```
    label='centroids'
)
plt.legend()
plt.show()

dist=[]
for i in range(1,11):
    km=KMeans(n_clusters=i,init='k-means++',n_init=10,max_iter=100)
    km.fit(x)
    dist.append(km.inertia_)
plt.plot(range(1,11),dist,marker='o')
plt.xlabel('Number of clusters (K)-->')
plt.ylabel('Distortions')
plt.show()

# Agglomerative
x=np.array([[1,2],[1,4],[1,0],[4,2],[4,4],[4,0]])
plt.scatter(x[:,0],x[:,1])
plt.show()
ac=AgglomerativeClustering(n_clusters=2,metric='euclidean',linkage='single')
 ↪  # 'complete'
labels=ac.fit_predict(x)
print(labels)

x=np.array([[1,2],[1,4],[1,0],[4,2],[4,4],[4,0]])
z=linkage(x,'single') # 'complete'
dendrogram(z)
plt.title('Hierarchical clustering')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()

x,y=make_moons(n_samples=200,noise=0.05,random_state=0)
plt.scatter(x[:,0],x[:,1])
plt.show()

km=KMeans(n_clusters=2,random_state=0)
y_km=km.fit_predict(x)
plt.scatter(x[y_km==0,0],x[y_km==0,1],c='blue',s=50,marker='o')
plt.scatter(x[y_km==1,0],x[y_km==1,1],c='red',s=50,marker='d')
plt.scatter(
    km.cluster_centers_[:,0],
    km.cluster_centers_[:,1],
    c='black',s=50,marker='*'
)
plt.show()

km=AgglomerativeClustering(n_clusters=2,linkage='single') #'compete
```