

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

df=pd.read_csv("/content/Iris.csv",header="infer").values

x=df[:,1:-1]
y=df[:, -1]

test_split=float(input("Enter the test_split in 0 to 1 :- "))

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=test_split,stratify=

k=int(input("Enter the number of clusters you want :- "))
n=int(input("Enter the number of iterations :- "))

centroids=np.zeros(shape=(k,x_train.shape[1]))
per=np.random.permutation(x_train.shape[0])

for i in range(k):
    centroids[i,:]=x_train[per[i],:]

for it in range(n):
    dist=np.zeros(shape=(k,x_train.shape[0]))
    for i in range(k):
        dist[i,:]=np.sqrt(np.sum((x_train-centroids[i,:])**2,axis=1))
    membership=[np.argmin(dist,axis=0)]
    #for i in range(k):
        #centroids[i,:]=np.mean(x_train[membership==i,:],axis=0)

print("Centroids after ",n," iterations :- ")
print(centroids)

dist=np.zeros(shape=(k,x_test.shape[0]))
for i in range(k):
    dist[i]=np.sqrt(np.sum((x_test-centroids[i])**2,axis=1))
    membership=np.argmin(dist,axis=0)

print(y_test.astype(int))
print(membership)

```

```

Enter the test_split in 0 to 1 :- 0.2
Enter the number of clusters you want :- 3
Enter the number of iterations :- 2
Centroids after 2 iterations :-
[[5.  3.4 1.5 0.2]
 [6.5 3.  5.5 1.8]
 [5.6 3.  4.1 1.3]]
[1 0 0 2 0 1 2 1 1 0 0 0 2 1 0 2 2 2 2 0 1 1 2 2 1 0 0 1 2 1]
[1 0 0 1 0 2 1 2 2 0 0 0 1 2 0 1 1 1 1 0 2 2 1 1 2 0 0 2 1 2]

```

✓ 7s completed at 7:21 PM

● ×