

# Lloyd's K means algorithm

```
from matplotlib.pyplot import *
from sklearn.datasets import load_digits
import numpy as np
dig = load_digits()
def kmean(data, k, updates=100):
    cost = []
    n = data.shape[0]
    # randomly initialize k centroids
    centroids = data[np.random.choice(n, k, replace=False)]
    for _ in range(updates):
        # assign each point to the closest centroid
        distances = np.linalg.norm(data[:, np.newaxis, :] - centroids, axis=-1)
        labels = np.argmin(distances, axis=1)
        #calculating cost
        c = (np.min(distances,axis=1))**2
        cost.append(np.sum(c))
        # update centroids
        for i in range(k):
            centroids[i] = np.mean(data[labels == i], axis=0)
    return centroids, labels, cost
updates = 100
C,_,cost = kmean(dig["data"],10,updates)
for i in range(10):
    subplot(2,5,i+1)
    im = np.split(C[i],8)
    imshow(im,"gray_r")
suptitle(f"Centers after {updates} updates")
figure()
print(cost)
plot(np.arange(updates),cost)
title("Cost vs iterations")
xlabel("iterations")
ylabel("cost")
show()
```

Centers after 100 updates

