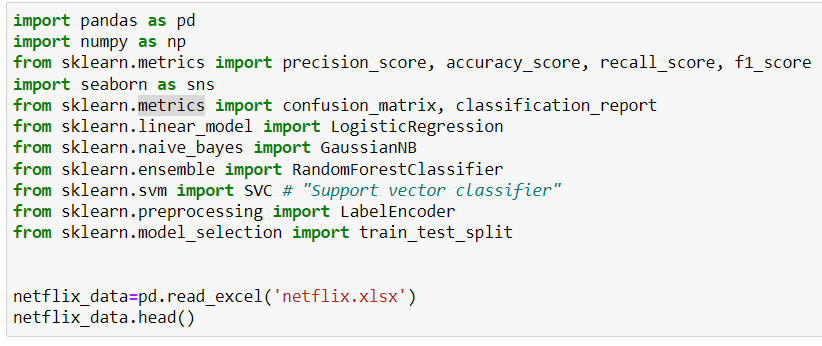
****

Figure.4.1 Importing Library Packages and Export Dataset

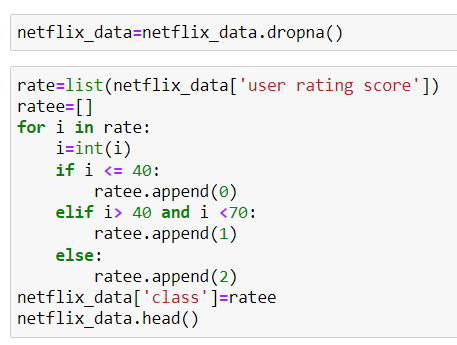
****

Figure.4.2 Missing values and creating the class

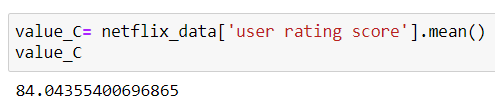


Figure.4.3 Finding average rating score

The total average Rating for all of the films is around 60 on a scale of 1 to 100. The next step is to find an appropriate value for m, which stands for the minimum number of votes required to be included in the chart. We will choose the 90th percentile as our cut-off. To put it another way, a picture must have gotten more votes than at least 90% of the other films on the list in order to be included in the rankings.

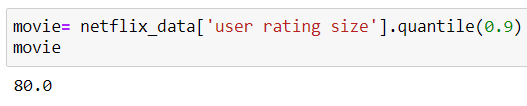


Figure.4.4 Average user rating size

With this knowledge, we can narrow our choices to only those films that qualify for the chart.

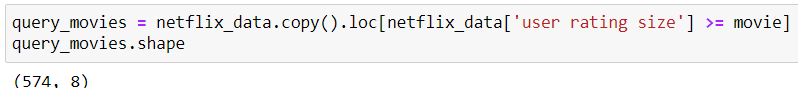


Figure.4.5 Copying data frame based on the user rating size.

It has come to our knowledge that a total of 574 movies satisfy the criteria to be listed here. We need to determine our metric for each qualifying movie now that we have them all. We will first create a method called weighted Rating () and then create a new feature called score to achieve this. We will use weighted Rating () to our Data Frame of qualified movies to determine the value of the score feature.

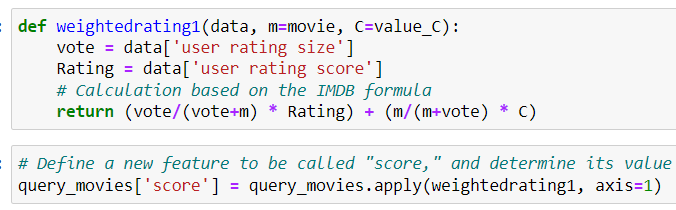


Figure.4.6 Weighted Rating

Finally, let's sort the Data Frame using the score function before displaying the top 10 movies' titles, vote totals, vote averages, and weighted ratings.

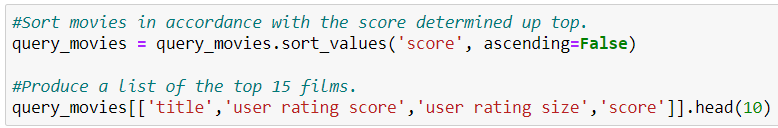


Figure.4.7 Sorting movies based on attributes

Hurray! Even though it is quite basic, we have developed our very first recommender. These systems' "Trending Now" tab allows us to find movies that are now experiencing a rise in popularity; all that is required to access these films is to sort the dataset according to the popularity column.

Based on the summary of each film's narrative, we will calculate pairwise similarity scores for each one, and then we will base our suggestions for films on those results. The plot description is part of a feature in our dataset called "overview" that is included in the dataset. Let's examine the data we currently have.



Figure.4.8 TV shows based on rating level

Everyone in this room which has even a passing familiarity with text processing knows that we must transform the word vector of each overview. For each overview, we will now create Term Frequency-Inverse Document Frequency (TF-IDF) vectors. Term Frequency-Inverse Document Frequency is referred to as TF-IDF.

If you are wondering what term frequency is, it refers to the relative frequency of a word inside a text and is calculated as the ratio of the term's frequency to the word's overall frequency. The Inverse Document Frequency is calculated using the formula log (number of documents/documents with the term). The relative number of documents that include the phrase is represented by this figure.

Each words overall significance in relation to other words in the documents in which it appears is equal to TF times IDF. The same as previously, this will give you a matrix where each row stands for a movie, and each column represents a term from the overview vocabulary (all the words that are found in at least one document). The purpose of doing this is to reduce the importance of words that are repeated frequently in plot summaries and, as a result, their weight in the total computation of similarity. You will be glad to find that scikit-learn comes with a built-in TfIdf Vectorizer class that quickly and easily creates the TF-IDF matrix.

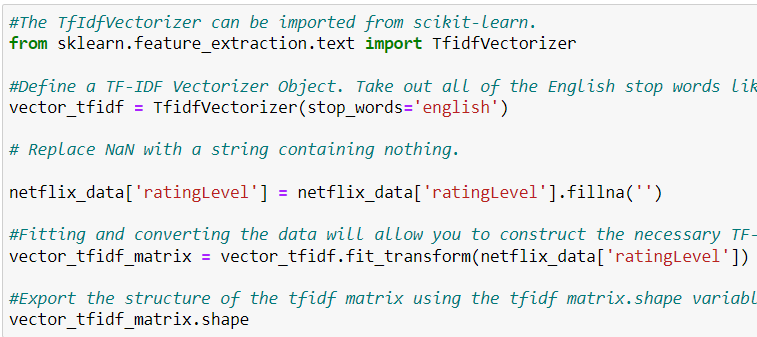


Figure.4.9 Vectorizer

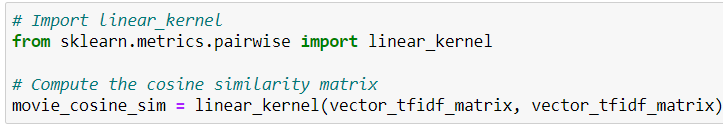


Figure.4.10 Cosine Similarity Index

We're going to create a function that accepts a movie title as an input and returns a list of the ten films that are most comparable to it. First, we require a reverse mapping between data Frame indices and movie titles. In other words, we require a method to recognize, given a movie's title, the index of the film in our metadata data Frame.

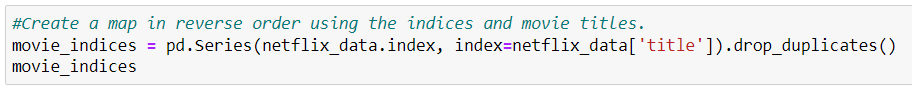


Figure.4.11 Movies Indices based on movies title

At this moment, an acceptable definition of our recommendation function may be made. The stages that we shall keep going through are as follows:

Using the movie's title as a guide, find the index. Get a list of the cosine similarity scores contrasting that particular film with all of the other films. Transform it into a list of tuples, with the location as the first member and the similarity score as the second. The second component is to rank the tuples in the aforementioned list according to their similarity scores.

The top 10 items on this list should be noted. Ignore the first element as it concerns you (yourself) (the movie most similar to a particular movie is the movie itself). If you could provide the titles that match the top items indices, that would be useful.

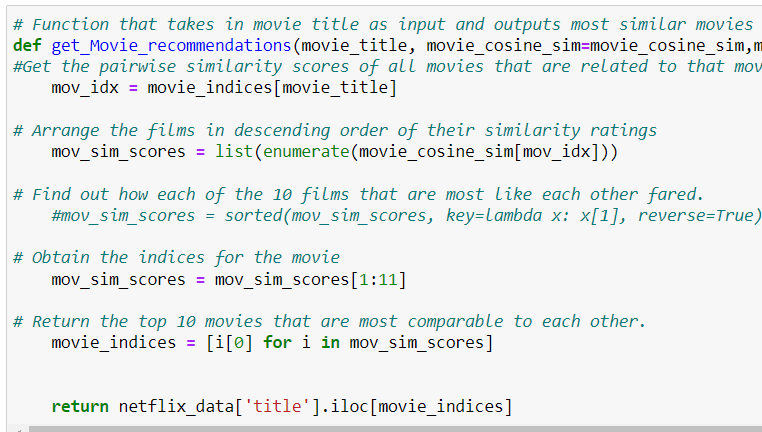


Figure.4.12 Get movie recommendations.

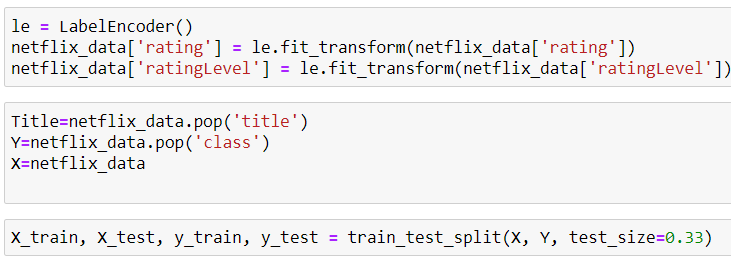


Figure.4.13 Label Encoder & Train and Test Split

**4.1 Support Vector Machine**

The Support Vector Machine, or SVM for short, is a well-known technique of supervised learning that is used to tackle problems involving classification and regression. This approach was developed at Stanford University in the 1980s. The bulk of its applications, on the other hand, are in categorization difficulties and include machine learning. Finding the best line or decision boundary that divides an n-dimensional space into classes is the goal of the technique known as support vector machine, or SVM for short. Because of this, we will be able to easily categorise any new data points that are brought to our attention in the future. The well-defined decision boundary is referred to as a "hyperplane," which is a word used in computer science. The support vector machine (SVM) is used to identify the extreme vectors and places that will be contributing to the construction of the hyperplane. Because of how well it works in such difficult conditions, the method is often referred to as a "support vector machine."

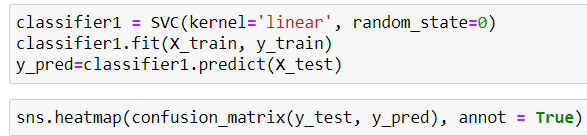


Figure.4.14 Support Vector Machine

**4.2 Random Forest Classifier**

In the field of machine learning, an approach called supervised learning makes use of an algorithm called Random Forest. This algorithm is well-known and widely employed. In the field of machine learning, the use of this method might be helpful in finding solutions to classification and regression problems. It is dependent on the idea of ensemble learning, which is a strategy that includes integrating several distinct classifiers in order to find a solution to a challenging problem and improve the model's overall performance.

According to what its name suggests, Random Forest is an ensemble of unpruned classification or regression trees created by bootstrapping training data samples and random feature selection in tree induction. (Svetnik V et al.,2003). The Random Forest algorithm is a classifier that works by constructing a number of decision trees based on different sections of the information that is provided. In place of relying on just one decision tree, the random forest takes the projections that have been made by all of the decision trees and compiles them into a single set. From there, it determines which projections have the most votes and uses that information to make a prediction about the result.

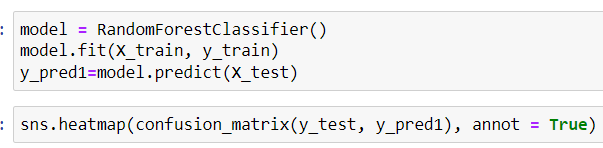


Figure.4.15 Random Forest Classifier

**4.3 Naive Bayes**

The Bayes theorem provides the foundation for the Naive Bayes algorithm, which is a kind of supervised learning that may be used to classification problems. Most of its applications revolve on text classification using extensive data sets for training. One of the classification algorithms that is currently accessible, the Naive Bayes Classifier is known for being one of the simplest and most effective. It is helpful in the process of developing quick machine learning models that can make correct predictions.

As a probabilistic classifier, it determines the probability that an event will take place and then makes predictions based on that likelihood. A few examples of common applications for the Naive Bayes algorithm include the filtering of spam, the examination of sentiment, and the categorization of articles.

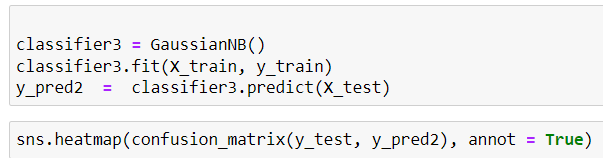


Figure.4.16 Naïve Bayes

**4.4 Logistic Regression**

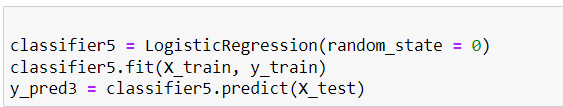


Figure.4.17 Logistic Regression

One of the most common forms of Supervised Learning is known as logistic regression. It uses the independent components to make a prediction about a categorical dependent variable. A categorical output may be predicted using logistic regression. It is required that the output be either discrete or categorical. You may answer with Yes or No, 0 or 1, true or false, etc., but probabilistic values fall somewhere between 0 and 1. The Logistic Regression method is similar to the Linear Regression method; however, it is used in a different way. Problems with regression can be solved using linear regression, whereas problems with classification may be solved using logistic regression. Logistic regression corresponds to a logistic function in the form of a "S," which predicts two maximum values (0 or 1). The curve of the logistic function may determine whether or not cells are cancerous, whether or not an animal is obese based on its weight, and so on. Logistic Regression is an effective method of machine learning that can classify both continuous and discrete data sets.

**Chapter 5: Software and Tools**

The required software, tools, and library packages for this proposed work are listed in the below list.

* Python
* Anaconda
* Jupyter Notebook
* NumPy
* Pandas
* Matplotlib
* Sklearn
* Seaborn

**5.1 Python**

Python has been a popular research language for the last decade. Its large online community and free, open-source nature have helped it succeed. Python has improved research productivity in several domains, including data science, AI, and scientific research. This article helps users start Python by installing and configuring Anaconda. Users will be led through developing their first script to learn Python. This lesson includes a quick review of Python, directions for obtaining Anaconda, a breakdown of the installation's components, and an example of how to execute a Python script.

Python is object-oriented, interpreted, and mid-level. It's easy to learn and use but adaptable enough to tackle many tasks (Helmus & Collis, 2016). Since 1991 (Van Rossum & Drake Jr., 1995), its open-source nature has skyrocketed its popularity, and it is now considered one of the greatest programming languages to learn (Saabith, Fareez, & Vinothraj, 2019).

Python is an open-source programming language that anybody may learn for free and with low system requirements. This community has created data science, machine learning, AI, app and game development, and scientific research initiatives. Because of the open-source community, you may search for these projects by using the phrase "Python." This makes identifying projects easy. This community includes courses, source codes, issue solutions, video tutorials, and more.

**5.2 Anaconda**

The Anaconda is a kind of snake that... The open-source program known as Anaconda provides you access to a toolkit that has been developed with the express purpose of facilitating scientific research and analysis. After installing Anaconda, you will have access to a range of environments, each of which allows you to develop code in either Python or R. These environments will become available to you after Anaconda has been installed. These environments, which are sometimes referred to as integrated development environments (IDEs) on occasion, are software programs or platforms that simplify the process of creating code. IDE stands for an integrated development environment. They serve a role that is analogous to that of text processors such as Microsoft Word, Google Docs, and Pages when it comes to the creation of text; yet, in reality they are a great deal more than that. Integrated development environments, or IDEs, have a wide variety of helpful features that can be used to write, modify, and debug code, view and examine data; save variables; show findings, and collaborate on projects. Some of these features include viewing examining data, viewing and examining data, saving variables, showing findings, and collaborating on projects. The programming language that is used does not change, despite possible changes in the presentation and other components of an integrated development environment (IDE). Therefore, changing the Python integrated development environment (IDE) that Anaconda uses will not cause substantial changes to the code that you have previously written. The bulk of the time required to climb the learning curve is spent getting comfortable with the syntax of Python. When you have mastered coding in one integrated development environment (IDE), transferring those talents to another IDE won't be too difficult for you. It is not always the case that one IDE is preferable to another; each one has its own set of benefits and drawbacks, and the option of which one to use comes down, in the end, to individual preference.

**5.2.1 Anaconda Navigator and IDEs**

As was indicated before, the Anaconda Navigator has a graphical user interface (GUI), as can be seen in figure 4. Imagine this as Anaconda's "menu" page, which allows you to quickly access and run a variety of integrated development environments (IDEs) without the need to write instructions into a terminal window. There are several other integrated development environments (IDEs) that have been specifically built for Python programming, and Anaconda will install the following IDEs for you by default. Simply click on the IDE icon located on the home tab of the Navigator to open it.

**5.2.2** **Spyder**

Spyder is a strong integrated development environment (IDE) that focuses on data analysis. It is feasible to conceive of it as an integrated development environment (IDE), which comprises a console for Ipython programming, an editor for authoring code scripts, a variable explorer for quickly reviewing data, as well as a lot of other features. In addition to this, it is equipped with extremely efficient tools for code inspection and debugging, which allow the user to examine the code in its whole or line by line in order to identify and fix any errors that may have been introduced. You have a lot of say over the way the interface looks since you can modify things like the position of the different panes and even the color scheme of the layout. You also have a lot of control over the functionality of the interface (we recommend Spyder Dark). You may get further information on the software by going to the Spyder website or reading through the Spyder documentation.

**5.3 The Jupyter Notebook as well as the Jupyter Lab**

Jupyter Notebook is an integrated development environment (IDE) that you may access using the web browser you normally use and which runs in the cloud. Because every portion of code may be run independently, the system offers a high degree of flexibility and is not difficult to manipulate in any way. Because of this, it is possible to use a wide range of text formats while working inside the limits of a single Notebook. As a result of this, code outputs, visualizations, equations, and plain text may all be used in the same spot at the same time.

This enables it to be feasible to successfully produce and share documents, as well as to exhibit your code and results in a manner that is both well-organized and aesthetically attractive. Additionally, this makes it possible to display your findings in a way that is visually appealing. Not only is it easy to share Notebooks with other people because of the fact that it is web-based, but it also works exceptionally well for group projects and collaboration thanks to its excellent functionality. Jupyter Lab is an extension of the Jupyter Notebook that adds a significant amount of new capability. It is a part of the Jupyter ecosystem. Within Jupyter Lab, Jupyter Notebooks may be integrated with a variety of additional applications, including a command-line Terminal, a code Console, and a Text Editor, amongst others.

**5.4 Numpy**

When it comes to scientific computing, the Numpy package is the Python library that is considered to be the most significant. This Python package includes not just an object that can store data in many dimensions but also array operations methods and derivative objects (such as masked arrays and matrices). In addition to that, an object that is a multidimensional array may be utilized instead. This topic includes a variety of subjects, including mathematics, logic, manipulating shapes, sorting, selecting, input/output, discrete Fourier transforms, basic linear algebra, rudimentary statistical operations, and random simulation, among others.

The array object is the fundamental component of the Numpy programming language. This comprises arrays of different data types that are equal to one another, some of which may be n-dimensional, and compiled code is used for the majority of the operations. The following is a list of a few of the most important distinctions that can be made between NumPy arrays and Python sequences:

In contrast to Python lists, the size of a NumPy array is not variable at any point (which can grow dynamically). Memory utilization of a NumPy array is constant regardless of the array's size since all of the array's members are required to have the same data type. When the dimensions of an array are changed, the previous array is replaced with this new one, and the old one is discarded. Arrays that are constructed using Python and NumPy objects are the solitary and one exception to this rule. NumPy arrays make it much simpler to do complex mathematical calculations and extensive data manipulations on a big scale. When using Python's built-in sequences, it is frequently necessary to write extra code in order to get the same degree of efficiency that might be reached by making direct use of the sequences themselves. This is because using Python's built-in sequences is typically inefficient. These Python-based programs do take Python sequences as input, but before the processing can begin, the sequences must first be transformed into NumPy arrays. Recent years have seen an increase in the number of Python-based scientific and mathematical programs that make use of arrays created using NumPy. In addition to being comfortable with Python, one has to be knowledgeable with NumPy arrays in order to utilize a considerable number of scientific and mathematical applications that are built on Python effectively. This may even be the case for the majority of these apps. Arrays created using NumPy are used by the overwhelming majority of applications.

**5.5 Pandas**

Managing "relational" or "labeled" data is made much easier with the assistance of Python's panda's module, which offers data structures that are brisk, diverse, and expressive. This makes working with such data much more manageable. It is a high-level building component for data analysis in Python that has the potential to be employed in the real world. Its mission is to evolve into an open-source data analysis and manipulation tool that is superior to all others in terms of its capabilities and power, and this is regardless of the language used. At this stage, everything is going forward at a rather brisk pace.

Pandas can analyze many different kinds of data types.

* Information is presented in a tabular style with several column types comparable to those found in Excel or SQL; this may be the case.
* Time series in both an ordered and an unsorted format to choose from (though not always with a set frequency).
* Data is stored in the form of a coded matrix with row and column labels, which may include additional statistical or observational information.
* Pandas are able to store data that has not been labeled.

Both Series (1-dimensional) and DataFrame can handle most financial, stats, ss, and engineering use cases (2-dimensional). DataFrame accesses R's data. The frame offers the features listed. NumPy's basis makes Pandas interoperable with many scientific packages.

The following is a list of the panda's strengths:

* Higher-dimensional DataFrame elements and columns may be added or removed.
* Missing data is easy to handle in floating point and non-floating point formats (represented as NaN). Series, DataFrame, and other data structures may automatically align data if the user ignores labels.
* A strong, adaptive group may split, employ, and combine data set activities to obtain and transform data.
* Create DataFrame objects with ad hoc indexing from Python and NumPy data sets.
* Massive data sets need label-based slicing, indexing, subsetting, combining, and merging.

These powerful IO tools handle Excel, databases, flat files (both CSV and delimited), and the speedy HDF5 format, amongst other file types. Flexibility in data collection manifests itself in the form of the capacity to rotate and reformat the labels on hierarchy axes (the possibility to have multiple labels per tick).

In terms of time series characteristics, some examples include lagging, moving window statistics, date shifting, date range creation, and frequency conversion. These proposals provide responses to challenges that come up while doing scientific research employing a variety of languages. Data scientists will gather the data, clean it, analyze it, model it, and then plot or table the findings of the study while they are working with it. Each of the pursuits requires the participation of pandas.

**5.6 Matplot**

Matplotlib is a library for the Python programming language that provides data visualization and graphical charting capabilities. It is an effective substitute for the MATLAB program. The Matplotlib APIs provide graphical user interfaces (Application Programming Interfaces). Using a Python script that calls on the matplotlib module, a graphical data plot might be created in only a few lines of code. Matplotlib has support for two different application programming interfaces inside its scripting layer.

The Matplotlib library is at the pinnacle of the Python code object hierarchy, that is, the Pyplot Application Programming Interface (API). This hierarchy also includes additional OO API components that are easier to create; these components are more straightforward. Through the use of this application programming interface (API), you will have access to the Matplotlib backend layers. Both Python's Matplotlib and Pyplot MATLAB stand to benefit from the stateful interface that is made available by pyplot. matplotlib is a free and open-source alternative to MATLAB that you may use instead. The user interface (UI) and application programming interface (API) of OO are more robust and diverse than those of pyplot; nonetheless, OO is more difficult to use. Given that the pyplot interface is the approach that is used most often, it will be utilized for the whole of this session.

It is necessary for you to be acquainted with the matplotlib pyplot API in order to be able to deal with graphs. The figure is the name of the primary container inside Matplotlib and plot. There is at least one axis to the plot, and there may be more than one. Axes, which are used to convey coordinates, may be represented by a number of different notations, such as Axis, Tick, Line2D, and Text. A conversation is taking place about the visualization of data. X-, Y-, and Z-axes are axes.

**5.7 Scikit-learn**

The scikit-learn package contains both supervised and unsupervised methods of learning Python. Both of these approaches are covered. It is provided with the Linux operating system and a broad, simplified version of the BSD licence, both of which allow both academic and commercial usage of the software. In order to use the scikit-learn package, SciPy must first be installed on your computer.

* The NumPy standard library for n-dimensional arrays is one of the objects that can be found in this particular category.
* SciPy for mathematical symbols is a difficult programming language.
* Data structures and analysis by using the SciKits Pandas SciPy modules or extensions;
* Comprehensive 2D/3D charting by utilizing Matplotlib's extended interactive terminal. Therefore, the scikit-learn module provides teaching strategies.

The reliability and support that is necessary for the library to be used in production systems are expected to be provided by the library. In order to do this, it is necessary to place a high value on efficiency, teamwork, documentation, and the readability of code. Despite the presence of the Python interface, performance may be improved by using C-library for array and matrix operations. Examples of such libraries are LibSVM, LAPACK, and NumPy.

**5.8 Seaborn**

Users of Python who are interested in making statistical visuals may make use of the Seaborn module that is made accessible to them. This is accomplished by expanding the features offered by matplotlib and by integrating closely with the data structures that are made available by pandas. Seaborn makes it easier for you to analyze and comprehend the data you're dealing with, saving you time and effort in the process. Its charting methods work on data frames and arrays that encompass whole datasets, and in order to generate meaningful charts, it accomplishes the essential semantic mapping and statistical aggregation on its own internally. These routines can operate on arrays as well as data frames. These functions perform their operations on arrays and data frames that have been provided to them as sources of data to deal with. Because it is based on datasets and has a declarative application programming interface, you will be able to concentrate more on the significance of the various components of your plots rather than on the technicalities of how to create them. This will allow you to get a better understanding of your data (API).

**Chapter 6: Exploratory Analysis and Visualization**

The exploratory data analysis has been performed using excel, where we have obtained the visual as follows.

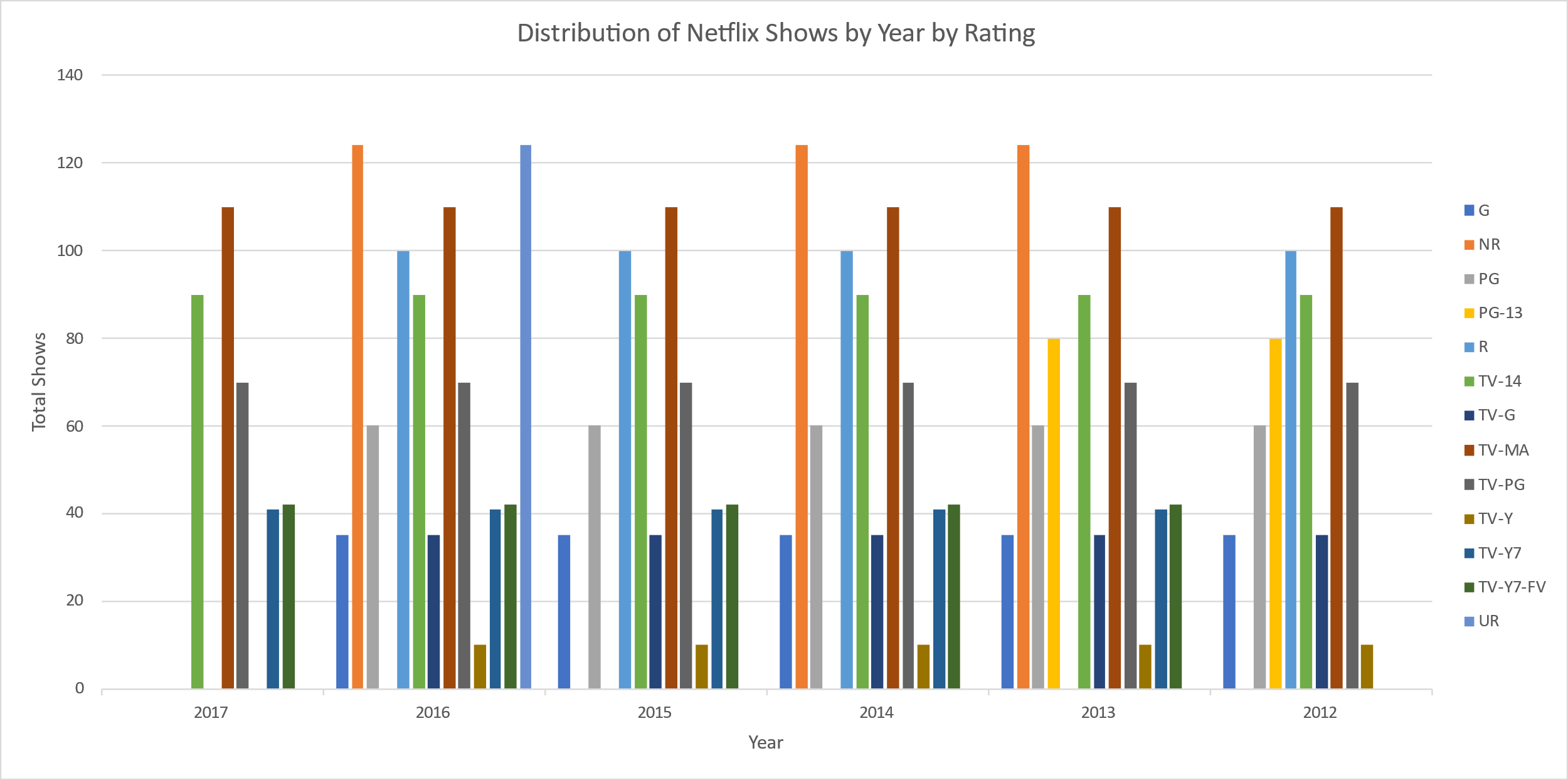


Figure.6.1 Distribution of Netflix Shows by Year by Rating

The above figure shows a visual of the distribution of Netflix shows by year by Rating, where the Rating is given based on the type of TV shows and movies.

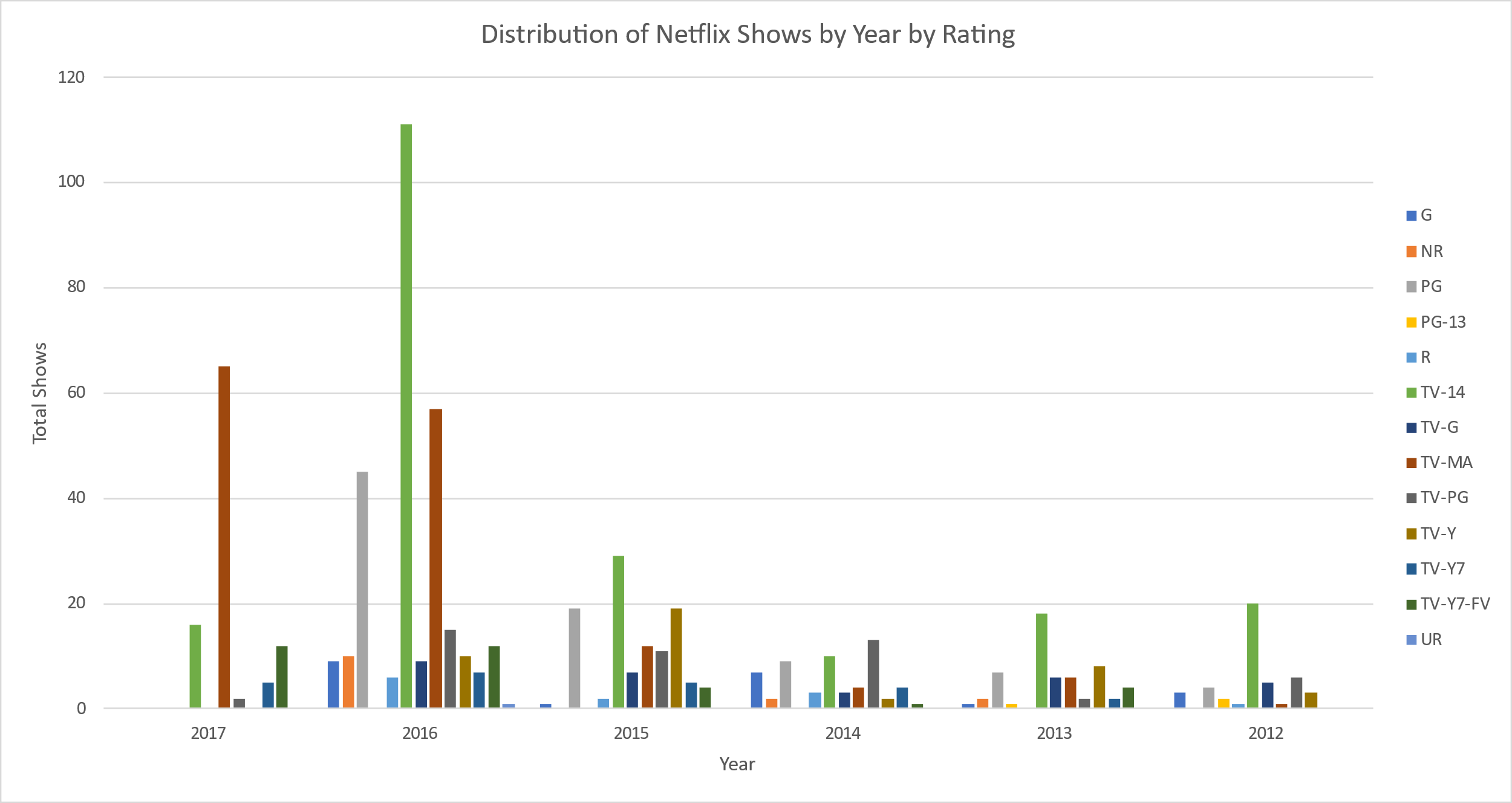


Figure.6.2 Distribution of Netflix Shows by Year by Single Rating

The above figure shows a visual of the distribution of Netflix shows by year by single Rating.

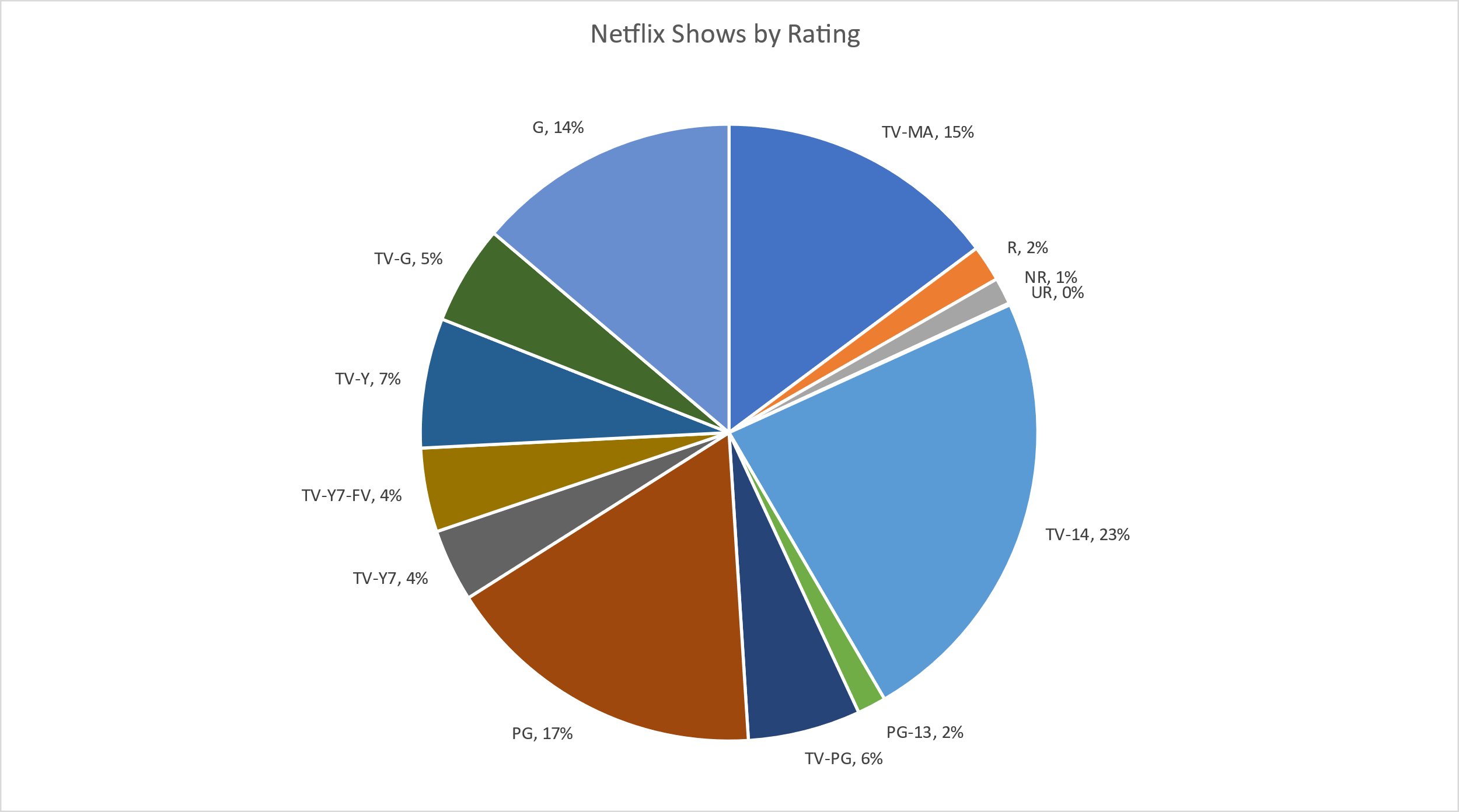


Figure.6.3 Netflix shows by Rating

The above figure is used to know which kind of rated movies or TV shows have more Ratings, where TV-14 has the highest rating than others.

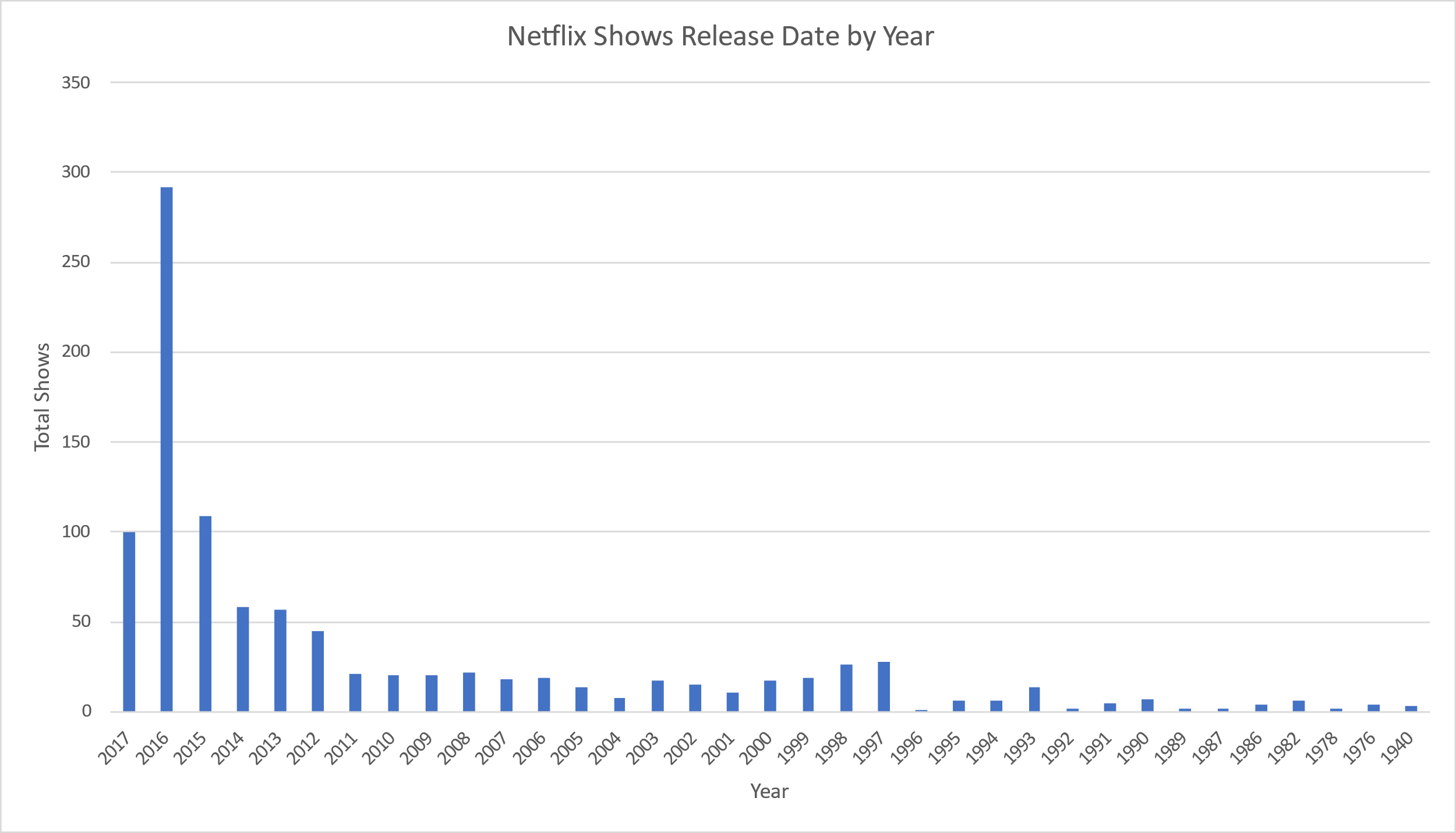


Figure.6.4 Netflix Shows Release Date by Year

The above figure shows Netflix shows release date by a year; from this visual, we can see that there is a gradual increase in the number of movie releases per year. And 2016 has the highest number of movies released than others.

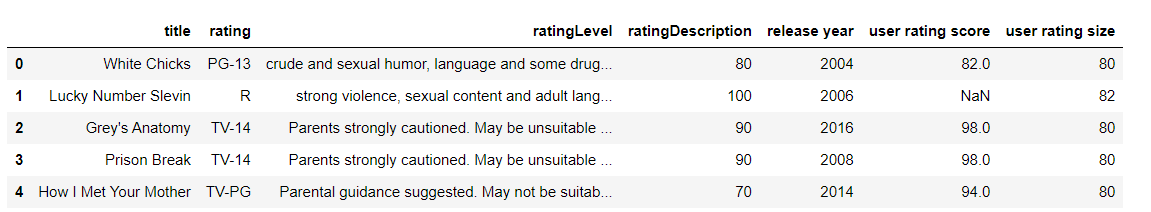


Figure.6.5 Sample Netflix dataset

The above figure shows the sample dataset of the Netflix movies shows with its column attributes. A new column name was created using a class for the classification, as shown in the below figure.

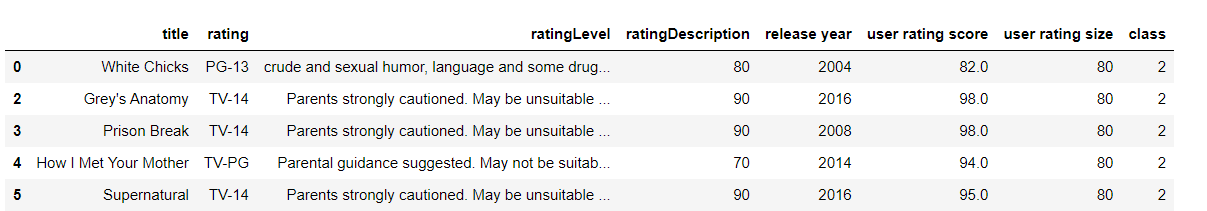


Figure.6.6 Data frame after adding class attributes

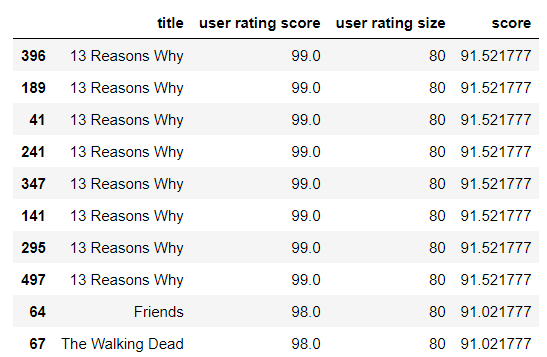


Figure.6.7 Movie shows based on Ratings.

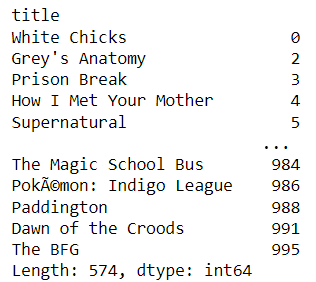


Figure.6.8 Movies Indices

The recommended movies list for the input show “THE PRISIONER PARK” is shown in the below figure.

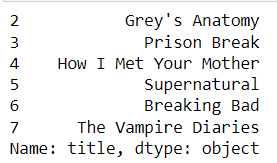


Figure.6.9 Recommended movies

The below four figures shows the confusion matrix for all four machine learning algorithms.

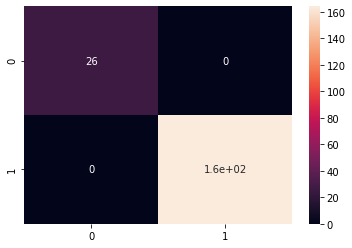


Figure.6.10 Confusion matrix for Support vector machine

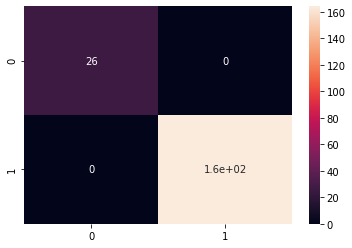


Figure.6.11 Confusion matrix Random Forest

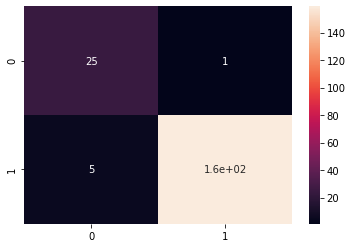


Figure.6.12 Confusion matrix for Naïve bayes

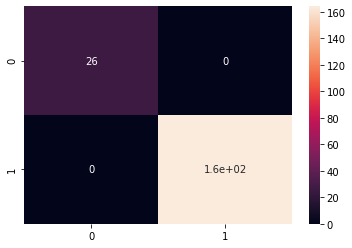


Figure.6.13 Confusion matrix for Logistic Regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SVM | Random Forest | Naïve Bayes | Logistic Regression |
| Accuracy | 100 | 100 | 97 | 100 |
| Precision | 100 | 100 | 83 | 100 |
| Recall | 100 | 100 | 96 | 100 |
| F1-Score | 100 | 100 | 89 | 100 |

Table.1 Classification Report

The above table shows the performance matrix for the SVM, RF, and NB, and Logistic regression such as accuracy, precision, recall, and F1-score for the movie's recommendation and rating prediction. From the above results, Naïve Bayes provides 97% accuracy, whereas the other three algorithms provide 100% accuracy. So we can use any of these algorithms.

**Conclusion:**

Netflix is one of the most widely accessed “Over the Top” (OTT) platform that provides various internet entertainment services for watching movies and television shows. Netflix broadcasts in 30 languages and in around 190 countries all over the world. Netflix maintains a humongous lists of movies, shows, documentaries, etc. Since Netflix is a cloud-based application, it is not possible to project the entire available data consisting of movies, shows to the end user and moreover it is unnecessary to the consumer as well. Because of this whenever a consumer wants to watch a movie/show/documentary and cannot decide on what exactly he/she wants to watch, Netflix using a recommendation system analyses the user watching behaviour and make predictions from the acquired data using various machine learning algorithms and only project the movies, shows that are most likely to be watched by the user since these predictions are made based on the user requirements and previous behavioural patterns. In summary, we can summarize Netflix adapts to recommendation system based on user initial requirements, make predictions, and visualize the data as in movies, shows which are most likely to be watched by the user.

Though there have been many researches on several recommendations’ framework using various techniques such as collaborative filtering, categorization filtering, Hybrid approach. These approaches have their shortcoming and lack the accuracy in providing the most reliable suggestion to the user. In the study, the main aim is to develop a recommendation system that can predict and project more accurate suggestions to the user based on requirements.

Here, we devised and constructed a recommendation system which uses KNN (K nearest neighbour Algorithm) and collaborative filtering algorithm both forming a knowledge -acquisition system which helps in movie suggestions to the user. Random Responsive technique is an approach of collaborative filtering for binary facts is used. Here we are utilised the Netflix dataset that has been collected through internet sources. Data cleaning, Data processing and data analysis on processed data is conducted. Visualized data is the made used to train the machine learning algorithms and based on the results the algorithm with high accuracy rate is advised.

We can say that the research aims are achieved. We critically analysed the Netflix data set containing data of various movies and shows. From the performance metrics of each of the four machines learning models we can say that all the three machine learning models as in Support Vector Machine, Logistic Regression and Random Forest models have returned 100% accuracy in projecting movie, show suggestions that are most likely to be viewed by the user. Therefore, we can use any of the three models to the make predictions and provide suggestions to the user.

**References**

[1] Choi, Sang-Min, Sang-Ki Ko, and Yo-Sub Han. "A movie recommendation algorithm based on genre correlations." Expert Systems with Applications 39.9 (2012): 8079-8085.

[2] Lekakos, George, and Petros Caravelas. "A hybrid approach for movie recommendation." Multimedia tools and applications 36.1 (2008): 55-70.

[3] Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." International Journal of Computer Applications 160.7 (2017).

[4] Zhang, Jiang, et al. "Personalized real-time movie recommendation system: Practical prototype and evaluation." Tsinghua Science and Technology 25.2 (2019): 180-191.

[5] Rajarajeswari, S., et al. "Movie Recommendation System." Emerging Research in Computing, Information, Communication, and Applications. Springer, Singapore, 2019. 329-340.

[6] Ahmed, Muyeed, Mir Tahsin Imtiaz, and Raiyan Khan. "Movie recommendation system using clustering and pattern recognition network." 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2018.

[7] Arora, Gaurav, et al. "Movie recommendation system based on users’ similarity." International Journal of Computer Science and Mobile Computing 3.4 (2014): 765-770.

[8] Subramaniyaswamy, V., et al. "A personalized movie recommendation system based on collaborative filtering." International Journal of High-Performance Computing and Networking 10.1-2 (2017): 54-63.

[9] Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (this) 5.4 (2015): 1-19.

[10] R. Lavanya, U. Singh, and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, DOI: 10.1109/ICAIS50930.2021.9395759.

[11] N. Immaneni, I. Padmanaban, B. Ramasubramanian, and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, DOI: 10.1109/ICACCI.2017.8126171.

[12] M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (IEEE ICT), 2018, pp. 443-448, DOI: 10.1109/CEEICT.2018.8628128.

[13] Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use. DOI: 10.1051/,04008 (2017) 712012ITA 2017 ITM Web of Conferences Autoconf/201 40084 In Proc. of the Ca RR 2011. New York: ACM Press, 2011. 19- 22.http://dl.acm.org/citation.cfm?id=1961639[doi:10.1145/1961 634.1961

[14] Bilge, A., Kaleli, C., Yakut, I., Gunes, I., Polat, H.: A survey of privacy-preserving collaborative filtering schemes. Int. J. Softw. Eng. Knowl. Eng. 23(08), 1085–1108 (2013)

[15] Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.:(2018) You might also like the privacy risks of collaborative filtering. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 231–246, Oakland, CA.

[16] Okkalioglu, M., Koc, M., Polat, H.: On the discovery of fake binary ratings. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC 2015, pp. 901–907. ACM, USA (2015).

[17] Kaleli, C., Polat, H.: Privacy-preserving naïve bayesian classifier based recommendations on distributed data. Comput. Intell. 31(1), 47–68(2015).

[18] Munoz-Organero, Mario, Gustavo A. Ramíez-González, Pedro J. Munoz-Merino, and Carlos Delgado Kloos. "A Collaborative Recommender System Based on Space-Time Similarities", IEEE Pervasive Computing, 2010.

[19] Peng, Xiao, Shao Liangshan, and Li Xiuran (2015). "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie.