Case Study 1 – Recommender System

Problem Statement:

Sam's next exam would be to build a "Recommender System" using the Singular Value Decomposition (SVD) algorithm. Questions would be asked on the basis of what you've learnt in the respective module.

Tasks To Be Performed:

1. Implementing User-Based Recommender System using SVD (Singular Value Decomposition) method:

- a. Load the 'ratings' and 'movies' datasets which is a part of 'MovieLense'
- b. Find the unique number of users and movies in the 'ratings' dataset
- c. Create a rating matrix for the 'ratings' dataset and store it in 'Ratings'
- d. Load the 'ratings' dataset as SVD's Dataset object and compute 3-fold cross-validation using the SVD object
- e. Find all the movies rated as 5 stars by user id '5' and store it in 'ratings_1' data frame
- f. Create a shallow copy of the 'movies' dataset and store the result in 'user_5'
- g. Train a recommender system using the SVD object and predict the ratings for user id '5'
- h. Print the top10 movie recommendations for the user id '5'

In [1]:
```
import pandas as pd
from surprise import Reader, Dataset, SVD
from surprise.model_selection import cross_validate
```

In [2]:
```
ratings = pd.read_csv(r"csv files/ratings.csv")
ratings
```

Out[2]:

	userId	movieId	rating	timestamp
0	1	2	3.5	1112486027
1	1	29	3.5	1112484676
2	1	32	3.5	1112484819
3	1	47	3.5	1112484727
4	1	50	3.5	1112484580
...	...	...	...	...
1048570	7120	168	5.0	1175543061
1048571	7120	253	4.0	1175542225
1048572	7120	260	5.0	1175542035
1048573	7120	261	4.0	1175543376
1048574	7120	266	3.5	1175542454

1048575 rows × 4 columns

In [3]:
```
movies = pd.read_csv(r"csv files/movies.csv")
movies
```

Out[3]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27275	131258	The Pirates (2014)	Adventure
27276	131260	Rentun Ruusu (2001)	(no genres listed)
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 3 columns

In [4]:
```
n_users = ratings['userId'].nunique()
n_users
```

Out[4]:
```
7120
```

In [5]:
```
n_movies = ratings['movieId'].nunique()
n_movies
```

Out[5]:
```
14026
```

In [6]:
```
Ratings = ratings.pivot(index='userId', columns='movieId', values='rating').fillna(0)
Ratings
```

Out[6]:

movieId	1	2	3	4	5	6	7	8	9	10	...	129350	129354	129428	129707	130052	130073	130219	130462	130490	130642
userId																					
1	0.0	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7116	4.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7117	4.0	0.0	4.0	0.0	0.0	5.0	3.0	0.0	1.0	3.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7118	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7119	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7120	4.5	4.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

7120 rows × 14026 columns

In [7]:
```
#conda install -c conda-forge scikit-surprise
```

In [8]:
```
reader = Reader()
data = Dataset.load_from_df(ratings[['userId','movieId','rating']], reader)
```

In [9]:
```
svd = SVD()
```

In [10]:
```
cross_validate(svd, data, measures=['RMSE', 'MAE'], cv=3, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm SVD on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    0.8432  0.8452  0.8471  0.8452  0.0016
MAE (testset)     0.6462  0.6474  0.6484  0.6474  0.0009
Fit time          0.66    0.50    0.45    0.53    0.09
Test time         3.47    3.41    3.15    3.34    0.14
```

Out[10]:
```
{'test_rmse': array([0.84323337, 0.84516507, 0.84714116]),
 'test_mae': array([0.64624967, 0.64741827, 0.64843356]),
 'fit_time': (0.6597347259521484, 0.4951817989349437, 0.44805502128601),
 'test_time': (3.465430736541748, 3.412599802017212, 3.147939443588257)}
```

In [11]:
```
ratings.head()
```

Out[11]:

	userId	movieId	rating	timestamp
0	1	2	3.5	1112486027
1	1	29	3.5	1112484676
2	1	32	3.5	1112484819
3	1	47	3.5	1112484727
4	1	50	3.5	1112484580

In [12]:
```
# Find all the movies rated as 5 stars by user id '5' and store it in 'ratings_1' data frame
ratings_1 = ratings[(ratings['userId']==5) * (ratings['rating']==5)]
ratings_1 = ratings_1.set_index('movieId')
ratings_1 = ratings_1.join(movies)['title']
ratings_1
```

```
C:\Users\Roy\AppData\Local\Temp\ipykernel_14516\4114181218.py:2: UserWarning: evaluating in Python space because the '*' operator is not supported by numexpr for the bool dtype, use '&' instead.
  ratings_1 = ratings[(ratings['userId']==5) * (ratings['rating']==5)]
```

Out[12]:
```
movieId
11                        Dracula: Dead and Loving It (1995)
62        Don't Be a Menace to South Central While Drink...
141                                             Gospa (1995)
150                                  Addiction, The (1995)
260                                Ladybird Ladybird (1994)
318       Strawberry and Chocolate (Fresa y chocolate) (...
364                                             Maverick (1994)
368                                      Reality Bites (1994)
377                    When a Man Loves a Woman (1994)
380                                         Bad Company (1995)
440                    Even Cowgirls Get the Blues (1993)
454                   Geronimo: An American Legend (1993)
457                                          Go Fish (1994)
500                                        No Escape (1994)
508                              Puppet Masters, The (1994)
531                                        Short Cuts (1993)
588                 Snow White and the Seven Dwarfs (1937)
589                          Beauty and the Beast (1991)
590                                        Pinocchio (1940)
594                                 Love and a .45 (1994)
595              Wooden Man's Bride, The (Yan shen) (1994)
671                   Coup de torchon (Clean Slate) (1981)
720                              Original Gangstas (1996)
736                          Man from Down Under, The (1943)
780       My Life and Times With Antonin Artaud (En comp...
832                                     Big Squeeze, The (1996)
1028                       Long Kiss Goodnight, The (1996)
1035                              Shadow Conspiracy (1997)
1036                                            Jude (1996)
1079                                         Top Gun (1986)
1080                                  American Strays (1996)
1097                    People vs. Larry Flynt, The (1996)
1136                                      Love in Bloom (1935)
1196                               Full Metal Jacket (1987)
1198                                          Henry V (1989)
1210      Seventh Seal, The (Sjunde inseglet, Det) (1957)
1291                           Alien³ (a.k.a. Alien 3) (1992)
1393                                     Turbulence (1997)
Name: title, dtype: object
```

In [13]:
```
user_5 = movies.copy()
user_5 = user_5.reset_index()
user_5
```

Out[13]:

	index	movieId	title	genres
0	0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	2	3	Grumpier Old Men (1995)	Comedy|Romance
3	3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...	...
27273	27273	131254	Kein Bund für's Leben (2007)	Comedy
27274	27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27275	27275	131258	The Pirates (2014)	Adventure
27276	27276	131260	Rentun Ruusu (2001)	(no genres listed)
27277	27277	131262	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 4 columns

In [14]:
```
data = Dataset.load_from_df(ratings[['userId','movieId','rating']], reader)

trainset = data.build_full_trainset()
svd.fit(trainset)
```

Out[14]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x1cd367ea020>
```

In [15]:
```
user_5['Estimate_Score'] = user_5['movieId'].apply(lambda x: svd.predict(5,x).est)
user_5 = user_5.drop(['movieId', 'genres', 'index'], axis=1)
user_5 = user_5.sort_values('Estimate_Score', ascending=False)
user_5.head(10)
```

Out[15]:

	title	Estimate_Score
2239	Life Is Beautiful (La Vita è bella) (1997)	5.000000
8937	Decalogue, The (Dekalog) (1989)	5.000000
1173	Raiders of the Lost Ark (Indiana Jones and the...	5.000000
936	It's a Wonderful Life (1946)	5.000000
17874	Avengers, The (2012)	4.999758
523	Schindler's List (1993)	4.977937
1944	Saving Private Ryan (1998)	4.953859
15208	Cosmos (1980)	4.935604
10286	Serenity (2005)	4.915887

148	Apollo 13 (1995)	4.902070

In [ ]: