Final Project-Part 4

May 3, 2025

0.0.1 Part 2

Data: Prepare a data analysis report that includes:

Data definition for each parameter including measurement units

Plan for missing data for each parameter (if needed)

Plan for additional parameters or data (if needed)

Any transformations necessary

Plan for separating

Visualization of data if possible

[3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[4]:
```
anime_df = pd.read_csv("anime.csv")
```

[5]:
```
rating_df = pd.read_csv("rating.csv")
```

[6]:
```
anime_rt_df = pd.merge(anime_df, rating_df, on = 'anime_id')
```

[7]:
```
anime_rt_df.head()
```

[7]:
```
   anime_id            name                                    genre   type  \
0     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
1     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
2     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
3     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
4     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie

  episodes  rating_x  members  user_id  rating_y
0        1      9.37   200630       99         5
1        1      9.37   200630      152        10
2        1      9.37   200630      244        10
3        1      9.37   200630      271        10
4        1      9.37   200630      278        -1
```

[ ]:

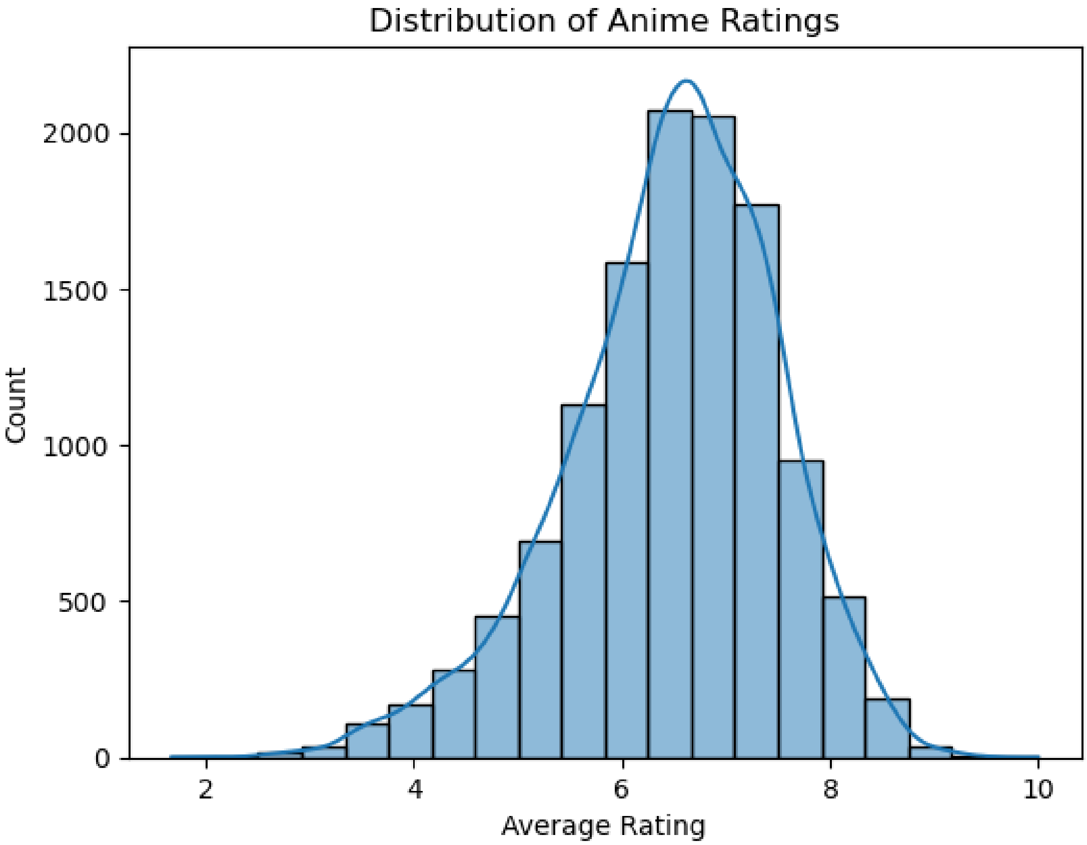
```
 
```

[8]:

```
sns.histplot(anime_df['rating'], bins=20, kde=True)
plt.title("Distribution of Anime Ratings")
plt.xlabel("Average Rating")
plt.ylabel("Count")
```

[8]:

```
Text(0, 0.5, 'Count')
```

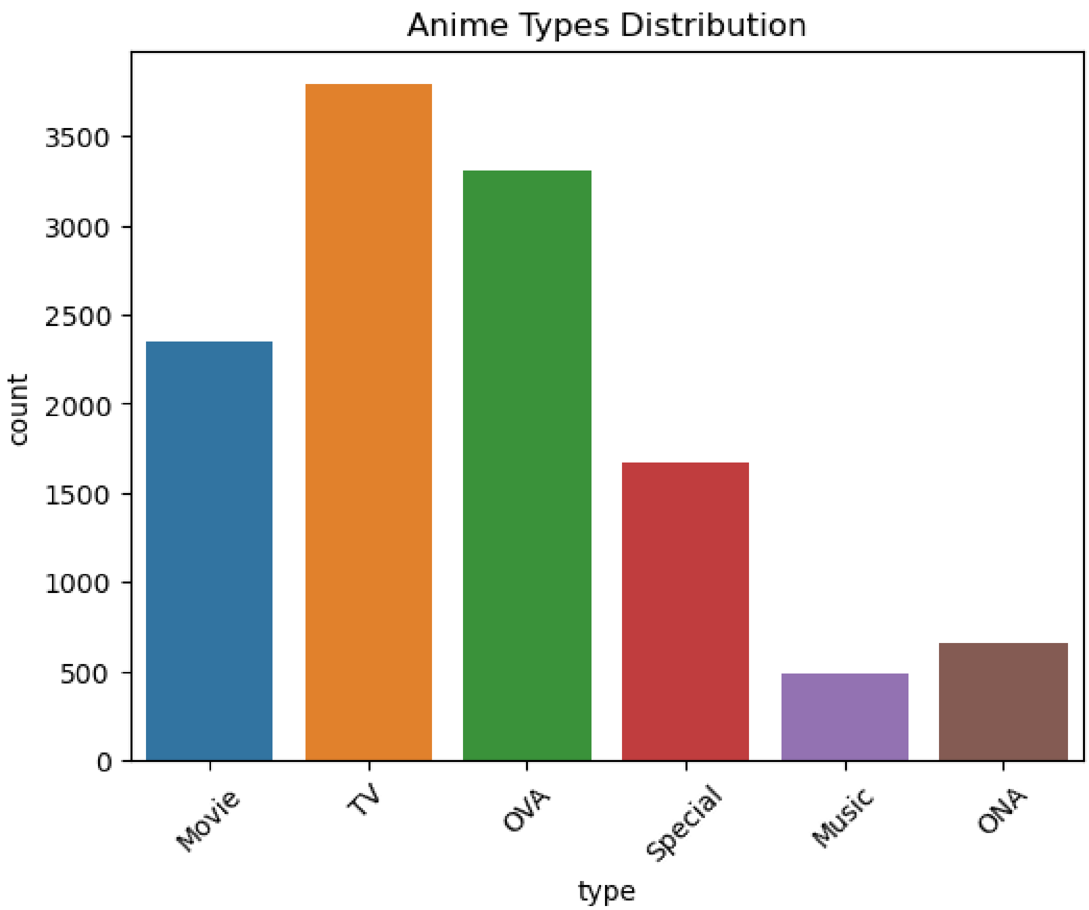


[9]:

```
sns.countplot(data=anime_df, x='type')
plt.title("Anime Types Distribution")
plt.xticks(rotation=45)
```

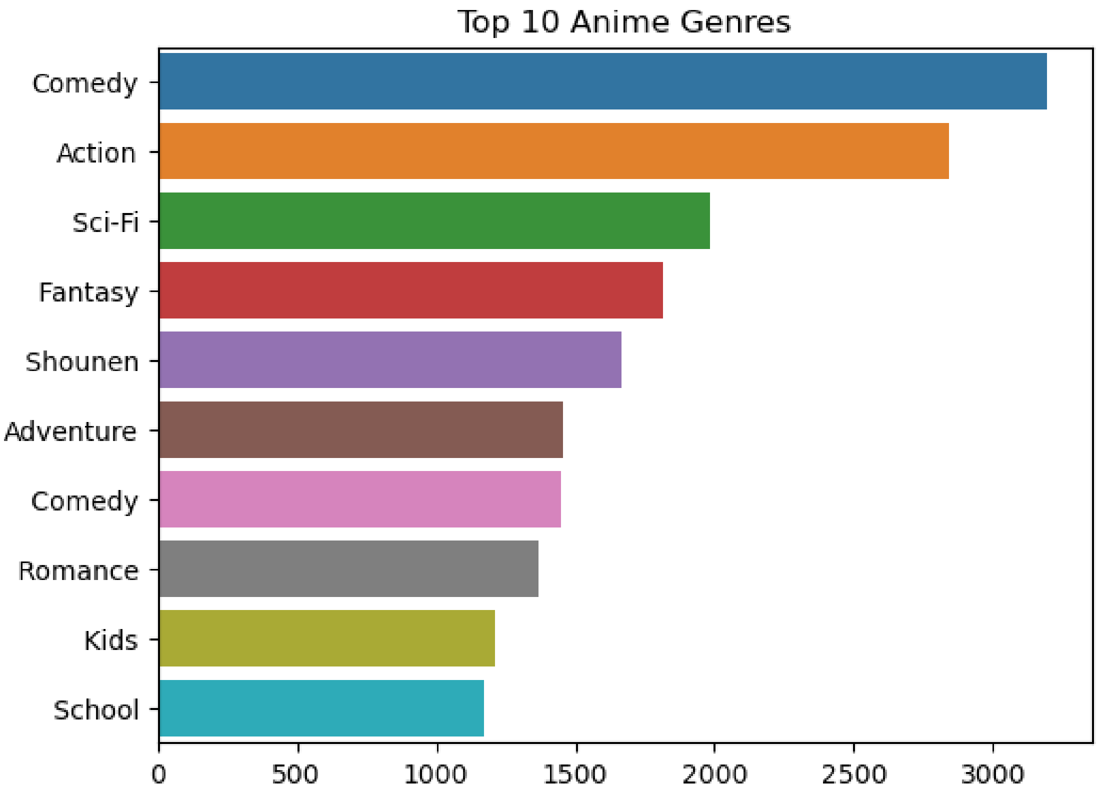
[9]:

```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Movie'),
  Text(1, 0, 'TV'),
  Text(2, 0, 'OVA'),
  Text(3, 0, 'Special'),
  Text(4, 0, 'Music'),
  Text(5, 0, 'ONA')])
```



[10]:
```
from collections import Counter
genres = anime_df['genre'].str.split(',').explode()
top_genres = Counter(genres).most_common(10)
sns.barplot(x=[g[1] for g in top_genres], y=[g[0] for g in top_genres])
plt.title("Top 10 Anime Genres")
```

[10]:
```
Text(0.5, 1.0, 'Top 10 Anime Genres')
```



```
[11]: anime_rt_df.isna().sum()
```

```
[11]: anime_id        0
      name            0
      genre         110
      type            4
      episodes        0
      rating_x        6
      members         0
      user_id         0
      rating_y        0
      dtype: int64
```

```
[12]: anime_rt_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7813727 entries, 0 to 7813726
Data columns (total 9 columns):
 #   Column     Dtype
---  ------     -----
 0   anime_id   int64
 1   name       object
 2   genre     object
 3   type      object
 4   episodes  object
 5   rating_x  float64
 6   members   int64
 7   user_id   int64
 8   rating_y  int64
dtypes: float64(1), int64(4), object(4)
memory usage: 596.1+ MB
```

[13]:

```
anime_rt_df[anime_rt_df['type'] == 'TV']
```

[13]:

```
         anime_id                            name  \
2199         5114  Fullmetal Alchemist: Brotherhood
2200         5114  Fullmetal Alchemist: Brotherhood
2201         5114  Fullmetal Alchemist: Brotherhood
2202         5114  Fullmetal Alchemist: Brotherhood
2203         5114  Fullmetal Alchemist: Brotherhood
…             …                                …
7689811      7808  Zukkoke Knight: Don De La Mancha
7689812      7808  Zukkoke Knight: Don De La Mancha
7689813      7808  Zukkoke Knight: Don De La Mancha
7689814      7808  Zukkoke Knight: Don De La Mancha
7689816     13455                 Zumomo to Nupepe

                                                    genre type episodes  \
2199     Action, Adventure, Drama, Fantasy, Magic, Mili…   TV       64
2200     Action, Adventure, Drama, Fantasy, Magic, Mili…   TV       64
2201     Action, Adventure, Drama, Fantasy, Magic, Mili…   TV       64
2202     Action, Adventure, Drama, Fantasy, Magic, Mili…   TV       64
2203     Action, Adventure, Drama, Fantasy, Magic, Mili…   TV       64
…                                                    …    …      …
7689811            Adventure, Comedy, Historical, Romance   TV       23
7689812            Adventure, Comedy, Historical, Romance   TV       23
7689813            Adventure, Comedy, Historical, Romance   TV       23
7689814            Adventure, Comedy, Historical, Romance   TV       23
7689816                                            Comedy   TV       32

         rating_x  members  user_id  rating_y
2199         9.26   793665        3        10
2200         9.26   793665       10        10
2201         9.26   793665       11         8
2202         9.26   793665       12         9
2203         9.26   793665       17        10
…             …       …       …        …
7689811      6.47      172    25856         4
7689812      6.47      172    48766        -1
```

```
7689813      6.47      172    49127          6
7689814      6.47      172    51693          7
7689816      7.00      120    48766         -1

[5283596 rows x 9 columns]
```

Genre's that are not labeled.

```
[40]: anime_rt_df[anime_rt_df['type'].isna()]
```

```
[40]: Empty DataFrame
      Columns: [anime_id, name, genre, type, episodes, rating_x, members, user_id,
      rating_y]
      Index: []
```

```
[41]: anime_rt_df.loc[(anime_rt_df['name'] == 'Steins;Gate 0') & (anime_rt_df['type'].
       ↪isna()), 'type'] = 'TV'
```

```
[16]: anime_rt_df[anime_rt_df['episodes'] == 'Unknown']
```

```
[16]:          anime_id               name                   genre     type  \
      7686348     32237      Burutabu-chan  Comedy, Fantasy, Magic       TV
      7686525     27551  Deko Boko Friends            Comedy, Kids       TV
      7688629     21981           Pic-lele                    Kids       TV
      7689759     33846           Yurugaro         Comedy, Fantasy  Special
      7689817     30484      Steins;Gate 0        Sci-Fi, Thriller       TV
      7689818     30484      Steins;Gate 0        Sci-Fi, Thriller       TV
      7689819     30484      Steins;Gate 0        Sci-Fi, Thriller       TV
      7689820     30484      Steins;Gate 0        Sci-Fi, Thriller       TV

              episodes  rating_x  members  user_id  rating_y
      7686348  Unknown      5.00       37    12431         1
      7686525  Unknown      7.14       61    48766        -1
      7688629  Unknown      6.33       59    48766        -1
      7689759  Unknown      4.74      153    13954         4
      7689817  Unknown       NaN    60999    31437         9
      7689818  Unknown       NaN    60999    37475         7
      7689819  Unknown       NaN    60999    42372         9
      7689820  Unknown       NaN    60999    45904         9
```

```
[17]: anime_rt_df[anime_rt_df['name'] == 'Steins;Gate 0']
```

```
[17]:          anime_id           name             genre type episodes  rating_x  \
      7689817     30484  Steins;Gate 0  Sci-Fi, Thriller   TV  Unknown       NaN
      7689818     30484  Steins;Gate 0  Sci-Fi, Thriller   TV  Unknown       NaN
      7689819     30484  Steins;Gate 0  Sci-Fi, Thriller   TV  Unknown       NaN
      7689820     30484  Steins;Gate 0  Sci-Fi, Thriller   TV  Unknown       NaN
```

```
         members  user_id  rating_y
7689817    60999    31437         9
7689818    60999    37475         7
7689819    60999    42372         9
7689820    60999    45904         9
```

[18]:
```
anime_rt_df.loc[(anime_rt_df['name'] == 'Steins;Gate 0') &␣
 ↪(anime_rt_df['episodes'] == 'Unknown'), 'episodes'] = 23
```

[19]:
```
anime_rt_df[anime_rt_df['name'] == 'Steins;Gate 0']
```

[19]:
```
         anime_id           name             genre type episodes  rating_x  \
7689817     30484  Steins;Gate 0  Sci-Fi, Thriller   TV       23       NaN
7689818     30484  Steins;Gate 0  Sci-Fi, Thriller   TV       23       NaN
7689819     30484  Steins;Gate 0  Sci-Fi, Thriller   TV       23       NaN
7689820     30484  Steins;Gate 0  Sci-Fi, Thriller   TV       23       NaN

         members  user_id  rating_y
7689817    60999    31437         9
7689818    60999    37475         7
7689819    60999    42372         9
7689820    60999    45904         9
```

[20]:
```
anime_rt_df = anime_rt_df.dropna(subset=['genre'])
```

[21]:
```
anime_rt_df.isna().sum()
```

[21]:
```
anime_id    0
name        0
genre       0
type        0
episodes    0
rating_x    6
members     0
user_id     0
rating_y    0
dtype: int64
```

[22]:
```
anime_rt_df[anime_rt_df['rating_x'].isna()]
```

[22]:
```
         anime_id                  name  \
7689817     30484         Steins;Gate 0
7689818     30484         Steins;Gate 0
7689819     30484         Steins;Gate 0
7689820     30484         Steins;Gate 0
7689821     33674  No Game No Life Movie
7689822      9488           Cencoroll 2
```

```
                                                  genre   type episodes  \
7689817                                Sci-Fi, Thriller     TV       23
7689818                                Sci-Fi, Thriller     TV       23
7689819                                Sci-Fi, Thriller     TV       23
7689820                                Sci-Fi, Thriller     TV       23
7689821  Adventure, Comedy, Ecchi, Fantasy, Game, Super…  Movie        1
7689822                                  Action, Sci-Fi  Movie        1

         rating_x  members  user_id  rating_y
7689817       NaN    60999    31437         9
7689818       NaN    60999    37475         7
7689819       NaN    60999    42372         9
7689820       NaN    60999    45904         9
7689821       NaN    32041    43480        -1
7689822       NaN    15181    43873         7
```

```
[23]: anime_rt_df.loc[(anime_rt_df['name'] == 'Steins;Gate 0') &␣
        ↪(anime_rt_df['rating_x'].isna()), 'rating_x'] = 8.54
```

```
[24]: anime_rt_df.loc[(anime_rt_df['name'] == 'No Game No Life Movie') &␣
        ↪(anime_rt_df['rating_x'].isna()), 'rating_x'] = 6.79
```

```
[25]: anime_rt_df.loc[(anime_rt_df['name'] == 'Cencoroll 2') &␣
        ↪(anime_rt_df['rating_x'].isna()), 'rating_x'] = 7.19
```

```
[26]: anime_rt_df.head()
```

```
[26]:    anime_id            name                                 genre   type  \
      0     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
      1     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
      2     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
      3     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
      4     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie

        episodes  rating_x  members  user_id  rating_y
      0        1      9.37   200630       99         5
      1        1      9.37   200630      152        10
      2        1      9.37   200630      244        10
      3        1      9.37   200630      271        10
      4        1      9.37   200630      278        -1
```

```
[27]: anime_rt_df = anime_rt_df[anime_rt_df['episodes'] != 'Unknown']
```

```
[28]: anime_rt_df['episodes'].value_counts()
```

```
[28]: 1       1771175
      12      1734393
```

```
13       953125
26       542351
24       536798
         …
475           1
83            1
71            1
132           1
283           1
Name: episodes, Length: 184, dtype: int64
```

[29]:
```
cleaned_df = anime_rt_df[anime_rt_df['rating_y'] != -1]
```

[30]:
```
# Filter for TV anime only
tv_anime = cleaned_df[cleaned_df['type'] == 'TV']

# Further filter for anime with members > 100,000
popular_tv_anime = tv_anime[tv_anime['members'] > 100000]
```

[31]:
```
print(f"After popularity filter: {popular_tv_anime.shape}")
```

```
After popularity filter: (3000924, 9)
```

[32]:
```
sample_1000 = popular_tv_anime.sample(n = 1000, random_state = 20)
```

0.0.2 Part 3: Models

Supervised Learning: Rating Prediction(Regression)

[42]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

[43]:
```
# Feature engineering
# Feature Engineering
# Convert genre strings into dummy variables (one-hot encoding)
genre_dummies = sample_1000['genre'].str.get_dummies(sep=',')

# Convert type to dummy variables (though we only have TV now)
type_dummy = pd.get_dummies(sample_1000['type'], prefix='type')

# Select numerical features
numerical_features = sample_1000[['episodes', 'members', 'rating_x']]

# Combine all features
X = pd.concat([genre_dummies, type_dummy, numerical_features], axis=1)
y = sample_1000['rating_y']  # Target variable (user ratings)
```

```
[44]: # Train-test split (80-20 split)
      from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
       ↪random_state=42)

      # Initialize and train Random Forest Regressor
      from sklearn.ensemble import RandomForestRegressor
      rf = RandomForestRegressor(n_estimators=100,
                                 max_depth=10,
                                 random_state=42,
                                 min_samples_split=5)

      rf.fit(X_train, y_train)
```

```
[44]: RandomForestRegressor(max_depth=10, min_samples_split=5, random_state=42)
```

```
[45]: y_pred = rf.predict(X_test)
      mse = mean_squared_error(y_test, y_pred)

      # Feature importance
      importances = pd.DataFrame({'feature':X.columns,'importance':rf.
       ↪feature_importances_})\
                    .sort_values('importance',ascending=False)
```

```
[46]: print(f"Mean Squared Error: {mse:.2f}")
      print("\nTop 10 Feature Importances:")
      print(importances.head(10).to_string(index=False))
```

```
Mean Squared Error: 1.96

Top 10 Feature Importances:
 feature  importance
rating_x    0.469334
 members    0.178327
episodes    0.062803
  Comedy    0.015848
 Mystery    0.015568
   Drama    0.011985
 Fantasy    0.011830
 Romance    0.010913
 Romance    0.010877
   Drama    0.010794
```

Kmeans

```
[47]: from sklearn.cluster import KMeans
      from sklearn.preprocessing import StandardScaler
```

```
# Prepare the data
# Convert genre to dummy variables and select features
genre_dummies = sample_1000['genre'].str.get_dummies(',')
X = pd.concat([sample_1000[['members', 'episodes', 'rating_x']],␣
 ↪genre_dummies], axis=1)

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[48]:

```
# Initialize variables for elbow method
wcss = []  # Within-cluster sum of squares
k_range = range(1, 200)  # Testing k from 1 to 10

# Compute WCSS for each k
for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(50, 20))
plt.plot(k_range, wcss, marker='o', linestyle='--')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.title('Elbow Method for Optimal k')
plt.xticks(k_range)
plt.grid()
plt.show()
```

```
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
```

```
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
```

```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
```

```
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
```

```
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
```

```
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
```

```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
```

```
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
```

```
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
```

```
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
```

```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
```

```
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
```

```
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
```

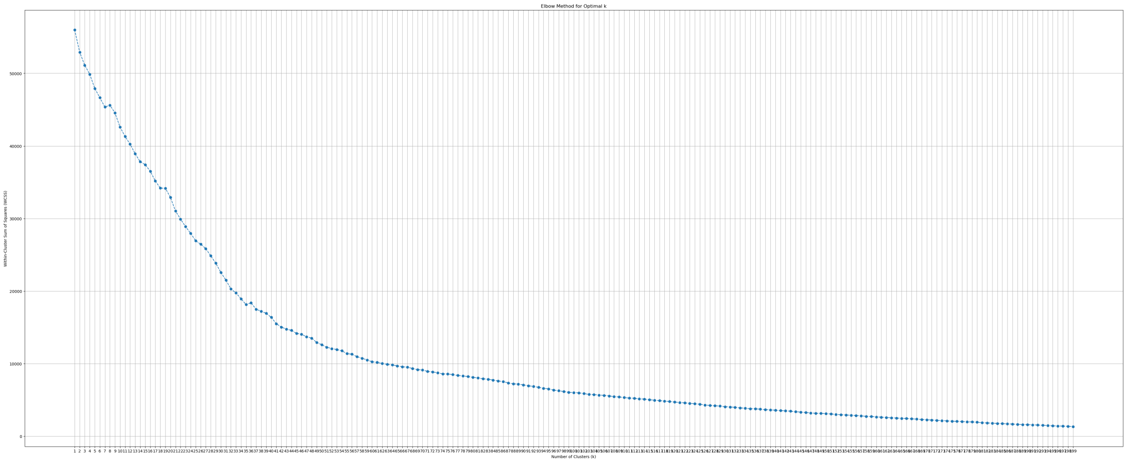
```
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
```

```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
```

```
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
```

```
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
```

```
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
```



[49]:

```python
optimal_k = 62  # Change this based on your elbow plot observation
print(f"Optimal number of clusters: {optimal_k}")

# Fit K-Means with optimal k
kmeans = KMeans(n_clusters=optimal_k, random_state=20)
clusters = kmeans.fit_predict(X_scaled)

# Add cluster labels to your DataFrame
sample_1000['cluster'] = clusters

# Analyze the clusters
cluster_profile = sample_1000.groupby('cluster').agg({
    'members': 'mean',
    'episodes': 'mean',
    'rating_x': 'mean'
}).round(1)

# Print cluster profiles
print("\nCluster Profiles:")
print(cluster_profile)

# Print top genres in each cluster
print("\nTop Genres by Cluster:")
for cluster in range(optimal_k):
    print(f"\nCluster {cluster}:")
    top_genres = sample_1000[sample_1000['cluster'] == cluster]['genre']\
        .str.split(',').explode().value_counts().head(5)
    print(top_genres)
```

```
Optimal number of clusters: 62

Cluster Profiles:
          members       episodes  rating_x
cluster
0        196336.8  1.985944e+116       7.9
1        722456.2   2.711393e+36       8.9
2        298416.2   5.912519e+79       8.4
3        395821.7   1.126259e+22       7.9
4        562962.0   3.030303e+16       8.8
…           …            …       …
57       212122.8   4.040408e+57       8.0
58       184846.8   7.789008e+31       7.7
59       410218.6   1.101928e+20       7.6
60       223662.6   2.805612e+34       7.6
61       257296.8   8.207008e+29       7.5

[62 rows x 3 columns]

Top Genres by Cluster:

Cluster 0:
Comedy            61
 School           42
 Romance          30
 Slice of Life    28
 Supernatural     13
Name: genre, dtype: int64

Cluster 1:
Action        19
 Adventure    19
 Drama        19
 Fantasy      19
 Magic        19
Name: genre, dtype: int64

Cluster 2:
Drama             41
 Slice of Life    28
 Supernatural     28
 Romance          24
 Fantasy          14
Name: genre, dtype: int64

Cluster 3:
Action        12
 Adventure    12
 Comedy       12
 Fantasy      10
 Shounen       8
Name: genre, dtype: int64

Cluster 4:
Action        9
 Adventure    9
 Comedy       9
 Mecha        9
 Sci-Fi       9
Name: genre, dtype: int64

Cluster 5:
Sci-Fi       10
 Thriller    10
Name: genre, dtype: int64

Cluster 6:
Action            29
 Horror           29
 Psychological    26
 Drama            26
 Supernatural     26
Name: genre, dtype: int64

Cluster 7:
Harem       1
 School     1
 Shoujo     1
 Vampire    1
Name: genre, dtype: int64
```

```
Cluster 8:
 Comedy          15
 Shounen         15
 Super Power     15
Action           13
 Martial Arts    12
Name: genre, dtype: int64

Cluster 9:
Action         12
 Samurai       12
 Adventure     11
 Comedy        10
 Historical    10
Name: genre, dtype: int64

Cluster 10:
Action            4
 Dementia         4
 Drama            4
 Mecha            4
 Psychological    4
Name: genre, dtype: int64

Cluster 11:
 Super Power     40
Action           36
 Sci-Fi          31
 Magic           14
 Supernatural     8
Name: genre, dtype: int64

Cluster 12:
 School       3
 Shoujo Ai    3
Comedy        2
 Parody       2
Action        1
Name: genre, dtype: int64

Cluster 13:
Ecchi       4
 School     4
 Shounen    4
Name: genre, dtype: int64

Cluster 14:
```

```
Mystery           9
 Psychological    5
 School           4
 Slice of Life    4
 Supernatural     3
Name: genre, dtype: int64

Cluster 15:
Action        7
 Drama        7
 Sci-Fi       7
 Space        7
 Adventure    6
Name: genre, dtype: int64

Cluster 16:
Action           30
 Comedy          30
 School          24
 Supernatural    12
 Shounen          9
Name: genre, dtype: int64

Cluster 17:
Psychological    3
 Thriller        3
Name: genre, dtype: int64

Cluster 18:
Comedy         4
 Drama         4
 Romance       4
 Shounen Ai    4
Name: genre, dtype: int64

Cluster 19:
Action        7
 Adventure    7
 Comedy       7
 Fantasy      7
 Kids         7
Name: genre, dtype: int64

Cluster 20:
 Josei            3
 Slice of Life    3
Comedy            2
 Drama            2
 Music            2
Name: genre, dtype: int64

Cluster 21:
Horror           18
 Mystery         18
 Supernatural    17
 Thriller        11
 School          10
Name: genre, dtype: int64

Cluster 22:
Adventure    12
 Game        12
 Fantasy     11
 Comedy      10
 Ecchi       10
Name: genre, dtype: int64

Cluster 23:
Action           40
 Adventure       35
 Fantasy         30
 Shounen         19
 Supernatural    16
Name: genre, dtype: int64

Cluster 24:
 Demons          23
Action           20
 Fantasy         20
 Shounen         17
 Supernatural    16
Name: genre, dtype: int64

Cluster 25:
Action            9
 Mystery          9
 Psychological    9
 Shounen          9
 Supernatural     9
Name: genre, dtype: int64

Cluster 26:
Military    2
 School     2
Name: genre, dtype: int64
```

```
Cluster 27:
Dementia          2
 Drama            2
 Mystery          2
 Psychological    2
 Sci-Fi           2
Name: genre, dtype: int64

Cluster 28:
Romance           4
 School           4
 Slice of Life    4
 Shoujo           2
Name: genre, dtype: int64

Cluster 29:
Comedy            36
 Seinen           36
 Slice of Life    24
 Romance          14
 Sci-Fi            7
Name: genre, dtype: int64

Cluster 30:
 Martial Arts    20
Action           17
 Comedy          13
 Shounen         11
 School           9
Name: genre, dtype: int64

Cluster 31:
Fantasy           6
 Slice of Life    5
 Supernatural     5
 Mystery          1
Name: genre, dtype: int64

Cluster 32:
 Music            11
Comedy             8
 School            8
 Slice of Life     8
 Romance           6
Name: genre, dtype: int64

Cluster 33:
 Sports           9
```

```
Comedy            7
 School           7
 Shounen          5
 Slice of Life    4
Name: genre, dtype: int64

Cluster 34:
 Parody     8
 Sci-Fi     8
Comedy      6
 Mystery    6
 School     6
Name: genre, dtype: int64

Cluster 35:
Josei             1
 Slice of Life    1
Name: genre, dtype: int64

Cluster 36:
Action           11
 Ecchi           11
 Harem           11
 Supernatural    11
 Comedy           9
Name: genre, dtype: int64

Cluster 37:
Mystery           6
 Police           6
 Psychological    6
 Supernatural     6
 Thriller         6
Name: genre, dtype: int64

Cluster 38:
 Ecchi      43
 Romance    43
Comedy      42
 Harem      40
 School     38
Name: genre, dtype: int64

Cluster 39:
 Historical    9
 Seinen        5
 Mystery       5
 Romance       4
```

```
Drama          4
Name: genre, dtype: int64

Cluster 40:
Action        14
 Game         14
 Adventure    12
 Fantasy      12
 Romance      12
Name: genre, dtype: int64

Cluster 41:
 Vampire         20
Action           18
 Supernatural    18
 School           9
 Shounen          8
Name: genre, dtype: int64

Cluster 42:
 Romance          36
 Shoujo           36
Comedy            34
 School           24
 Slice of Life    13
Name: genre, dtype: int64

Cluster 43:
Action          14
 Mecha          14
 Military       14
 Sci-Fi         14
 Super Power    13
Name: genre, dtype: int64

Cluster 44:
Adventure    6
 Comedy      6
 Drama       6
 Fantasy     6
 Magic       6
Name: genre, dtype: int64

Cluster 45:
Mystery          7
 Supernatural    7
 Vampire         7
 Romance         6
 Thriller        1
Name: genre, dtype: int64

Cluster 46:
 Mystery         27
 Supernatural    19
Action           17
 Sci-Fi           8
 Thriller         7
Name: genre, dtype: int64

Cluster 47:
 Police           9
 Psychological    7
Action            6
 Sci-Fi           6
 Mystery          5
Name: genre, dtype: int64

Cluster 48:
Action          13
 Drama          13
 Fantasy        13
 Shounen        13
 Super Power    13
Name: genre, dtype: int64

Cluster 49:
Adventure      10
 Fantasy       10
 Historical    10
 Romance        8
 Mystery        2
Name: genre, dtype: int64

Cluster 50:
 Demons     14
 Ecchi      14
 Harem      14
 Romance    14
Action      12
Name: genre, dtype: int64

Cluster 51:
 Fantasy         32
Action           30
 Magic           25
 Supernatural    21
 Romance         19
Name: genre, dtype: int64

Cluster 52:
Action        22
 Shounen      22
 Comedy       19
 Adventure    18
 Fantasy      12
Name: genre, dtype: int64

Cluster 53:
Comedy            42
 School           41
 Drama            38
 Slice of Life    38
 Romance          37
Name: genre, dtype: int64

Cluster 54:
 Seinen     27
Action      26
 Sci-Fi      9
 Mystery     6
 Drama       4
Name: genre, dtype: int64

Cluster 55:
Drama             10
 Psychological    10
 Thriller         10
 Magic             7
 Game              2
Name: genre, dtype: int64

Cluster 56:
Drama            7
 Mystery         7
 Romance         7
 Shoujo          7
 Supernatural    7
Name: genre, dtype: int64

Cluster 57:
Comedy           30
 Romance         30
 Shounen         25
 Harem           21
 Supernatural    12
Name: genre, dtype: int64

Cluster 58:
 Parody          17
Comedy           13
 School           8
 Romance          7
 Supernatural     4
Name: genre, dtype: int64

Cluster 59:
 Horror          11
Action            8
 Sci-Fi           6
 Supernatural     6
 Ecchi            4
Name: genre, dtype: int64

Cluster 60:
 Mecha      18
 Sci-Fi     18
Action      12
 Romance     6
 Harem       6
Name: genre, dtype: int64

Cluster 61:
Action      16
 Comedy     16
 Ecchi      16
 Fantasy    16
 Harem      16
Name: genre, dtype: int64
```

```
D:\Python Projects\lib\site-packages\sklearn\cluster\_kmeans.py:1419:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=4.
  warnings.warn(
```

Collaborative Filtering

[50]:

```python
from surprise import Dataset, Reader, KNNBasic
from surprise.model_selection import cross_validate

# Prepare data for Surprise
reader = Reader(rating_scale=(1, 10))
```

```
data = Dataset.load_from_df(sample_1000[['user_id', 'anime_id', 'rating_y']],␣
 ↪reader)

# User-based collaborative filtering
sim_options = {'name': 'cosine', 'user_based': True}
algo = KNNBasic(k=20, sim_options=sim_options)

# Cross-validation - modified to handle results properly
cv_results = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3,␣
 ↪verbose=True)

# Train on full dataset
trainset = data.build_full_trainset()
algo.fit(trainset)
```

```
Computing the cosine similarity matrix…
Done computing similarity matrix.
Computing the cosine similarity matrix…
Done computing similarity matrix.
Computing the cosine similarity matrix…
Done computing similarity matrix.
Evaluating RMSE, MAE of algorithm KNNBasic on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.4964  1.3069  1.4897  1.4310  0.0878
MAE (testset)     1.1367  1.0122  1.1430  1.0973  0.0602
Fit time          0.01    0.01    0.00    0.01    0.00
Test time         0.00    0.00    0.00    0.00    0.00
Computing the cosine similarity matrix…
Done computing similarity matrix.
```

[50]:

```
<surprise.prediction_algorithms.knns.KNNBasic at 0x278806294c0>
```

[51]:

```
print("""
COLLABORATIVE FILTERING RESULTS
------------------------------""")
print(f"Average RMSE across folds: {cv_results['test_rmse'].mean():.2f}")
print(f"Average MAE across folds: {np.mean(cv_results['test_mae']):.2f}")
print(f"Fit time: {np.mean(cv_results['fit_time']):.2f}s")
print(f"Test time: {np.mean(cv_results['test_time']):.2f}s")

# Example predictions
test_user = sample_1000['user_id'].iloc[0]
test_anime = sample_1000['anime_id'].iloc[0]
pred = algo.predict(test_user, test_anime).est
actual = sample_1000[(sample_1000['user_id'] == test_user) &
```

```
                    (sample_1000['anime_id'] == test_anime)]['rating_y'].
 ↪values[0]
print(f"\nExample Prediction:")
print(f"User {test_user} on Anime {test_anime}:")
print(f"Actual rating: {actual} | Predicted: {pred:.1f}")
```

```
COLLABORATIVE FILTERING RESULTS
------------------------------
Average RMSE across folds: 1.43
Average MAE across folds: 1.10
Fit time: 0.01s
Test time: 0.00s

Example Prediction:
User 51514 on Anime 12413:
Actual rating: 10 | Predicted: 9.0
```

0.0.3 Part 4

[59]:

```
# Consolidate similar genres
def consolidate_genres(genre_str):
    if pd.isna(genre_str):
        return "Unknown"
    genres = genre_str.split(',')
    consolidated = []
    for g in genres:
        g = g.strip()
        if g in ['Comedy', 'Parody']:
            consolidated.append('Comedy')
        elif g in ['Drama', 'Tragedy']:
            consolidated.append('Drama')
        elif g in ['Shounen', 'Shoujo']:
            consolidated.append('Demographic')
        else:
            consolidated.append(g)
    return ','.join(list(set(consolidated)))
```

[60]:

```
from sklearn.metrics import make_scorer, mean_squared_error

sample_1000['genre_consolidated'] = sample_1000['genre'].
 ↪apply(consolidate_genres)
genre_dummies = sample_1000['genre_consolidated'].str.get_dummies(',')
type_dummies = pd.get_dummies(sample_1000['type'], prefix='type')
numerical_features = sample_1000[['episodes', 'members', 'rating_x']]
X = pd.concat([genre_dummies, type_dummies, numerical_features], axis=1)
y = sample_1000['rating_y']
```

[63]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Initialize and train Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators=200,
                          max_depth=20,
                          random_state=42,
                          min_samples_split=5)

rf.fit(X_train, y_train)
```

[63]:
```
RandomForestRegressor(max_depth=20, min_samples_split=5, n_estimators=200,
                      random_state=42)
```

[64]:
```
y_pred = rf.predict(X_test)
mse = mean_squared_error(y_test, y_pred)

# Feature importance
importances = pd.DataFrame({'feature':X.columns,'importance':rf.
 ↪feature_importances_})\
               .sort_values('importance',ascending=False)

print(f"Mean Squared Error: {mse:.2f}")
print("\nTop 10 Feature Importances:")
print(importances.head(10).to_string(index=False))
```

```
Mean Squared Error: 2.11

Top 10 Feature Importances:
      feature  importance
     rating_x    0.439310
      members    0.185923
     episodes    0.068948
      cluster    0.043809
      Mystery    0.017473
      Romance    0.015462
        Drama    0.014804
Slice of Life    0.014747
 Supernatural    0.013525
       School    0.013358
```

[62]:
```
from surprise import KNNWithMeans, SVD
from surprise.model_selection import GridSearchCV

# Define parameter grid
param_grid = {
    'k': [20, 30, 50],
    'sim_options': {
        'name': ['msd', 'cosine', 'pearson'],
        'user_based': [True, False]
    },
    'min_k': [1, 5]
}

# Grid search
gs = GridSearchCV(KNNWithMeans, param_grid, measures=['rmse'], cv=3, n_jobs=-1)
gs.fit(data)

# Best model
best_knn = gs.best_estimator['rmse']
cross_validate(best_knn, data, measures=['rmse'], cv=3, verbose=True)

# Try matrix factorization as comparison
svd = SVD()
cross_validate(svd, data, measures=['rmse'], cv=3, verbose=True)
```

```
Computing the msd similarity matrix…
Done computing similarity matrix.
Computing the msd similarity matrix…
Done computing similarity matrix.
Computing the msd similarity matrix…
Done computing similarity matrix.
Evaluating RMSE of algorithm KNNWithMeans on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.4173  1.4386  1.4271  1.4277  0.0087
Fit time          0.00    0.00    0.00    0.00    0.00
Test time         0.00    0.00    0.00    0.00    0.00
Evaluating RMSE of algorithm SVD on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    1.4639  1.3429  1.3665  1.3911  0.0524
Fit time          0.01    0.01    0.01    0.01    0.00
Test time         0.00    0.00    0.00    0.00    0.00
```

```
[62]: {'test_rmse': array([1.46393543, 1.34286789, 1.36649242]),
       'fit_time': (0.012560129165649414,
        0.009973526000976562,
        0.010988473892211914),
       'test_time': (0.001993417739868164,
        0.0019943714141845703,
        0.0009970664978027344)}
```

```
[ ]:
```