Movie Recommendation System

Team Members:

Danielle Aras

Soumyadeb Maity

Jhansi Saketa B V

About the Project

In a crowded entertainment market, movie streaming services and advertisers need to present customers with the most relevant recommendations possible to maintain customer interest and loyalty. This project will use a database of user-submitted movie ratings to explore ways to generate movie recommendations and predict how users may rate future movies.

Data

MovieLens Dataset

"Latest Full" set will be used (27 million data points)

URL: https://grouplens.org/datasets/movielens/

The dataset has been downloaded to Danielle's computer

Tools

- Python
- Pandas
- Scikit-learn
- Matplotlib

Classification

SVD

Singular Value Decomposition

- Unsupervised
- Collaborative filtering
- Factorization of the matrix into three matrices
- Method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning

Classification

SVD

Imported the SVD from Surprise library

↓

Load the rating.csv into the surprise reader object (setting the rating scale 0 to 5)

↓

Train the model

↓

Prediction

Evaluation - SVD

```
from surprise.model_selection import train_test_split
from surprise import accuracy
# define train test function
def train_test_algo(algo, label):
    training_set, testing_set = train_test_split(rating_dataframe, test_size = 0.25)
    algo.fit(training_set)
    test_output = algo.test(testing_set)
    test_dataframe = pd.DataFrame(test_output)
    print("RMSE -",label, accuracy.rmse(test_output, verbose = False))
    print("MAE -", label, accuracy.mae(test_output, verbose=False))
    print("MSE -", label, accuracy.mse(test_output, verbose=False))

    return test_dataframe
```

```
train_test_SVD = train_test_algo(SVD_algo, "SVD_algo")
print(train_test_SVD.head())
```

```
RMSE - SVD_algo 0.7979897029603923
MAE - SVD_algo 0.6025927732135373
MSE - SVD_algo 0.6367875660308151
      uid     iid  r_ui       est                    details
0   39569   59107   2.0  2.652985  {'was_impossible': False}
1  135415   59784   2.5  3.090653  {'was_impossible': False}
2  180345   36529   5.0  4.189357  {'was_impossible': False}
3  103615   45728   3.5  2.758480  {'was_impossible': False}
4  237509  133281   3.0  3.443806  {'was_impossible': False}
```

Train-Test Split Evaluation Method:

- 75 Percent for Training Data and 25 Percent for Testing Data.
- RMSE - 0.8
- MAE - 0.6
- MSE - 0.6

Results

SVD

```
Top Three Movies Recommendations for first 5 users
       userId  movieId    rating  imdbId  \
0           1      318  4.905599  111161
1           1      527  4.848964  108052
2           1     1704  4.686185  119217
11986       2      318  4.527728  111161
11987       2    26086  4.481109   56300
11988       2     7153  4.462035  167260
23972       3      318  4.646206  111161
23973       3     2858  4.617762  169547
23974       3      632  4.583413  114671
35958       4     5418  5.000000  258463
35959       4     4011  5.000000  208092
35960       4     3275  5.000000  144117
47944       5     2858  5.000000  169547
47945       5     7153  4.996867  167260
47946       5     5618  4.975823  245429

                                                   title
0                       Shawshank Redemption, The (1994)
1                                  Schindler's List (1993)
2                                 Good Will Hunting (1997)
11986                   Shawshank Redemption, The (1994)
11987  Occurrence at Owl Creek Bridge, An (La rivière...
11988  Lord of the Rings: The Return of the King, The...
23972                   Shawshank Redemption, The (1994)
23973                            American Beauty (1999)
23974        Land and Freedom (Tierra y libertad) (1995)
35958                       Bourne Identity, The (2002)
35959                                     Snatch (2000)
35960                       Boondock Saints, The (2000)
47944                            American Beauty (1999)
47945  Lord of the Rings: The Return of the King, The...
47946  Spirited Away (Sen to Chihiro no kamikakushi) ...
```

Classification

Nearest Neighbor

Nearest Neighbor (KNN)

- Unsupervised
- Collaborative filtering
- Finding similarity between searched movie name and other existing movie in database
- Cosine metric is chosen over Euclidean to calculate distance among data points.

Results - KNN

```
Other recommendations for movie: God Father:



Get the Gringo (2012), similarity or distance : 0.7862867116928101
Cold in July (2014), similarity or distance : 0.784233808517456
Imperium (2016), similarity or distance : 0.7790570855140686
The Trust (2016), similarity or distance : 0.7748041152954102
Forsaken (2016), similarity or distance : 0.7747822999954224
Triple 9 (2016), similarity or distance : 0.7712504267692566
Walk Among the Tombstones, A (2014), similarity or distance : 0.75550585
Bastille Day (2016), similarity or distance : 0.7426487803459167
In a Valley of Violence (2016), similarity or distance : 0.7412719726562
Criminal (2016), similarity or distance : 0.7377589344978333
```

```
Other recommendations for movie: Titanic:



Jaws (1975), similarity or distance : 0.8694257736206055
Lethal Weapon (1987), similarity or distance : 0.8685681819915771
Romancing the Stone (1984), similarity or distance : 0.8664568662643433
Patriot Games (1992), similarity or distance : 0.8655930757522583
Backdraft (1991), similarity or distance : 0.8655616044998169
Red Dawn (1984), similarity or distance : 0.8647364974021912
Superman (1978), similarity or distance : 0.8645337820053101
Presidio, The (1988), similarity or distance : 0.8639472126960754
Thelma & Louise (1991), similarity or distance : 0.8606656193733215
Night to Remember, A (1958), similarity or distance : 0.8453669548034668
```

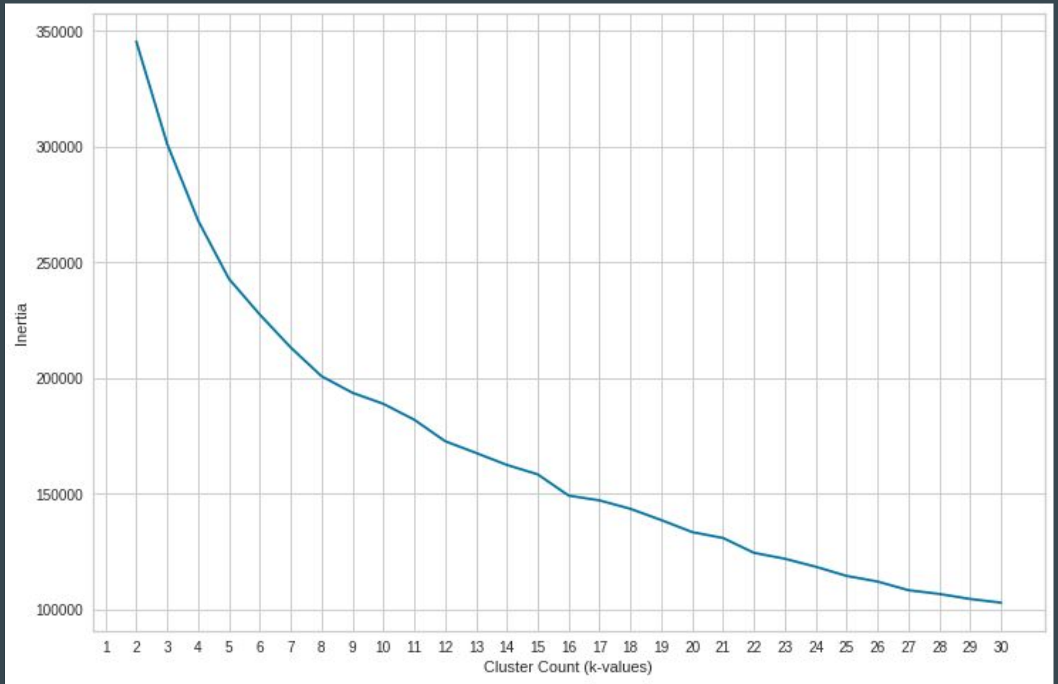
Clustering

K-Means

K-Means clustering

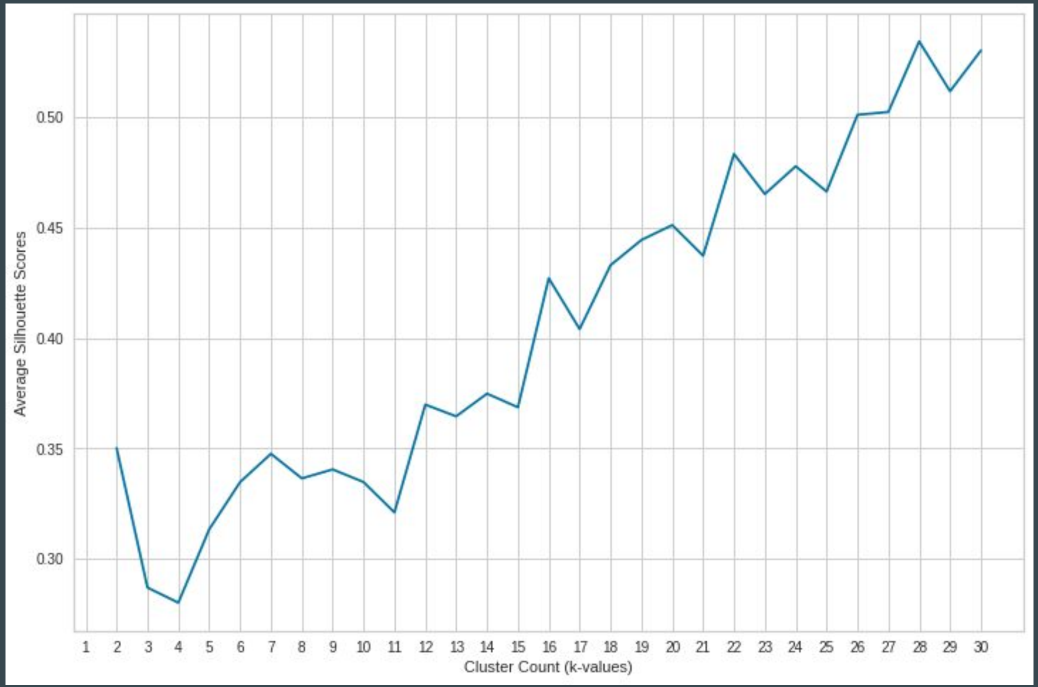
- Unsupervised
- Group users into clusters based on their movie ratings
- Points are assigned to the nearest cluster based on Euclidean distance from centroids
- Once grouped, recommendations can be generated for users based on the highest rated movies in their cluster

Evaluation - K Means



Inertia (Sum of Squared Error)

A measure of similarity between points in the same cluster, used for the “Elbow Method”



Silhouette Coefficient

A measure of distance between different clusters and distance between points within each cluster

Knowledge Gained (Results)

Clustering

```
def get_top_recs(cluster, df_model, df_ratings, min_ratings, top_n):
    cluster_ids = df_model[df_model['Cluster'] == cluster]['userId'].tolist()
    df_cluster = df_ratings[df_ratings['userId'].isin(cluster_ids)]
    com_ratings = df_cluster.groupby('movieId').filter(lambda x: len(x) > min_ratings)
    return com_ratings.groupby('title').mean()['rating'].reset_index().sort_values('rating', ascending=False).head(top_n)
```

```
get_top_recs(1, df_avg, df_c, 10, 10)
```

	title	rating
287	It's a Wonderful Life (1946)	4.725000
283	Intouchables (2011)	4.642857
298	Killing Fields, The (1984)	4.590909
454	Shawshank Redemption, The (1994)	4.584906
565	Wallace & Gromit: The Wrong Trousers (1993)	4.576923
77	Bonnie and Clyde (1967)	4.576923
0	12 Angry Men (1957)	4.571429
223	Good, the Bad and the Ugly, The (Buono, il bru...	4.562500
31	Apocalypse Now (1979)	4.562500
315	Life Is Beautiful (La Vita è bella) (1997)	4.545455

Using a simple function and existing dataframes, top-rated recommendations for each cluster can be generated.

Applications

We can use a hybrid model to combine all the discussed methods like k-means clustering, SVD, and KNN for use in a recommendation system for any movie library.

Consider how services like Netflix, Hulu, etc. have multiple types of recommendations.

Reference

[1] Carlos A. Gomez-Uribe and Neil Hunt. 2015. The Netflix recommender system: Algorithms, business value, and innovation. ACM Trans. Manage. Inf. Syst. 6, 4, Article 13 (December 2015), 19 pages. DOI: http://dx.doi.org/10.1145/2843948

[2] Eyrun A. Eyjolfsdottir, Gaurangi Tilak, Nan Li (2008), "MovieGEN: A Movie Recommendation System", 2008 Conference Proceedings.

[3] Roman, Victor (2019), "Unsupervised Classification Project: Building a Movie Recommender with Clustering Analysis and K-Means", Towards Data Science, https://towardsdatascience.com/unsupervised-classification-project-building-a-movie-recommender-with-clustering-analysis-and-4bab0738efe6

[4] Nixon, Alex Escola (2020), "Building a movie content based recommender using tf-idf", Towards Data Science, https://towardsdatascience.com/content-based-recommender-systems-28a1dbd858f5