# SeekRoommateReccomender

April 10, 2022

[1]:
```
import pandas as pd
from sklearn.cluster import AgglomerativeClustering
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import datasets
from kmodes.kprototypes import KPrototypes
from kmodes.kmodes import KModes
from sklearn import preprocessing
import numpy as np
from sklearn.cluster import KMeans
```

[2]:
```
df = pd.read_csv('userdata.csv')
interests = ['Sports', 'Gaming', 'Coding', 'Outdoors', 'Movies',  'Art',␣
 ↪'Anime', 'Foodie',  'Music', 'Travel']
```

[3]:
```
for i in interests:
    df[i] = int(0)
```

[4]:
```
i1 = []
i2 = []
i3 = []
for i, row in df.iterrows():
    temp = (row["Describe Yourself"]).split(",")
    i1.append(temp[0].strip())
    i2.append(temp[1].strip())
    i3.append(temp[2].strip())
```

[5]:
```
df['Interest 1'] = i1
df['Interest 2'] = i2
df['Interest 3'] = i3
df = df.drop(columns=['Describe Yourself'])
```

[6]:
```
num = ['Interest 1', 'Interest 2', 'Interest 3']
for i, row in df.iterrows():
    for j in num:
        df.loc[i, row[j]] = 1
```

[28]:
```
temp = pd.DataFrame()
for i in interests:
    temp[i] = df[i]
```

[28]:
```
     Sports  Gaming  Coding  Outdoors  Movies  Art  Anime  Foodie  Music  \
0         0       0       1         0       1    0      0       0      0
1         0       0       0         0       1    0      1       0      1
2         0       0       0         1       0    0      0       1      1
3         1       0       1         0       0    0      0       0      1
4         1       0       1         0       0    0      0       1      0
..      ...     ...     ...       ...     ...  ...    ...     ...    ...
995       1       0       0         0       0    0      1       1      0
996       0       1       1         1       0    0      0       0      0
997       0       1       0         0       1    0      0       1      0
998       0       0       1         0       0    1      1       0      0
999       1       0       0         0       0    0      0       1      0

     Travel
0         1
1         0
2         0
3         0
4         0
..      ...
995       0
996       0
997       0
998       0
999       1

[1000 rows x 10 columns]
```

[29]:
```
kmeans = KMeans(n_clusters=5, random_state=0).fit(temp)
kmeans.predict(temp)
```

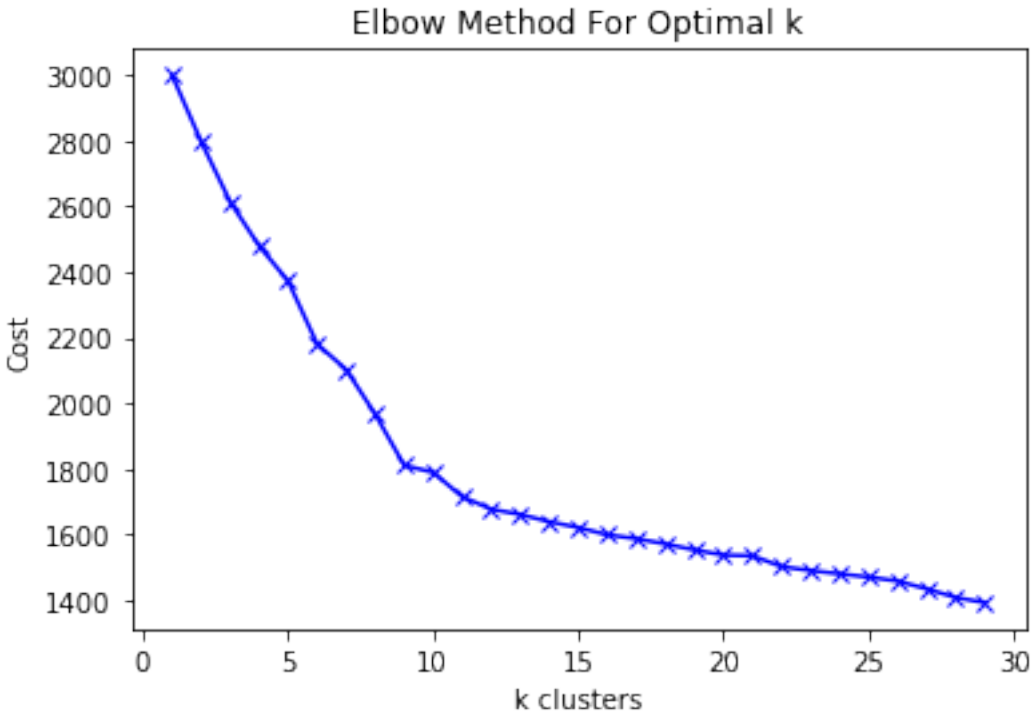
[29]:
```
array([3, 0, 0, 0, 2, 0, 4, 1, 0, 2, 0, 3, 3, 0, 0, 3, 4, 0, 2, 2, 0, 4,
       4, 2, 4, 4, 4, 2, 4, 2, 0, 4, 0, 4, 4, 3, 2, 2, 4, 4, 2, 0, 4, 1,
       2, 1, 0, 2, 0, 4, 3, 1, 0, 0, 4, 2, 1, 2, 0, 2, 0, 0, 2, 2, 3, 4,
       2, 4, 0, 2, 3, 0, 0, 4, 1, 4, 0, 3, 0, 1, 2, 3, 1, 4, 2, 2, 4, 1,
       4, 3, 2, 1, 1, 1, 0, 1, 4, 1, 4, 4, 4, 1, 1, 1, 3, 2, 1, 0, 0, 4,
       1, 2, 4, 0, 2, 1, 0, 2, 0, 0, 4, 1, 2, 1, 1, 2, 3, 4, 0, 4, 1, 3,
       2, 1, 2, 3, 2, 2, 1, 0, 1, 0, 4, 4, 0, 1, 4, 4, 2, 4, 2, 3, 0, 0,
       0, 0, 0, 0, 2, 0, 2, 2, 1, 4, 1, 4, 2, 2, 1, 0, 3, 0, 1, 0, 0, 0,
       1, 1, 1, 3, 0, 1, 1, 2, 4, 2, 0, 4, 2, 0, 0, 4, 3, 1, 0, 2, 0, 3,
       1, 4, 0, 4, 4, 0, 3, 1, 2, 0, 2, 1, 0, 1, 4, 4, 4, 0, 4, 2, 3, 1,
       1, 1, 4, 1, 2, 1, 2, 4, 2, 4, 0, 0, 1, 4, 1, 3, 3, 2, 0, 0, 2, 1,
       1, 0, 1, 3, 3, 0, 4, 0, 0, 1, 3, 1, 4, 4, 1, 0, 4, 2, 1, 4, 2, 1,
```

```
1, 4, 4, 1, 0, 4, 0, 2, 2, 0, 0, 1, 0, 3, 0, 4, 2, 4, 1, 3, 2, 2,
0, 0, 0, 3, 1, 0, 3, 0, 1, 2, 0, 3, 2, 2, 1, 1, 0, 4, 0, 4, 2, 0,
4, 2, 0, 3, 4, 2, 4, 0, 2, 3, 3, 4, 1, 0, 2, 4, 4, 2, 0, 0, 3, 2,
2, 4, 3, 1, 2, 0, 2, 2, 2, 4, 3, 0, 3, 4, 3, 2, 1, 0, 0, 0, 0, 1,
2, 0, 2, 0, 0, 2, 0, 2, 0, 2, 1, 4, 0, 2, 0, 4, 0, 4, 1, 2, 1, 0,
3, 0, 3, 0, 0, 1, 2, 3, 0, 0, 1, 4, 2, 0, 1, 0, 2, 0, 0, 1, 2, 0,
2, 2, 0, 2, 1, 3, 1, 4, 0, 0, 0, 4, 3, 1, 2, 4, 1, 1, 1, 1, 0, 2,
3, 0, 2, 4, 4, 4, 2, 3, 4, 3, 3, 0, 0, 1, 1, 1, 0, 3, 0, 1, 2, 4,
1, 3, 2, 1, 3, 0, 0, 2, 2, 0, 2, 4, 0, 0, 1, 0, 4, 0, 4, 0, 4, 0,
4, 2, 4, 1, 2, 4, 1, 1, 2, 2, 4, 0, 3, 3, 0, 0, 3, 1, 4, 0, 3, 4,
4, 0, 4, 2, 0, 1, 3, 3, 4, 1, 1, 1, 0, 0, 4, 3, 4, 2, 2, 2, 3, 0,
0, 0, 2, 0, 2, 2, 1, 3, 1, 3, 0, 2, 0, 0, 0, 1, 2, 0, 2, 2, 0, 4,
2, 0, 4, 0, 2, 2, 0, 2, 0, 0, 4, 1, 0, 0, 0, 0, 3, 0, 0, 1, 2, 0,
3, 2, 0, 4, 4, 3, 0, 0, 3, 3, 4, 3, 1, 4, 1, 4, 2, 3, 0, 2, 4, 4,
4, 0, 4, 4, 3, 0, 2, 4, 1, 0, 4, 1, 2, 2, 0, 0, 0, 0, 3, 3, 1, 4,
0, 1, 0, 0, 4, 2, 2, 4, 0, 3, 3, 0, 0, 2, 2, 1, 2, 2, 0, 2, 4, 0,
2, 4, 1, 0, 3, 0, 1, 1, 4, 2, 1, 4, 3, 4, 0, 2, 2, 3, 4, 4, 1, 0,
4, 0, 4, 1, 0, 3, 4, 3, 2, 1, 1, 0, 3, 2, 1, 0, 4, 2, 0, 2, 0, 0,
4, 0, 3, 4, 4, 0, 3, 0, 1, 1, 1, 3, 0, 3, 1, 2, 3, 1, 2, 4, 3, 3,
3, 2, 1, 0, 0, 1, 1, 4, 4, 4, 0, 3, 0, 2, 2, 3, 4, 2, 3, 2, 2, 0,
0, 2, 4, 2, 1, 3, 0, 0, 3, 3, 4, 3, 2, 2, 0, 3, 2, 0, 1, 1, 0, 4,
3, 4, 3, 3, 4, 3, 3, 0, 2, 2, 0, 0, 2, 1, 4, 2, 2, 3, 1, 4, 2, 0,
2, 2, 2, 2, 2, 0, 1, 1, 0, 0, 1, 0, 2, 2, 3, 0, 2, 3, 2, 0, 2, 2,
3, 3, 0, 1, 2, 1, 0, 0, 4, 3, 4, 4, 4, 2, 0, 0, 1, 3, 1, 4, 4, 3,
2, 0, 2, 0, 1, 0, 2, 3, 3, 2, 4, 4, 0, 1, 4, 4, 4, 0, 2, 3, 4, 0,
4, 0, 1, 3, 1, 2, 0, 3, 1, 3, 2, 1, 2, 2, 4, 0, 0, 2, 3, 4, 2, 1,
0, 2, 0, 2, 3, 0, 0, 4, 3, 0, 4, 4, 2, 4, 0, 0, 0, 0, 3, 1, 0, 0,
2, 0, 3, 2, 0, 3, 1, 1, 2, 4, 2, 2, 4, 1, 0, 2, 4, 4, 1, 3, 4, 2,
4, 4, 0, 2, 2, 3, 0, 2, 2, 3, 2, 0, 4, 0, 2, 4, 3, 3, 0, 4, 4, 2,
4, 3, 1, 1, 0, 2, 3, 0, 2, 3, 0, 4, 4, 1, 0, 0, 4, 2, 0, 2, 4, 4,
0, 3, 1, 1, 1, 3, 0, 2, 2, 3, 4, 1, 4, 0, 1, 0, 1, 0, 3, 0, 3, 0,
0, 0, 2, 0, 0, 0, 3, 4, 3, 4, 0, 1, 0, 0, 3, 0, 2, 0, 3, 0, 0, 4,
0, 4, 0, 1, 0, 1, 3, 0, 0, 0, 3, 2, 0, 3, 1, 0, 1, 0, 2, 2, 3, 0,
2, 4, 4, 2, 1, 1, 4, 2, 3, 2])
```

[30]:
```
cost = []
K = range(1,30)
for num_clusters in list(K):
    model = KModes(n_clusters=num_clusters, random_state=0, n_init = 20,␣
 ↪max_iter = 20).fit(temp)
    model.fit(temp)
    cost.append(model.cost_)
```

[31]:
```
plt.plot(K, cost, 'bx-')
plt.xlabel('k clusters')
plt.ylabel('Cost')
plt.title('Elbow Method For Optimal k')
```

```
plt.show()
```




[32]:
```
model = KModes(n_clusters=9, random_state=0, n_init = 20, max_iter = 20).
 ↪fit(temp)
clusters = model.fit_predict(temp)
clusters
```

[32]:
```
array([1, 2, 0, 1, 1, 2, 8, 1, 6, 1, 2, 7, 8, 8, 6, 1, 8, 4, 0, 0, 2, 3,
       7, 1, 8, 8, 0, 0, 2, 0, 8, 7, 4, 3, 8, 3, 0, 4, 7, 3, 0, 0, 2, 1,
       5, 1, 0, 5, 6, 7, 1, 1, 4, 2, 8, 0, 1, 5, 1, 0, 3, 2, 0, 8, 8, 3,
       0, 7, 4, 2, 7, 1, 2, 2, 5, 3, 3, 3, 4, 1, 1, 7, 1, 7, 4, 5, 0, 1,
       0, 4, 0, 1, 5, 6, 1, 0, 7, 1, 8, 3, 8, 1, 0, 7, 1, 0, 5, 1, 2, 2,
       1, 8, 3, 7, 0, 8, 3, 0, 2, 6, 3, 1, 0, 5, 1, 0, 1, 8, 1, 0, 7, 2,
       0, 2, 4, 1, 5, 5, 5, 4, 7, 2, 7, 0, 2, 2, 3, 7, 0, 3, 5, 1, 2, 6,
       2, 4, 4, 6, 8, 2, 0, 0, 1, 0, 3, 3, 0, 0, 1, 6, 2, 1, 7, 6, 6, 1,
       1, 5, 3, 1, 2, 3, 1, 0, 3, 0, 2, 3, 1, 2, 2, 3, 8, 5, 8, 1, 0, 7,
       7, 7, 2, 0, 7, 2, 7, 1, 1, 8, 0, 6, 0, 1, 0, 0, 3, 4, 3, 0, 1, 2,
       1, 7, 8, 1, 0, 1, 5, 0, 0, 8, 0, 6, 6, 7, 2, 2, 2, 5, 0, 2, 0, 6,
       1, 7, 5, 2, 3, 6, 3, 0, 6, 1, 7, 1, 8, 0, 2, 2, 7, 0, 3, 3, 0, 8,
       3, 8, 8, 1, 2, 2, 1, 0, 1, 6, 2, 1, 4, 1, 2, 0, 5, 3, 1, 1, 0, 0,
       8, 1, 4, 8, 1, 2, 1, 2, 3, 0, 1, 7, 5, 0, 6, 1, 4, 0, 2, 0, 0, 2,
       2, 0, 2, 8, 3, 0, 0, 1, 4, 1, 2, 3, 5, 1, 5, 8, 0, 0, 2, 2, 8, 8,
       0, 7, 2, 0, 0, 2, 0, 0, 0, 0, 1, 4, 3, 2, 8, 0, 5, 2, 4, 0, 2, 1,
```

```
       5, 2, 0, 2, 1, 0, 6, 8, 7, 0, 2, 7, 6, 0, 6, 3, 4, 0, 1, 0, 1, 2,
       1, 4, 2, 2, 8, 3, 0, 2, 2, 2, 2, 0, 2, 4, 7, 2, 0, 4, 1, 7, 0, 8,
       5, 0, 3, 0, 2, 4, 2, 3, 4, 2, 2, 0, 2, 1, 0, 7, 1, 1, 1, 7, 4, 0,
       8, 0, 0, 8, 0, 8, 0, 2, 7, 8, 1, 6, 2, 1, 7, 7, 2, 1, 6, 6, 0, 0,
       7, 1, 0, 5, 1, 4, 2, 0, 0, 8, 0, 7, 6, 4, 1, 6, 7, 4, 2, 6, 3, 2,
       7, 0, 0, 2, 1, 8, 8, 7, 1, 0, 0, 2, 2, 8, 2, 2, 3, 1, 7, 4, 8, 3,
       3, 2, 0, 5, 4, 5, 2, 1, 8, 6, 5, 1, 2, 4, 0, 7, 8, 0, 1, 0, 2, 2,
       4, 2, 0, 2, 3, 5, 1, 7, 3, 1, 6, 5, 4, 2, 0, 5, 2, 4, 0, 7, 2, 2,
       2, 0, 7, 3, 2, 1, 2, 0, 2, 6, 8, 1, 2, 2, 4, 2, 1, 2, 7, 7, 0, 4,
       4, 0, 3, 3, 8, 1, 6, 2, 7, 1, 2, 1, 1, 7, 6, 0, 0, 1, 6, 0, 0, 3,
       8, 4, 3, 0, 1, 2, 0, 8, 7, 6, 7, 7, 1, 0, 6, 7, 2, 7, 7, 4, 2, 8,
       2, 5, 4, 4, 2, 1, 0, 0, 4, 3, 1, 8, 2, 5, 5, 1, 0, 5, 3, 3, 8, 4,
       5, 3, 1, 7, 1, 6, 5, 0, 7, 0, 1, 0, 8, 0, 2, 4, 0, 7, 0, 2, 7, 6,
       3, 2, 3, 1, 0, 7, 3, 2, 4, 1, 1, 1, 1, 0, 7, 6, 3, 7, 2, 0, 6, 0,
       3, 4, 8, 0, 7, 2, 1, 1, 7, 1, 7, 7, 2, 1, 6, 0, 8, 0, 5, 0, 1, 1,
       3, 5, 7, 8, 2, 1, 5, 3, 3, 3, 2, 7, 2, 4, 5, 8, 3, 0, 8, 7, 1, 2,
       1, 0, 0, 5, 2, 8, 6, 2, 1, 1, 7, 1, 0, 0, 2, 1, 5, 0, 7, 1, 4, 3,
       1, 8, 1, 2, 0, 1, 1, 2, 0, 5, 2, 8, 0, 8, 8, 0, 5, 8, 1, 7, 0, 8,
       4, 1, 0, 1, 0, 2, 5, 7, 2, 2, 5, 2, 7, 0, 1, 3, 0, 7, 0, 0, 0, 1,
       1, 8, 1, 7, 0, 2, 6, 7, 2, 7, 8, 8, 3, 1, 4, 2, 1, 7, 1, 3, 7, 8,
       8, 2, 5, 3, 5, 4, 0, 1, 7, 3, 7, 3, 4, 6, 0, 0, 3, 4, 3, 8, 3, 0,
       7, 4, 1, 3, 1, 5, 2, 1, 7, 8, 2, 2, 2, 0, 0, 3, 4, 0, 1, 7, 0, 7,
       2, 0, 1, 2, 8, 2, 6, 3, 2, 4, 3, 3, 0, 3, 2, 4, 6, 2, 1, 1, 2, 4,
       0, 4, 1, 1, 2, 2, 5, 1, 0, 7, 1, 0, 7, 7, 4, 0, 3, 0, 8, 1, 8, 0,
       2, 8, 2, 0, 0, 1, 2, 0, 0, 8, 0, 2, 0, 4, 5, 7, 4, 2, 8, 8, 0, 7,
       7, 1, 5, 1, 4, 5, 8, 1, 0, 2, 6, 0, 0, 7, 8, 2, 0, 2, 0, 4, 0, 3,
       8, 3, 6, 1, 5, 2, 2, 7, 0, 8, 7, 0, 0, 2, 2, 2, 7, 6, 8, 0, 7, 0,
       1, 2, 4, 6, 4, 6, 1, 3, 1, 0, 4, 8, 2, 1, 7, 0, 5, 6, 1, 1, 4, 7,
       2, 7, 2, 1, 0, 1, 8, 6, 6, 6, 4, 0, 2, 8, 6, 2, 1, 1, 0, 2, 8, 2,
       3, 8, 3, 0, 1, 5, 8, 4, 2, 0], dtype=uint16)
```

[33]:
```
seeks = df.copy(deep=True)
seeks.insert(0, "Cluster", clusters, True)
```

[34]:
```
```

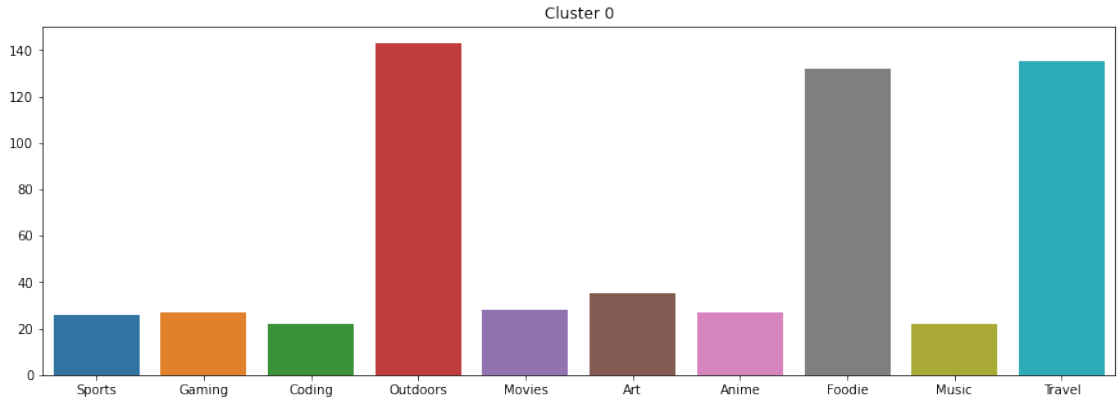
[34]:

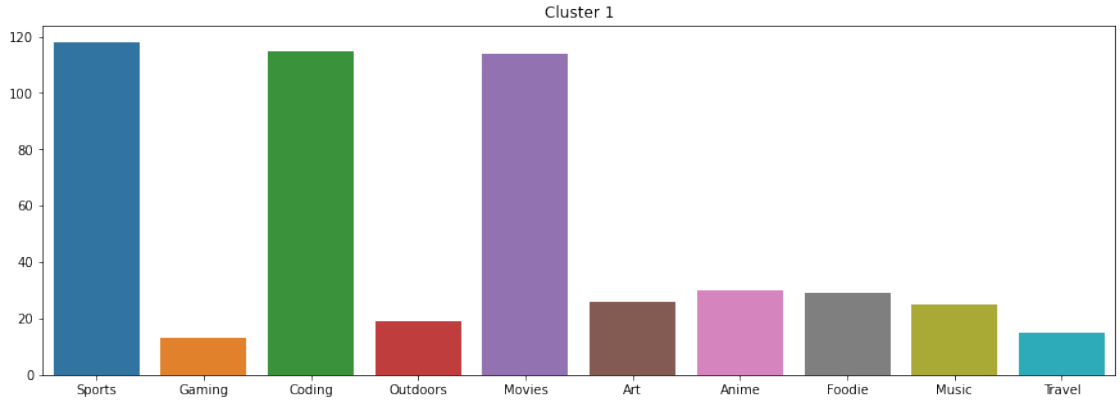
| | Cluster | first_name | last_name | email |
|---|---|---|---|---|
| 0 | 1 | Cash | Foxley | cfoxley0@i2i.jp |
| 1 | 2 | Rooney | Gitthouse | rgitthouse1@cpanel.net |
| 2 | 0 | Roseanne | Graffham | rgraffham2@tumblr.com |
| 3 | 1 | Conny | Flintoff | cflintoff3@theglobeandmail.com |
| 4 | 1 | Grove | Szymanek | gszymanek4@phoca.cz |
| .. | ... | ... | ... | ... |
| 995 | 5 | Allyn | Yerson | ayersonrn@sbwire.com |
| 996 | 8 | Susanna | Corsan | scorsanro@berkeley.edu |
| 997 | 4 | Lind | Gladtbach | lgladtbachrp@merriam-webster.com |
| 998 | 2 | Theo | Risdale | trisdalerq@washingtonpost.com |

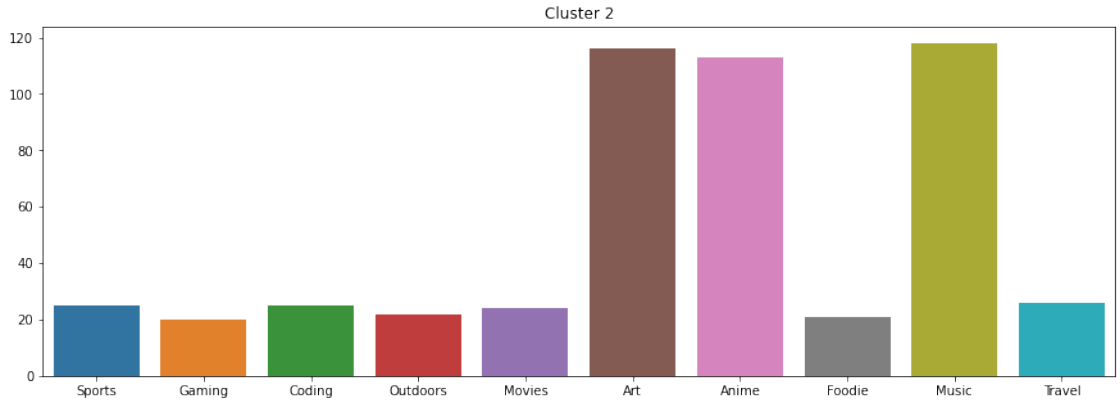
```
999        0    Sergent   Ingledow         singledowrr@csmonitor.com

          gender University  Age  Class Graduation Year  Lease Term  \
0           Male         PU   22                   2025          12
1           Male       UIUC   41                   2022           6
2    Genderfluid       UMAA   22                   2024          12
3        Agender         GT   25                   2023           6
4           Male       UCLA   30                   2025           6
..           ...        ...  ...                    ...         ...
995  Genderqueer      UPenn   40                   2025           6
996       Female         PU   21                   2025          12
997     Bigender       UIUC   50                   2022           6
998         Male        CIT   22                   2025          12
999         Male         UW   33                   2024          12

                Major  ...  Outdoors  Movies  Art  Anime  Foodie  Music  \
0             Nursing  ...         0       1    0      0       0      0
1             Nursing  ...         0       1    0      1       0      1
2         Engineering  ...         1       0    0      0       1      1
3         Engineering  ...         0       0    0      0       0      1
4    Computer Science  ...         0       0    0      0       1      0
..                ...  ...       ...     ...  ...    ...     ...    ...
995  Computer Science  ...         0       0    0      1       1      0
996          Business  ...         1       0    0      0       0      0
997             Other  ...         0       1    0      0       1      0
998          Business  ...         0       0    1      1       0      0
999           Nursing  ...         0       0    0      0       1      0

     Travel  Interest 1  Interest 2  Interest 3
0         1      Movies      Travel      Coding
1         0       Music       Anime      Movies
2         0    Outdoors       Music      Foodie
3         0      Sports      Coding       Music
4         0      Sports      Coding      Foodie
..      ...         ...         ...         ...
995       0      Foodie      Sports       Anime
996       0    Outdoors      Gaming      Coding
997       0      Movies      Foodie      Gaming
998       0      Coding       Anime         Art
999       1      Travel      Sports      Foodie

[1000 rows x 23 columns]
```

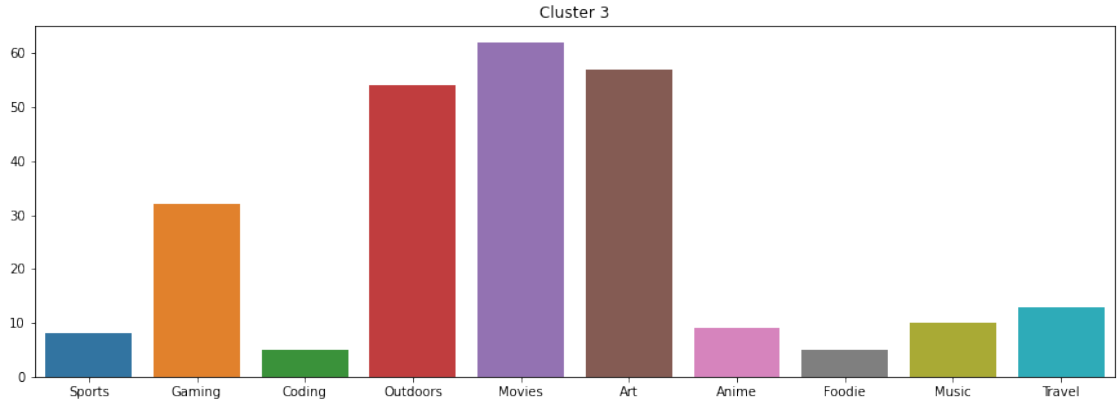
[14]:
```python
for cluster in range(0,9):
    temp = seeks.loc[seeks['Cluster'] == cluster]
    count = {}
    for i, row in temp.iterrows():
```

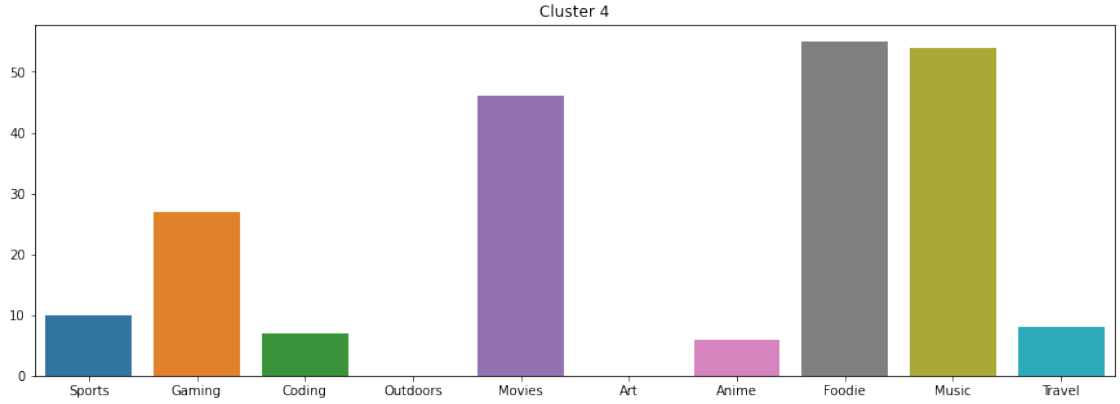
```
        for j in interests:
            if j in count.keys():
                count[j]+= row[j]
            else:
                count[j] = row[j]
    plt.subplots(figsize = (15,5))
    keys = list(count.keys())
    vals = [int(count[k]) for k in keys]
    sns.barplot(x=keys, y=vals).set(title="Cluster " + str(cluster))
    plt.show()
```

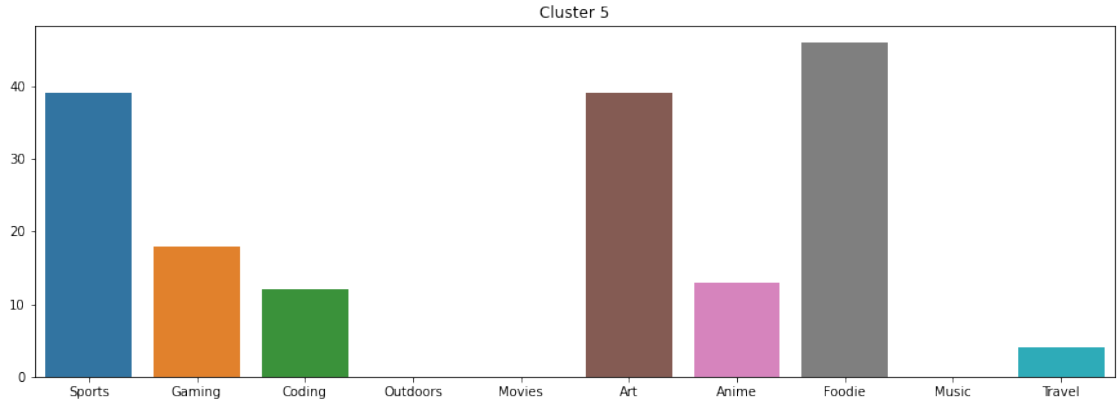


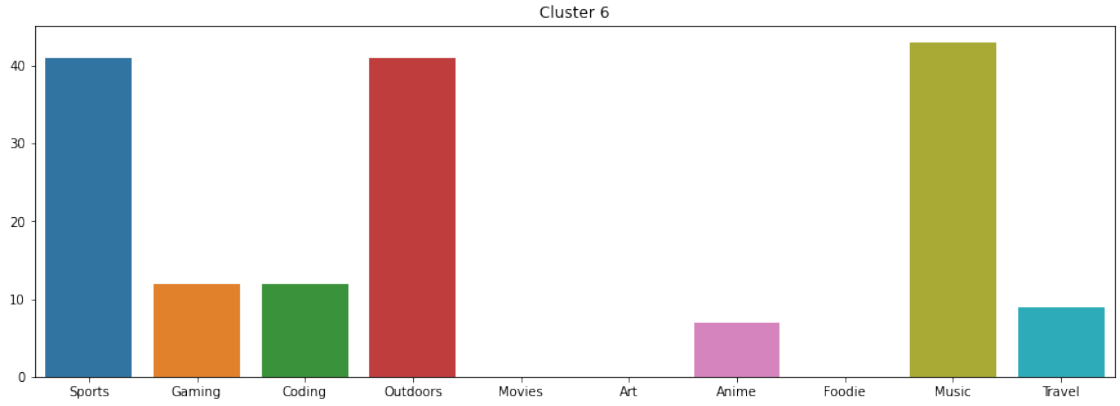



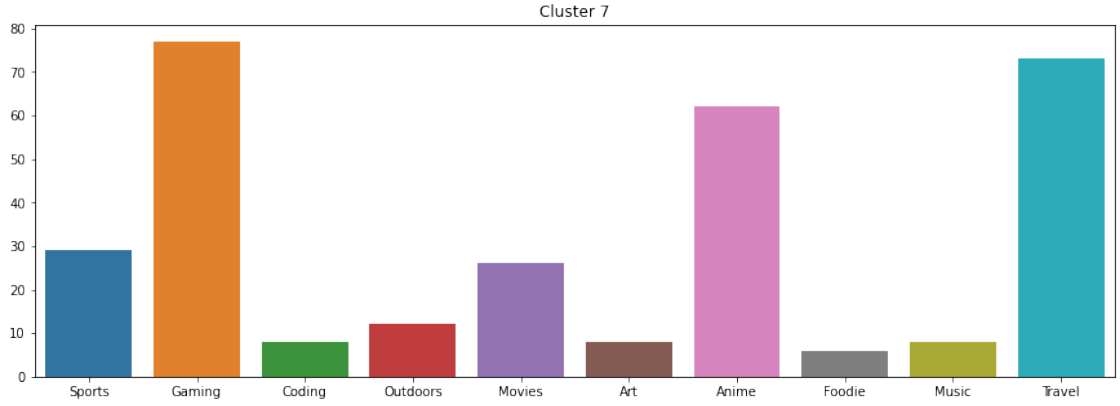

Cluster 2
120
100
80
60
40
20
0
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel

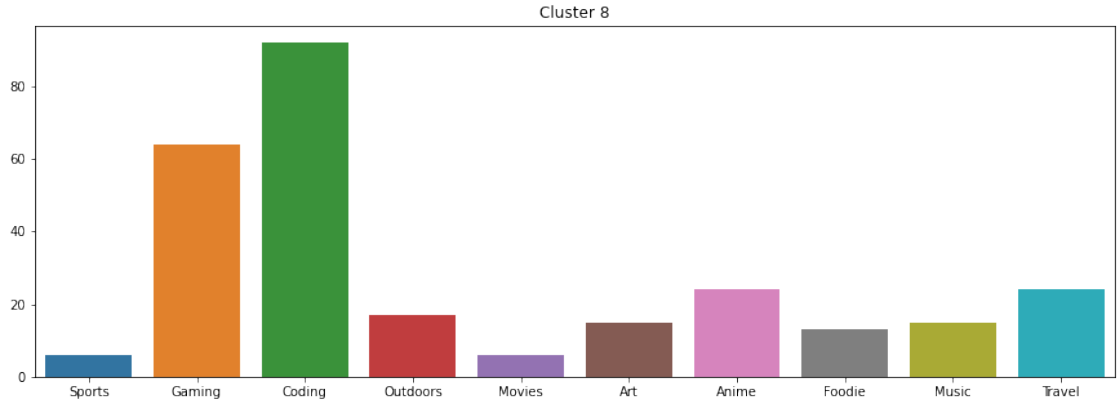

Cluster 3
60
50
40
30
20
10
0
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel


Cluster 4
50
40
30
20
10
0
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel


Cluster 5
0
10
20
30
40
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel


Cluster 6
0
10
20
30
40
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel


Cluster 7
0
10
20
30
40
50
60
70
80
Sports
Gaming
Coding
Outdoors
Movies
Art
Anime
Foodie
Music
Travel




[15]:
```
# plt.subplots(figsize = (15,5))
# sns.countplot(x='Interest 1',hue=col, data = seeks)
# plt.show()
```

[40]:
```
dalton = [['Dalton', 'Pang', 'dspangp@gmail.com', 'Male', 'UMD', 19, 2023, 12,␣
 ↪'Computer Science', 'Gaming', 'Music', 'Coding']]
data = [[1,0,1,1,0,0,0,0,0,0]]
```

[42]:
```
model.predict(data)
```

[42]:
```
array([1], dtype=uint16)
```

[ ]:
```
pd.set_option("display.max_rows", None, "display.max_columns", None)
```

[ ]:
```
seeks
```

[26]:
```
def get_cluster(interests):
    return model.predict(interests)
```

[46]:
```
from pathlib import Path
filepath = Path('C:/Users/Dalton/Documents/Project Seek/out.csv')
filepath.parent.mkdir(parents=True, exist_ok=True)
seeks.to_csv(filepath)
```

[ ]:
```

```