Recommendation Systems

Mini Project AI 705

IMT2021010 Raj Bunsha
IMT2021068 Adithya Sunil
IMT2021080 Pannaga Bhat
IMT2021095 Jashwanth Kadaru

Neighbourhood based Collaboration System

We tried different ways of getting recommendations using neighbourhood based collaborative system.

The specifications of which are given ahead:

1. Cosine Correlation coefficient
2. Pearson’s Correlation coefficient
3. Weighted Correlation coefficient

Neighbourhood based Collaboration System(Input)

Movies were given as input to get output

Movies:

	Title	genre	ratings
378	Wolf (1994)	[Drama, Horror]	1.0
374	Speechless (1994)	[Comedy, Romance]	5.0
373	Speed (1994)	[Action, Romance, Thriller]	5.0
385	Colonel Chabert, Le (1994)	[Drama, Romance, War]	5.0
376	True Lies (1994)	[Action, Adventure, Comedy, Romance]	5.0

Example of romantic movies inputted

Cosine Similarity

Horror movies:- given an input with horror movies having higher rating and children's movies having very low rating it gives the following results

```
Recommendations based on cosine similarity
('Matewan (1987)', ['Drama'])
('Spellbound (1945)', ['Mystery', 'Romance', 'Thriller'])
('Gold Diggers: The Secret of Bear Mountain (1995)', ['Adventure', "Children's"])
('Body Snatcher, The (1945)', ['Horror'])
('Welcome to the Dollhouse (1995)', ['Comedy', 'Drama'])
```

It doesn't give us great results as there are many children's movies which were very low rated by the user

Pearson's Correlation

On giving the same input and using pearson's correlation we get better results.

```
Recommendations based on pearson similarity
('Nightmare on Elm Street, A (1984)', ['Horror'])
('Repulsion (1965)', ['Thriller'])
('Omen, The (1976)', ['Horror'])
('Torso (Corpi Presentano Tracce di Violenza Carnale) (1973)', ['Horror'])
('Vampires (1998)', ['Horror'])
```

So pearson's correlation gives us better recommendations
when compared to cosine similarity

Weighted Pearson correlation

Upon giving the same inputs and getting the movies, we get a bit of variety along with horror, although the top 3 movies remain the same, other movies are replaced by movies from other genre's, from the ones that are a bit similar or the complementary genre's of the original movies. It replaced movies that many people have generally watched

```
Recommendations based on weighted pearson similarity
('Nightmare on Elm Street, A (1984)', ['Horror'])
('Repulsion (1965)', ['Thriller'])
('Omen, The (1976)', ['Horror'])
('Amos & Andrew (1993)', ['Comedy'])
('Superman II (1980)', ['Action', 'Adventure', 'Sci-Fi'])
```

Some other examples (children's movies)

```
Recommendations based on cosine similarity
('Matewan (1987)', ['Drama'])
('Surviving the Game (1994)', ['Action', 'Adventure', 'Thriller'])
('Body Snatcher, The (1945)', ['Horror'])
('Welcome to the Dollhouse (1995)', ['Comedy', 'Drama'])
('Mad Max Beyond Thunderdome (1985)', ['Action', 'Sci-Fi'])

Recommendations based on pearson similarity
('Boiling Point (1993)', ['Action', 'Drama'])
('To Sir with Love (1967)', ['Drama'])
('Life Less Ordinary, A (1997)', ['Romance', 'Thriller'])
('Pokémon: The First Movie (1998)', ['Animation', "Children's"])
('Crow: City of Angels, The (1996)', ['Action', 'Thriller'])

Recommendations based on weighted pearson similarity
('Boiling Point (1993)', ['Action', 'Drama'])
('Crow: City of Angels, The (1996)', ['Action', 'Thriller'])
('To Sir with Love (1967)', ['Drama'])
('Pokémon: The First Movie (1998)', ['Animation', "Children's"])
('Surviving the Game (1994)', ['Action', 'Adventure', 'Thriller'])
```

Some other examples (Romance)

```
Recommendations based on cosine similarity
('Oliver! (1968)', ['Musical'])
('Body Snatcher, The (1945)', ['Horror'])
('Welcome to the Dollhouse (1995)', ['Comedy', 'Drama'])
('European Vacation (1985)', ['Comedy'])
('Airplane! (1980)', ['Comedy'])

Recommendations based on pearson similarity
('Boiling Point (1993)', ['Action', 'Drama'])
('Life Less Ordinary, A (1997)', ['Romance', 'Thriller'])
('Superman II (1980)', ['Action', 'Adventure', 'Sci-Fi'])
('Blood In, Blood Out (a.k.a. Bound by Honor) (1993)', ['Crime', 'Drama'])
("Clara's Heart (1988)", ['Drama'])

Recommendations based on weighted pearson similarity
('Boiling Point (1993)', ['Action', 'Drama'])
('Life Less Ordinary, A (1997)', ['Romance', 'Thriller'])
('Superman II (1980)', ['Action', 'Adventure', 'Sci-Fi'])
("Clara's Heart (1988)", ['Drama'])
('From Here to Eternity (1953)', ['Drama', 'Romance', 'War'])
```

SVD and K-Means: Genre matrix

Genre matrix stores the average ratings of movies of a particular genre by every user. We cannot use the user-movie matrix as it is sparse.

This matrix is reduced using SVD and K-means is performed, to cluster users based on their genre preferences.

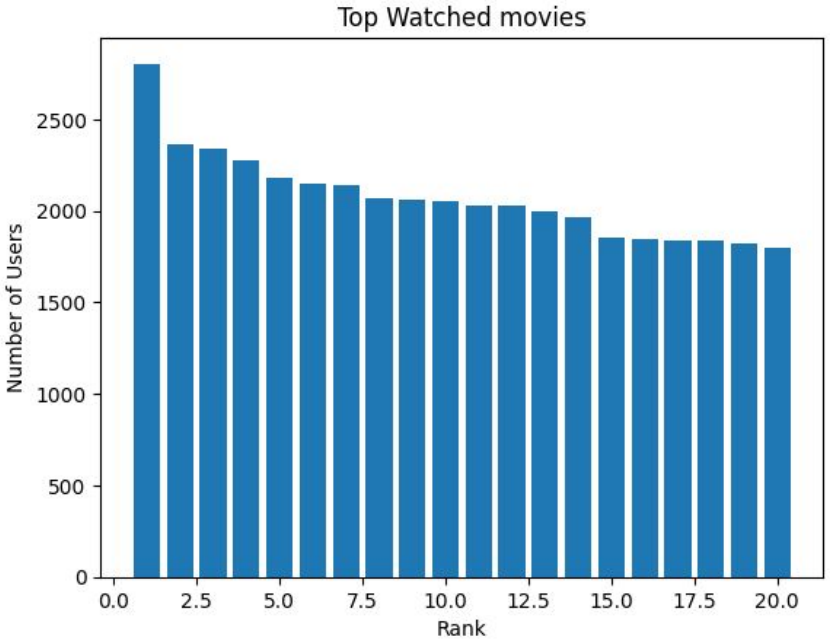
When a new user is added, the closest centroid is found in terms of our distance metric.

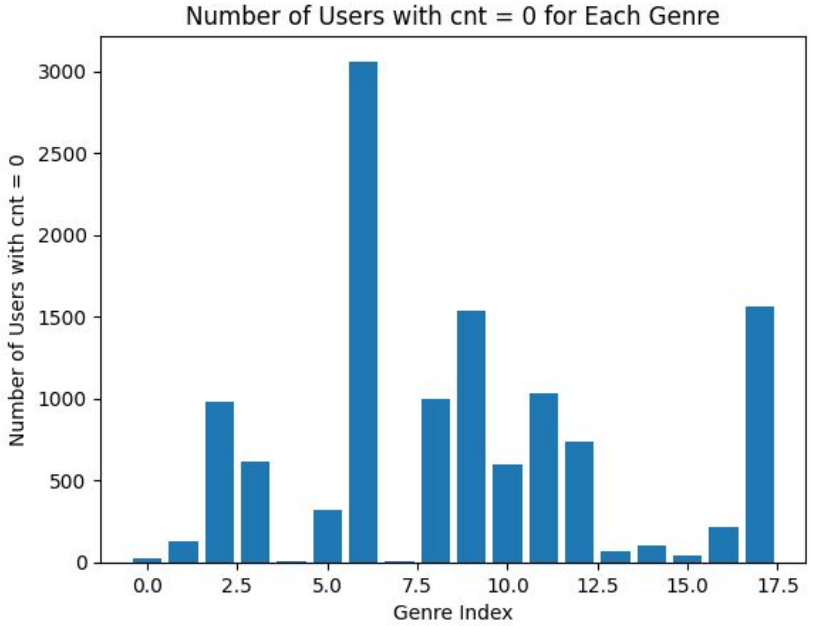
The predicted ratings for a user is the average of the ratings of all users belonging to the cluster.

SVD and K-Means: Preprocessing

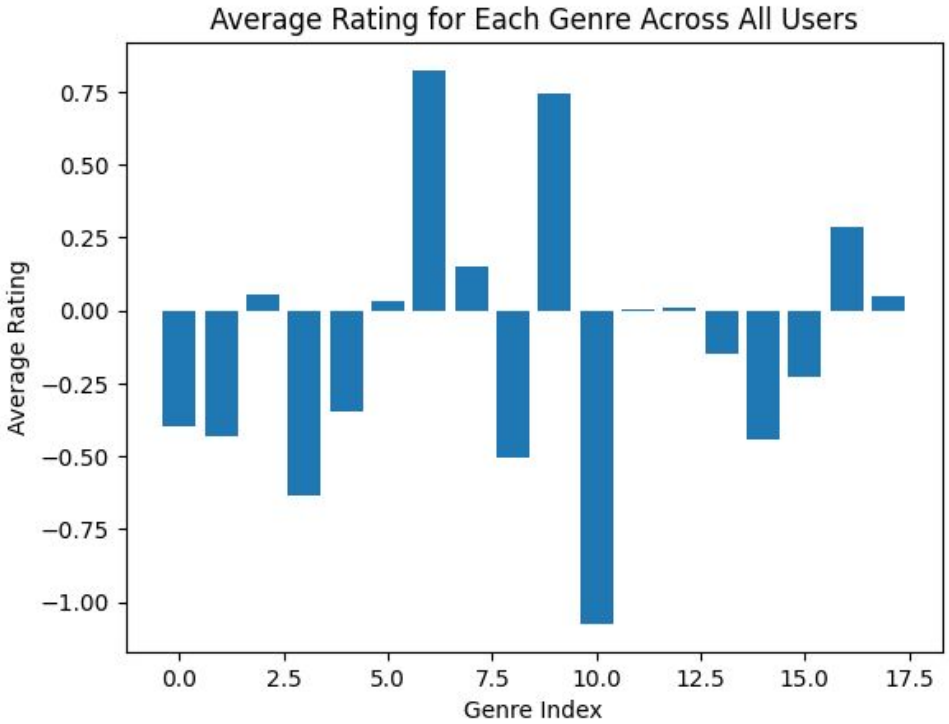
- Mean Shifting
- Standardization
- Normalization
- Weighted Reviews

SVD and K-means: EDA





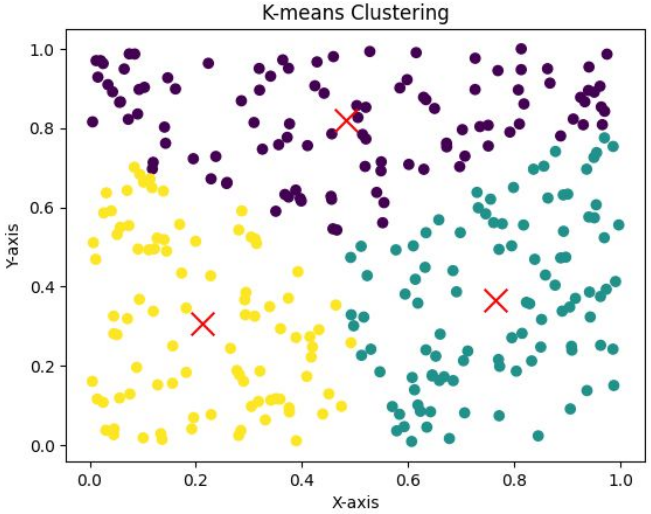
SVD and K-means: EDA

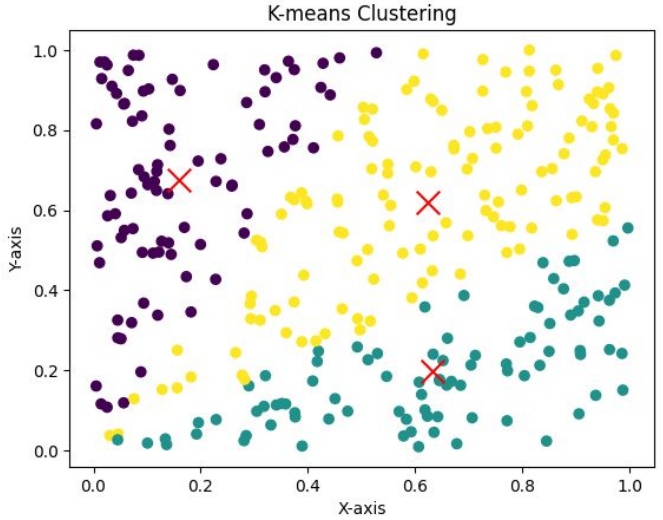


K-means

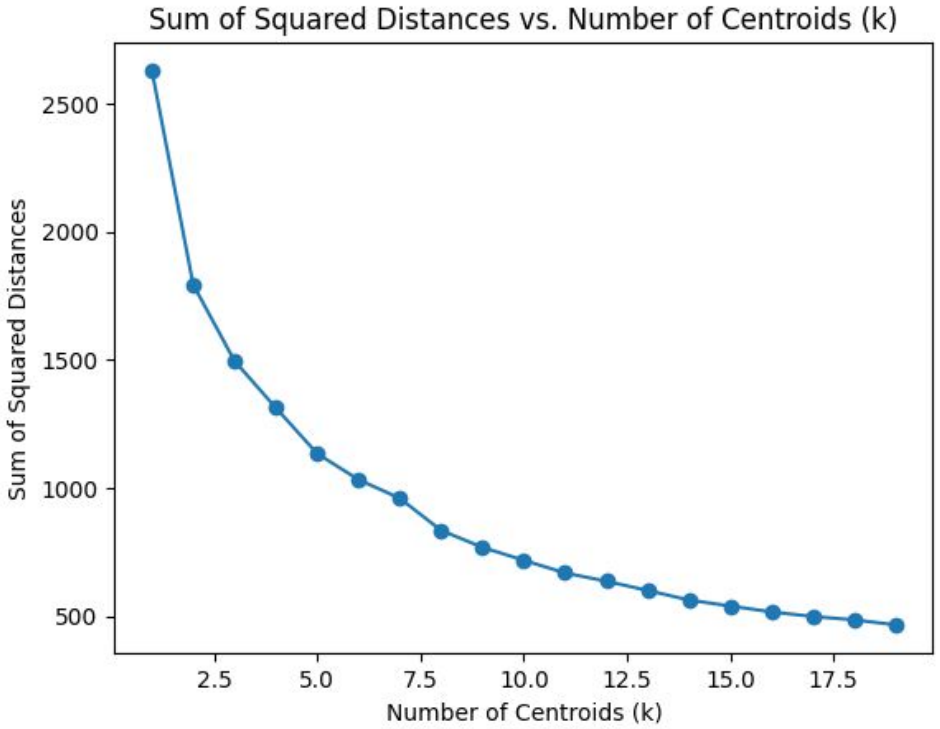
K-Means ++ algorithm used.

2 different distance metrics: euclidean and cosine.





K-means: Elbow graph



SVD: Eigenvector convergence

When multiplied by a matrix repeatedly, the direction of a vector converges to the that of the primary eigenvector.

```
def power_iteration(A, max_iter = 100, epsilon = 1e-6):
    n = A.shape[0]
    curr = np.random.rand(n)
    curr = curr / norm(curr)

    for i in range(0, max_iter):
        next = np.dot(A, curr)
        val = norm(next)
        if val < epsilon:
            return np.zeros(n)
        next = next / val
        if abs(np.dot(next, curr)) > 1 - epsilon:
            break
        curr = next

    #val = np.dot(curr.T, np.dot(A, curr)) / np.dot(curr.T, curr)
    #print(val**0.5)

    return curr
```

SVD Optimizations

- We don’t need to calculate the eigenvectors for both A^TA and AA^T
- If we know one, we can use the fact that matrices U and V are orthonormal to calculate the other.
- Therefore the time complexity only depends on the smaller of the two, (the number of rows and the number of columns)
- This is very useful for the Genre Matrix as it only has 18 columns

Clustering

```
Cluster 3
182
[ 1 14  0 15 16 10  8 13 17  6  3 12  5  4  2 11  9  7]
Shawshank Redemption, The (1994) | Drama,
0.2004977197136982
Usual Suspects, The (1995) | Crime,Thriller,
0.1943619990987534
Schindler's List (1993) | Drama,War,
0.18543951822964372
Boys Don't Cry (1999) | Drama,
0.18266911591884483
This Is Spinal Tap (1984) | Comedy,Drama,Musical,
0.1806580830947652
GoodFellas (1990) | Crime,Drama,
0.17440386958163542
Room with a View, A (1986) | Drama,Romance,
0.17397632751621278
Sixth Sense, The (1999) | Thriller,
0.17260932937883722
Singin' in the Rain (1952) | Musical,Romance,
0.16909374312486447
American Beauty (1999) | Comedy,Drama,
0.16798034139351228
```

Drama Cluster

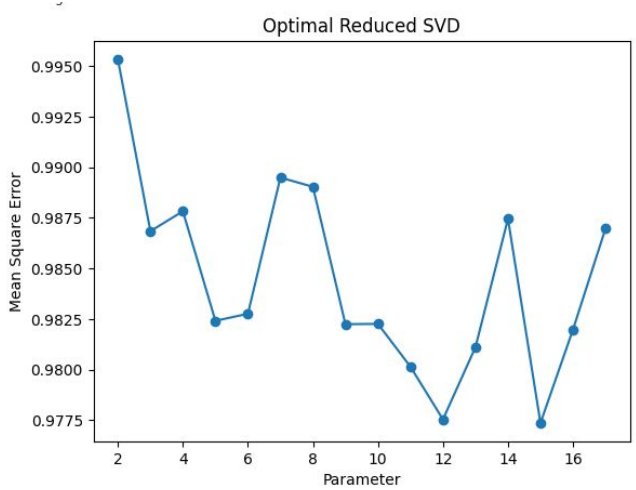
```
Cluster 8
161
[10  0 17  3  1 14 13 15  4  8  5 16  2  6  7  9 12 11]
Singin' in the Rain (1952) | Musical,Romance,
0.28846278327243824
Shawshank Redemption, The (1994) | Drama,
0.24952041867044428
Godfather, The (1972) | Action,Crime,Drama,
0.2486063985273529
My Fair Lady (1964) | Musical,Romance,
0.2404158283433095
All That Jazz (1979) | Musical,
0.23463012175920694
Schindler's List (1993) | Drama,War,
0.23150747878620345
Sixth Sense, The (1999) | Thriller,
0.2209569001355528
Dancer in the Dark (2000) | Drama,Musical,
0.21239744405644928
This Is Spinal Tap (1984) | Comedy,Drama,Musical,
0.20560485853444982
Sound of Music, The (1965) | Musical,
0.20504401640208184
```

Musical Cluster

Sometimes, movie popularity overcomes genre preference!

Error and Hyperparameter Tuning

- Data was split into test (20%) and train (80%)
- Mean square error between (Non-NULL) input data and predicted rating values was evaluated
- Error was plotted against number of singular values (for k = 18)



Optimal number of Singular Values

- The graph on the previous slide shows dips in error at 12 and 15
- We decided to fix the number of singular values at 12, to capture the latent features while also removing noise.
- All 18 singular values are shown on the right

```
154.8557140377255
133.2576050663377
130.2664510380285
99.27498064064756
94.73064894549186
92.22955326516153
87.38617630354483
74.72455838007753
72.03904252394065
70.81020280915115
70.22145007428787
66.90346569502115
61.857209863924886
53.33238655466719
51.322079894111525
46.25248541508946
42.54160735850884
40.01419771293211
```

PCA as an alternative to SVD

- PCA can be used instead of SVD for filtering noise and dimensionality reduction.
- The PCA function was tested along with the k-means algorithm on 18 clusters.
- The mean square error given by this method is slightly higher than that of SVD: 1.0233205132375414