

In [1]:

```
import numpy as np
import pandas
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import NearestCentroid
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

In [2]:

```
X,Y = load_digits(return_X_y=True)
classes = load_digits().target_names
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.30, random_state=0
)
```

In [3]:

```
class myKNeighborsClassifier:
    def __init__(self, k=5):
        self.k = k
        self.Xtrain = None
        self.Ytrain = None
    def fit(self,Xtrain,Ytrain):
        self.Xtrain = Xtrain
        self.Ytrain = Ytrain
    def predict(self,X):
        #1. Calcular la distancia de Los datos X a Los datos Xtrain
        #2. Ordenar Las distancias de menor a mayor
        #3. Calcular La clase de acuerdo a La mayoría de Los k vecinos más cercanos
        #4. Regresar valores
        pass
```

In [4]:

```

model = KNeighborsClassifier()
model.fit(Xtrain,Ytrain)
Ypred = model.predict(Xtest)

print('Accuracy',accuracy_score(Ytest,Ypred))
m = confusion_matrix(Ytest,Ypred)
df = pandas.DataFrame(m,index=classes,columns=classes)
df

```

Accuracy 0.9814814814814815

Out[4]:

	0	1	2	3	4	5	6	7	8	9
0	45	0	0	0	0	0	0	0	0	0
1	0	51	0	0	0	1	0	0	0	0
2	0	0	52	0	0	0	0	1	0	0
3	0	0	1	53	0	0	0	0	0	0
4	0	0	0	0	47	0	0	1	0	0
5	0	0	0	0	0	55	1	0	0	1
6	0	0	0	0	0	0	60	0	0	0
7	0	0	0	0	0	0	0	53	0	0
8	0	1	0	1	0	0	1	0	58	0
9	0	0	0	0	0	1	0	0	0	56

In [5]:

```

class myNearestCentroid:
    def __init__(self):
        self.centroids = None

    def fit(self,Xtrain,Ytrain):
        Xtrain = Xtrain
        Ytrain = Ytrain
        # 1. calcular el número de clases
        # 2. calcular los centroides de cada clase y guardarlo en self.centroids
    def predict(self,X):
        # 1. calcular la distancia de cada muestra en X a cada uno de los centroides
        # 2. asignar la clase de acuerdo al centroide más cercano
        pass

```

In [6]:

```

model = NearestCentroid()
model.fit(Xtrain,Ytrain)
Ypred = model.predict(Xtest)

print('Accuracy',accuracy_score(Ytest,Ypred))
m = confusion_matrix(Ytest,Ypred)
df = pandas.DataFrame(m,index=classes,columns=classes)
df

```

Accuracy 0.8925925925925926

Out[6]:

	0	1	2	3	4	5	6	7	8	9
0	45	0	0	0	0	0	0	0	0	0
1	0	42	1	0	0	1	1	0	1	6
2	1	2	44	3	0	0	0	2	0	1
3	0	0	1	44	0	0	0	2	2	5
4	0	1	0	0	44	0	0	3	0	0
5	0	0	0	0	0	47	1	0	0	9
6	0	1	0	0	0	0	59	0	0	0
7	0	0	0	0	0	0	0	52	1	0
8	0	3	0	1	0	1	0	1	51	4
9	0	0	0	0	0	1	0	2	0	54