In [88]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from collections import Counter
```

In [89]:

```
data=pd.read_csv('diabetes.csv')
data.head()
```

Out[89]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [90]:

```
not_zero=['Glucose','BloodPressure','SkinThickness','Insulin','BMI']
for colum in not_zero:
    mean=int(data[colum].mean())
    data[colum]=data[colum].replace(0,mean)
```

In [91]:

```
x=data.iloc[:,:8]
y=data.iloc[:,8]
```

In [92]:

```
X = data.to_numpy()
X = X[:,:8]

Y = data.to_numpy()
Y = Y[:,8]
```

In [93]:

```
def numpy_distance(x,y):
    return np.linalg.norm(x-y)
```

In [95]:

```
X_train, X_test, y_train, y_test = train_test_split( X, Y, test_size = 0.3)
k=11
```

In [98]:

```
prediction = []
coorect_count = 0
for i in range(len(X_test)):
        distances = []
        for j in range(len(X_train)):
            dist = numpy_distance(X_test[i], X_train[j])
            distances.append((X_train[j], dist, y_train[j]))
        distances.sort(key=lambda x: x[1])
        neighbors = distances[:k]
        class_counter = Counter()
        for neighbor in neighbors:
            class_counter[neighbor[2]] += 1
        prediction.append(class_counter.most_common(1)[0][0])
        if(y_test[i] == prediction[i]):
            coorect_count = coorect_count + 1
acc = coorect_count/float(len(X_test))
```

In [99]:

```
acc
```

Out[99]:

```
0.7056277056277056
```

In [ ]: