import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

np.random.seed(200)

k = 3

# centroids[i] = [x, y]

centroids = {

i+1: [np.random.randint(0, 80), np.random.randint(0, 80)]

for i in range(k)

}

path="/Users/shushruthsheshadri/Desktop/data4.csv"

headernames = ['age', 'gender', 'bpm', 'result']

dataset = pd.read\_csv(path, names = headernames)

dataset.head()

fig = plt.figure(figsize=(5, 5))

plt.scatter(dataset['age'], dataset['bpm'], color='k')

colmap = {1: 'r', 2: 'g', 3: 'b'}

for i in centroids.keys():

plt.scatter(\*centroids[i], color=colmap[i])

plt.xlim(0, 80)

plt.ylim(0, 80)

plt.show()

def assignment(dataset, centroids):

for i in centroids.keys():

# sqrt((x1 - x2)^2 - (y1 - y2)^2)

dataset['distance\_from\_{}'.format(i)] = (

np.sqrt(

(dataset['age'] - centroids[i][0]) \*\* 2

+ (dataset['bpm'] - centroids[i][1]) \*\* 2

)

)

centroid\_distance\_cols = ['distance\_from\_{}'.format(i) for i in centroids.keys()]

dataset['closest'] = dataset.loc[:, centroid\_distance\_cols].idxmin(axis=1)

dataset['closest'] = dataset['closest'].map(lambda x: int(x.lstrip('distance\_from\_')))

dataset['color'] = dataset['closest'].map(lambda x: colmap[x])

return dataset

dataset = assignment(dataset, centroids)

print(dataset.head())

fig = plt.figure(figsize=(5, 5))

plt.scatter(dataset['age'], dataset['bpm'], color=dataset['color'], alpha=0.5, edgecolor='k')

for i in centroids.keys():

plt.scatter(\*centroids[i], color=colmap[i])

plt.xlim(0, 80)

plt.ylim(0, 80)

plt.show()

import copy

old\_centroids = copy.deepcopy(centroids)

def update(k):

for i in centroids.keys():

centroids[i][0] = np.mean(dataset[dataset['closest'] == i]['age'])

centroids[i][1] = np.mean(dataset[dataset['closest'] == i]['bpm'])

return k

centroids = update(centroids)

fig = plt.figure(figsize=(5, 5))

ax = plt.axes()

plt.scatter(dataset['age'], dataset['bpm'], color=dataset['color'], alpha=0.5, 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 old\_centroids.keys():

old\_x = old\_centroids[i][0]

old\_y = old\_centroids[i][1]

dx = (centroids[i][0] - old\_centroids[i][0]) \* 0.75

dy = (centroids[i][1] - old\_centroids[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()

dataset = assignment(dataset, centroids)

# Plot results

fig = plt.figure(figsize=(5, 5))

plt.scatter(dataset['age'], dataset['bpm'], color=dataset['color'], alpha=0.5, edgecolor='k')

for i in centroids.keys():

plt.scatter(\*centroids[i], color=colmap[i])

plt.xlim(0, 80)

plt.ylim(0, 80)

plt.show()

while True:

closest\_centroids = dataset['closest'].copy(deep=True)

centroids = update(centroids)

dataset = assignment(dataset, centroids)

if closest\_centroids.equals(dataset['closest']):

break

fig = plt.figure(figsize=(5, 5))

plt.scatter(dataset['age'], dataset['bpm'], color=dataset['color'], alpha=0.5, edgecolor='k')

for i in centroids.keys():

plt.scatter(\*centroids[i], color=colmap[i])

plt.xlim(0, 80)

plt.ylim(0, 80)

plt.show()

dataset = pd.read\_csv(path, names = headernames)

dataset.head()

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=3)

kmeans.fit(dataset)