```
import math
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import numpy as numpy
```

```
Data= pd.read_csv("bank.csv", sep=None, engine="python")
Data.head()
```

	age	job	marital	education	default	balance	housing	loan	contact	da
0	58	management	married	tertiary	no	2143	yes	no	unknown	
1	44	technician	single	secondary	no	29	yes	no	unknown	
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	
4	33	unknown	single	unknown	no	1	no	no	unknown	

```
cols= ["age","balance","day","duration","campaign","pdays","previous"]
data_encode= Data.drop(cols, axis= 1)
data_encode= data_encode.apply(LabelEncoder().fit_transform)
data_rest= Data[cols]
Data= pd.concat([data_rest,data_encode], axis= 1)
```

```
Data.head()
```

	age	balance	day	duration	campaign	pdays	previous	job	marital	education
0	58	2143	5	261	1	-1	0	4	1	2
1	44	29	5	151	1	-1	0	9	2	1
2	33	2	5	76	1	-1	0	2	1	1
3	47	1506	5	92	1	-1	0	1	1	3
4	33	1	5	198	1	-1	0	11	2	3

```
data_train, data_test= train_test_split(Data, test_size= 0.33, random_state= 4)
X_train= data_train.drop("y", axis= 1)
Y_train= data_train["y"]
X_test= data_test.drop("y", axis=1)
Y_test= data_test["y"]
```

```
scaler= StandardScaler()
scaler.fit(X_train)
```

```
X_train= scaler.transform(X_train)
X_test= scaler.transform(X_test)



K_cent= 8
km= KMeans(n_clusters= K_cent, max_iter= 100)
km.fit(X_train)
cent= km.cluster_centers_


max=0
for i in range(K_cent):
    for j in range(K_cent):
        d= numpy.linalg.norm(cent[i]-cent[j])
        if(d> max):
            max= d
d= max
sigma= d/math.sqrt(2*K_cent)


shape= X_train.shape
row= shape[0]
column= K_cent
G= numpy.empty((row,column), dtype= float)
for i in range(row):
    for j in range(column):
        dist= numpy.linalg.norm(X_train[i]-cent[j])
        G[i][j]= math.exp(-math.pow(dist,2)/math.pow(2*sigma,2))


GTG= numpy.dot(G.T,G)
GTG_inv= numpy.linalg.inv(GTG)
fac= numpy.dot(GTG_inv,G.T)
W= numpy.dot(fac,Y_train)


row= X_test.shape[0]
column= K_cent
G_test= numpy.empty((row,column), dtype= float)
for i in range(row):
    for j in range(column):
        dist= numpy.linalg.norm(X_test[i]-cent[j])
        G_test[i][j]= math.exp(-math.pow(dist,2)/math.pow(2*sigma,2))



prediction= numpy.dot(G_test,W)
prediction= 0.5*(numpy.sign(prediction-0.5)+1)
score= accuracy_score(prediction,Y_test)
print("Accuracy: ", score.mean())
```

```
Accuracy:  0.8876675603217158
```