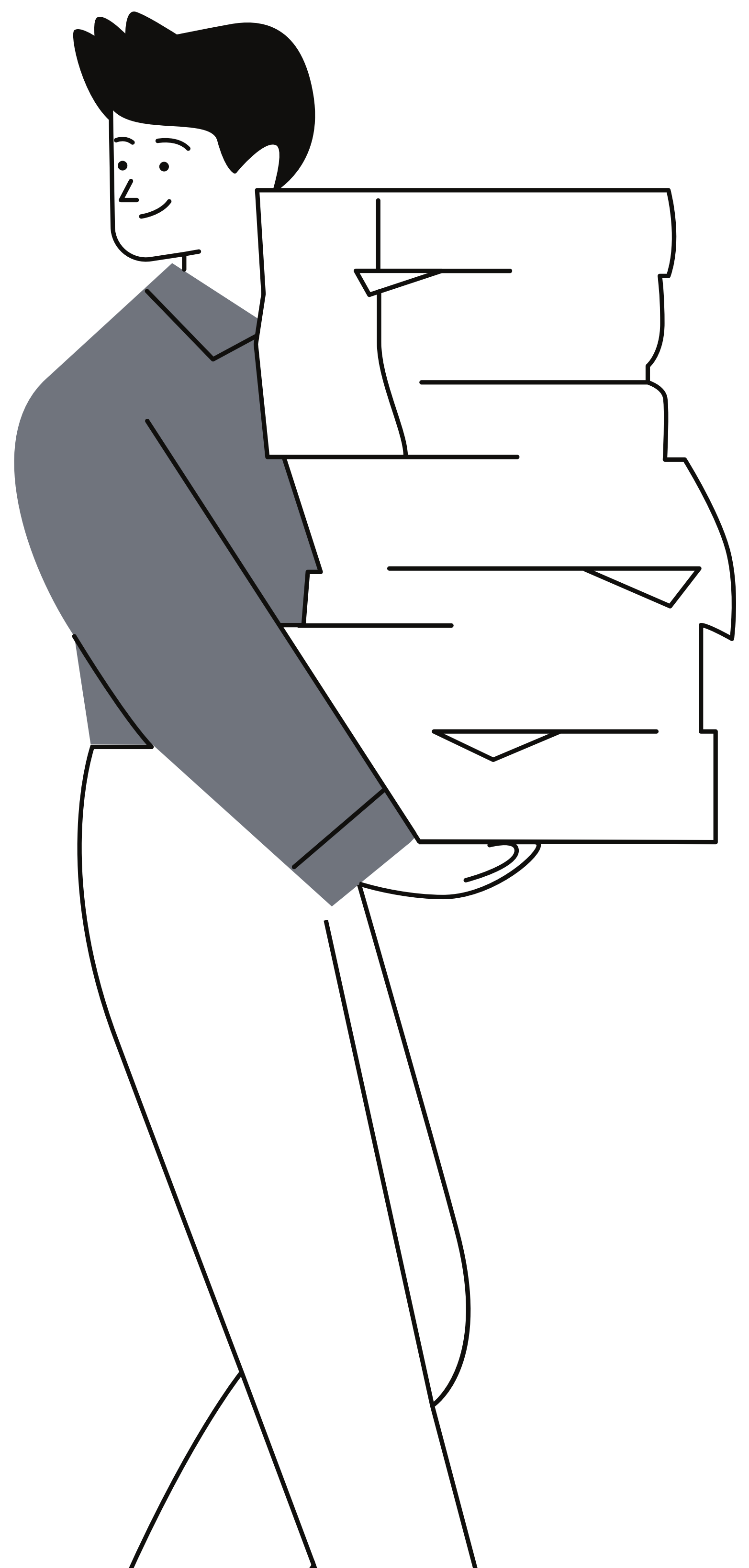
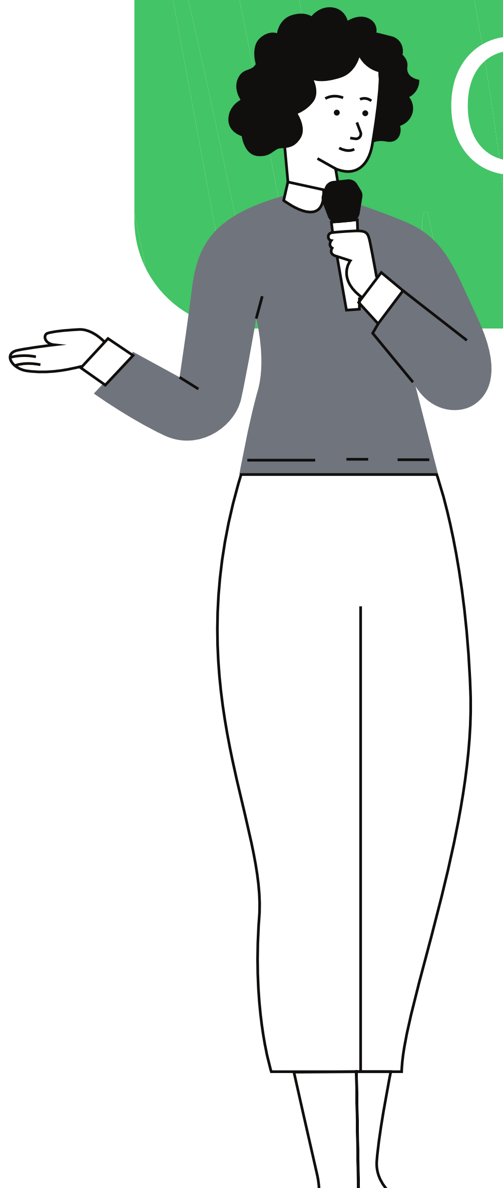
CodeX

Movie Recommendation Engine



MDM Company

Slide Contents



Objectives

- To develop a system of movie engine analysis.
- Analyzing the rating of movies helps to group the similar movies with each other for recommendations for other similar movies than the that has been watched.

Methodology

How the system is developed?

1 **Data Scraping**

We utilized the API to scrape from a website a total of almost 10000 rows of metadata.
Language: Python

2 **Data Cleaning**

We cleaned the data using Alteryx. In the context of our data, we remove unwanted characters and emojis from the scraped metadata.
Software: Alteryx
Language: Python

3 **Data Science Process**

Since we are doing a recommendation engine, we used the collaborative filtering which utilizes the use of matrix.
Language: Python

4 **Database**

Connect to MySQL Database

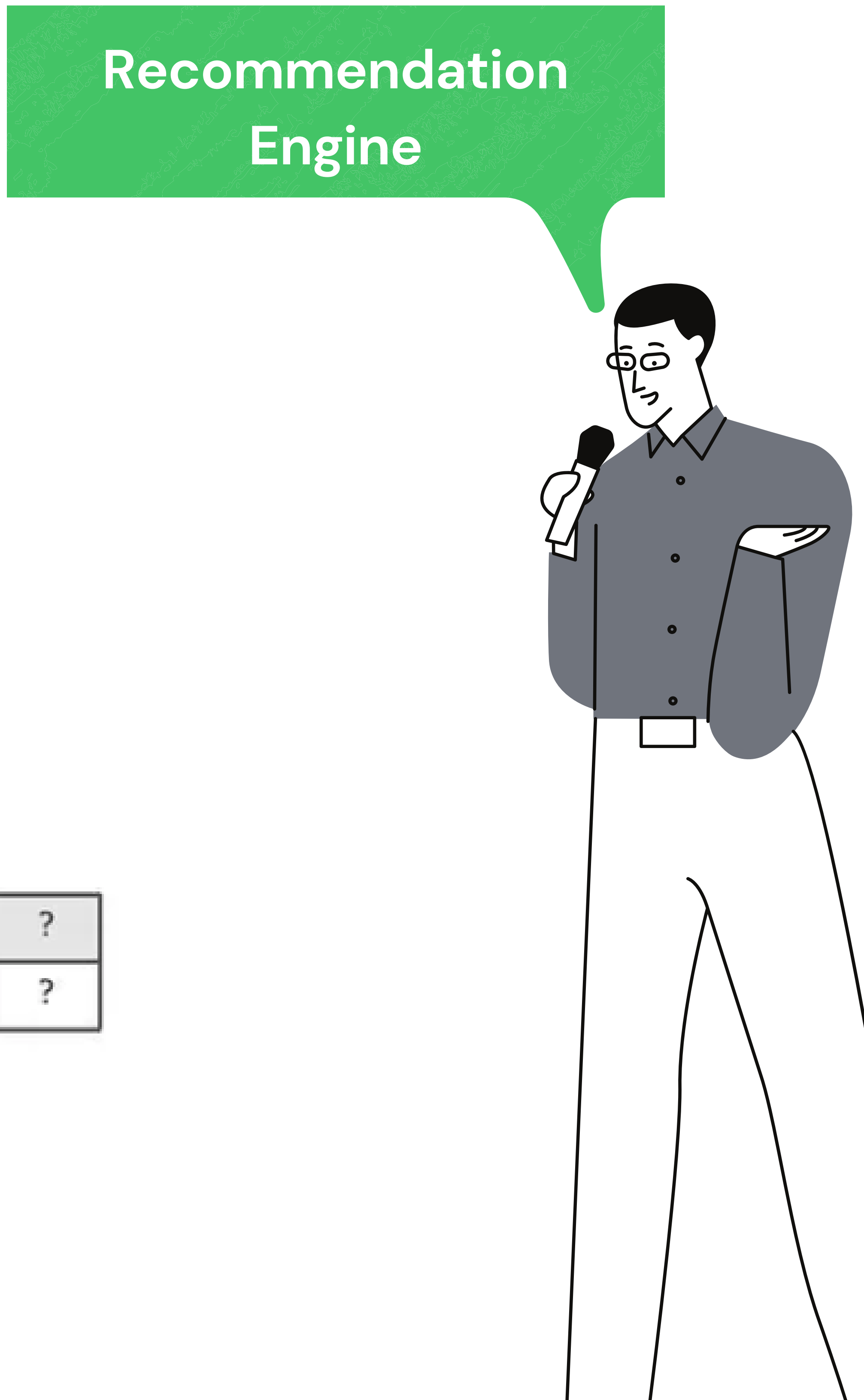
5 **Data Visualisation**

Use the Tableau visualizations and features to create a dashboard that meets your analysis objectives.
Software: Tableau

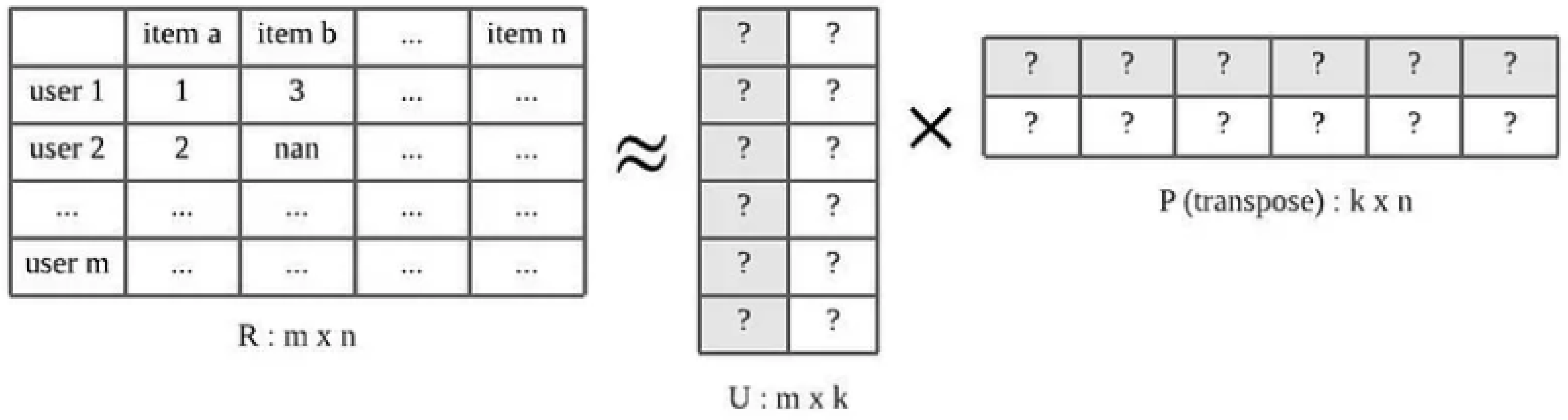
6 **System Development**

Developed a system using php.
Software: Sublime, Visual Studio Code

Data Analysis Techniques



- Collaborative Filtering: Used to find the relationship between the subset of the observation (user) and the observation (rating).
- Utilizing the use of matrix factorization as the intersection between the columns and rows are the score of how well the users rate the movie.
- After getting the score, we will use it to make a prediction through the production of clusters. In this project, we are using the K-Means algorithm to group the movies in clusters. Some movies can be found in different clusters



	item a	item b	...	item n
user 1	1	3	...	...
user 2	2	nan	...	...
...	...	...	...	...
user m	...	...	...	...

R : m x n

≈

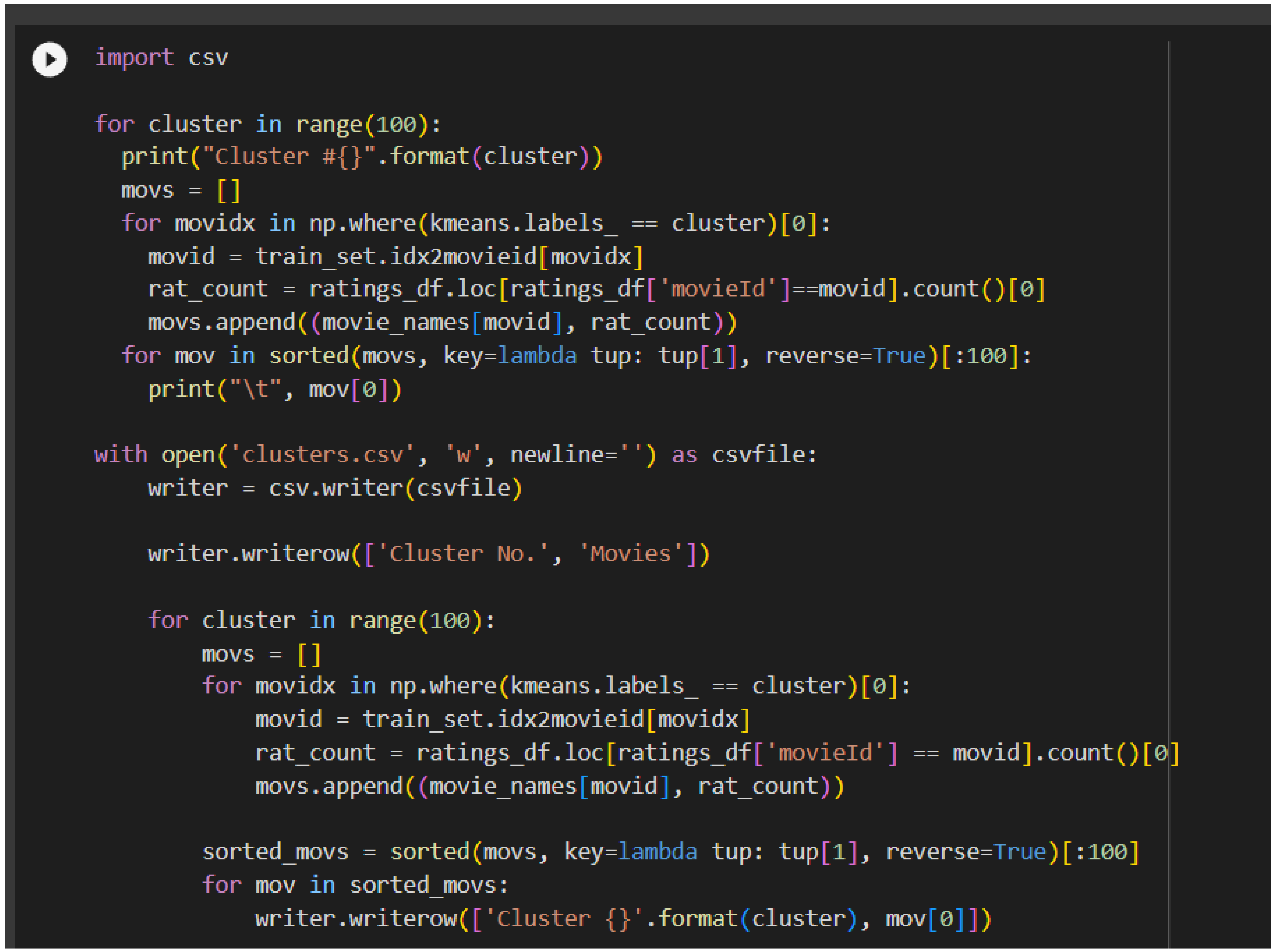
?	?
?	?
?	?
?	?
?	?
?	?

U : m x k

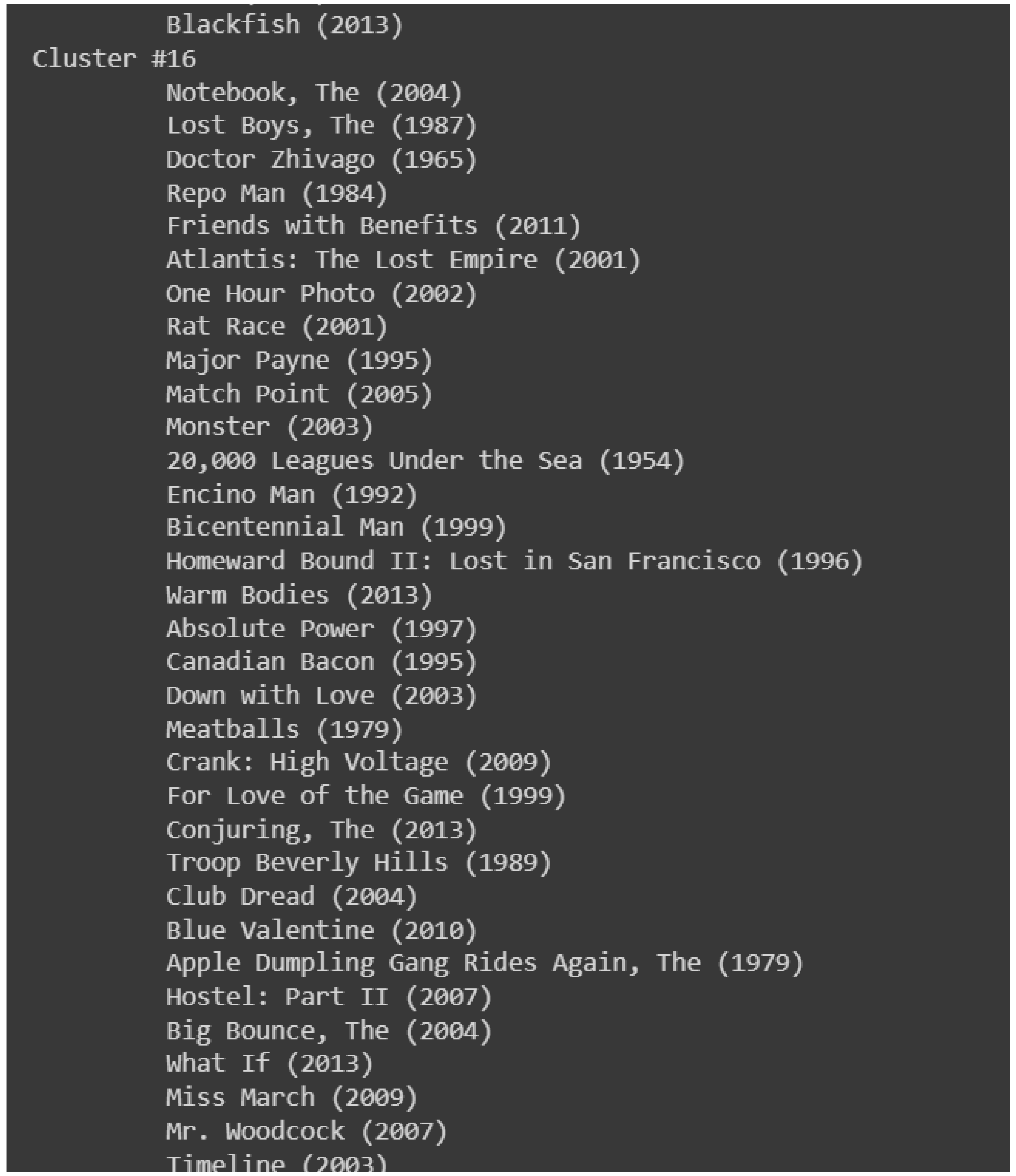
×

?	?	?	?	?	?
?	?	?	?	?	?

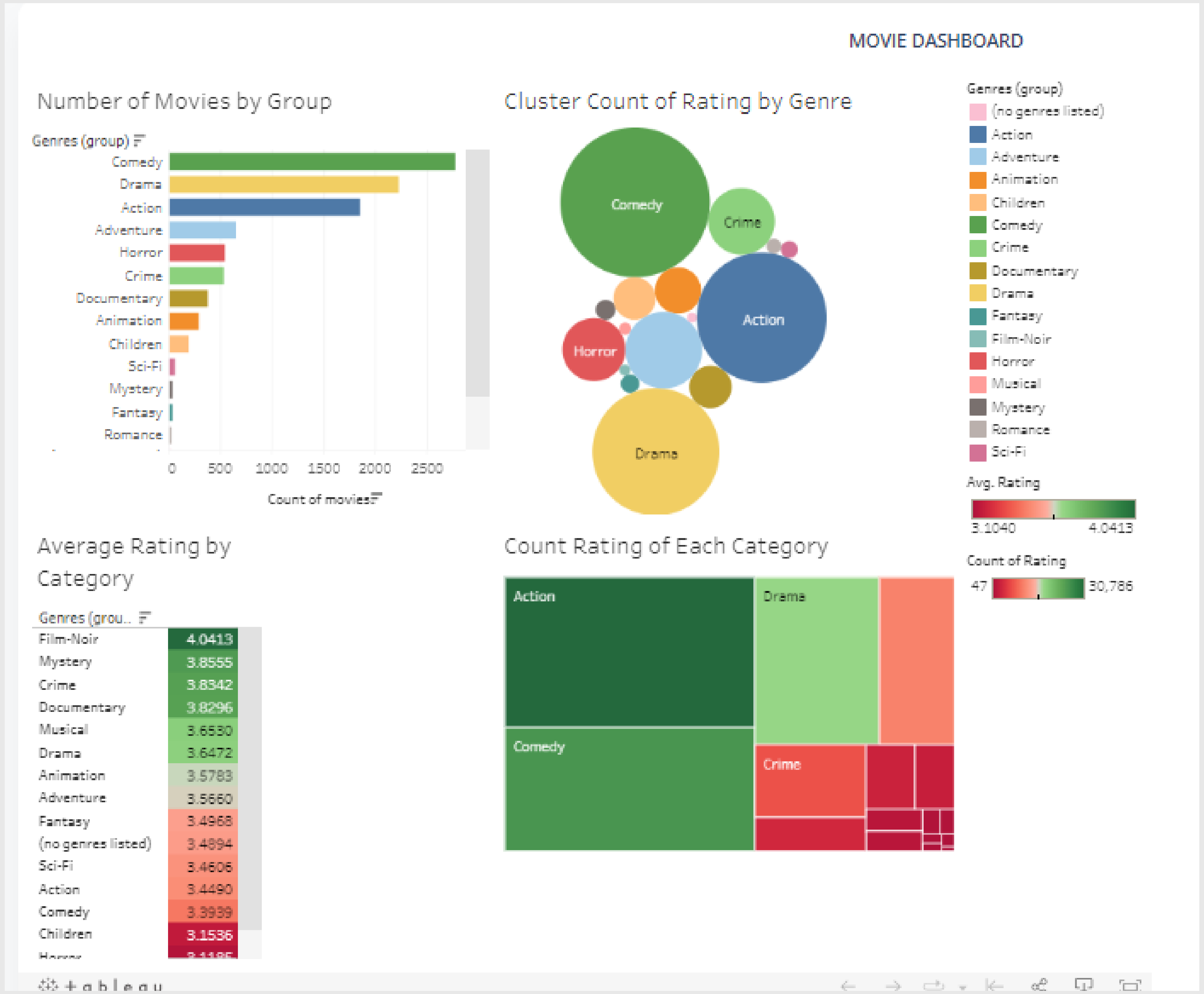
P (transpose) : k x n



```python
import csv

for cluster in range(100):
  print("Cluster #{}".format(cluster))
  movs = []
  for movidx in np.where(kmeans.labels_ == cluster)[0]:
    movid = train_set.idx2movieid[movidx]
    rat_count = ratings_df.loc[ratings_df['movieId']==movid].count()[0]
    movs.append((movie_names[movid], rat_count))
  for mov in sorted(movs, key=lambda tup: tup[1], reverse=True)[:100]:
    print("\t", mov[0])

with open('clusters.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)

    writer.writerow(['Cluster No.', 'Movies'])

    for cluster in range(100):
        movs = []
        for movidx in np.where(kmeans.labels_ == cluster)[0]:
            movid = train_set.idx2movieid[movidx]
            rat_count = ratings_df.loc[ratings_df['movieId'] == movid].count()[0]
            movs.append((movie_names[movid], rat_count))

        sorted_movs = sorted(movs, key=lambda tup: tup[1], reverse=True)[:100]
        for mov in sorted_movs:
            writer.writerow(['Cluster {}'.format(cluster), mov[0]])
```



```
         Blackfish (2013)
Cluster #16
         Notebook, The (2004)
         Lost Boys, The (1987)
         Doctor Zhivago (1965)
         Repo Man (1984)
         Friends with Benefits (2011)
         Atlantis: The Lost Empire (2001)
         One Hour Photo (2002)
         Rat Race (2001)
         Major Payne (1995)
         Match Point (2005)
         Monster (2003)
         20,000 Leagues Under the Sea (1954)
         Encino Man (1992)
         Bicentennial Man (1999)
         Homeward Bound II: Lost in San Francisco (1996)
         Warm Bodies (2013)
         Absolute Power (1997)
         Canadian Bacon (1995)
         Down with Love (2003)
         Meatballs (1979)
         Crank: High Voltage (2009)
         For Love of the Game (1999)
         Conjuring, The (2013)
         Troop Beverly Hills (1989)
         Club Dread (2004)
         Blue Valentine (2010)
         Apple Dumpling Gang Rides Again, The (1979)
         Hostel: Part II (2007)
         Big Bounce, The (2004)
         What If (2013)
         Miss March (2009)
         Mr. Woodcock (2007)
         Timeline (2003)
```



Average Rating by Category

Genres (grou..	
Film-Noir	4.0413
Mystery	3.8555
Crime	3.8342
Documentary	3.8296
Musical	3.6530
Drama	3.6472
Animation	3.5783
Adventure	3.5660
Fantasy	3.4968
(no genres listed)	3.4894
Sci-Fi	3.4606
Action	3.4490
Comedy	3.3939
Children	3.1536
Horror	[illegible]

Visualisation

Thanks.