EDS_4_Michael_Friderich

January 8, 2021

1 Unüberwachtes Lernen mit Twitter

Semesterarbeit 4

1.1 EDS-Einführung in Data Science

Klasse: BSc INF-P-IN010, BE1, HS20/21 Dozent: Dr. Tim vor der Brück Autor: Michael Friderich Datum: 08.01.2021

1.2 Einleitung

Viel häufiger als beim überwachten Lernen, treffen wir auf das unüberwachte Lernen. Dabei sind die Daten nicht klassifiziert, was natürlich auf die meisten zugänglichen Daten zutrifft. In dieser Arbeit werden wir Daten von Twitter herunterladen. Diese sollen anschliessend in Cluster verteilt werden. Um die Anzahl Cluster zu berechnen, werden wir die Ellbow-Methode anwenden. Anschliessend werden die Daten mittels K-Means Algorithmus auf die Cluster verteilt. Zum Schluss erhalten die einzelnen Cluster einen Centroiden und die Daten werden solange optimiert bis die Zuteilung optimal ist.

1.3 Verbindung mit der Twitter API

Als Erstes richten wir den Zugriff zu der Twitter API ein. Die benötigten credentials befinden sich in einem separaten File (Twitter_credentials.py) welches hier importiert wird. Dies ermöglicht uns Daten mittels der angebotenen API von Twitter abzurufen.

[2]:
```
import twitter
import Twitter_credentials

# connect to Twitter's API
auth = twitter.oauth.OAuth(Twitter_credentials.OAUTH_TOKEN,
                           Twitter_credentials.OAUTH_TOKEN_SECRET,
                           Twitter_credentials.CONSUMER_KEY,
                           Twitter_credentials.CONSUMER_SECRET)

twitter_api = twitter.Twitter(auth=auth)
print(twitter_api)
```

```
<twitter.api.Twitter object at 0x7fa7c2ffb0d0>
```

1.4 Datensatz

Der erste wichtige Punkt der beachtet werden muss, sind die Daten. Sie müssen in geeigneter Form und in passendem Format vorhanden sein. Zum Beispiel sind bei den weltweite trendigen Topics auch Zeichen ausserhalb des UTF-8 Zeichensatzes enthalten. Dies muss bei der Datenbearbeitung berücksichtigt werden und die Daten gegebenenfalls zuerst angepasst werden.

Ich habe mich entschieden die 58 trendigen Topis aus der Schweiz abzurufen. Dafür bietet die Twitter API eine Methode an. Die WORLD_WOE_ID kann auf die gewünschte Region angepasst werden. Anschliessend werden die erhaltenen Topics im JSON Format ausgegeben.

[3]:
```
import json

# finde the 50 trending topics in Switzerland
WORLD_WOE_ID = 782538
world_trends = twitter_api.trends.place(_id = WORLD_WOE_ID)
print("Top trends from Switzerland:")
print(json.dumps(world_trends, indent=1))
print()
```

```
Top trends from Switzerland:
[
 {
  "trends": [
   {
    "name": "#WeHaveAPlan",
    "url": "http://twitter.com/search?q=%23WeHaveAPlan",
    "promoted_content": null,
    "query": "%23WeHaveAPlan",
    "tweet_volume": null
   },
   {
    "name": "#Adelboden",
    "url": "http://twitter.com/search?q=%23Adelboden",
    "promoted_content": null,
    "query": "%23Adelboden",
    "tweet_volume": null
   },
   {
    "name": "Markt",
    "url": "http://twitter.com/search?q=Markt",
    "promoted_content": null,
    "query": "Markt",
    "tweet_volume": null
   },
   {
    "name": "#climatechange",
    "url": "http://twitter.com/search?q=%23climatechange",
    "promoted_content": null,
```

```
 "query": "%23climatechange",
 "tweet_volume": null
},
{
 "name": "Arzt",
 "url": "http://twitter.com/search?q=Arzt",
 "promoted_content": null,
 "query": "Arzt",
 "tweet_volume": null
},
{
 "name": "Threema",
 "url": "http://twitter.com/search?q=Threema",
 "promoted_content": null,
 "query": "Threema",
 "tweet_volume": null
},
{
 "name": "Umsatz",
 "url": "http://twitter.com/search?q=Umsatz",
 "promoted_content": null,
 "query": "Umsatz",
 "tweet_volume": null
},
{
 "name": "Einsatz",
 "url": "http://twitter.com/search?q=Einsatz",
 "promoted_content": null,
 "query": "Einsatz",
 "tweet_volume": null
},
{
 "name": "#PMTnein",
 "url": "http://twitter.com/search?q=%23PMTnein",
 "promoted_content": null,
 "query": "%23PMTnein",
 "tweet_volume": null
},
{
 "name": "Verf\u00fcgung",
 "url": "http://twitter.com/search?q=Verf%C3%BCgung",
 "promoted_content": null,
 "query": "Verf%C3%BCgung",
 "tweet_volume": null
},
{
 "name": "Branchen",
 "url": "http://twitter.com/search?q=Branchen",
```

```
 "promoted_content": null,
 "query": "Branchen",
 "tweet_volume": null
},
{
 "name": "China",
 "url": "http://twitter.com/search?q=China",
 "promoted_content": null,
 "query": "China",
 "tweet_volume": 508508
},
{
 "name": "Tisch",
 "url": "http://twitter.com/search?q=Tisch",
 "promoted_content": null,
 "query": "Tisch",
 "tweet_volume": null
},
{
 "name": "#fisalpine",
 "url": "http://twitter.com/search?q=%23fisalpine",
 "promoted_content": null,
 "query": "%23fisalpine",
 "tweet_volume": null
},
{
 "name": "Januar 2021",
 "url": "http://twitter.com/search?q=%22Januar+2021%22",
 "promoted_content": null,
 "query": "%22Januar+2021%22",
 "tweet_volume": null
},
{
 "name": "Wunder",
 "url": "http://twitter.com/search?q=Wunder",
 "promoted_content": null,
 "query": "Wunder",
 "tweet_volume": null
},
{
 "name": "WhatsApp",
 "url": "http://twitter.com/search?q=WhatsApp",
 "promoted_content": null,
 "query": "WhatsApp",
 "tweet_volume": 561474
},
{
 "name": "Zurich",
```

```
 "url": "http://twitter.com/search?q=Zurich",
 "promoted_content": null,
 "query": "Zurich",
 "tweet_volume": null
},
{
 "name": "Schuhe",
 "url": "http://twitter.com/search?q=Schuhe",
 "promoted_content": null,
 "query": "Schuhe",
 "tweet_volume": null
},
{
 "name": "Unterst\u00fctzung",
 "url": "http://twitter.com/search?q=Unterst%C3%BCtzung",
 "promoted_content": null,
 "query": "Unterst%C3%BCtzung",
 "tweet_volume": null
},
{
 "name": "Kontext",
 "url": "http://twitter.com/search?q=Kontext",
 "promoted_content": null,
 "query": "Kontext",
 "tweet_volume": null
},
{
 "name": "Kontakte",
 "url": "http://twitter.com/search?q=Kontakte",
 "promoted_content": null,
 "query": "Kontakte",
 "tweet_volume": null
},
{
 "name": "Patienten",
 "url": "http://twitter.com/search?q=Patienten",
 "promoted_content": null,
 "query": "Patienten",
 "tweet_volume": null
},
{
 "name": "Kunden",
 "url": "http://twitter.com/search?q=Kunden",
 "promoted_content": null,
 "query": "Kunden",
 "tweet_volume": null
},
{
```

```
 "name": "Podcast",
 "url": "http://twitter.com/search?q=Podcast",
 "promoted_content": null,
 "query": "Podcast",
 "tweet_volume": 161980
},
{
 "name": "Pfizer",
 "url": "http://twitter.com/search?q=Pfizer",
 "promoted_content": null,
 "query": "Pfizer",
 "tweet_volume": 124661
},
{
 "name": "Impfung",
 "url": "http://twitter.com/search?q=Impfung",
 "promoted_content": null,
 "query": "Impfung",
 "tweet_volume": null
},
{
 "name": "Suisses",
 "url": "http://twitter.com/search?q=Suisses",
 "promoted_content": null,
 "query": "Suisses",
 "tweet_volume": null
},
{
 "name": "Grafik",
 "url": "http://twitter.com/search?q=Grafik",
 "promoted_content": null,
 "query": "Grafik",
 "tweet_volume": null
},
{
 "name": "Gegenteil",
 "url": "http://twitter.com/search?q=Gegenteil",
 "promoted_content": null,
 "query": "Gegenteil",
 "tweet_volume": null
},
{
 "name": "Risiko",
 "url": "http://twitter.com/search?q=Risiko",
 "promoted_content": null,
 "query": "Risiko",
 "tweet_volume": null
},
```

```
{
 "name": "Argument",
 "url": "http://twitter.com/search?q=Argument",
 "promoted_content": null,
 "query": "Argument",
 "tweet_volume": 87242
},
{
 "name": "Schritt",
 "url": "http://twitter.com/search?q=Schritt",
 "promoted_content": null,
 "query": "Schritt",
 "tweet_volume": null
},
{
 "name": "Jesus",
 "url": "http://twitter.com/search?q=Jesus",
 "promoted_content": null,
 "query": "Jesus",
 "tweet_volume": 330809
},
{
 "name": "Wirtschaft",
 "url": "http://twitter.com/search?q=Wirtschaft",
 "promoted_content": null,
 "query": "Wirtschaft",
 "tweet_volume": null
},
{
 "name": "J'avais",
 "url": "http://twitter.com/search?q=J%27avais",
 "promoted_content": null,
 "query": "J%27avais",
 "tweet_volume": 35188
},
{
 "name": "Konzept",
 "url": "http://twitter.com/search?q=Konzept",
 "promoted_content": null,
 "query": "Konzept",
 "tweet_volume": null
},
{
 "name": "Donald",
 "url": "http://twitter.com/search?q=Donald",
 "promoted_content": null,
 "query": "Donald",
 "tweet_volume": 1539124
```

```
},
{
 "name": "Druck",
 "url": "http://twitter.com/search?q=Druck",
 "promoted_content": null,
 "query": "Druck",
 "tweet_volume": null
},
{
 "name": "Gr\u00fcnden",
 "url": "http://twitter.com/search?q=Gr%C3%BCnden",
 "promoted_content": null,
 "query": "Gr%C3%BCnden",
 "tweet_volume": null
},
{
 "name": "Conseil",
 "url": "http://twitter.com/search?q=Conseil",
 "promoted_content": null,
 "query": "Conseil",
 "tweet_volume": 19274
},
{
 "name": "Seiten",
 "url": "http://twitter.com/search?q=Seiten",
 "promoted_content": null,
 "query": "Seiten",
 "tweet_volume": null
},
{
 "name": "Syria",
 "url": "http://twitter.com/search?q=Syria",
 "promoted_content": null,
 "query": "Syria",
 "tweet_volume": 157385
},
{
 "name": "Spotify",
 "url": "http://twitter.com/search?q=Spotify",
 "promoted_content": null,
 "query": "Spotify",
 "tweet_volume": 733882
},
{
 "name": "Australia",
 "url": "http://twitter.com/search?q=Australia",
 "promoted_content": null,
 "query": "Australia",
```

```
    "tweet_volume": 127209
   },
   {
    "name": "Referendum",
    "url": "http://twitter.com/search?q=Referendum",
    "promoted_content": null,
    "query": "Referendum",
    "tweet_volume": 30203
   },
   {
    "name": "krankheit",
    "url": "http://twitter.com/search?q=krankheit",
    "promoted_content": null,
    "query": "krankheit",
    "tweet_volume": null
   },
   {
    "name": "Liverpool",
    "url": "http://twitter.com/search?q=Liverpool",
    "promoted_content": null,
    "query": "Liverpool",
    "tweet_volume": 50709
   },
   {
    "name": "European",
    "url": "http://twitter.com/search?q=European",
    "promoted_content": null,
    "query": "European",
    "tweet_volume": 57374
   },
   {
    "name": "Wirkung",
    "url": "http://twitter.com/search?q=Wirkung",
    "promoted_content": null,
    "query": "Wirkung",
    "tweet_volume": null
   }
  ],
  "as_of": "2021-01-08T11:56:55Z",
  "created_at": "2020-12-11T15:26:52Z",
  "locations": [
   {
    "name": "Geneva",
    "woeid": 782538
   }
  ]
 }
]
```

In einem weiteren Schritt bringen wir die erhaltenen Daten in ein für uns passendes Format. Wir behalten den “name” und das “tweet_volume”.

[136]:
```
# slice "name" and "tweet_volume"
for trends in world_trends:
    dict = {}
    for trend in trends["trends"]:
        dict[trend["name"]] = trend["tweet_volume"]

    print('Name Tweet Volume')
    for x,y in dict.items():
        print(x, y)
```

```
Name Tweet Volume
#WeHaveAPlan None
#Adelboden None
Markt None
#climatechange None
Arzt None
Threema None
Umsatz None
Einsatz None
#PMTnein None
Verfügung None
Branchen None
China 508508
Tisch None
#fisalpine None
Januar 2021 None
Wunder None
WhatsApp 561474
Zurich None
Schuhe None
Unterstützung None
Kontext None
Kontakte None
Patienten None
Kunden None
Podcast 161980
Pfizer 124661
Impfung None
Suisses None
Grafik None
Gegenteil None
Risiko None
Argument 87242
```

```
Schritt None
Jesus 330809
Wirtschaft None
J'avais 35188
Konzept None
Donald 1539124
Druck None
Gründen None
Conseil 19274
Seiten None
Syria 157385
Spotify 733882
Australia 127209
Referendum 30203
krankheit None
Liverpool 50709
European 57374
Wirkung None
```

Da ich auch nach langem recherchieren und probieren keine Lösung gefunden habe, den Erstellungsort des Topics, abzurufen. Habe ich mich entschieden, das "tweet_volume" mit einer Random Zahl zwischen 1 und 100 zu ergänzen. Diese beiden Zahlen bilden die Daten für diese Thesis. Dieser Datensatz wird nun in eine csv Datei (Data.csv) gespeichert.

[137]:
```
import csv
import random

# trends in CSV File speichern
with open('Data.csv', 'w') as csv_file:
    fieldnames = ['name', 'tweet_volume', 'nummer']
    csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
    csv_writer.writeheader()

    # create a random number between 1-100
    for x,y in dict.items():
        i = random.randint(1, 100)
        csv_writer.writerow({'name': x, 'tweet_volume': y, 'nummer': i})
```

Die Daten sind in die drei Spalten "name", "tweet_volume" und "nummer" gegliedert.

[164]:
```
import pandas as pd

# show data from csv file
df = pd.read_csv("Data.csv")
df.head()
```

[164]:

	name	tweet_volume	nummer
0	Donald	1539124	9

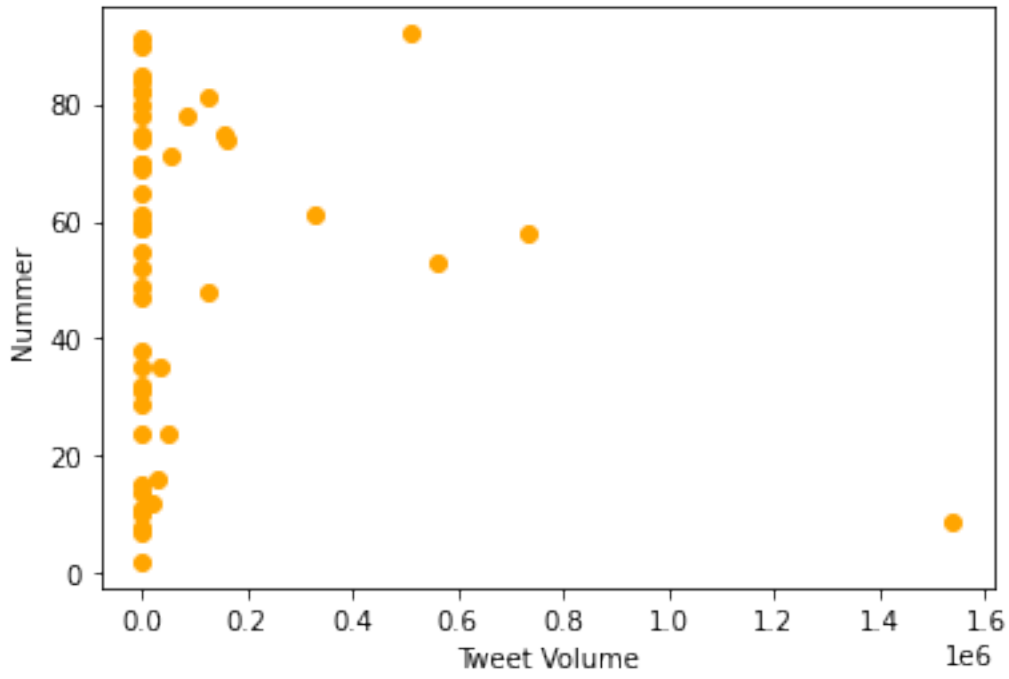
1	J'avais	35188	35
2	WhatsApp	561474	53
3	Spotify	733882	58
4	#Adelboden	1	70

1.5 K-Means Algorithms

Die Daten können nun in einem Diagramm ausgegeben werden.

[167]:
```
from matplotlib import pyplot as plt
%matplotlib inline

plt.scatter(df['tweet_volume'],df['nummer'], color='orange')
plt.xlabel('Tweet Volume')
plt.ylabel('Nummer')
```

[167]:
```
Text(0, 0.5, 'Nummer')
```



Nun wird der K-Means Algorithmus auf dem Datensatz angewendet. Zu Beginn wurden zwei Cluster gewählt. Der Algorithmus teilt nun die Daten auf die zwei Cluster 0 und 1 auf. Die Aufteilung ist auf der Tabelle ersichtlich.

[205]:
```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=2)
```

```
y_predicted = km.fit_predict(df[['tweet_volume', 'nummer']])

df['cluster'] = y_predicted
df.head()
```

[205]:

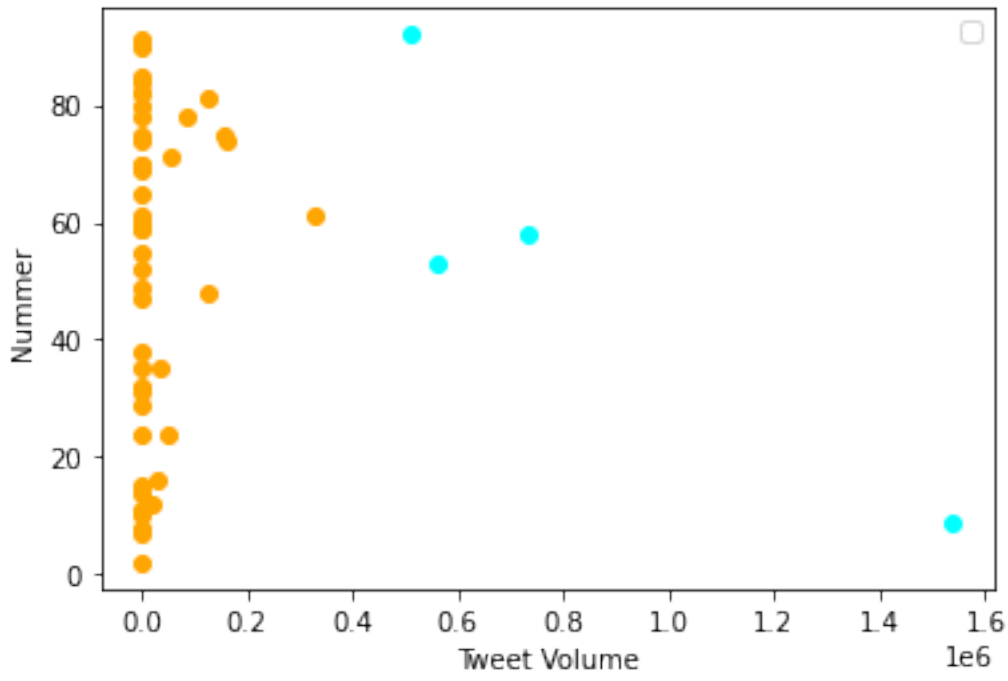
	name	tweet_volume	nummer	cluster
0	Donald	1539124	9	1
1	J'avais	35188	35	0
2	WhatsApp	561474	53	1
3	Spotify	733882	58	1
4	#Adelboden	1	70	0

Anhang der Daten und der Aufteilung durch den K-Means Algorithmus, werden die Daten ausgegeben. Die orangen Punkte beziehen sich auf die Daten des ersten Clusters, die grünen Punkte auf die Daten des zweiten Clusters.

[206]:
```
df1 = df[df.cluster == 0]
df2 = df[df.cluster == 1]

plt.scatter(df1.tweet_volume, df1['nummer'], color='orange')
plt.scatter(df2.tweet_volume, df2['nummer'], color='cyan')

plt.xlabel('Tweet Volume')
plt.ylabel('Nummer')
plt.legend()
```

```
No handles with labels found to put in legend.
```

[206]:
```
<matplotlib.legend.Legend at 0x7fa7598a3eb0>
```



1.6 Centroiden

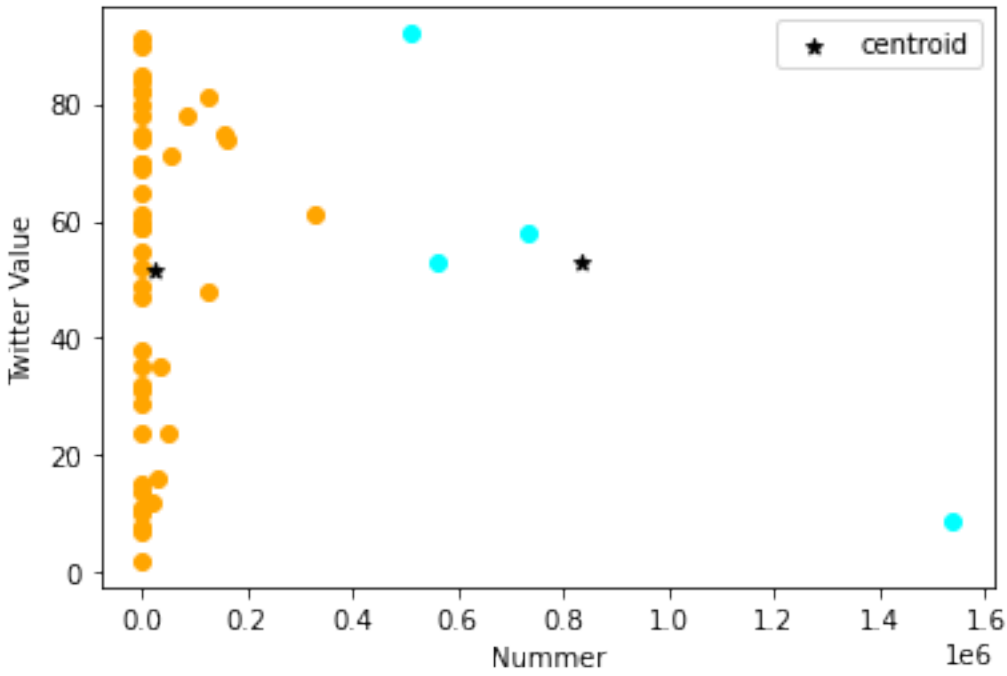
Nun werden die Centroiden (Mittelpunkte) der zwei Cluster berechnet und im Diagramm eingefügt. Das Array enthält die zwei Koordinaten der Centroiden.

[207]:
```
km.cluster_centers_
```

[207]:
```
array([[2.56971522e+04, 5.14782609e+01],
       [8.35747000e+05, 5.30000000e+01]])
```

[208]:
```
df1 = df[df.cluster == 0]
df2 = df[df.cluster == 1]

plt.scatter(df1.tweet_volume, df1['nummer'], color='orange')
plt.scatter(df2.tweet_volume, df2['nummer'], color='cyan')

plt.scatter(km.cluster_centers_[:,0],
            km.cluster_centers_[:,1],
            color='Black',
            marker='*',
            label='centroid')

plt.xlabel('Nummer')
plt.ylabel('Twitter Value')
plt.legend()
```

[208]: <matplotlib.legend.Legend at 0x7fa7595712b0>



Im nächsten Schritt werden nun die Punkte zu den Centroiden optimiert.

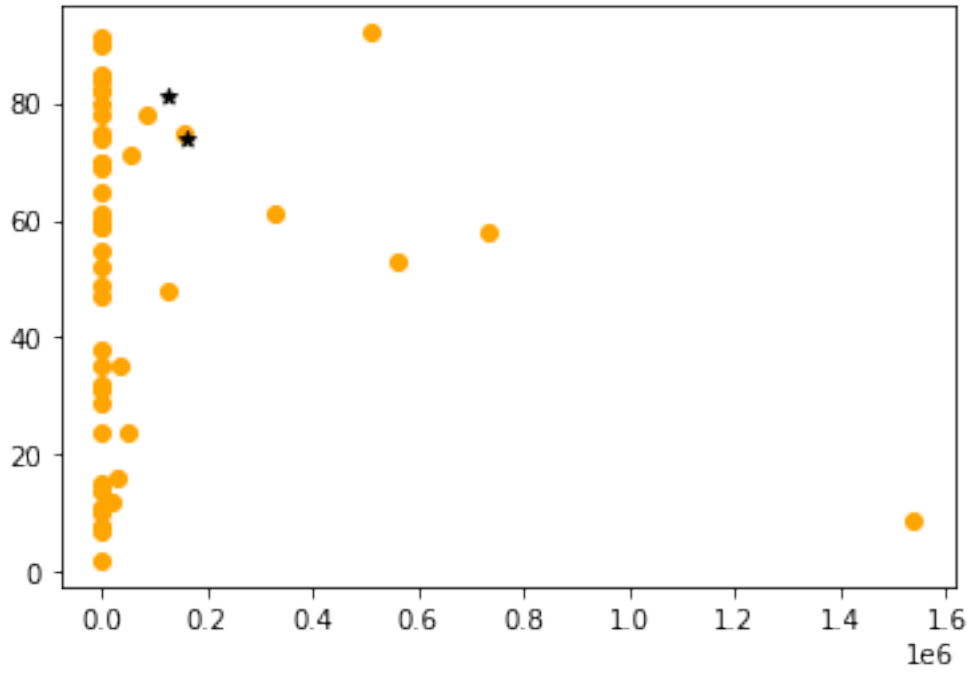
[209]:
```
import numpy as np

def plot_clusters(data_arg,li_cluster_indices,colors,centroids):
    data_clusters=[]
    for j in range(0,len(li_cluster_indices)):
        data_clusters.append([])
    data_centroid=[]

    for i in range(0,len(data_arg)):
        for j in range(0,len(li_cluster_indices)):
            cluster_indices=li_cluster_indices[j]
            if i in cluster_indices:
                data_clusters[j].append(data_arg[i])

    for cluster, color in zip(data_clusters,colors):
        x,y=np.array(cluster).T
        plt.scatter(x,y,color=color)
    x,y=np.array(centroids).T
    plt.scatter(x,y,color="black", marker='*', label='centroid')
    plt.show()
```

```
def euclidean_distance2(pt1, pt2):
    return np.linalg.norm(pt1 - pt2)
```

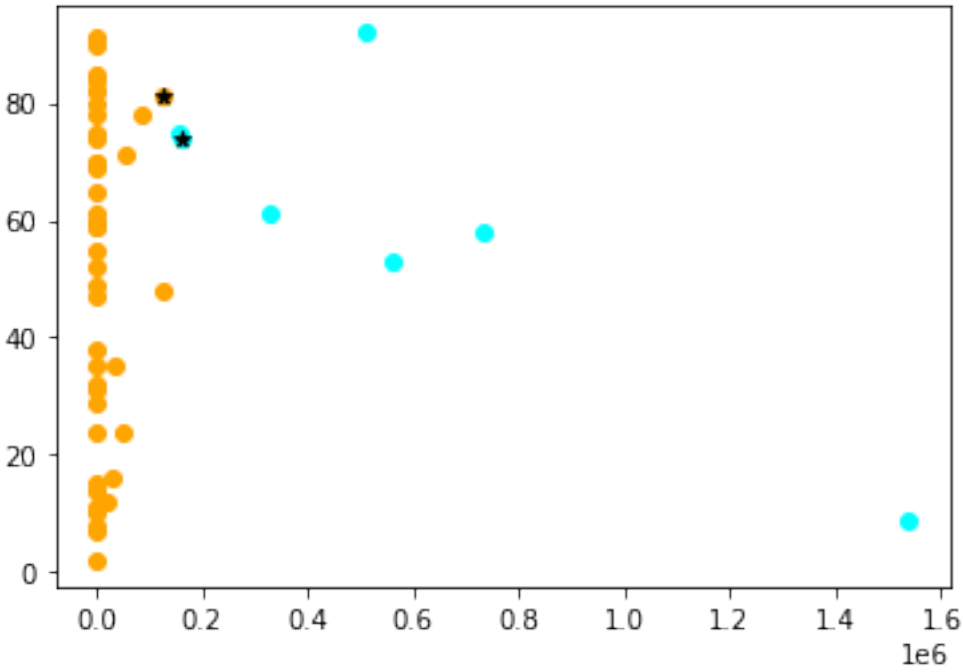
[210]:
```
data=␣
↪[[1539124,9],[35188,35],[561474,53],[733882,58],[1,70],[1,2],[1,82],[1,80],[1,55],[1,78],[1
    ␣
↪[1,52],[1,65],[508508,92],[1,15],[1,59],[1,10],[1,61],[1,24],[1,74],[1,32],[1,59],[1,14],[1
    ␣
↪[124661,81],[1,60],[1,8],[1,70],[1,11],[1,31],[87242,78],[1,29],[330809,61],[1,91],[1,82],[
    ␣
↪[1,90],[157385,75],[127209,48],[30203,16],[1,85],[50709,24],[57374,71],[1,49]
     ]
```

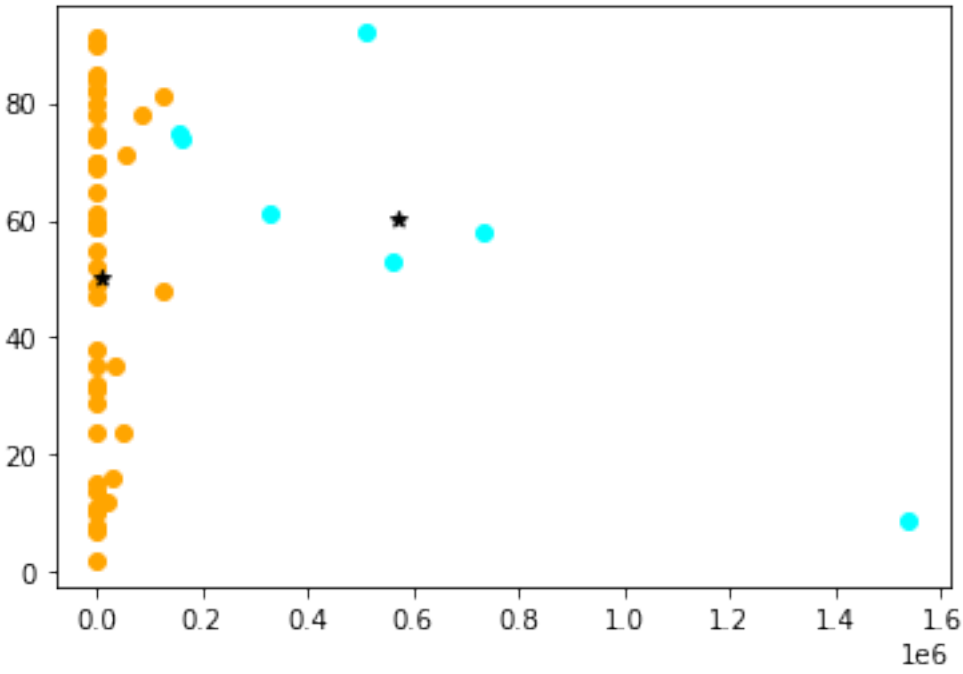
[201]:
```
random.shuffle(data)
c1=np.array(data[0])
c2=np.array(data[1])
rest=[]
for i in range(2,len(data)):
    rest.append(i)
plot_clusters(data,[rest],["orange"],[c1,c2])

cluster1=[]
cluster2=[]
centroid_index1=0
centroid_index2=1
oldc1=c1
oldc2=c2
for k in range(0,10):
        l=0
        cluster1=[]
        cluster2=[]
        for point in data:
            distance_c1=euclidean_distance2(np.array(point),c1)
            distance_c2=euclidean_distance2(np.array(point),c2)
            if distance_c1<distance_c2:
                cluster1.append(l)
            else:
                cluster2.append(l)
            l=l+1
        print ("adjust clusters")
        plot_clusters(data,[cluster1,cluster2],["orange","cyan"],[c1,c2,])

        c1=np.array([0.0])
        for i in cluster1:
            c1=c1+np.array(data[i])
        c1=c1/len(cluster1)
```

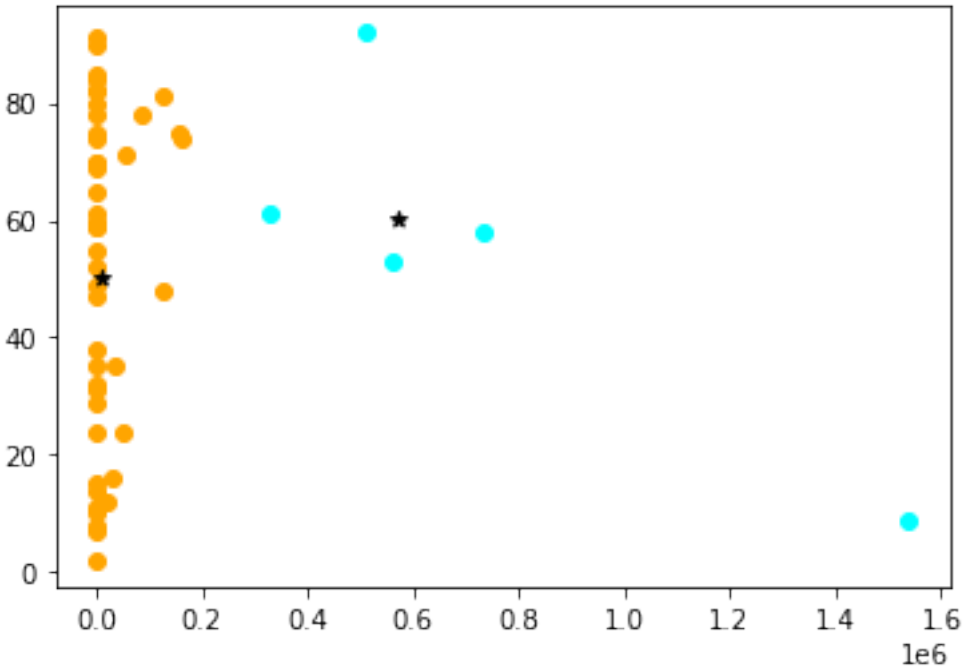
```
        c2=np.array([0.0])
        for i in cluster2:
            c2=c2+np.array(data[i])
        c2=c2/len(cluster2)

        print ("adjust centroids")
        plot_clusters(data,[cluster1,cluster2],["orange","cyan"],[c1,c2])

        if np.linalg.norm(c1-oldc1)+np.linalg.norm(c2-oldc2)<0.00001:
            print ("convergence!")
            break
        oldc1=c1
        oldc2=c2
```

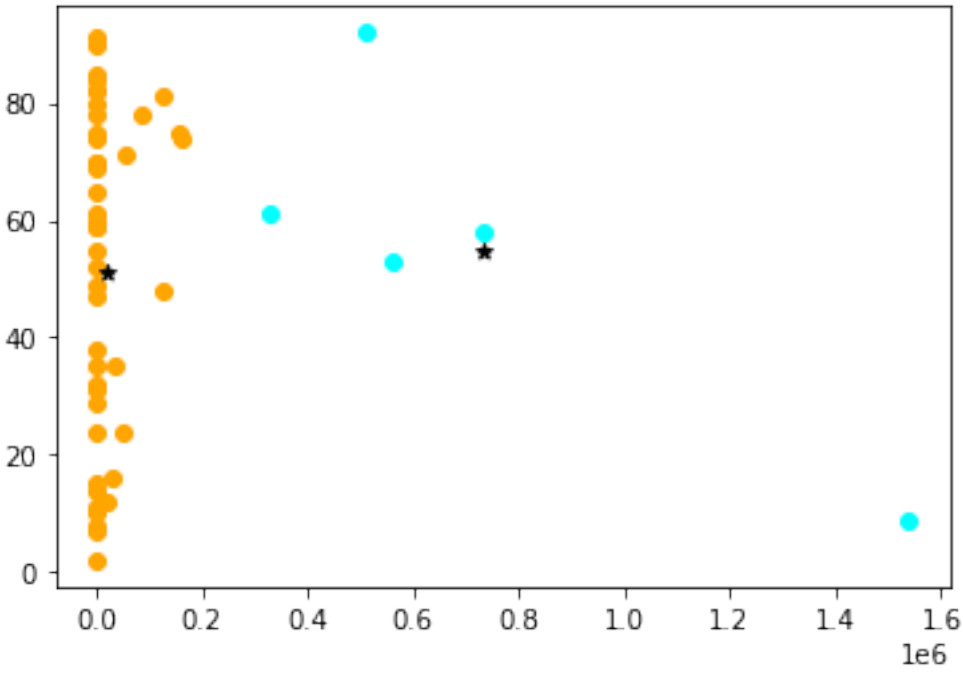


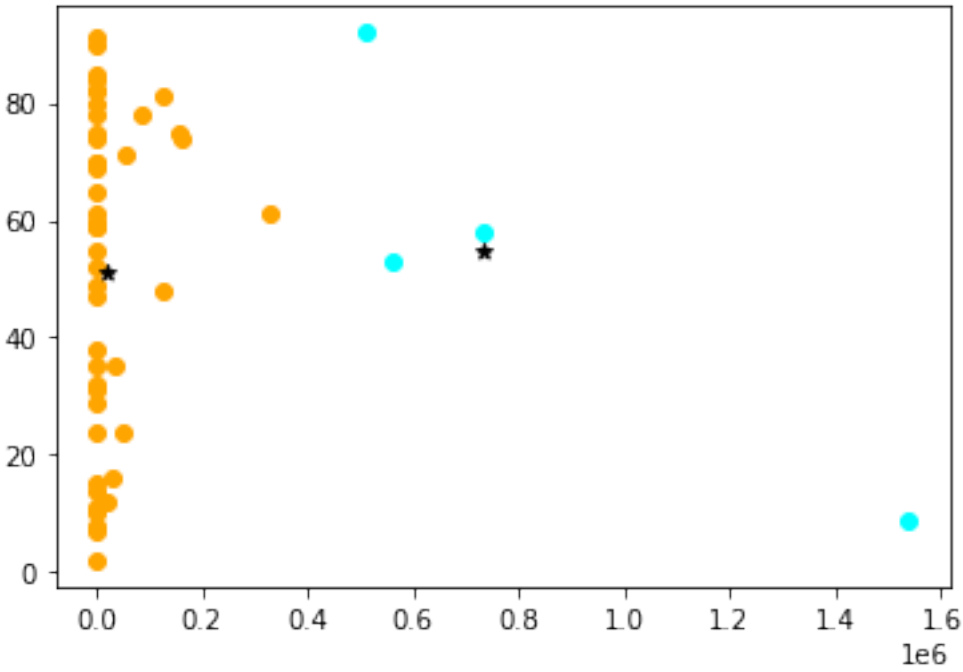

```
adjust clusters
adjust centroids
adjust clusters
adjust centroids
adjust clusters
adjust centroids
adjust clusters
adjust centroids
convergence!
```

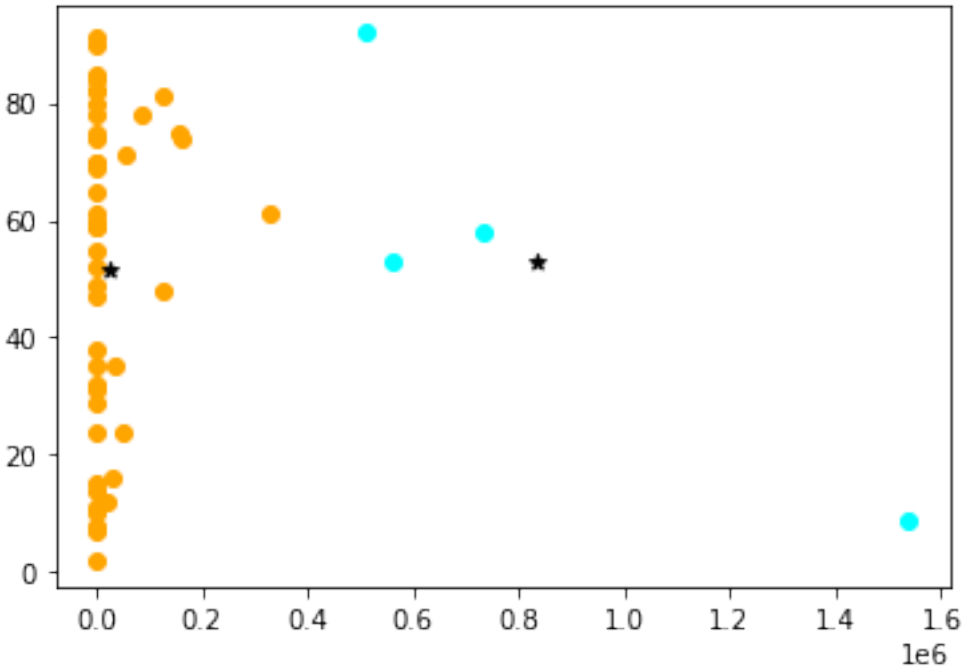

80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6

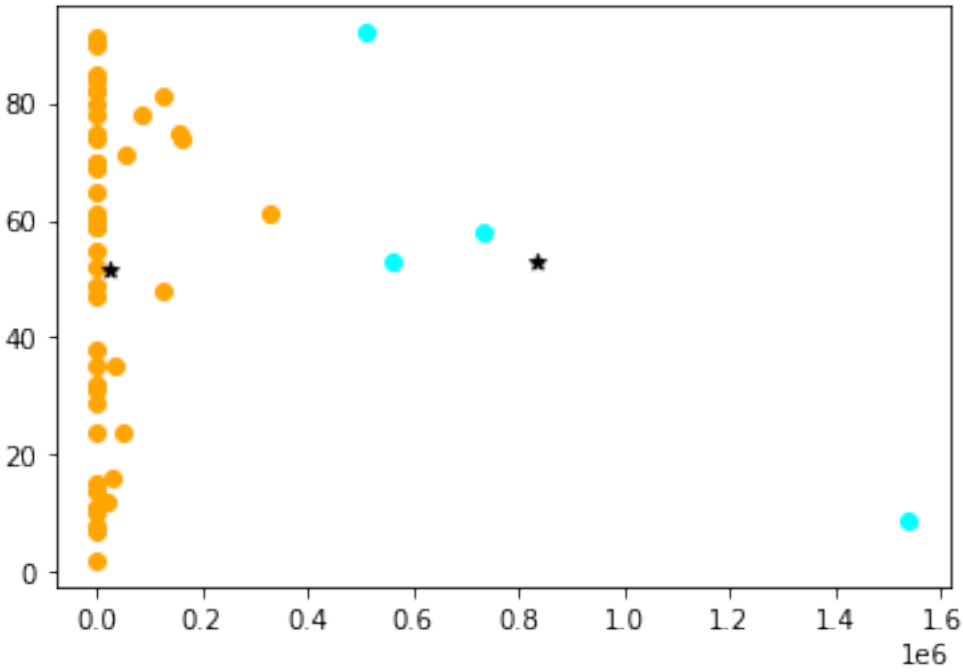

80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6

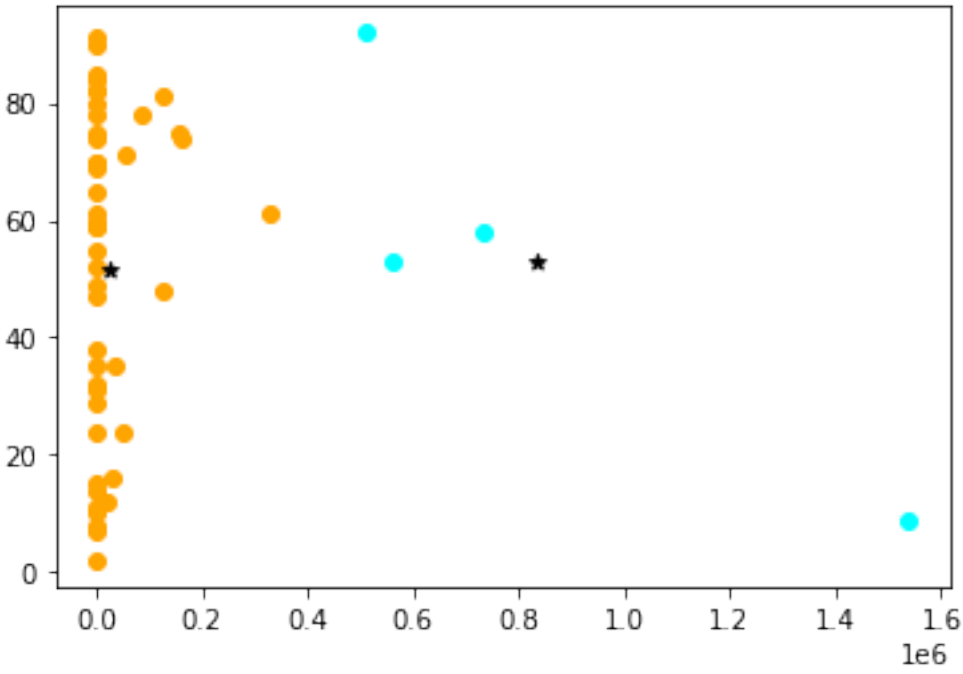

80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6


80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6


80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6


80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6


80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6

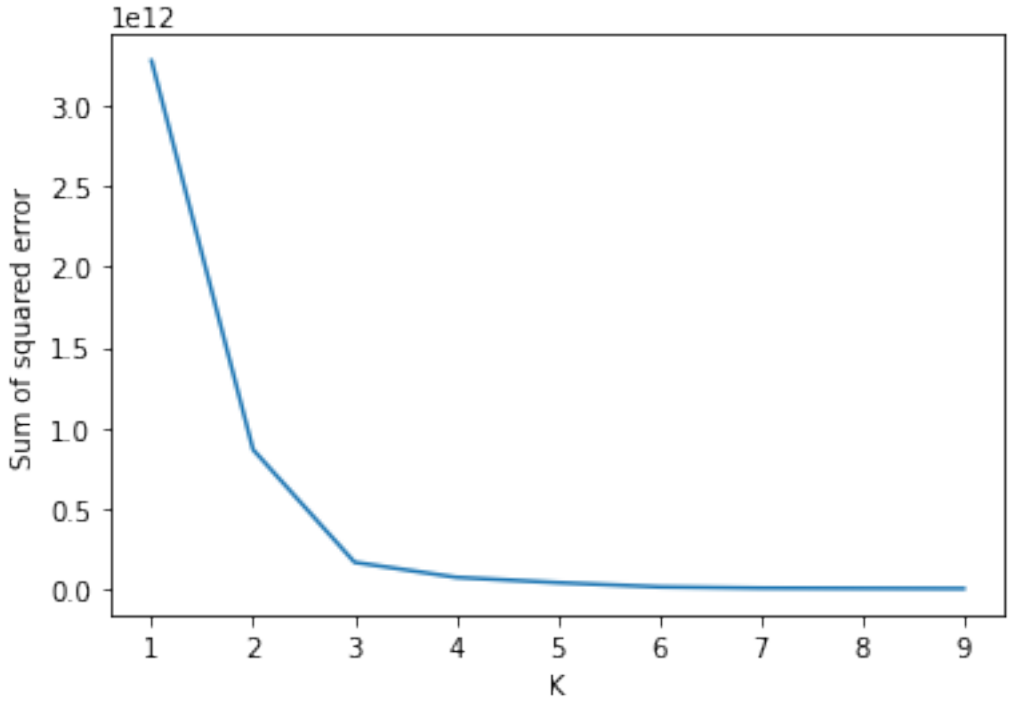

80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6

1.7 Ellbow Methode

Wir erkennen auf dem Diagramm, das die zwei Cluster nicht optimal gewählt wurden. Der Punkt unten rechts passt nicht recht zu den zwei Centroiden. Dies kann rechnerisch überprüft werden. Dafür wird die Ellbow-Methode angewandt. Aus der Berechnung ist ersichtlich das drei Cluster gewählt werden sollen. Dies untermauert die Vermutung.

[211]:
```
# ellbow plot method
k_rng = range(1,10)
sse = []
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['tweet_volume', 'nummer']])
    sse.append(km.inertia_)

plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng, sse)
```

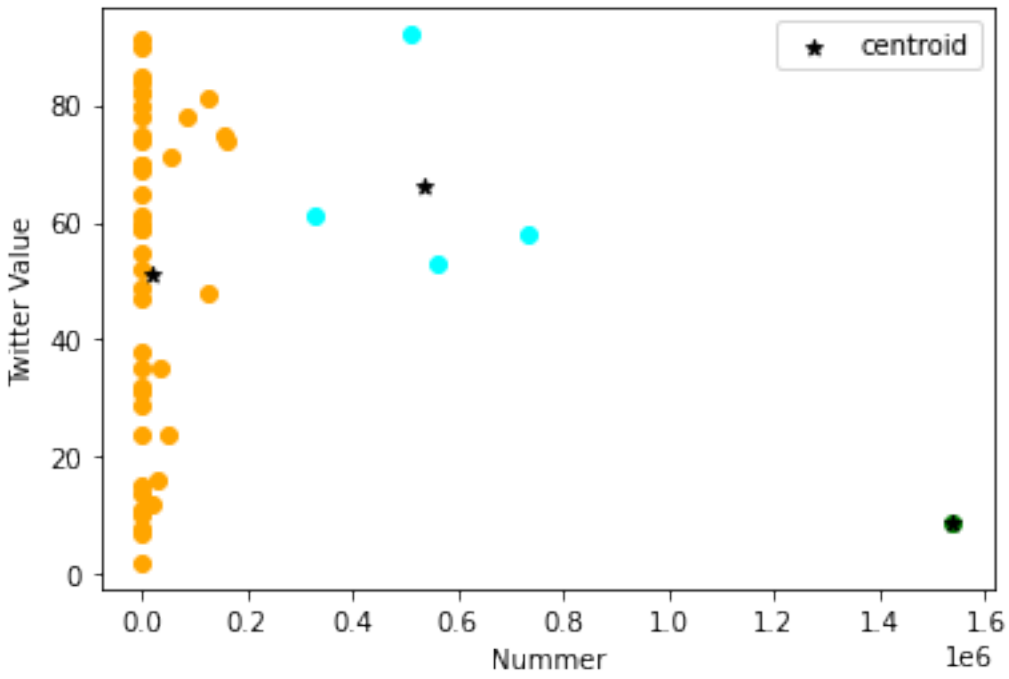
[211]:
```
[<matplotlib.lines.Line2D at 0x7fa759b8e160>]
```



Die Ellbow-Methode hat ergeben das sich drei Cluster besser eignen als zwei, daher wiederholen wir den K-Means Algorithmus für drei Cluster und geben das Resultat aus.

[219]:
```
df1 = df[df.cluster == 0]
df2 = df[df.cluster == 1]
df3 = df[df.cluster == 2]

plt.scatter(df1.tweet_volume, df1['nummer'], color='orange')
plt.scatter(df2.tweet_volume, df2['nummer'], color='green')
plt.scatter(df3.tweet_volume, df3['nummer'], color='cyan')

plt.scatter(km.cluster_centers_[:,0],
            km.cluster_centers_[:,1],
            color='Black',
            marker='*',
            label='centroid')

plt.xlabel('Nummer')
plt.ylabel('Twitter Value')
plt.legend()
```

[219]: <matplotlib.legend.Legend at 0x7fa75985dca0>



Das Diagramm zeigt nun eine sinnvolle Cluster aufteilung mit drei Centroiden.

1.8 Findings

Der wichtigste Punkt der beachtet werden muss, sind meines erachtens, die Daten. Sie müssen in geeigneter Form und in passendem Format vorhanden sein. Ansonsten müssen diese zuerst angepasst werden was aufwendig sein kann. Zum Beispiel sind bei den weltweite trendigen Topics auch Zeichen ausserhalb des UTF-8 Zeichensatzes enthalten. Ist der Datensatz vollständig und in passendem Format vorhanden, kann mit der Berechnung begonnen werden. Die Berechnung empfang ich als weniger aufwendiger Punkt.

1.9 Quellenverzeichnis

1.9.1 Literaturverzeichnis

[1] Russell, Matthew A. / Mikhail Klassen (2019): Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More, 3. Aufl., Sebastopol, USA, California: O'Reilly Media.