Business Case Study : Movie Recommender Systems

Problem Description

A Movie Recommender System aims to improve the user experience by suggesting personalized movie recommendations based on user ratings and preferences. The system leverages the data provided by multiple users to identify patterns and recommend movies that a particular user might enjoy. The goal is to build a system that accurately predicts a user's rating for unseen movies, enhancing their interaction with the platform by offering relevant suggestions.

Objectives:

- **Personalized Movie Recommendations:** Suggest movies based on a user's historical ratings and similar users' preferences.
- **Improve User Experience:** Provide better recommendations using techniques like Pearson Correlation, Cosine Similarity, and Matrix Factorization.
- **Efficient and Scalable System:** Develop a system that can handle large datasets and compute recommendations in a scalable way.

Dataset:

The dataset comprises three files:

1. **ratings.dat:** Contains user ratings for movies, (Each user has at least 20 ratings) including the following information in the format: UserID::MovieID::Rating::Timestamp
    - **UserID:** Unique identifier for users. (range between 1 and 6040)
    - **MovieID:** Unique identifier for movies. (range between 1 and 3952)
    - **Rating:** User rating on a 5-star scale. (whole-star ratings only)
    - **Timestamp:** Time of rating. (represented in seconds)
2. **users.dat:** Contains demographic information for users and is in the following format: UserID::Gender::Age::Occupation::Zip-code
    - **UserID:** Unique identifier for users.
    - **Gender:** is denoted by a "M" for male and "F" for female

- **Age:** The age group and the corresponding code used to represent each age group are summarized in Table 3.1.

Table 3.1 : Age group details of the Users

Age Group Code	Age Range
1	Under 18
18	18-24
25	25-34
35	35-44
45	45-49
50	50-55
56	56+

- **Occupation:** Profession of the user. The occupation details and the corresponding occupation code used to represent each occupation are summarized in Table 3.2.
- **Zip-code:** Zip code of the user.

3. **movies.dat:** Contains metadata about movies and is in the following format: MovieID::Title::Genres
    - **MovieID:** Unique identifier.
    - **Title:** Movie title, including the release year.
    - **Genres:** List of genres the movie belongs to Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western

Business Questions to be answered from Analysis

1. Users of which age group have watched and rated the most number of movies?
2. Users belonging to which profession have watched and rated the most movies?
3. Most of the users in our dataset who’ve rated the movies are Male. (T/F)

Table 3.2 : Occupation Details of the Users

Occupation Code	Occupation Description
0	other or not specified
1	academic/educator

2	artist
3	clerical/admin
4	college/grad student
5	customer service
6	doctor/health care
7	executive/managerial
8	Farmer
9	homemaker
10	K-12 student
11	lawyer
12	programmer
13	retired
14	sales/marketing
15	scientist
16	self-employed
17	technician/engineer
18	tradesman/craftsman
19	unemployed
20	writer

4. Most of the movies present in our dataset were released in which decade?
    1. 70s b. 90s c. 50s d.80s
5. The movie with maximum no. of ratings is ___.
6. Name the top 3 movies similar to ‘Liar Liar’ on the item-based approach.
7. On the basis of approach, Collaborative Filtering methods can be classified into ___-based and ___-based.
8. Pearson Correlation ranges between ___ to ___ whereas, Cosine Similarity belongs to the interval between ___ to ___.
9. Mention the RMSE and MAPE that you got while evaluating the Matrix Factorization model.

Methodology

The methodology for building a movie recommendation system involves the following series of steps

1. **Data Collection and Preparation**
    - **Dataset:** Start with the provided dataset, which includes three key files: ratings.dat, users.dat, and movies.dat.

    - **Data Formatting:** The first task is to load the data and bring it into a workable format. You can use Python libraries like pandas to read and manipulate the data.
        - ratings.dat contains user ratings of movies.
        - users.dat provides demographic information like age, gender, and occupation.
        - movies.dat contains movie metadata such as title and genres.

2. **Exploratory Data Analysis (EDA) and Feature Engineering**

    - **Exploratory Data Analysis (EDA):** Analyze the structure of the dataset, inspect missing values, outliers, and check for inconsistencies.
    - Perform summary statistics to understand the distribution of ratings and the number of ratings per movie.
    - Performing necessary type conversion and deriving new features
    - Investigating the data for any inconsistency
    - **Data Merging:** Merge the three datasets on the common columns like MovieID and UserID to create a consolidated dataset for further analysis.
    - **Feature Engineering:** Extract new features like the release year from the movie titles.
    - Analyze the distribution of ratings by age, gender, or occupation to understand the preferences of different demographic groups.
    - Group the data according to the average rating and no. of ratings

3. **Building the Recommender System**

- Movie recommendation can be done using various approaches. The two most common methods are Collaborative Filtering and Matrix Factorization.

  a. **Collaborative Filtering** relies on the idea that users with similar preferences in the past will like similar items in the future.

     i. **Item-Based Collaborative Filtering**

        - **Pivot Table Creation:** Create a user-item matrix where rows represent users and columns represent movies. The values are the ratings given by users to movies.
        - **Pearson Correlation:** Compute the correlation between movies based on user ratings using Pearson correlation. The goal is to recommend movies that are highly correlated to the ones the user has already rated highly.
        - **Cosine Similarity:** Calculate the Cosine Similarity between movies based on their ratings. This is useful in the nearest-neighbor approach to find similar movies. For example, recommend movies that are closest in similarity to the one the user has rated highly.

  b. Matrix Factorization (Model-Based Approach)

  Matrix Factorization techniques decompose the user-item matrix into two lower-dimensional matrices, which reveal the latent features of users and movies. Singular Value Decomposition (SVD) or Alternating Least Squares (ALS): These algorithms learn latent factors from the rating matrix and predict missing ratings. Each user and movie is represented by a set of latent features. A dot product of these features gives a predicted rating for unseen movies. The performance of the matrix factorization model is evaluated using metrics like Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

4. Recommender System Implementation

a. Recommendations Using Pearson Correlation

The system should accept a movie title as input from the user. Based on the Pearson correlation between movies, recommend 5 movies that are most similar to the input movie.

b. **Recommendations Using Cosine Similarity (K-Nearest Neighbors)**
Implement a K-Nearest Neighbors (KNN) algorithm to find the nearest similar movies based on Cosine Similarity. Given a movie title, recommend 5 movies that have the highest similarity scores.

c. **Recommendations Using Matrix Factorization**
Train a matrix factorization model using libraries such as cmfrec or Surprise. For each user, predict their rating for unseen movies and recommend the top-rated ones. Evaluate the model using RMSE to ensure accurate recommendations.

5. **Embedding Visualization**

a. **Embeddings:** The matrix factorization process produces user and movie embeddings (low-dimensional representations).
b. **Visualization:** Visualize these embeddings to explore relationships between users and movies.
c. **Analysis:** Analyze the structure and relationships in the embedding space, comparing movies that are closer to each other in terms of their latent features.

Analysis

1. **Data Collection and Preparation**
   - **Data Formatting:** The first task is to load the data and bring it into a workable format. The data samples of ratings.dat, users.dat, and movies.dat are shown in Figures 3.1, 3.2, and 3.3 respectively.

	UserID::MovieID::Rating::Timestamp
0	1::1193::5::978300760
1	1::661::3::978302109
2	1::914::3::978301968
3	1::3408::4::978300275
4	1::2355::5::978824291

Figure 3.1: Rating.dat data sample in the format: UserID::MovieID::Rating::Timestamp

	UserID::Gender::Age::Occupation::Zip-code
0	1::F::1::10::48067
1	2::M::56::16::70072
2	3::M::25::15::55117
3	4::M::45::7::02460
4	5::M::25::20::55455

Figure 3.2: Users.dat data sample in the format: UserID::Gender::Age::Occupation::Zip-code

- **Splitting the data into workable form:** All the third data are split into workable forms and shown in Figures 3.4, 3.5, and 3.6 respectively.

	Movie ID::Title::Genres
0	1::Toy Story (1995)::Animation|Children's|Comedy
1	2::Jumanji (1995)::Adventure|Children's|Fantasy
2	3::Grumpier Old Men (1995)::Comedy|Romance
3	4::Waiting to Exhale (1995)::Comedy|Drama
4	5::Father of the Bride Part II (1995)::Comedy

Figure 3.3: Movies.dat data sample in the format: MovieID::Title::Genres

	UserID	MovieID	Rating	Timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

Figure 3.4: Rating.dat data is split into columns UserID, MovieID, Rating and Timestamp

	UserID	Gender	Age	Occupation	Zip-code
0	1	F	1	10	48067
1	2	M	56	16	70072
2	3	M	25	15	55117
3	4	M	45	7	02460
4	5	M	25	20	55455

Figure 3.5: Users.dat data is split into columns UserID, Gender, Age, Occupation, and Zip-code

	Movie ID	Title	Genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy

Figure 3.6: Movies.dat data is split into columns MovieID, Title and Genres

2. Exploratory Data Analysis (EDA) and Feature Engineering

- Reviewing the shape and structure of the dataset. The ratings.dat consists of 1000209 movies with 6040 UserID unique values, 3706 unique values, 5 unique ratings (1,2,3,4,5), and 458455 unique values. The count, unique value count, top value, and frequency of the top value of rating.dat are displayed in Figure 3.7 (a).
- The rating value 4 has the highest frequency with 348971 Unique ratings and Figure 3.7 (b) lists the five movie ratings and the number of movies with each unique rating. User.dat consists of 6040 user information of which 4331 (71.7 %) are male and 1709 (28.3 %) of female users. The number of users in each age group and each occupation are cataloged in Table 3.3 and Table 3.4 respectively. The count, unique value count, top value, and frequency of the top value of Users.dat are displayed in Figure 3.8. The count, unique value count, top value, and frequency of the top value of movies.dat are displayed in Figure 3.9.

	UserID	MovieID	Rating	Timestamp
count	1000209	1000209	1000209	1000209
unique	6040	3706	5	458455
top	4169	2858	4	975528402
freq	2314	3428	348971	30

(a)

Rating	count
4	348971
3	261197
5	226310
2	107557
1	56174

(b) .

Figure 3.7: (a) The count, unique value count, top value, and frequency of top value of rating.dat (b) Unique rating and number of movies with those ratings

	UserID	Gender	Age	Occupation	Zip-code
count	6040	6040	6040	6040	6040
unique	6040	2	7	21	3439
top	1	M	25	4	48104
freq	1	4331	2096	759	19

Figure 3.8: The count, unique value count, top value, and frequency of the top value of Users.dat

Table 3.3 : Number of users in each age group

Age Group Code	Age Range	Number of Users
1	Under 18	222
18	18-24	1103
25	25-34	2096
35	35-44	1193
45	45-49	550
50	50-55	496
56	56+	380

	Movie ID	Title	Genres
count	3883	3883	3883
unique	3883	3883	301
top	1	Toy Story (1995)	Drama
freq	1	1	843

Figure 3.9: The count, unique value count, top value, and frequency of the top value of movies.dat

- **Transforming Movie Genres into Binary Features:** This is done by creating a structured matrix-like table with Movie ID as rows, Genres as columns, and movie Titles as values and removed null values as illustrated in Figure 3.10. The 18 unique Genres are Animation, Children's, Comedy, Adventure, Fantasy, Romance, Drama, Action, Crime, Thriller, Horror, Sci-Fi, Documentary, War, Musical, Mystery, Film-Noir, and Western.

Genres	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	Western
Movie ID																		
1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Figure 3.10: Transforming Movie Genres into Binary Features

- **Merging the data files and creating a single consolidated data frame :** The data frame ratings and users are merged based on the common "UserID" column, storing the combined information where "UserID" matches in the df DataFrame as displayed in Figure 3.11. This data frame was merged with the “movies” data frame to the new data frame as shown in Figure 3.12

Table 3.4 : Number of users in each Occupation

Occupation Code	Occupation Description	Number of Users
0	other or not specified	711
1	academic/educator	528
2	artist	267
3	clerical/admin	173
4	college/grad student	759
5	customer service	112
6	doctor/health care	236
7	executive/managerial	679
8	Farmer	17
9	homemaker	92
10	K-12 student	195
11	lawyer	129
12	programmer	388
13	retired	142

14	sales/marketing	302
15	scientist	144
16	self-employed	241
17	technician/engineer	502
18	tradesman/craftsman	70
19	unemployed	72
20	writer	281

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code
0	1	1193	5	978300760	F	1	10	48067
1	1	661	3	978302109	F	1	10	48067

Figure 3.11: Merging rating and users dataset

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Movie ID	Title	Genres
0	1	1193	5	978300760	F	1	10	48067	1193	One Flew Over the Cuckoo's Nest (1975)	Drama
1	2	1193	5	978298413	M	56	16	70072	1193	One Flew Over the Cuckoo's Nest (1975)	Drama

Figure 3.12: Data frame formed after merging ratings, users, and movies data frame

- **Performing feature engineering steps type conversions and deriving new features like 'Release Year':** The hour, month, and year extracted from the Timestamp column in the DataFrame, where each timestamp is converted to an integer and then transformed into a datetime object to retrieve the specific components (hour, year, month).

- The Gender column is converted into integers, where 'F' is mapped to 1 and 'M' to 0. Additionally, the Age and Occupation columns are also converted into integers. Finally, any rows with missing values are dropped from the DataFrame. The Release Year was derived from the title column.

- The new dataframe after these feature engineering steps is represented in Figure 3.13. The histogram for the columns Gender, Occupation, Age, Rating, Hour, Year, Month, and Release Year is plotted in Figure 3.14.

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Title	Genres	Hour	Year	Month	Release Year
0	1	1193	5	978300760	1	1	10	48067	One Flew Over the Cuckoo's Nest (1975)	Drama	22	2000	12	1975
1	2	1193	5	978298413	0	56	16	70072	One Flew Over the Cuckoo's Nest (1975)	Drama	21	2000	12	1975

Figure 3.13: Data frame after these feature engineering steps

- **Group the data according to the average rating and no. of ratings:** The User ID, title of movies, age group, occupation code, and movie genres of top 5, bottom 5, top 5 mean, and bottom 5 mean count are summarized in Table 3.5, Table 3.6, Table 3.7, Table 3.8, and Table 3.9 respectively

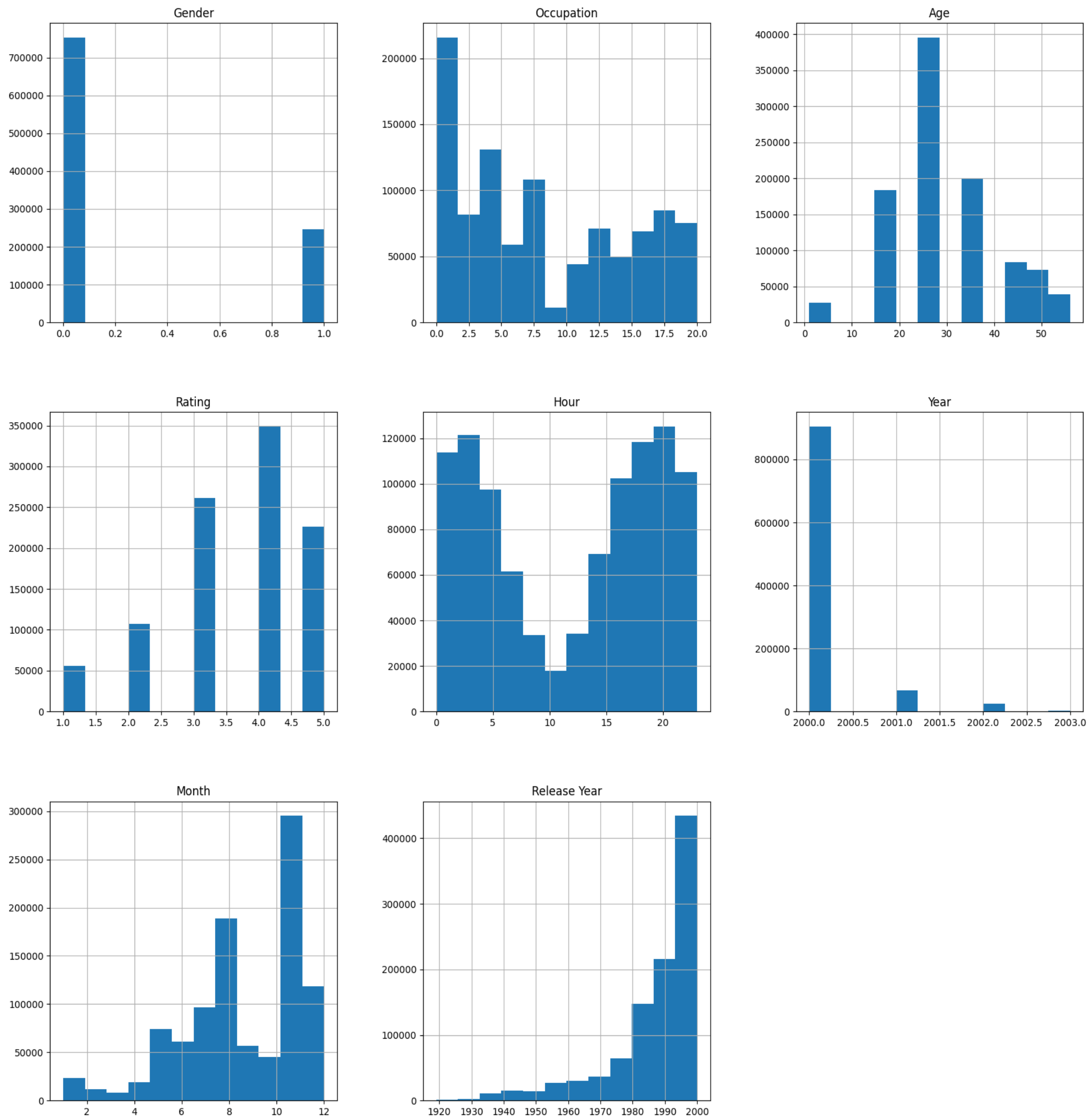

Figure 3.14: Histogram for the columns Gender, Occupation, Age, Rating, Hour, Year, Month, and Release Year

Table 3.5 : User ID of top 5, bottom 5, top 5 mean, and bottom 5 mean count

	Top 5 count	**Bottom 5 count**	**Top 5 mean**	**Bottom 5 mean**
1	4169	5725	283	5850
2	1680	3407	2339	4539
3	4277	1664	3324	2744
4	1941	4419	3902	4486
5	1181	3021	446	3598

Table 3.6 : Title of movies of top 5, bottom 5, top 5 mean, and bottom 5 mean

	Top 5 Count	**Bottom 5 Count**	**Top 5 Mean**	**Bottom 5 Mean**
1	American Beauty	Target	Ulysses (Ulisse)	Fantastic Night, The (La Nuit Fantastique)
2	Star Wars: Episode IV - A New Hope	I Don't Want to Talk About It (De eso no se habla)	Lured	Cheetah
3	Star Wars: Episode V - The Empire Strikes Back	An Unforgettable Summer	Follow the Bitch	Torso (Corpi Presentano Tracce di Violenza Carnale)
4	Star Wars: Episode VI - Return of the Jedi	Never Met Picasso	Bittersweet Motel	Mutters Courage
5	Jurassic Park	Full Speed	Song of Freedom	Windows

3. Build a Recommender System based on Pearson Correlation

- A pivot table of movie titles & user ID was created, imputing the NaN values with the UserID as index, Titles as columns, and Values as Ratings given by user to that particular movie title as as illustrated in Figure 3.15.
- The correlation of dataframe from the imputed dataframe was calculated.

Table 3.7 : Age group of top 5, bottom5, top 5 mean, and bottom 5 mean count

	Top 5 Count	**Bottom 5 Count**	**Top 5 Mean**	**Bottom 5 Mean**
1	25	18	56	45
2	35	45	50	35
3	18	50	45	1
4	45	56	35	25
5	50	1	1	18

Table 3.8 : Occupation code of top 5, bottom, top 5 mean, and bottom 5 mean count

	Top 5 Count	**Bottom 5 Count**	**Top 5 Mean**	**Bottom 5 Mean**
1	4	19	13	10
2	0	13	15	18
3	7	18	6	20
4	1	9	9	8
5	17	8	3	19

Table 3.9 : Movie genres of top 5, bottom, top 5 mean, and bottom 5 mean count

	Top 5 Count	**Bottom 5 Count**	**Top 5 Mean**	**Bottom 5 Mean**
1	Comedy	Drama	Romance	Western
2	Drama	Children’s	Fantasy	Sci-Fi
3	Comedy	Romance	Comedy	Film-Noir
4	Comedy	Drama	Film-Noir	Horror
5	Drama	Romance	Fantasy	Adventure

Title	$1,000,000 Duck (1971)	'Night Mother (1986)	'Til There Was You (1997)	'burbs, The (1989)	...And Justice for All (1979)	1-900 (1994)	10 Things I Hate About You (1999)	101 Dalmatians (1961)	101 Dalmatians (1996)	12 Angry Men (1957)	...	Young Poisoner's Handbook, The (1995)	Young Sherlock Holmes (1985)	Young and Innocent (1937)	Your Friends and Neighbors (1998)	Zachariah (1971)	Zed & Two Noughts, A (1985)	Zero Effect (1998)	Zero Kelvin (Kjærlighetens kjøtere) (1995)	Zeus and Roxanne (1997)	eXistenZ (1999)
UserID																					
1	3.0	4.0	3.2	3.2	4.6	2.5	4.0	4.0	3.8	4.6	...	3.4	4.0	3.8	2.6	3.5	3.4	3.0	3.5	3.0	3.2
2	3.6	3.0	2.8	2.8	4.2	2.5	3.6	3.8	3.6	4.6	...	3.8	3.6	3.8	2.8	3.5	3.6	3.0	3.5	2.0	3.4
3	3.0	3.4	2.4	2.8	4.2	2.5	4.4	4.2	4.2	4.4	...	4.0	4.2	4.2	3.4	3.5	4.0	4.4	3.5	2.0	3.2
4	3.4	3.4	3.2	3.8	4.4	2.5	3.6	3.4	4.0	4.8	...	3.8	2.8	4.2	3.6	3.5	3.8	3.8	3.5	2.2	3.8
5	2.4	3.0	2.4	3.4	4.0	2.5	3.6	4.0	2.6	4.2	...	3.8	3.0	3.6	3.8	3.5	3.6	4.4	3.5	2.6	3.4

Figure 3.15: Pivot table after imputation

- Item-based approach to create a simple recommender system that uses Pearson Correlation was built and tested for getting the top 3 recommendations for the movie title Liar Liar (1997) and got the recommended movies
   1. Jury Duty (1995)
   2. Big Bully (1996)
   3. Dumb & Dumber (1994)

4. **Build a Recommender System based on Cosine Similarity**.
   - Print the user similarity matrix and item similarity matrix

      1. **Cosine Similarity with Users:** A new data frame was formed by grouping UserID, Gender, Age, and Occupation, calculating the mean of the Rating, Hour, Month, and Year for each group then resetting the index to create a new DataFrame; afterward, it drops the UserID column from the DataFrame and displays the first few rows of the modified DataFrame in Figure 3.16.

	Gender	Age	Occupation	Rating	Hour	Month	Year
0	1	1	10	4.188679	22.245283	9.301887	2000.245283
1	0	56	16	3.713178	21.155039	12.000000	2000.000000
2	0	25	15	3.901961	21.000000	12.000000	2000.000000
3	0	45	7	4.190476	20.000000	12.000000	2000.000000
4	0	25	20	3.146465	6.015152	12.000000	2000.000000

Figure 3.16: Cosine Similarity with Users

      2. **Cosine Similarity with Items:** A copy of the movie data frame was created, and then the Genres column split into separate rows. It then removes duplicates and pivots the DataFrame to create a binary matrix

indicating the presence of each genre for each movie. After resetting the index and merging with the average ratings by grouping movies ID, it calculates the cosine similarity between the Movies based on this binary matrix and their average ratings.

5. Build a Recommender System based on Matrix Factorization

- Create a Recommender System using the Matrix Factorization method: A new data frame was created with userID, movieID, and rating
- Imported **cmfrec/Surprise** library to run matrix factorization.
- The Collaborative Matrix Factorization (CMF) model using the Alternating Least Squares (ALS) method with specified parameters, such as 2 latent factors (k), a regularization term (lambda_) of 0.1, and no user or item bias was used to test the new dataframe.
- Evaluated the model's performance are
  - RMSE: 0.76533128944678
  - MAE: 0.6878217940191732
- Use embeddings for visualization and similarity-based models: This calculates the overlap between the top recommended movies for each user and the movies they have already rated in the training data, storing the number of overlapping movies and the total recommended movies for each user in two lists, overlap and num_rec. It then computes the average percentage of overlap by dividing the mean of the overlap list by the mean of the number of recommendations. Finally, it visualizes the distribution of overlap counts by creating a histogram with 20 bins and is visualized in Figure 3.17

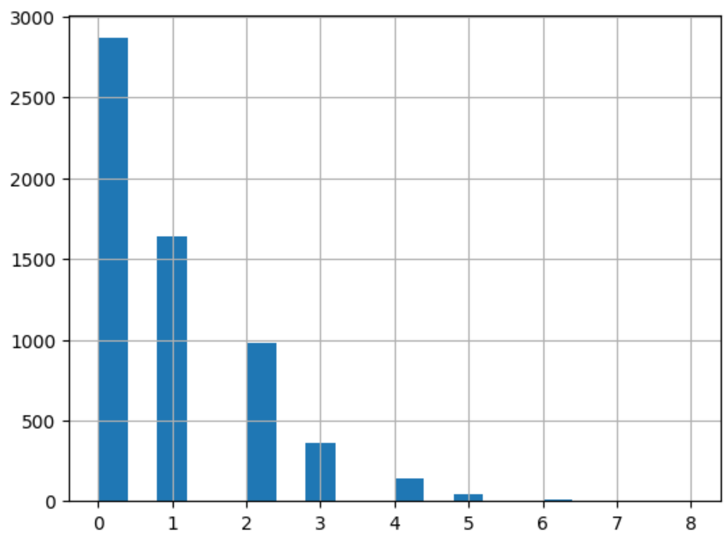

Figure 3.17: Use embeddings for visualization and similarity-based models

6. Build a Recommender System based Pearson Correlation

- A user-movie matrix was created using a pivot table from the DataFrame df, which organizes user ratings by UserID and MovieID.
- It then calculates the Pearson correlation matrix to determine the similarity between users and defines a function to find the top five most similar users for a given user ID.
- Finally, it retrieves the top five movie recommendations for the specified user based on high ratings from similar users, ensuring that recommended movies are not already rated by the user, and prints the recommendations as [1, 17, 34, 36, 161]

Business Questions to be answered from Analysis

1. Users of which age group have watched and rated the most number of movies?
    - 25: "25-34"
    - 25 395556

2. Users belonging to which profession have watched and rated the most movies?
    - The most common profession is "college/grad student" with a count of 759.

3. Most of the users in our dataset who've rated the movies are Male. (T/F)

- T(True)

4. Most of the movies present in our dataset were released in which decade?

   a. 70s

   b. 90s

   c. 50s

   d. 80s

   Answer : 90's

   The number of movies released in 70's is 82552

   The number of movies released in 90's is 532843

   The number of movies released in 50's is 35232

   The number of movies released in 80's is 224056

5. The movie with the maximum no. of ratings is American Beauty (1999)
   - American Beauty (1999)   14800

6. Name the top 3 movies similar to ‘Liar Liar’ on the item-based approach.
   1. Jury Duty (1995)
   2. Big Bully (1996)
   3. Dumb & Dumber (1994)

2. On the basis of approach, Collaborative Filtering methods can be classified into user-based and item-based.
   - Yes

2. Pearson Correlation ranges between -1 to 1 whereas, Cosine Similarity belongs to the interval between 0 to 1.
   - Yes

9. Mention the RMSE and MAPE that you got while evaluating the Matrix Factorization model.

RMSE: 0.76533128944678

MAE: 0.6878217940191732

Actionable Insights

- 18 unique Genres of movies: - 'Animation', "Children's", 'Comedy', 'Adventure', 'Fantasy', 'Romance', 'Drama', 'Action', 'Crime', 'Thriller', 'Horror', 'Sci-Fi', 'Documentary', 'War', 'Musical', 'Mystery', 'Film-Noir', 'Western'
- Out of 6040 Users, 4331 (71.7 %) users are male and 1709 (28.3 %) users are female (Ratings are mostly provided by males)
- The number of users in each age group and each occupation are cataloged in Table 3.3 and Table 3.4 respectively
- 72.7 % of users are aged between 18 and 44
- Out of 6040 users,
- Most watched category is college/grad student (12%)
- Mean of rating is 3.58
- 61% of users had given 3 – 4 rating
- 90% of the movies were released
- The least watched hour is between 10 and 11
- 60% of movies are watched during August, November, and December
- The User ID, title of movies, age group, occupation code, and movie genres of top 5, bottom 5, top 5 mean, and bottom 5 mean count are summarized in Table 3.5, Table 3.6, Table 3.7, Table 3.8, and Table 3.9 respectively
- Most movies belong to Comedy and Drama.

Recommendations

1. The case study recommends the service provider encourage users to provide more ratings, especially women.
2. More Comedy and Drama movies should be recommended.