

K-Means General Algorithm

```
# Clustering-k-Means algoirthm on a randomly generated dataset

import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.datasets.samples_generator import make_blobs

#Setup random seed
np.random.seed(0)

# making random clusters
## n_samples= total data points, centers= no. of centers to generate, cluster_std=standard deviation

X,y=make_blobs(n_samples=5000,centers=[[4,4],[-2,-1],[2,-3],[1,1]],cluster_std=0.9)

#Scatter plot of randomly generated data
plt.scatter(X[:,0],X[:,1],marker='.')

# setting up k-means
## init= initialization methods, n_clusters = no. of clusters to form, n_init=no. of times to run the algorithm

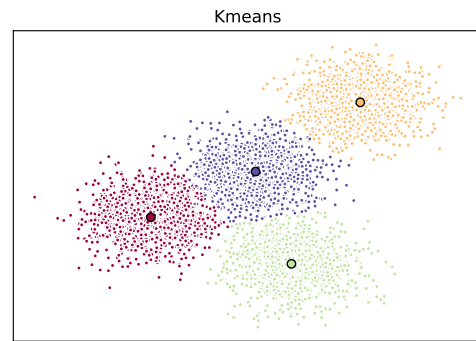
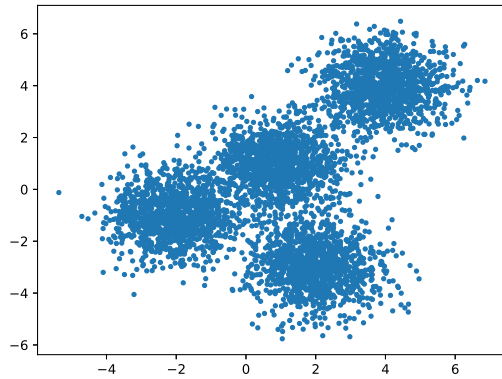
k_means=KMeans(init='k-means++',n_clusters=4,n_init=12)
k_means.fit(X)

k_means_labels=k_means.labels_ #labels for each point in the model using KMeans' .labels_
k_means_cluster_centers = k_means.cluster_centers_ #coordinates of the cluster centers using KMeans' .cluster_centers_
with open('Kmeans_general.txt','a') as f:
    print(k_means_cluster_centers,file=f)
    print(k_means_labels,file=f)

# Creating visula plot for clusters using k-means #

fig=plt.figure(figsize=(6,4))
colors=plt.cm.Spectral(np.linspace(0,1,len(set(k_means_labels))))
ax=fig.add_subplot(1,1,1)

for k, col in zip(range(len([[4,4],[-2,-1],[2,-3],[1,1]])),colors):
    members=(k_means_labels==k)
    cluster_center=k_means_cluster_centers[k]
```



```
ax.plot(X[members,0],X[members,1], 'w',markerfacecolor=col,marker='.')
ax.plot(cluster_center[0],cluster_center[1], 'o',markerfacecolor=col,markeredgecolor=
ax.set_title('Kmeans')
ax.set_xticks(())
ax.set_yticks(())

#Display plot
plt.show()
```

Solution:

```
[[-2.03375169 -0.99827293]
 [ 3.97334234  3.98758687]
 [ 1.99876902 -3.01796355]
 [ 0.96959198  0.98543802]]
[0 2 2 ... 1 0 0]
```