Keene
STATE COLLEGE

Movie Recommender

Author: Quinlan Wood

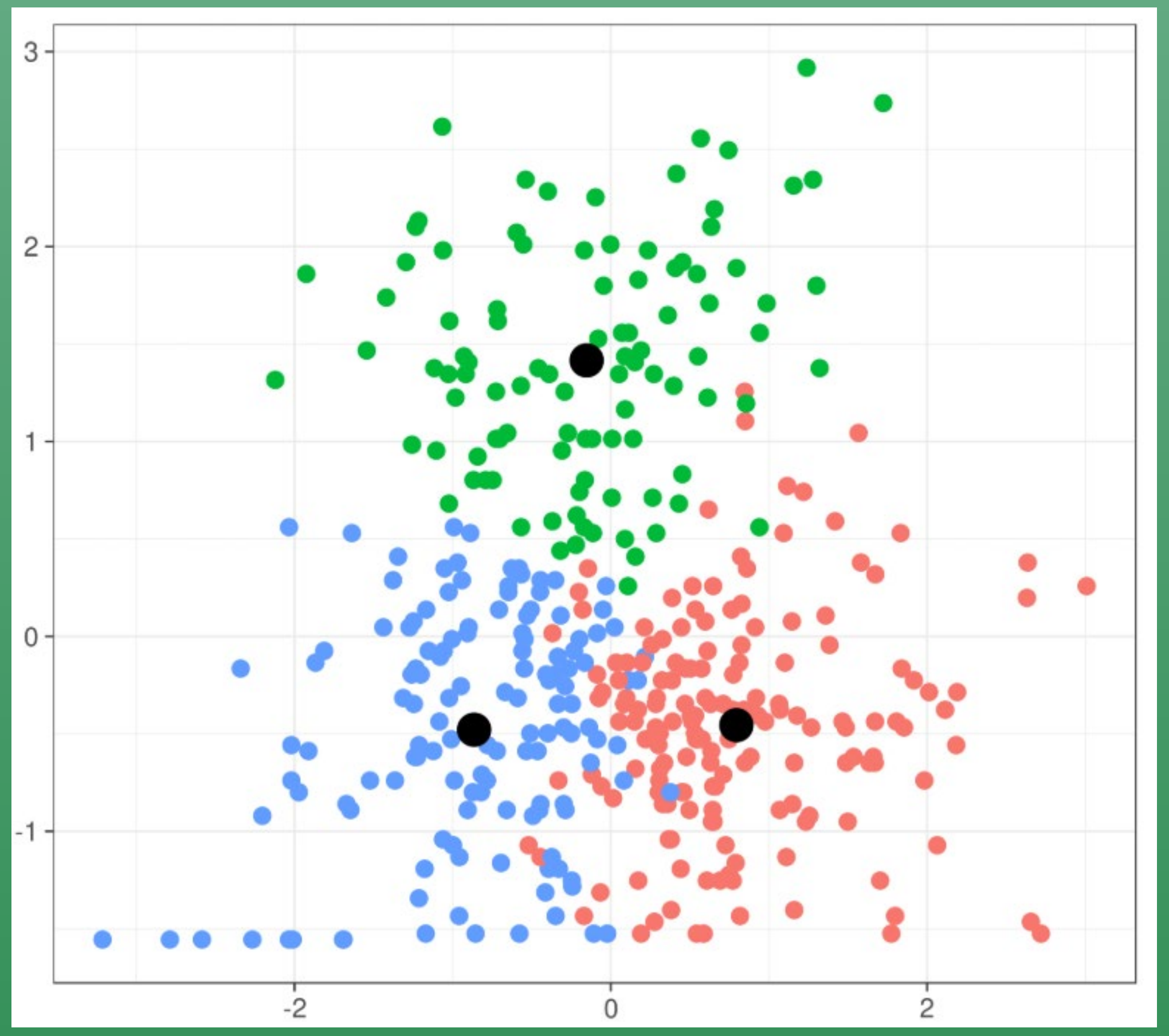
Acknowledgements:
https://grouplens.org/datasets/movielens/

Abstract

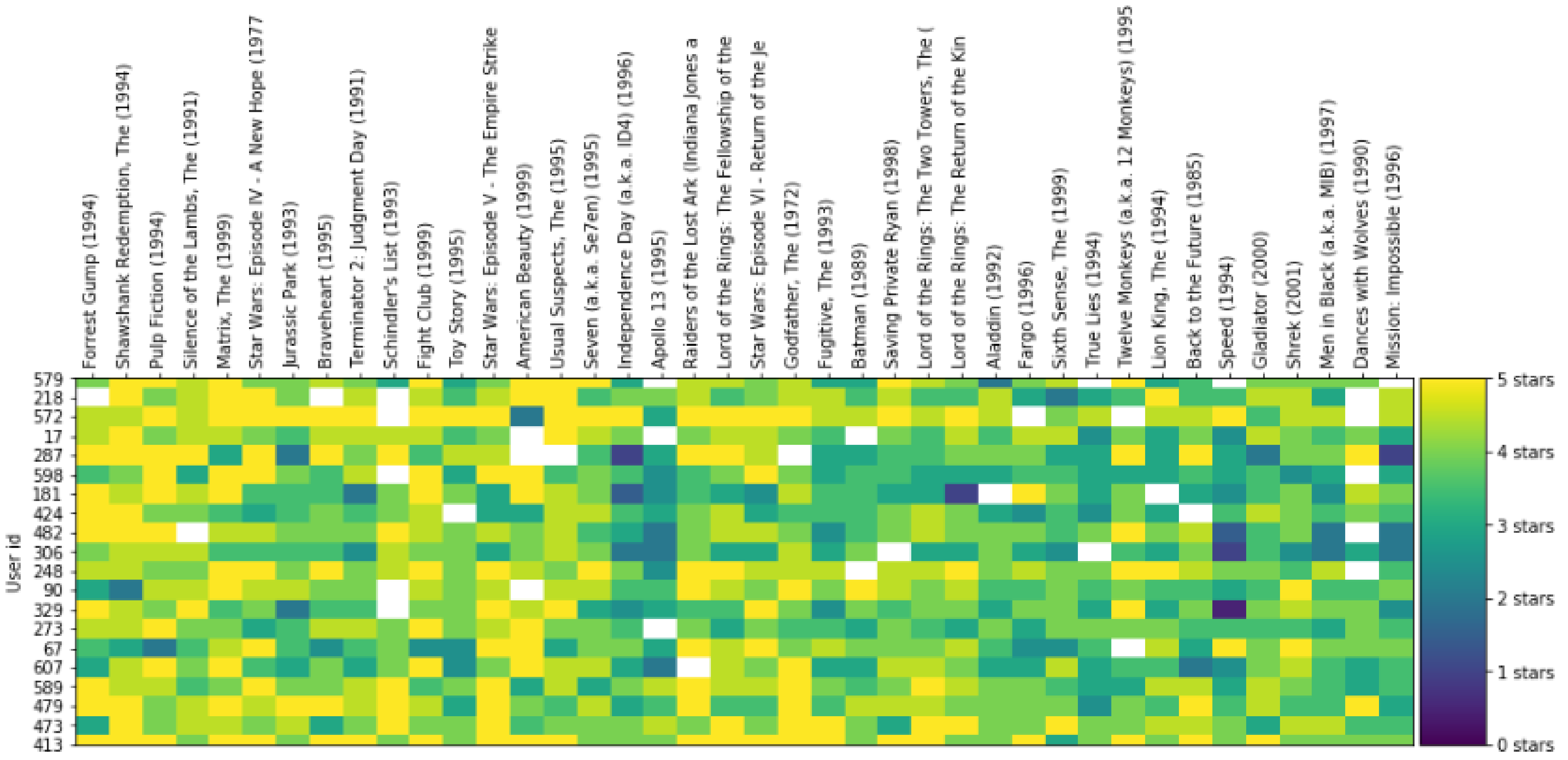
Have you every noticed that when you watch a video on Netflix or YouTube that soon after you will begin to see suggestion similar to what you recently watched? This is a type of market segmentation in which these streaming sites log all the recent movies and episode that you have been watching. Then by comparing you with other users with similar tastes they can suggest options that they think you would enjoy, but what if you recently watched variety of videos that you did not enjoy? This would result in your suggested videos being related to topics you did not enjoy. Therefore, I intend to create a market segmentation algorithm that will not only consider the recently watch videos of users, but it will also give higher priority to the rating the user gives the video. With more data being fed into the algorithm it will be able to more accurately give users suggestions that will be tailored to their liking.

K-Means Algorithm

- A popular clustering algorithm
- Centroid based
  - Each datapoint will be assigned a cluster based on it distance from that clusters centroid
- Will automatically generate different sets of clusters on the value of k provided
- Two inputs needed
  1. Number of clusters
  2. Number of training example

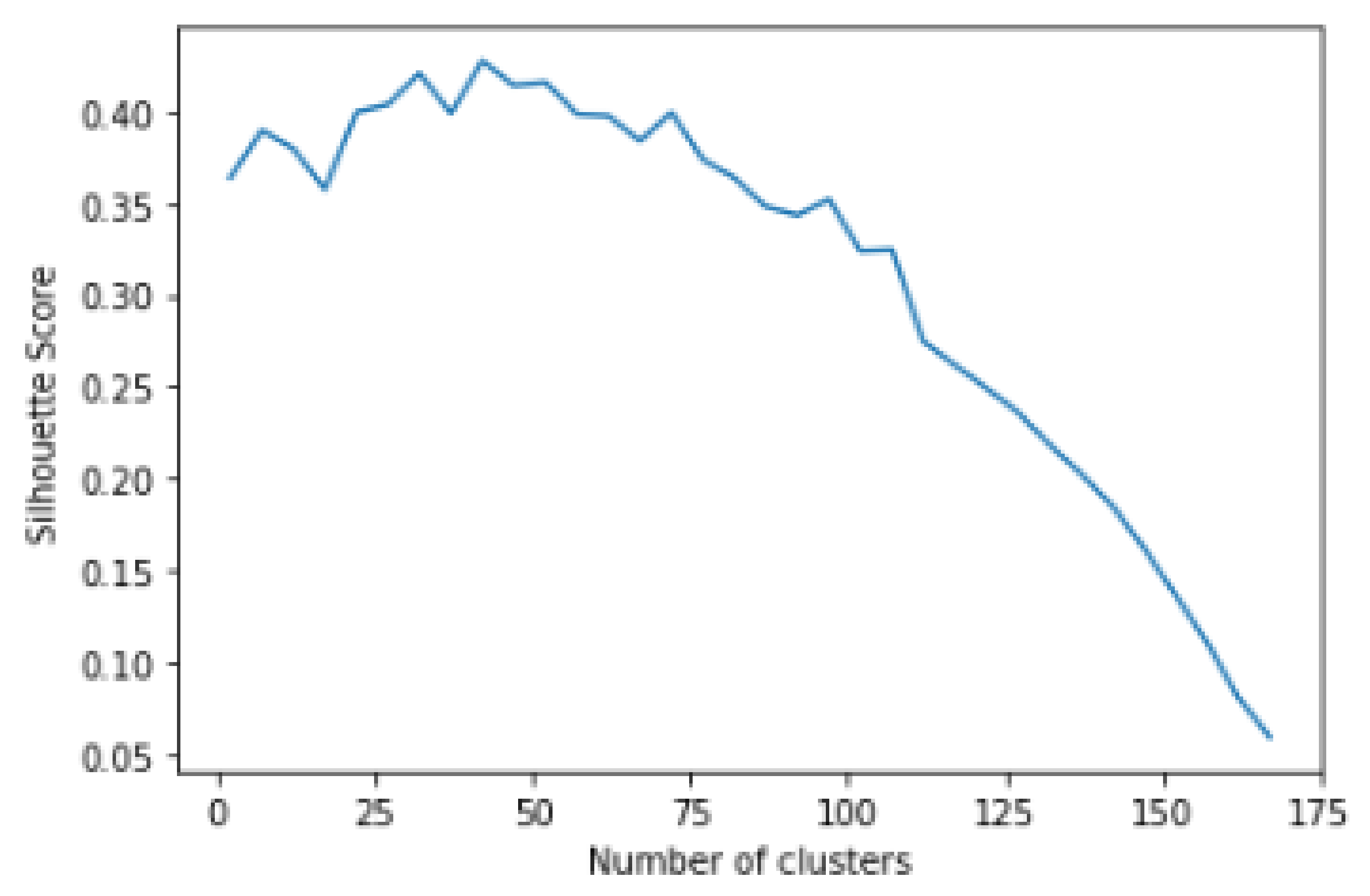


Raw data

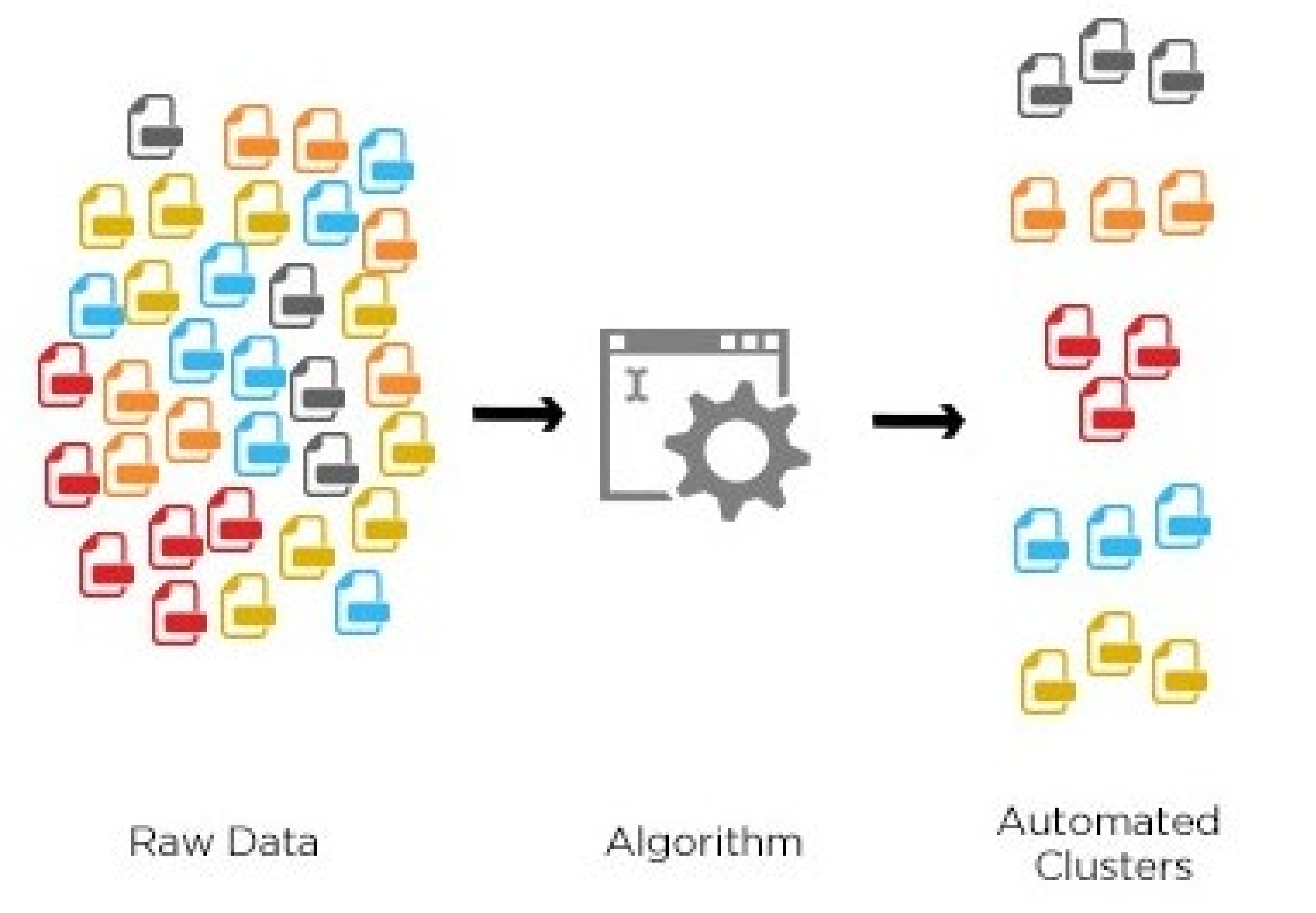


Finding number of clusters

When creating a K-Means model on the fields that must be given is a number of clusters. This tells the model how many groups to make from the data. In order to choose the best possible number of clusters I have compared the silhouette scores of the possible cluster values (higher is better) by running this test a number of times I have found that on average 25 clusters gives the best results for my model.



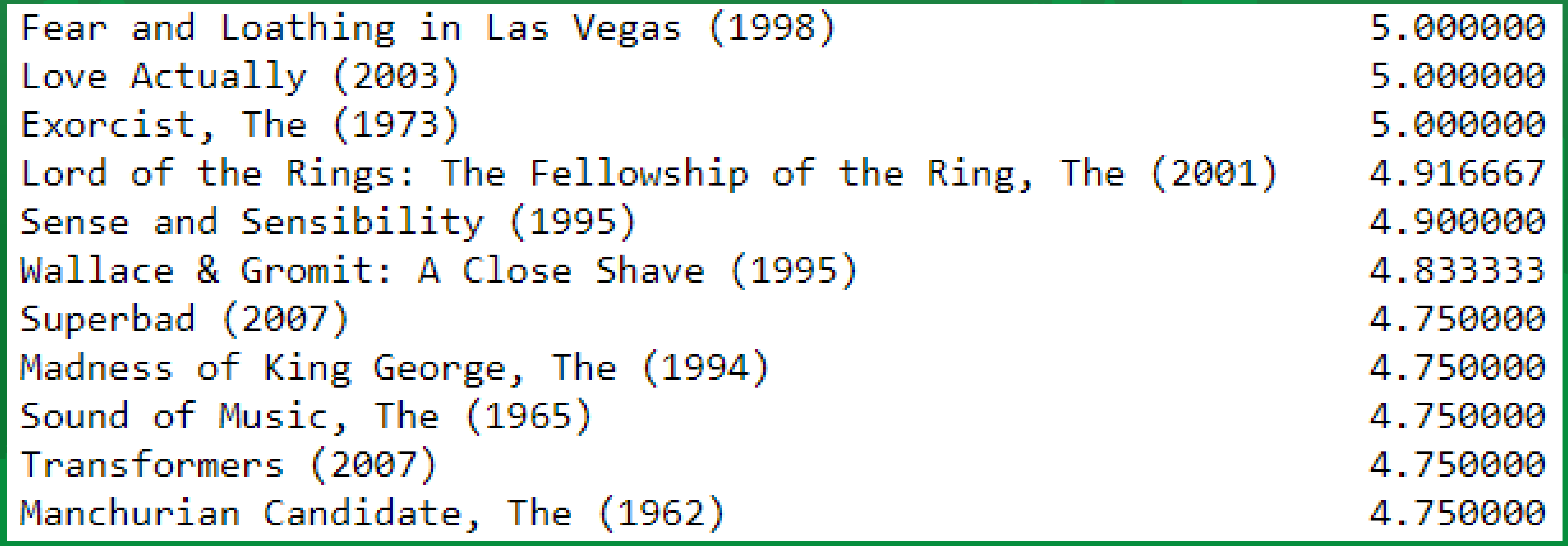
How clusters work



The K-Means algorithm takes the following steps to calculate the clusters:

1. Choose initial centroid values
2. Assign each data point to a cluster depending upon the distance between it and the centroids
3. The algorithm will then update the centroids and repeat steps 1 and 2
4. It will repeat steps 1 – 3 until the clusters in the previous iteration match the current clusters

Results



```
Fear and Loathing in Las Vegas (1998)                      5.000000
Love Actually (2003)                                       5.000000
Exorcist, The (1973)                                       5.000000
Lord of the Rings: The Fellowship of the Ring, The (2001)  4.916667
Sense and Sensibility (1995)                               4.900000
Wallace & Gromit: A Close Shave (1995)                     4.833333
Superbad (2007)                                            4.750000
Madness of King George, The (1994)                         4.750000
Sound of Music, The (1965)                                 4.750000
Transformers (2007)                                        4.750000
Manchurian Candidate, The (1962)                           4.750000
```