Anime Recommendation System

Pranay Ponkiya -17BCP041 Zeel Prajapati - 17BCP042

Prerak Shah - 17BCP044

Problem Statement

Recommend a list of animes to the user, which he/she would like by using the reviews of the said user for the animes he/she watched previously.

Methodology Used

- Identify the liked animes by the user according to the ratings given by him/her.

  Definition of Like :

  Anime which got rating higher than mean rating which the user has given will assign as like.

- Combine the anime and user datasets and create a cross table which will show the animes which each user likes.

Methodology Used (Contd.)

- We used Principal Component Analysis to convert our original variables to a new set of variables, which are a linear combination of the original set of variables. Our main goal is to reduce dimension of data for clustering and visualize.
- In order to apply K-Means Clustering we needed to select the optimal value of K. We choose the optimal value of K using the Silhouette Score and Inertia for K ranging between (2 to 8) and we found that K = 4 was the optimal number of cluster for our data.

Dataset Used

We took the dataset containing reviews of 76,000 users from myanimelist.net .

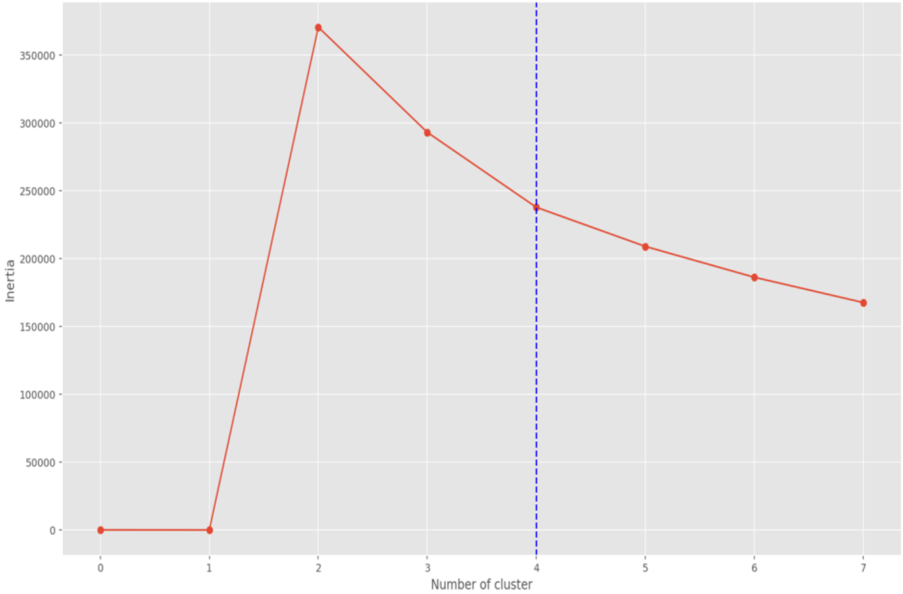
This data set contains information on user preference data from 73,516 users on 12,294 anime.

Files	Anime.csv	Rating.csv
Columns	● anime_id - myanimelist.net's unique id identifying an anime. ● name - full name of anime. ● genre - comma separated list of genres for this anime. ● type - movie, TV, OVA, etc. ● episodes - how many episodes in this show. (1 if movie). ● rating - average rating out of 10 for this anime. ● members - number of community members that are in this anime's "group".	● user_id - non identifiable randomly generated user id. ● anime_id - the anime that this user has rated. ● rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

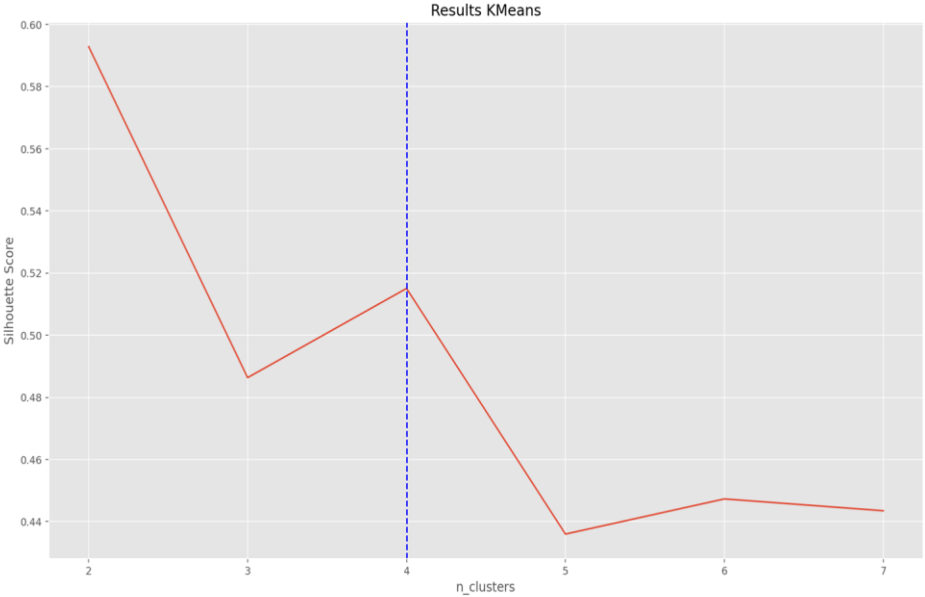
Steps to execute :

1. Open ipynb file in Google Colab
2. Upload Data (2 ways): a. Manually upload the 2 data files (anime and rating) (By clicking on left hand side file icon) b. Or upload the files into your google drive and than mount the drive
3. Don’t run all at the same time, run each cell individually (First time the session will crash due to memory issues, just rerun the entire code from start, it will run)

Screenshots of Output

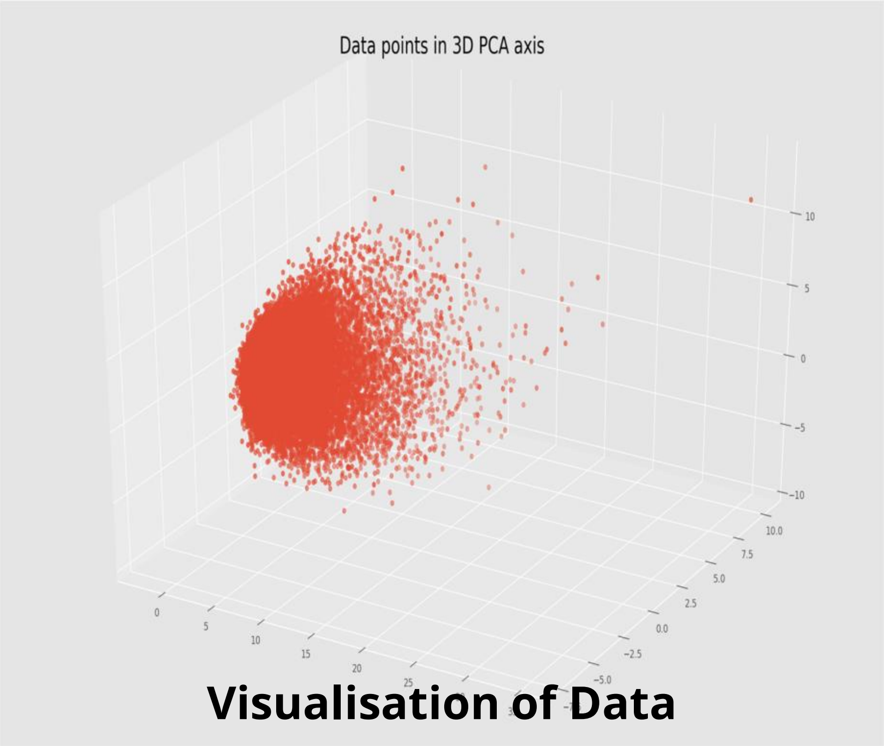


No. of Cluster Vs Inertia

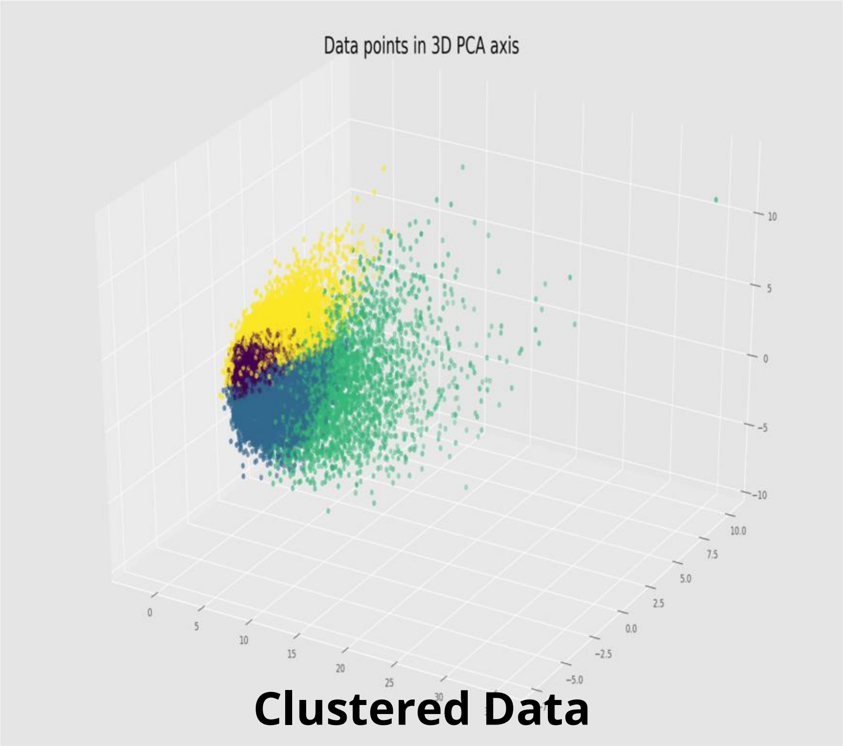


No. of Cluster Vs Silhouette Score

Screenshots of Output

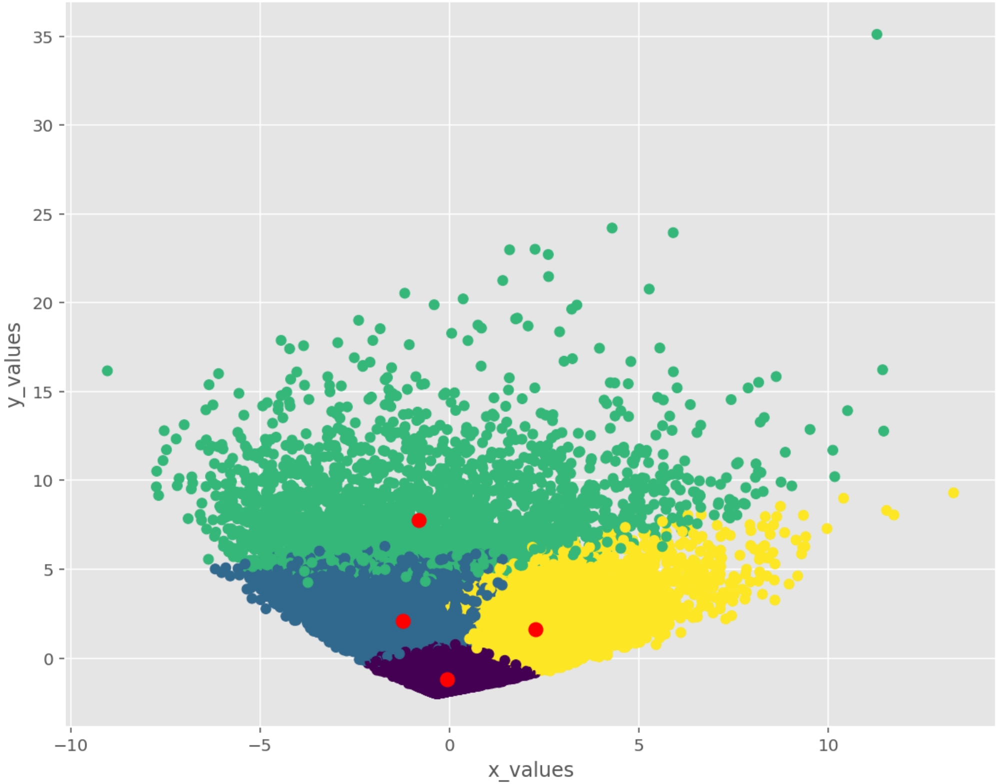


Visualisation of Data



Clustered Data

Screenshots of Output



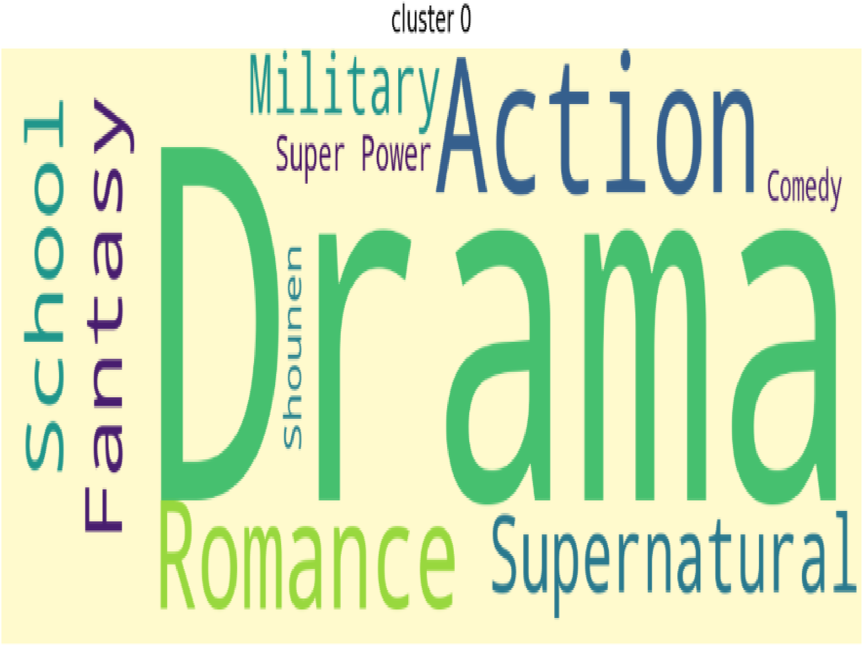
Data points in 2D PCA axis

Centers of all 4 clusters

```
[[-1.21436784 -0.07446326  0.09456676]
 [ 2.0789899  -1.23443221 -0.67653795]
 [ 7.77540598 -0.81045181  1.09154157]
 [ 1.64394214  2.26893543 -0.04665874]]
```

Screenshots of Output (Cluster 0)

1.	Death Note	0.344756
2.	Shingeki no Kyojin	0.206291
3.	Code Geass: Hangyaku no Lelouch	0.185705
4.	Sword Art Online	0.182254
5.	Fullmetal Alchemist: Brotherhood	0.179081
6.	Fullmetal Alchemist	0.174540
7.	Sen to Chihiro no Kamikakushi	0.173370
8.	Naruto	0.172973
9.	Elfen Lied	0.168987
10.	Angel Beats!	0.162700
11.	Ouran Koukou Host Club	0.155085
12.	Code Geass: Hangyaku no Lelouch R2	0.152646
13.	Toradora!	0.129085
14.	Howl no Ugoku Shiro	0.126210
15.	Clannad	0.121569



Outcomes

- After applying the PCA and K-Means Clustering the model defined four clusters of users having similar interests in animes.
- Below are the characteristics of the obtained clusters :
  - Top 15 Animes for each cluster
  - Top 10 Common Genres
  - Avg. No. of Episodes
  - Avg. User Rating
  - Avg. No of Members

Conclusions

- By applying PCA and K-Means clustering 4 clusters were created for the given dataset
- This model can be used to recommend any similar type of thing to users (ie. games, movies, products) when provided enough and relatable data.

References

- Scikit learn PCA : https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- Scikit learn KMeans : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Inertia and Silhouette Score : https://medium.com/@jyotiyadav99111/selecting-optimal-number-of-clusters-in-kmeans-algorithm-silhouette-score-c0d9ebb11308