|  |
| --- |
|  |
| # coding: utf-8 |
|  |  |
|  | # # K-Means Clustering |
|  |  |
|  | # In[1]: |
|  |  |
|  | import pandas as pd |
|  | import numpy as np |
|  | import matplotlib.pyplot as plt |
|  | get\_ipython().magic('matplotlib inline') |
|  |  |
|  |  |
|  | # In[2]: |
|  |  |
|  | data=pd.DataFrame({'x':[12,24,28,33,18,29,52,45,24,55,51,61,53,69,72,64,49,58], |
|  | 'y':[36,39,30,52,54,46,55,59,63,70,66,63,58,23,14,8,19,7]}) |
|  |  |
|  |  |
|  | # In[3]: |
|  |  |
|  | data |
|  |  |
|  |  |
|  | # In[4]: |
|  |  |
|  | k=3 |
|  | centroids={i+1: [np.random.randint(0,80), |
|  | np.random.randint(0,80)] for i in range(k)} |
|  |  |
|  |  |
|  | # In[5]: |
|  |  |
|  | centroids |
|  |  |
|  |  |
|  | # In[6]: |
|  |  |
|  | plt.scatter(data['x'],data['y'],color='k') |
|  | colmap = {1:'r',2:'g',3:'b'} |
|  |  |
|  | for i in centroids.keys(): |
|  | plt.scatter(\*centroids[i]) |
|  | plt.show() |
|  |  |
|  |  |
|  | # In[11]: |
|  |  |
|  | #Assignment Phase |
|  | def assignment(data,centroids): |
|  | for i in centroids.keys(): |
|  | #sqrt((x1-x2)^2 + (y1-y2)^2) |
|  | data['distance\_from\_{}'.format(i)]=(np.sqrt((data['x']-centroids[i][0]) \*\* 2 + |
|  | (data['y']-centroids[i][1]) \*\* 2)) |
|  |  |
|  | data['closest']=data.loc[:,'distance\_from\_1':'distance\_from\_3'].idxmin(axis=1) |
|  | data['closest']=data['closest'].map(lambda x: int(x.strip('distance\_from\_'))) |
|  | data['color']=data['closest'].map(lambda x: colmap[x]) |
|  | return data |
|  | data = assignment(data,centroids) |
|  | print(data.head()) |
|  |  |
|  |  |
|  |  |
|  | # In[8]: |
|  |  |
|  | a='distance\_from\_2' |
|  | a.lstrip('distance\_from\_') |
|  |  |
|  |  |
|  | # In[14]: |
|  |  |
|  | plt.scatter(data['x'],data['y'],color=data['color']) |
|  | for i in centroids.keys(): |
|  | plt.scatter(\*centroids[i],color=colmap[i]) |
|  | plt.show() |
|  |  |
|  |  |
|  | # In[16]: |
|  |  |
|  | #Update Stage |
|  | import copy |
|  | old\_centeroids = copy.deepcopy(centroids) |
|  |  |
|  | def update(k): |
|  | for i in centroids.keys(): |
|  | centroids[i][0] = np.mean(data[data['closest'] == i]['x']) |
|  | centroids[i][1] = np.mean(data[data['closest'] == i]['y']) |
|  |  |
|  | return k |
|  |  |
|  | centroids = update(centroids) |
|  |  |
|  |  |
|  | # In[19]: |
|  |  |
|  | centroids |
|  |  |
|  |  |
|  | # In[22]: |
|  |  |
|  | fig=plt.figure(figsize=(5,5)) |
|  | ax = plt.axes() |
|  | plt.scatter(data['x'],data['y'], edgecolor='k') |
|  | for i in centroids.keys(): |
|  | plt.scatter(\*centroids[i],color=colmap[i]) |
|  | plt.xlim(0,80) |
|  | plt.ylim(0,80) |
|  | for i in centroids.keys(): |
|  | old\_x=old\_centeroids[i][0] |
|  | old\_y=old\_centeroids[i][1] |
|  | dx = (centroids[i][0] - old\_centeroids[i][0]) \* 0.75 |
|  | dy = (centroids[i][1] - old\_centeroids[i][1]) \* 0.75 |
|  | ax.arrow(old\_x, old\_y, dx,dy, head\_width=2, head\_length=3, fc=colmap[i], ec=colmap[i]) |
|  | plt.show() |
|  |  |
|  |  |
|  | # In[23]: |
|  |  |
|  | #Repeat Assignment |
|  |  |
|  | data = assignment(data, centroids) |
|  |  |
|  |  |
|  | # In[25]: |
|  |  |
|  | plt.scatter(data['x'],data['y'],color=data['color']) |
|  | for i in centroids.keys(): |
|  | plt.scatter(\*centroids[i],color=colmap[i]) |
|  | plt.show() |
|  |  |
|  |  |
|  | # In[34]: |
|  |  |
|  | df=pd.DataFrame({'x':[12,24,28,33,18,29,52,45,24,55,51,61,53,69,72,64,49,58], |
|  | 'y':[36,39,30,52,54,46,55,59,63,70,66,63,58,23,14,8,19,7]}) |
|  |  |
|  |  |
|  | # In[35]: |
|  |  |
|  | from sklearn.cluster import KMeans |
|  | kmeans=KMeans(n\_clusters=3) |
|  | kmeans.fit(df) |
|  |  |
|  |  |
|  | # In[36]: |
|  |  |
|  | labels=kmeans.predict(df) |
|  |  |
|  |  |
|  | # In[37]: |
|  |  |
|  | labels |
|  |  |
|  |  |
|  | # In[38]: |
|  |  |
|  | centroids =kmeans.cluster\_centers\_ |
|  |  |
|  |  |
|  | # In[39]: |
|  |  |
|  | centroids |
|  |  |