Kmeans

November 8, 2022

[1]:
```
pip install kneed
```

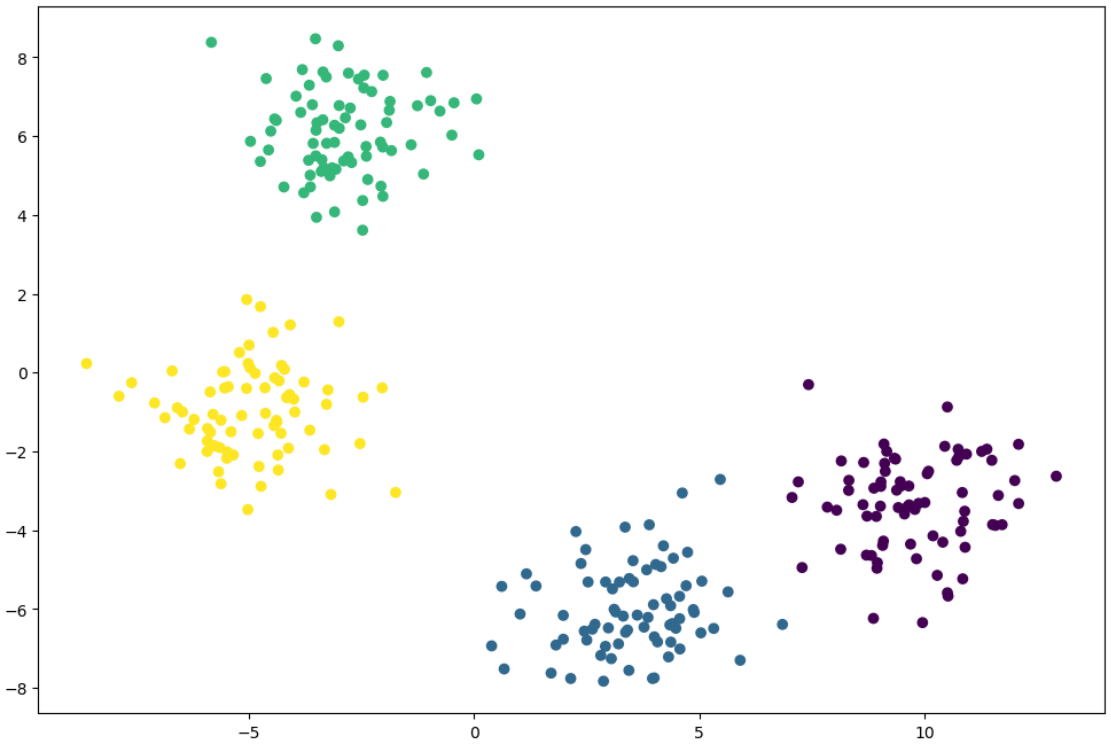
```
Defaulting to user installation because normal site-packages is not writeable
Collecting kneed
  Downloading kneed-0.8.1-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: scipy>=1.0.0 in /usr/lib/python3/dist-packages
(from kneed) (1.8.1)
Requirement already satisfied: numpy>=1.14.2 in /usr/lib/python3/dist-packages
(from kneed) (1.21.5)
Installing collected packages: kneed
Successfully installed kneed-0.8.1
Note: you may need to restart the kernel to use updated packages.
```

[1]:
```
import matplotlib.pyplot as plt #matplotlib
from kneed import KneeLocator #técnica do cotovelo
from sklearn.datasets import make_blobs #cria um grupo de dados sitético
from sklearn.cluster import KMeans #utiliza para agrupar os dados
from sklearn.metrics import silhouette_score #número de grupamentos
from sklearn.preprocessing import StandardScaler #padronização dos dados
```

[20]:
```
Caracteristicas, Labels = make_blobs(
n_samples= 300,
centers= 4,
cluster_std= 1.2,
random_state= 56)
```

[21]:
```
plt.figure(figsize=(12,8))
plt.scatter(Caracteristicas[:,0], Caracteristicas[:,1], c=Labels)
```

[21]:
```
<matplotlib.collections.PathCollection at 0x7f74e9689360>
```



0.1 Preprocessamento dos dados

[22]:
```
scalar = StandardScaler()
```

[23]:
```
scalar_caracteristicas = scalar.fit_transform(Caracteristicas)
```

[24]:
```
scalar_caracteristicas[:10]
```

[24]:
```
array([[ 1.40584388, -0.49763139],
       [ 1.59046813, -0.19333487],
       [ 1.39924977, -0.50902901],
       [ 1.17915555, -0.36344099],
       [ 1.37452803, -0.39530045],
       [-0.87173292,  1.20953894],
       [-1.1136886 ,  0.33562154],
       [-1.21304297, -0.17495737],
       [ 0.18906113, -0.74234596],
       [ 1.55001804, -0.97856452]])
```

[25]:
```
Caracteristicas[:10]
```

[25]:
```
array([[ 9.65903183, -3.35789179],
       [10.74860631, -1.94617861],
```

```
[ 9.62011617, -3.41076844],
[ 8.32121292, -2.73534655],
[ 9.47421891, -2.88315111],
[-3.78226616,  4.56212935],
[-5.2101866 ,  0.50779207],
[-5.79653418, -1.86092047],
[ 2.47809304, -4.49318832],
[10.50988696, -5.58906959]])
```

0.2 Inicializando o algoritmo Kmeans

[26]:
```
kmeans = KMeans(
init='random',
n_clusters=4,
n_init=20,
max_iter=800,
random_state=56)
```

[27]:
```
kmeans.fit(scalar_caracteristicas)
```

[27]:
```
KMeans(init='random', max_iter=800, n_clusters=4, n_init=20, random_state=56)
```

[28]:
```
kmeans.inertia_
```

[28]:
```
30.539143402608403
```

[29]:
```
kmeans.cluster_centers_
```

[29]:
```
array([[ 0.35873158, -1.07969699],
       [-1.07263754,  0.01882136],
       [ 1.41508637, -0.48994354],
       [-0.71526514,  1.54295579]])
```

[30]:
```
kmeans.n_iter_
```

[30]:
```
8
```

[31]:
```
kmeans.labels_[:10]
```

[31]:
```
array([2, 2, 2, 2, 2, 3, 1, 1, 0, 2], dtype=int32)
```

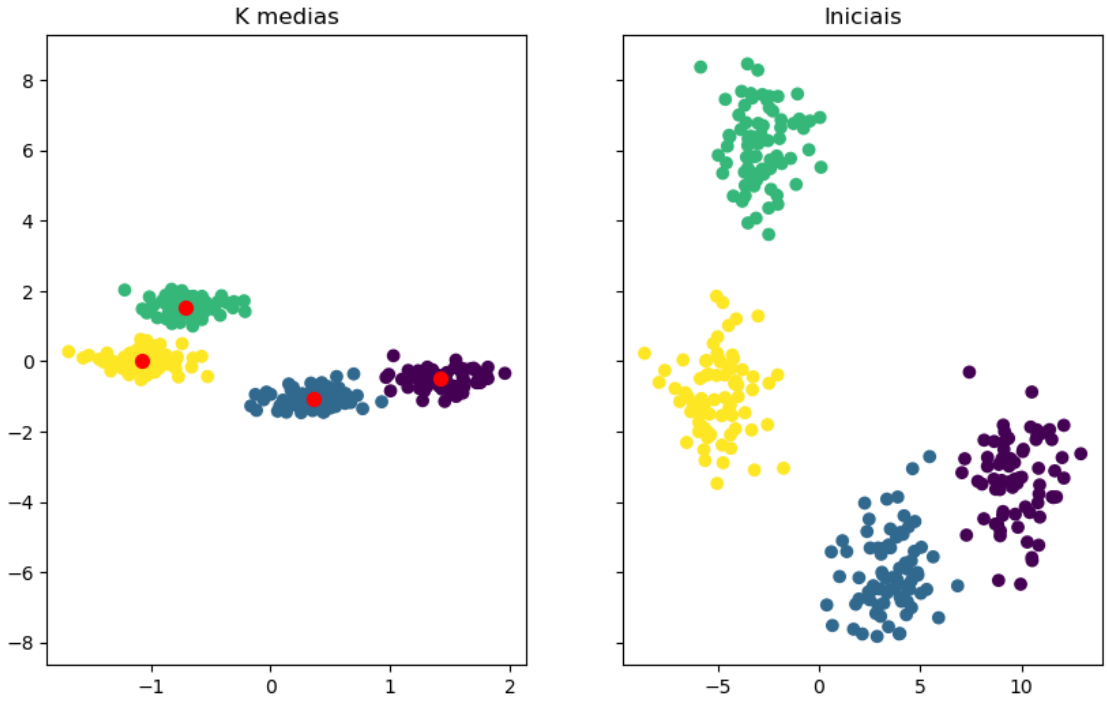
[32]:
```
Labels[:10]
```

[32]:
```
array([0, 0, 0, 0, 0, 2, 3, 3, 1, 0])
```

[35]:
```
f,(eixo1, eixo2)=plt.subplots(1, 2, sharey=True, figsize=(10,6))
eixo1.set_title('K medias')
eixo1.scatter(scalar_caracteristicas[:,0],
```

```
                scalar_caracteristicas[:,1],
               c=Labels)
eixo1.scatter(kmeans.cluster_centers_[:,0],
                kmeans.cluster_centers_[:,1],
               s=50, c='red', label='Centroides')
eixo2.set_title('Iniciais')
eixo2.scatter(Caracteristicas[:,0],Caracteristicas[:,1], c=Labels)
```

[35]: <matplotlib.collections.PathCollection at 0x7f74e982c4f0>



[36]:
```
from sklearn.metrics import confusion_matrix, classification_report
```

[37]:
```
print(confusion_matrix(Labels, kmeans.labels_))
```

```
[[ 0  0 75  0]
 [74  0  1  0]
 [ 0  0  0 75]
 [ 0 75  0  0]]
```

[39]:
```
print(classification_report(Labels, kmeans.labels_))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      75.0
           1       0.00      0.00      0.00      75.0
```

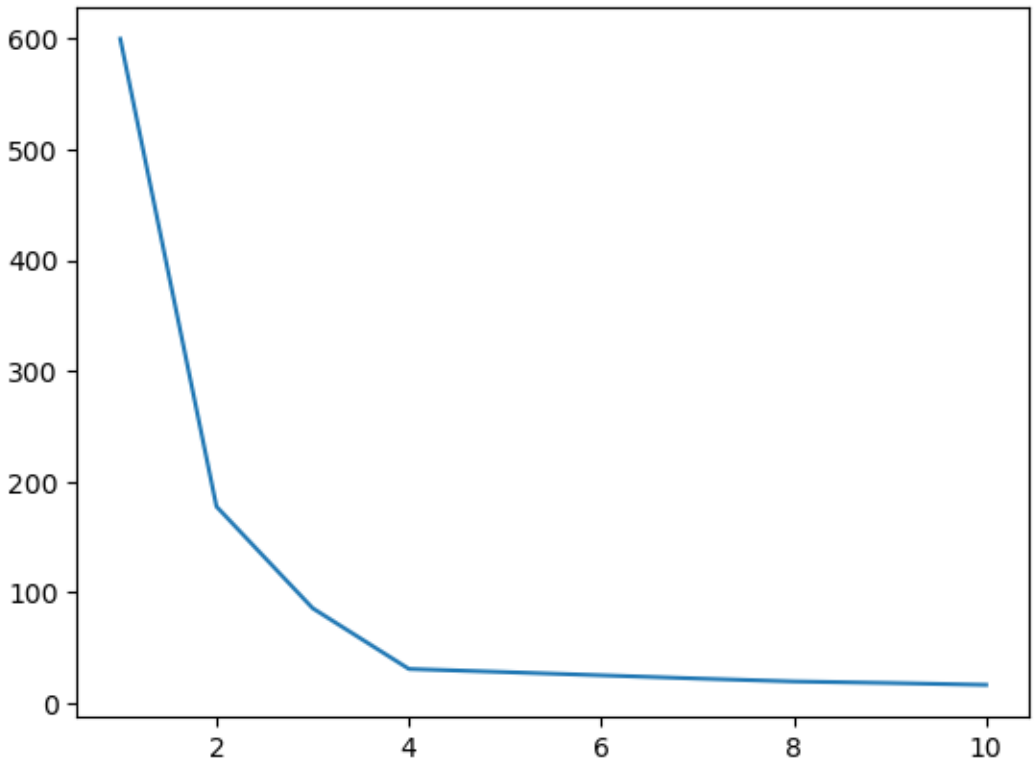
```
              2       0.00      0.00      0.00      75.0
              3       0.00      0.00      0.00      75.0

    accuracy                            0.00     300.0
   macro avg       0.00      0.00      0.00     300.0
weighted avg       0.00      0.00      0.00     300.0
```

[42]:
```
kmeans_valores = {
    'init':'random',
    'n_init':10,
    'max_iter':300,
    'random_state':56,
}
```

[43]:
```
SER=[]
for k in range (1,11):
    kmeansCT = KMeans(n_clusters=k, **kmeans_valores)
    kmeansCT.fit(scalar_caracteristicas)
    SER.append(kmeansCT.inertia_)
```

[44]:
```
plt.plot(range(1,11), SER)
```

[44]:
```
[<matplotlib.lines.Line2D at 0x7f74e035c430>]
```



[ ]: