Program 4:

Aim:Program to implement K-NN classification using any random dataset without using inbuilt packages.

Program:

```
from math import sqrt
def euclidean_distance(row1,row2):
    distance = 0.0
    for i in range(len(row1) - 1):
        distance += (row1[i] - row2[i] )**2
        return sqrt(distance)

def get_neighbors(train,test_row, num_neighbors):
    distances = list()
    for train_row in train:
        dist =euclidean_distance(test_row, train_row)
        distances.append((train_row,dist))
        distances.sort(key=lambda  tup:tup[1])
        neighbors = list()
        for i in range(num_neighbors):
            neighbors.append(distances[i][0])
            return neighbors
def predict_classification(train,test_row,
num_neighbors):
    neighbors = get_neighbors(train, test_row,
num_neighbors)
    output_values = [row[-1] for row in neighbors]
    prediction = max(set(output_values),
key=output_values.count)
    return prediction
dataset = [[2.7810836, 2.550537003,0],
          [1.465458936,2.64785645,0],
          [3.56789536,4.568555858,0],
          [1.468956556,3.1464756654,0],
          [5.135663212,2.621254545,0],
          [6.2545449552,5.1436870564,1],
          [8.4365631212,7.56655252636,1],
          [2.146589696,5.66655665555,1],
           [3.4664565252,5.46558866,1],
           [5.895525255,3.46565858,1]]
```

```
prediction = predict_classification(dataset,dataset[0],
5)
print('expected %d, Got %d. ' % (dataset[0][-1],
prediction))
```

Output: