Capstone Project

Netflix

Kavitha Palraj
kavipalraj.9@gmail.com

Table of Contents

Introduction

Netflix is an online media streaming platform which provides users with an abundance of movies, TV shows and documentaries (Lifewire, 2024). Despite this ease access, customers often face difficulty when attempting to find content aligned to their interests and preferences. To address this problem a recommendation system should be built that tailors movie suggestions to customer based on their previous usage data.

a. Netflix's Business Model
   Originally starting as a DVD rental business in 2007, Netflix shifted its fundamental business model to a subscription video on-demand over-the-top streaming service. This shift coincided with the availability of new technologies to make this possible along with the benefits this new business model provided. Today, Netflix has over 230 million users from 190 different countries (Study, 2023). This large-scale success experienced by Netflix comes down to a few key factors.

b. Reasons for this success
   Netflix provides the ability for people to enjoy watching movies, TV shows and documentaries from the comfort of their own home rather than in theatres. They are given access to content from various language, genre and cultural backgrounds which vary to their own. This customer centric experience provided by Netflix directly contributed to the success of the business and means that user experience and customer satisfaction are key points Netflix has had to focus on.

c. Key Value Propositions
   These points are addressed through the implementation of a user-friendly interface and more importantly the use of a content recommendation algorithm. This algorithm is one of the key value propositions provided by Netflix which suggests to users which movies, TV shows or documentaries they should watch next. Providing customised recommendations in this manner has helped significantly in customer retention and revenue generation for the business (Bstrategyhub, 2023).

1. Problem Statement

For the purpose of this project a Recommendation Engine will be created where every single user would be recommended a list of movies that are best suited for them based on their area of interest and ratings.

2. Project Objective

The objective of this project is to create a model to predict and provide users with the best suited movie recommendation. This will be provided based on features of the dataset and latent factors generated from the SVD algorithm. Alongside this main objective, additional aspects will be analysed such as the list of most popular movies, the number of movies released each year, highest and lowest movies released each year and how the ratings differ based on the movie and user.

3. Data Description

a. Overview

The dataset provided contains a list of movies from Netflix along with the 'Date' they were released, their 'Rating' and their respective 'Name' and 'ID'. This information has been split into two tables which can be seen below.

b. Dataset 1

Variable	Description	Data Values
Cust_Id	Unique identification number associated with each customer in the dataset.	6 to 2,649,429
Rating	User rating for each movie	The movie has been allocated ratings within the scale of 1 to 5

c. Dataset 2

Variable	Description	Data Values
Movie_Id	Unique identification number associated with each movie in the dataset.	1 to 4,499
Year	Indicates the year for the respective movie.	1876 to 2005
Name	Name of each movie.	17,358 different names

d. Insights from the data

A few insights can be made from the data before the following pre-processing, EDA and model building steps. As mentioned above, the data contains information about 2,649,429 individuals. The Netflix streaming data has 4,499 movies made available November 1999 to December 2005 each with different ratings. There are a total of 24,053,764 ratings.

4. Inspiration

As mentioned previously, the use of a Recommendation System will help in the retention of customers for Netflix and the attraction of new customers. As the business expands through this, more employees will be attracted to the company leading to further revenue generation. With the increase in revenue for Netflix, money can be invested into the business to improve services for customers. For example, more advanced algorithms could be created, thereby staying at the forefront of technological innovation and maintaining competitive edge in the market. Value-added features could also be introduced at higher price points and bundle offerings with strategic partnerships could be explored.

Data Pre-processing Steps

Data pre-processing is an important step in improving the reliability and interpretability of the dataset (IBM, 2022). **Exploratory Data Analysis (EDA)** helps in showing the initial structure of the data. **Data Cleaning** is included within EDA which ensures the data is well structured and free from issues such as outliers and missing values. This enables us to analyse the data to a higher accuracy and create machine learning models which perform optimally. EDA is also used to create visualizations are such as boxplots and correlation heatmaps which help in the subsequent model building steps.

The individual steps that were carried out and their results are elaborated upon below:

a. Importing the libraries

   The first step in EDA is importing all the libraries that will be necessary for the different process carried out.

   1. `numpy` is used mainly for scientific computation and mathematical operations. It provides features such as a numpy array which is a powerful N-dimensional array object and functions for linear algebra.

   2. `pandas` is used for data cleaning, pre-processing and manipulation. This is achieved using functions such as `.isnull()` and `groupby()` which will be seen later on. Series and DataFrames are used which structure data for manipulation and analysis.

   3. `matplotlib` and `seaborn` are both used for data visualization.

b. Loading the dataset

   There are two datasets provided for this project. Pandas is used to load and read the datasets from the two provided .csv files. This is done using the `pd.read_csv()` function. From these files the `dataset` and `df_title` DataFrames are created.

c. Information summary of the dataset

   The `dataset.info()` function is applied to the DataFrame which provides a breakdown of each column. There are a total of `24,058,263` entries. The dataset has the 2 columns shown in the Data Description section. ‘Cust_Id’ has type `float64` and ‘Rating’ has type `object.`

Using `df_title.info()`, the second dataset has 17,770 entries across 3 columns of datatypes `int64`, `float64` and `object`.

`df.dtypes` shows us the data type of each column that was seen with `df.info()`.

d. <u>Handling duplicate values</u>

Duplicate values can lead to a biased analysis, inaccurate results and a machine learning model which overfits due to learning patterns from duplicated data. Using `dataset.duplicated.sum()`. There are no duplicate values present.

e. <u>Creating a new DataFrame including Movie_Ids</u>

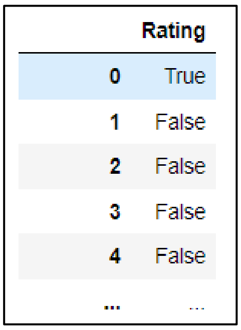
The first dataset provided has been stored in the `dataset` DataFrame. Each line within the DataFrame corresponds to a rating given by a customer for a specific movie.

The individual movies are also provided within the dataset. They are on the lines where the 'Cust_Id' column has values with an integer followed by ':' (for example '1:') and the 'Rating' column has a corresponding NaN value. The rows between each Movie_Id are all the ratings provided by the customer each movie.

These Movie_Ids need to be identified and placed in a separate column. To do this, the following steps were executed:

i. <u>Locating the null values in the 'Rating' column</u>

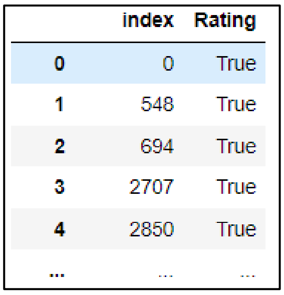
As explained above, the values which are 'NaN' in the 'Rating' column correspond to the different Movie_Ids. These are located using the `pd.isnull()` function and stored in a new DataFrame `df_nan`.



	Rating
0	True
1	False
2	False
3	False
4	False
...	...

ii. <u>Replacing DataFrame with only 'True' values</u>

Using `df_nan[df_nan['Rating'] == True]`, the values in the DataFrame are replaced with only 'Rating' values that are true which represent the Movie_Ids. `reset_index()` is used to store the index of each respective Movie_Id within the DataFrame. This new DataFrame has 4,499 rows.



	index	Rating
0	0	True
1	548	True
2	694	True
3	2707	True
4	2850	True
...	...	...

iii. Replacing DataFrame with only ‘True’ values

As mentioned previously, the rows between each Movie_Id are the ratings for each movie. The rows of `df_nan` are iterated through and the indexes for each Movie_Id are stored in variables `i` and `j`. Each Movie_Id is stored in the list `movie_np` for the amount of values between these two variables.

iv. Handling null values

The original `dataset` DataFrame had null values in the ‘Rating’ column where each Movie_Id begins. These values are removed using the `pd.notnull()` function.

v. Movie_Id column added to dataset DataFrame

The `movie_id` list is added as a column to the `dataset` DataFrame. Each row now has a rating and the respective movie the rating has been provided for.

f. Loading and merging the second dataset

The movie names are loaded from a csv file and stored in the DataFrame `df_title`. Using `pd.merge()`, the two DataFrames are merged into a final DataFrame `dataset1`. This is done by matching the ‘Movie_Id’ columns in both DataFrames.

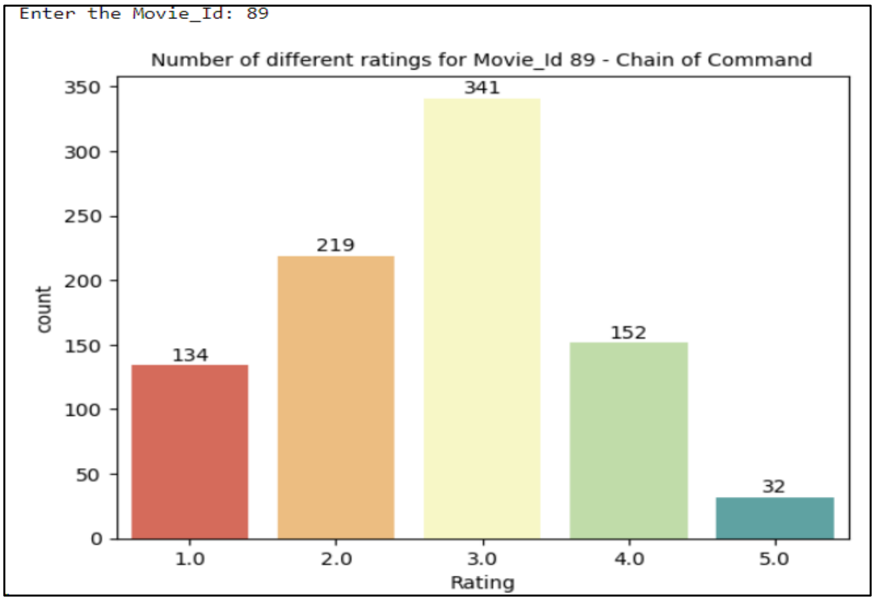
g. Outlier detection

The detection of outliers is a very important step in data pre-processing. It involves spotting values that deviate significantly from the overall pattern and removing them. Outliers can cause distortions in a dataset which lead to a machine learning model making inaccurate predictions. There is no requirement of outlier detection here.

Data Visualisation

Data visualisation is a very important step. It transforms information within the dataset into a visual format where it is easier to spot trends, patterns and draw insights from the data. Alongside the given problem statement, these visualisations provide further data exploration to understand different relationships present. These relationships can be explained and any outliers or anomalies can also be identified.

a. Count of different ratings for given single Movie_Id

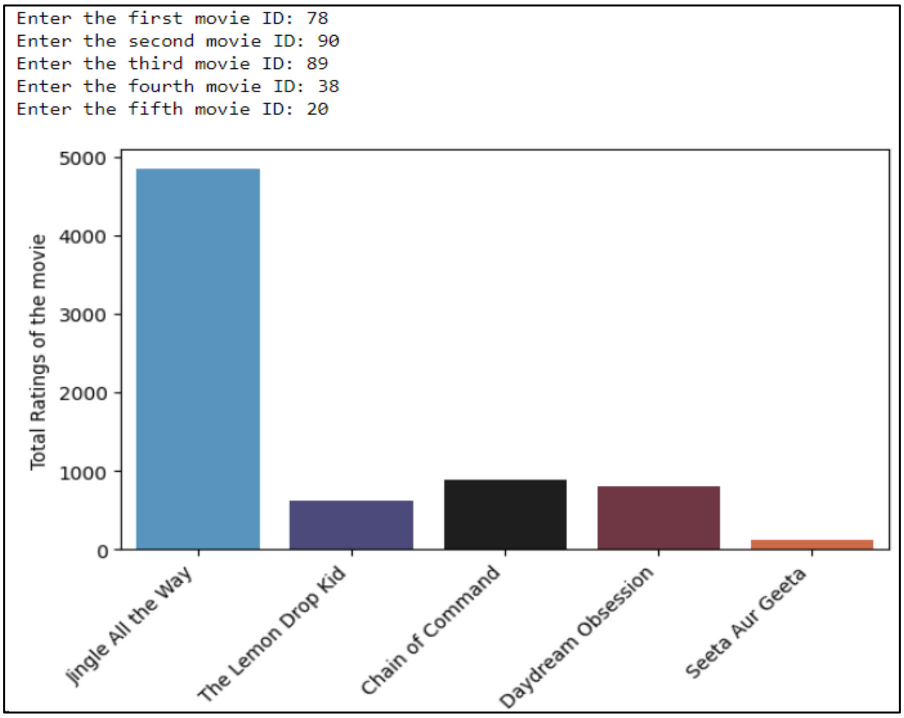


```
Enter the Movie_Id: 89
```

This graph shows the number of ratings provided for each rating level given for a specific 'Movie_Id' provided by the user. There are 4,499 movies and any 'Movie_Id' can be given as input. The example given was for movie 89. 341 people provided a Rating of 3 which was the largest number. The fewest number of people gave a Rating of 5.

Using `int(input('Enter the Movie_Id: '))`, the user is prompted to provide a movie ID value. `df2.groupby()` is used to group the movies with the same 'Movie_Id' together and provide the total number of ratings provided for each. `sns.barplot()` was used to visualise this data.

b. Analysis of ratings of five different Movies

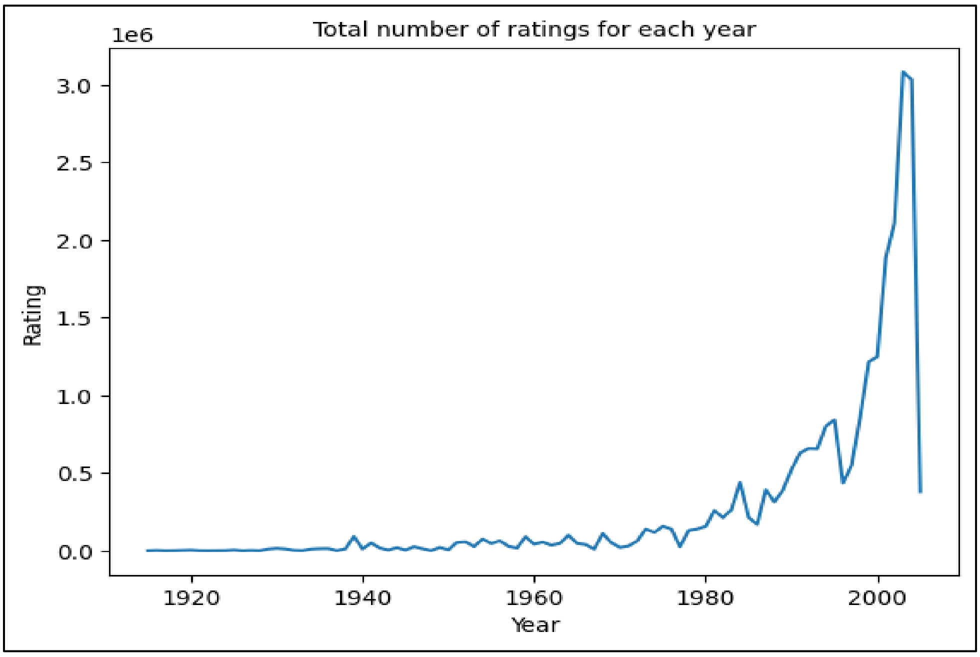


```
Enter the first movie ID: 78
Enter the second movie ID: 90
Enter the third movie ID: 89
Enter the fourth movie ID: 38
Enter the fifth movie ID: 20
```

The user is prompted to provide the ‘Movie_Id’s for 5 different movies. The total number of ratings for these movies are calculated and compared. In the example provided, ‘Movie_Id’s 78, 90, 89, 38 and 20 are provided. ‘Jingle All The Way’ has the highest number of ratings with 4,800 and ‘Seeta Aur Geeta’ has the least with 100. This implies ‘Seeta Aur Geeta’ was a much less popular movie and was seen by only a few people.

`int(input(‘Enter the Movie_Id: ’))` is used 5 times and the 5 user inputs are stored in variables `i` to `n` which are placed in list `inputs`. This list is iterated through using a for loop and the number of times each ‘Movie_Id’ appears in the DataFrame is stored in the variable `count_movie`. This represents the number of ratings given for each movie. This number along with the ‘Movie_Id’ is appended to the `data` and a DataFrame is created from this. This was plotted using `sns.barplot()`.

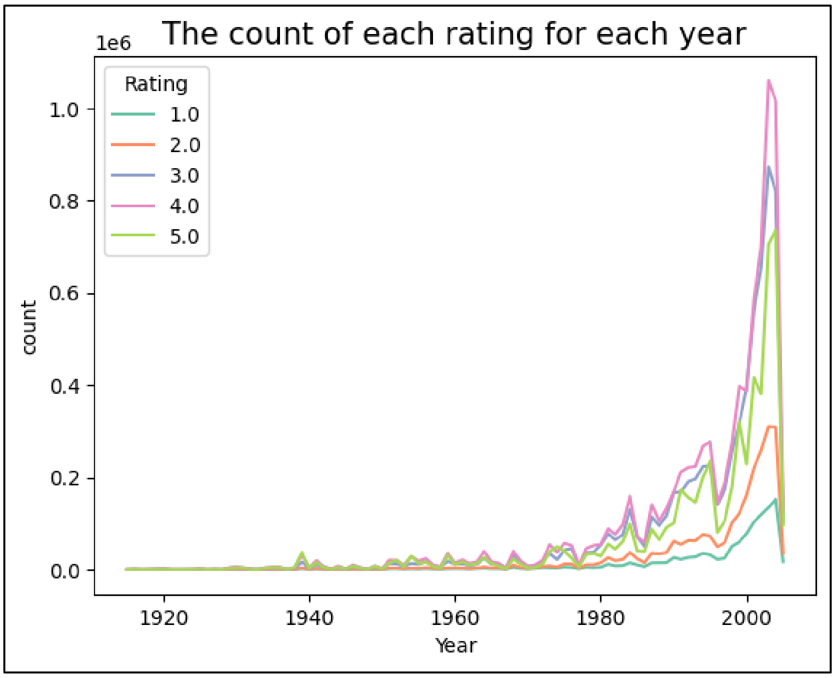
c. Customer Ratings for Each Year



This graph shows the total number of ratings stays low between the years 1920 and 1980. From 1980 to 2000 there is a gradual increase and then a much steeper increase until 2005 where it peaked with 3,000,000 ratings. This coincides with the sudden increase in the number of people having access to computers and having the ability to stream content using Netflix.

`groupby()` is used to group year and rating with the count aggregation to create a DataFrame with the total number of ratings for each year. This is stored and plotted using `sns.lineplot()` for visualisation.

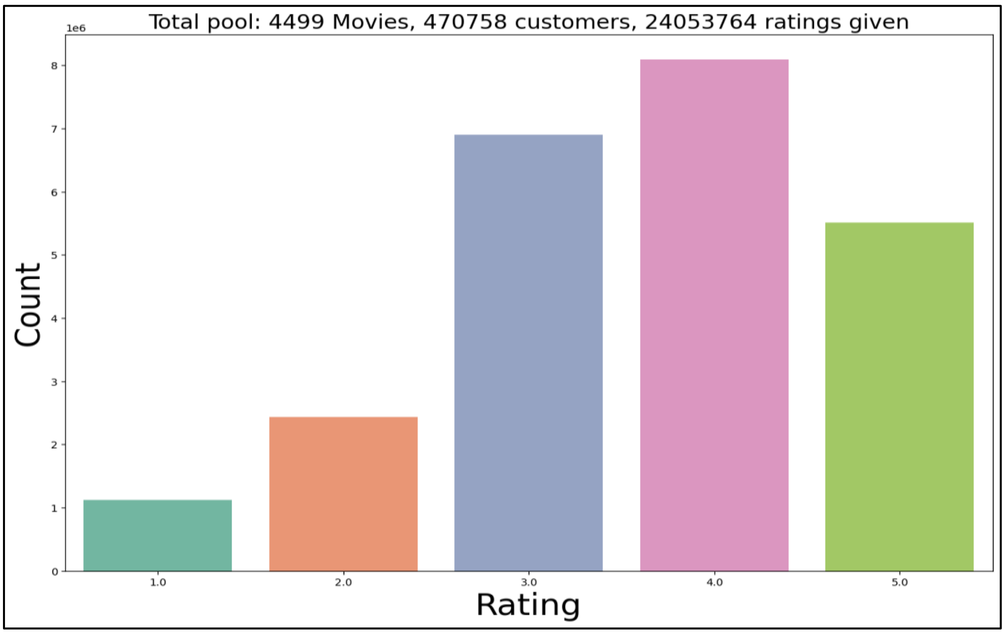
d. Count of Each Rating For Each Year



This graph shows all ratings increased at a similar rate between 1915 to 2005, with a steeper increase between 1995 and 2005. Rating 4 consistently was the highest rating given from years 1970 to 2005.

The `groupby()` function is used to group 'Rating' and 'Year' with 'Cust_Id' with an aggregation of count. This creates a DataFrame with the count of the number of customers that provided a rating for each year. This data is shown through a lineplot.

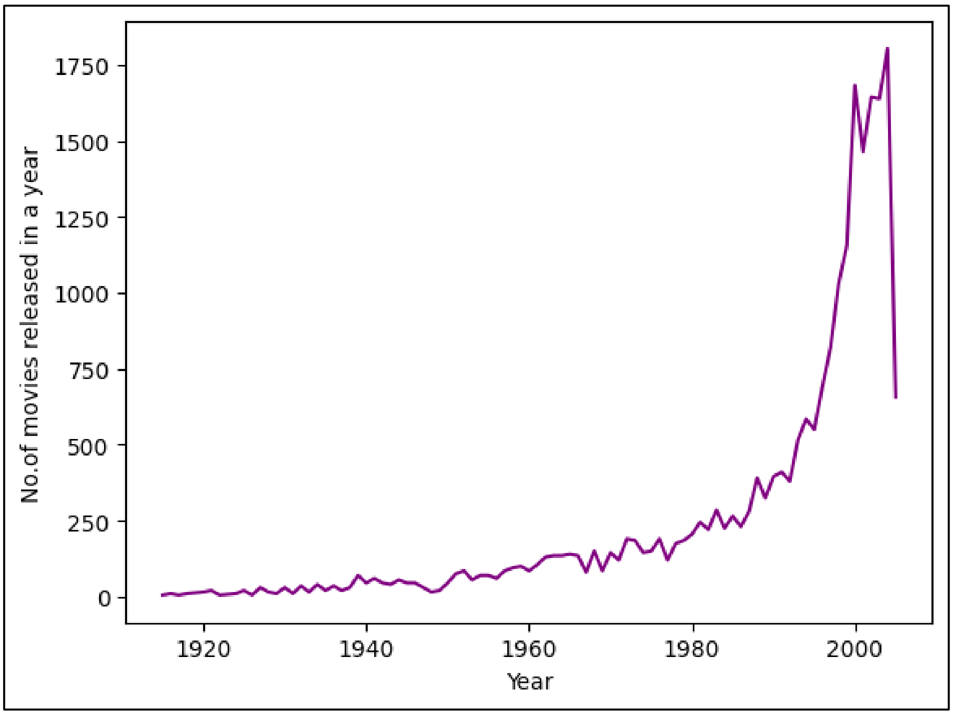
e. The number of customers for each Rating



There are a total of 24,053,764 ratings given by customers. The largest number of ratings were provided for a rating of 4 with 8,000,000 ratings with 1,000,000 ratings given for a rating of 1. There are a total of 4,499 movies that were rated and 470,758 distinct customers provided these ratings.

The number of ratings provided for each rating number was calculated using the `groupby()` function with 'Rating' and 'Cust_Id'. The number of unique customers was calculated by subtracting the 'movie_count' from the number of unique customers which was calculated using `nunique()`. This value was displayed in the title of the plot along with the total number of ratings.

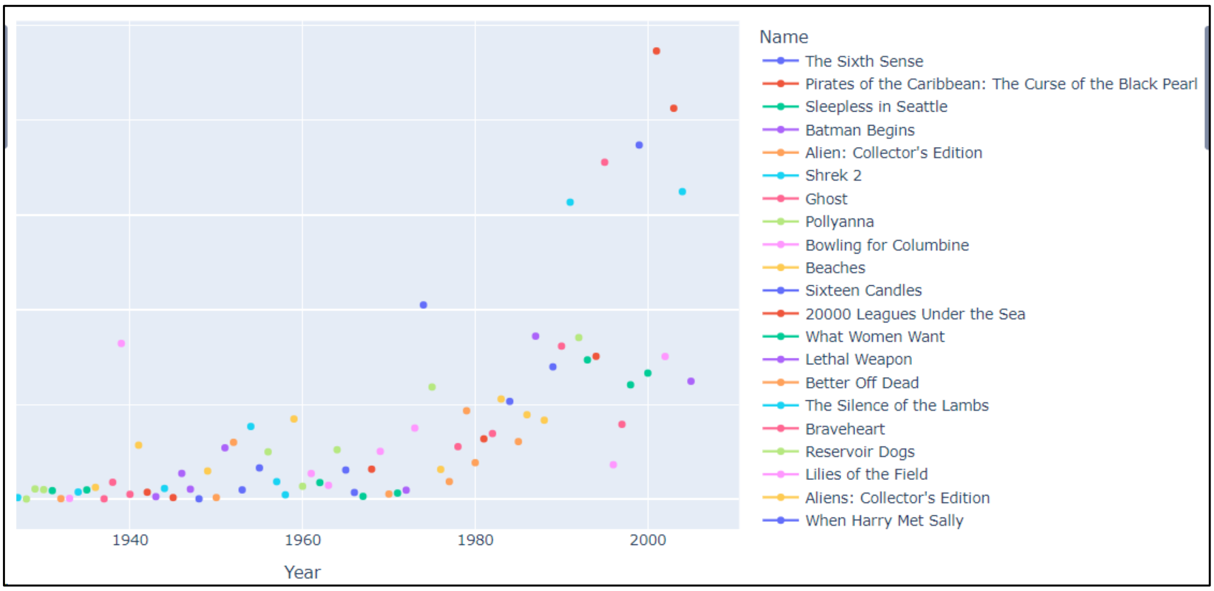
f. Number of Movies streamlined in a year



This graph shows the number of movies streamed gradually increased between 1920 and 1990. The number increased at a greater rate from 1991 to 2005, peaking at 1750 movies streamed in 2005.

A new DataFrame was created by using group by for the ‘Year’ and ‘movie_id’ columns with the count aggregation. This data was then visualised using a lineplot.

g. Best Rated Movies in Each Year



This graph shows the movie for each year which has the highest number of rating '5's along with the number of ratings given for this movie that were 5. There is a general upward trend with the number of 5 ratings gradually increasing between 1980 and 2005. 'Lord of the Rings: The Fellowship of the Ring' has the highest number of 5 ratings with 95,000. It was released in 2001.

A dataframe df7 is initially created which has all rows within the dataset where the rating given is 5. This data is then grouped by 'Name' and 'Year' based on the 'Cust_Id' with a count aggregation which provides the total number of 5 ratings given for each movie for each year. A for loop is then used to iterate through these values for each year and find the movie with the highest number of ratings for each year. These years, names and counts are stored in the list `values` which is then visualised using plotty express lineplot (`px.line()`).

Machine Learning Model Building

1. Data Processing

The data needs to be pre-processed before the subsequent model building steps. In creating a recommendation system, content should be recommended to the user based on content that has a high number of ratings.

a. Number of ratings to include

If recommendations are based on content with only a few ratings, inaccurate recommendations would be provided to users. If users are given recommendations they dislike, their interest in using the Netflix will reduce and they would be prompted to stop their subscription. This reduces the money generated for the business which would have many negative implications for Netflix. Therefore, only the movies with the highest 30% of ratings will be included in the model building.

b. Quantile method

To do this, the data is first grouped by the 'Movie_id' column and the count and mean are found for the ratings. `quantile(0.7)` is applied to this DataFrame based on the 'count' column which only takes the movies with the top 30% of rating count. The number of ratings provided at the 70th percentile is 1,799.

2. Machine Learning Models

The main step is the building the machine learning models, fitting and training these models and then using these models to predict the desired outputs.

Problem Statement: For the purpose of this project a Recommendation Engine will be created where every single user would be recommended a list of movies that are best suited for them based on their area of interest and ratings.

3. Choosing the Algorithm for the Project

Since this is a recommendation task, a model is required which will take the input features, apply latent features and produce an accurate output to address the problem statement.

a. What is Singular Value Decomposition?
SVD is a mathematical technique commonly used in various fields, including machine and recommendation systems (Towardsdatascience, 2020). It is particularly useful in decomposing a matrix into its constituent parts, revealing hidden patterns and structures within dataset. This dataset is given with customer id, rating , movie id based on this recommendation engine is to be created so SVD can find the hidden patterns of previous ratings. Movies can be recommended to customer based on these hidden patterns.

b. Collaborative filtering
SVD involves in Matrix Factorization for collaborative filtering, which breaks down the user-item interaction matrix into latent factors, allowing for a more effective representation of user preferences and item characteristics. SVD helps finding out undercover hidden patterns and reduce dimensionality (Scikit-learn, 2023).

c. Latent Factors
In the context of collaborative filtering, latent factors are hidden features or characteristics that the model learns from the user-item interactions in the training data. These factors are not explicitly provided in the input data but are inferred by the model during the training process. In Singular Value Decomposition (SVD), these latent factors are represented by the matrices U (user matrix) and V (item matrix).

With our specific problem statement, these latent factors would include aspects such as the director of the movie, the movie genre and actors. When the model has these different factors, it will be able to provide much more specialised recommendations to users, resulting in user retention and revenue generation for the business (AIM, 2022).

d. User Latent Factors (U matrix)
Each row in the U matrix represents a user.

The elements in each row of this matrix represent the user's association with different latent factors. For example, how much they prefer a specific movie genre or a specific director. Therefore, these latent factors can be used to understand individual preferences of users and tailor movie recommendations based on this. This would lead to user retention and revenue generation for the business.

e. Item Latent Factors (V matrix)
   Each column in the V matrix represents an item which is a movie in this case. These latent factors capture hidden characteristics or features of items.

f. Singular Values (Σ matrix)
   The diagonal matrix Σ contains singular values, which indicate the strength of each latent factor.

   Higher singular values correspond to more important latent factors. These values enable us to understand which factors are important to different users. For example, if a user has high singular values for directors, movie recommendations can be tailored specifically to this user based on the director.

g. Training process
   During the training process, the model learns the values of these latent factors that best approximate the original user-item ratings matrix. The latent factors are chosen in such a way that the reconstructed matrix (U * Σ * V) approximates the observed ratings as closely as possible. This allows the model to generalize well to unseen user-item pairs.

h. Benefits and drawbacks
   The SVD model is very powerful but has some limitations. The 2 main limitations are the cold start problem which is when insufficient data leads to the model struggling to recommend items for new users and scalability problems with larger datasets. Despite these negatives, SVD is still widely used for recommendation systems and will therefore be used in this project.

4. Motivation and Reasons For Choosing the Algorithm

From the analysis carried out above, **SVD** is the chosen algorithm because it provides the most accurate and robust prediction. This algorithm is simple to implement and is able to easily handle and process large amounts of data.

5. Assumptions

Defining assumptions is an important step before building machine learning models. It enables you to understand the performance and interpretation of the model and define parameters to ensure the predictions are accurate and robust.

a. Linear Independence
The columns of the matrix being decomposed are assumed to be linearly independent. Linear independence is essential for the uniqueness of the singular value decomposition.

b. Data Completeness
The input data is complete, having no missing values in the matrix. In recommendation systems, users must have provided ratings for all items and there no gaps in the user- item interaction matrix.

c. Linearity
Linear relationship between variables is required. This assumption implies that the underlying patterns and structures in the data can be effectively captured using linear combination of the singular vectors and values.

d. Orthogonality of Singular Vectors
The singular vectors obtained from SVD are assumed to be orthogonal. This orthogonality property simplifies the interpretation of decomposition and allows for efficient reconstruction of the original matrix.

i. Model Evaluation and Techniques

a. Cross Validation
Cross-validation is a technique used in machine learning to assess the performance of a predictive model. It helps in estimating how well the model will generalize to an independent dataset, which is crucial for evaluating its effectiveness and avoiding overfitting.

This technique divides the dataset into multiple subsets, called folds. The model is trained on some of these folds and tested on the remaining folds. This process is repeated multiple times, with different subsets used for training and testing in each iteration. The most common form of cross-validation is k-fold cross-validation.

b. Accuracy of the model

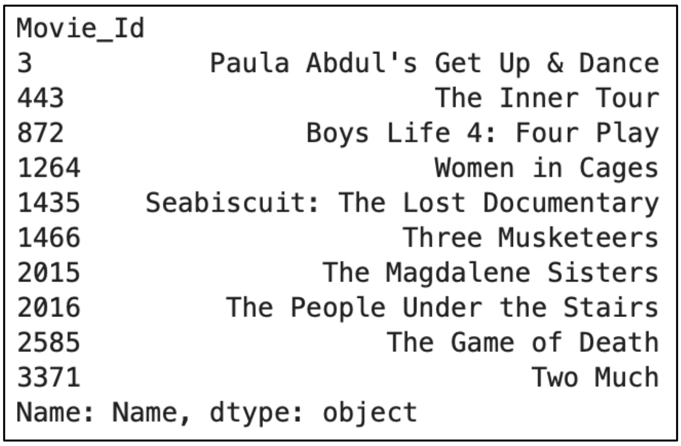
The summary of the accuracy of the 3 folds can be seen in the following table

SVD Algorithm	Accuracy
Fold 1	99%
Fold 2	100%
Fold 3	99%

c. Testing the model

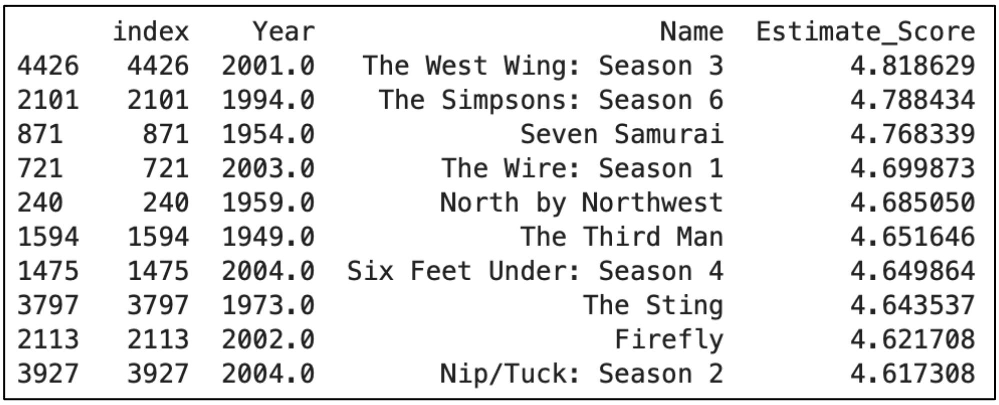
Following the creation and training of the SVD model, the next step is to test it against a specific user and provide them with custom movie recommendations.

The DataFrame `dataset_44937` is created which contains the movies rated as 5 by user 44937. The top 10 of these can be seen below:



```
Movie_Id
3           Paula Abdul's Get Up & Dance
443                       The Inner Tour
872                 Boys Life 4: Four Play
1264                      Women in Cages
1435    Seabiscuit: The Lost Documentary
1466                    Three Musketeers
2015                The Magdalene Sisters
2016           The People Under the Stairs
2585                    The Game of Death
3371                            Two Much
Name: Name, dtype: object
```

Using `svd.predict(44937, x)`, the movie predictions for user 44937 are generated using the SVD model. The dataset used only includes the top 30% of rated movies. The reasoning for this was explained previously. The results from the model are sorted by 'Estimate_Score' in descending order. The movies with the highest 10 estimate scores can be seen below:



```
      index    Year                      Name  Estimate_Score
4426   4426  2001.0   The West Wing: Season 3        4.818629
2101   2101  1994.0    The Simpsons: Season 6        4.788434
871     871  1954.0             Seven Samurai        4.768339
721     721  2003.0        The Wire: Season 1        4.699873
240     240  1959.0        North by Northwest        4.685050
1594   1594  1949.0             The Third Man        4.651646
1475   1475  2004.0  Six Feet Under: Season 4        4.649864
3797   3797  1973.0                 The Sting        4.643537
2113   2113  2002.0                   Firefly        4.621708
3927   3927  2004.0        Nip/Tuck: Season 2        4.617308
```

The high estimation scores for the top 10 movies range from 4.8 to 4.6. This indicates the SVD model has performed well and has provided accurate recommendations for this user.

j. K-Means

K-means is an unsupervised machine learning algorithm that is used to partition a dataset into distinct clusters where each relates to a different attribute or characteristic (Youguo, 2012). It is a versatile algorithm that can be applied to different problems and can efficiently handle large datasets.

Following the SVD implementation, K-Means has been implemented with the number of clusters determined using a Silhouette Score. However, when comparing the predictions produced with the predictions from the SVD model, there are large discrepancies. This may be because K-means is only trained on the features provided within the dataset. Using the movie name, year and rating to train the model are not sufficient in providing accurate predictions. K-means can therefore be disregarded.

k. Inferences

- The SVD model performs well with 99%, 100% and 99% accuracy scores in Folds 1, 2 and 3 respectively.
- When tested with user 44937, predictions were provided with the top 10 estimate scores ranging from 4.8 to 4.6.
- The recommendations provided from the K-means algorithm differ greatly to the recommendations generated by the SVD model. This suggests only SVD can be applied in addressing the problem statement.

Conclusion

a. Summary

The aim of this project was to address the problem statement and create a recommendation system for Netflix where users are recommended movies to watch based on the movies they have watched and the ratings they have given these movies.

The SVD model was trained, tested and evaluated. From the steps carried out, the 3 folds had accuracy ratings of 99%, 100% and 99% respectively. When tested against a specific user (44937), the model ran successfully and produced results with high estimate scores ranging from 4.8 to 4.6.

b. Future possibilities of the project

If this project were to be carried out again, it would be beneficial to use a dataset with more features such as movie genre, actors and directors. This would enable another algorithm such as K-Means to be implemented because there are enough features to train the model and provide personalised set of recommendations to users. Comparing the outcome of this model to SVD would enable us to optimise and choose a model which performs the best and provides the most accurate predictions for Netflix.

Furthermore, an overall better set of results could be achieved by using a combination of machine learning and deep learning models.

References

AIM, 2022. *Singular Value Decomposition.* [Online] Available at: https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/ [Accessed February 2024]

Bstrategyhub, 2023. *Netflix Business Model How Does Netflix Make Money.* [Online] Available at: https://bstrategyhub.com/netflix-business-model-how-does-netflix-make-money/#Introduction_to_Netflix_Inc [Accessed February 2024]

IBM, 2022. *What is Exploratory Data Analysis.* [Online] Available at: https://www.ibm.com/topics/exploratory-data-analysis [Accessed January 2024]

Lifewire, 2024. *Overview of the Netflix Streaming Service.* [Online] Available at: https://www.lifewire.com/overview-of-the-netflix-streaming-service-1847831 [Accessed January 2024]

Pyimagesearch, 2023. *Netflix Movies and Series Recommendation Systems.* [Online] Available at: https://pyimagesearch.com/2023/07/03/netflix-movies-and-series-recommendation-systems/ [Accessed February 2024]

Study, 2023. *Netflix History Founding Facts.* [Online] Available at: https://study.com/academy/lesson/netflix-history-founding-facts-created.html#:~:text=Netflix%20was%20founded%20on%20August,its%20streaming%20service%20became%20available [Accessed February 2024]

Scikit-Learn, 2023. *Cross Validation.* [Online] Available at: https://scikit-learn.org/stable/modules/cross_validation.html [Accessed February 2024]

Towardsdatascience, 2020. *Understanding Singular Value Decomposition.* [Online] Available at: https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d [Accessed February 2024]

Youguo Li, Haiyan Wu (2012). A Clustering Method Based on K-Means Algorithm, Physics Procedia, Volume 25, 2012, Pages 1104-1109.