# DATA WAREHOUSE AND DATA MINING

# LAB DA-6

NAME: HRITHIK HEM SUNDAR.B

REGNO: 19MID0021

**1.SINGLE LINK CLUSTERING**

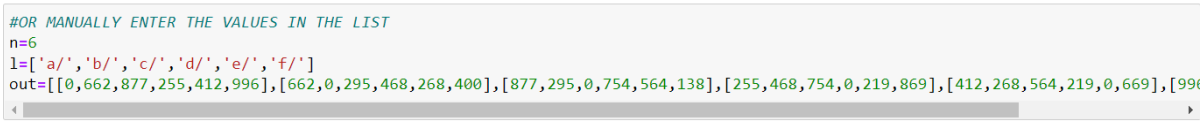
**CODE:**

```
n=6
l=['a/','b/','c/','d/','e/','f/']
out=[[0,662,877,255,412,996],[662,0,295,468,268,400],[877,295,0,754,564,138],[255
,468,754,0,219,869],[412,268,564,219,0,669],[996,400,138,869,669,0]]
import pandas as pd
print("\n")
print("Input Distance Matrix")
print("\n")
df=pd.DataFrame(out)
print(df)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
```

```
    c=[]
d=[]
g=[]
e=[]
h=[]
for i in range(len(minr)):
    d.append(min(newr[i]))
x=min(d)
for i in range(len(minr)):
    if(min(newr[i])==x):
        h.append(i)
for i in range(len(h)):
    for j in range(len(minr)):
        if(minr[h[i]][j]==x):
            g.append((h[i],j))
f=[]
ab=len(g)/2
for i in range(int(ab)):
    k=(g[i][1],g[i][0])
    del g[g.index(k)]
lar=[]
for i in range(int(ab)):
    lar.append(max(g[i][0],g[i][1]))
lar.sort(reverse=True)
for i in range(int(ab)):
    for j in range(int(ab)):
        if(g[j][1]==lar[i]):
            e.append(g[j])
for i in range(len(e)):
    for j in range(len(minr)):
```

```
            y=min(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
        minr[e[i][0]]=f
        l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
        f=[]
    ds=pd.DataFrame(minr)
    for i in range(len(e)):
         for j in range(len(minr)):
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
    for i in range(len(e)):
        del minr[e[i][1]]
        del l[e[i][1]]
    print(l)
    df=pd.DataFrame(minr)
    print(df)
    n=n-1
    s=s+1
```

**SCREENSHOT WITH OUTPUT:**



```
import pandas as pd
print("\n")
print("Input Distance Matrix")
print("\n")
df=pd.DataFrame(out)
print(df)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
        c=[]
    d=[]
    g=[]
    e=[]
    h=[]
    for i in range(len(minr)):
        d.append(min(newr[i]))
    x=min(d)
    for i in range(len(minr)):
        if(min(newr[i])==x):
            h.append(i)
```

```
    for i in range(len(h)):
        for j in range(len(minr)):
            if(minr[h[i]][j]==x):
                g.append((h[i],j))
    f=[]
    ab=len(g)/2
    for i in range(int(ab)):
        k=(g[i][1],g[i][0])
        del g[g.index(k)]
    lar=[]
    for i in range(int(ab)):
        lar.append(max(g[i][0],g[i][1]))
    lar.sort(reverse=True)
    for i in range(int(ab)):
        for j in range(int(ab)):
            if(g[j][1]==lar[i]):
                e.append(g[j])
    for i in range(len(e)):
        for j in range(len(minr)):
            y=min(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
        minr[e[i][0]]=f
        l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
        f=[]
    ds=pd.DataFrame(minr)
    for i in range(len(e)):
         for j in range(len(minr)):
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
    for i in range(len(e)):
        del minr[e[i][1]]
        del l[e[i][1]]
```

```
print(l)
df=pd.DataFrame(minr)
print(df)
n=n-1
s=s+1
```

```
Input Distance Matrix


     0    1    2    3    4    5
0    0  662  877  255  412  996
1  662    0  295  468  268  400
2  877  295    0  754  564  138
3  255  468  754    0  219  869
4  412  268  564  219    0  669
5  996  400  138  869  669    0


Iteration  1


['a/', 'b/', 'c/f/', 'd/', 'e/']
     0    1    2    3    4
0    0  662  877  255  412
1  662    0  295  468  268
2  877  295    0  754  564
3  255  468  754    0  219
4  412  268  564  219    0
```

```
Iteration  2


['a/', 'b/', 'c/f/', 'd/e/']
     0    1    2    3
0    0  662  877  255
1  662    0  295  268
2  877  295    0  564
3  255  268  564    0


Iteration  3


['a/d/e/', 'b/', 'c/f/']
     0    1    2
0    0  268  564
1  268    0  295
2  564  295    0


Iteration  4


['a/d/e/b/', 'c/f/']
     0    1
0    0  295
1  295    0
```

**COMPLETE LINK CLUSTERING**

**CODE:**

```
n=8
l=['1/','2/','3/','4/','5/','6/','7/','8/']
out=[[0.0, 5.0, 8.4, 3.6, 7.07, 7.21, 8.06, 2.23], [5.0, 0.0, 6.08, 4.24, 5.0, 4.12, 3.16, 4.47], [8.4, 6.08, 0.0, 5.0, 1.41, 2.0, 7.28, 6.4],
[3.6, 4.24, 5.0, 0.0, 3.6, 4.12, 7.21, 1.41],
[7.07, 5.0, 1.41, 3.6, 0.0, 1.41, 6.7, 5.0],
[7.21, 4.12, 2.0, 4.12, 1.41, 0.0, 5.38, 5.38],
[8.06, 3.16, 7.28, 7.21, 6.7, 5.38, 0.0, 7.61],
[2.23, 4.47, 6.4, 1.41, 5.0, 5.38, 7.61, 0.0]]
import pandas as pd
print("\n")
print("Input Distance Matrix")
print("\n")
df1=pd.DataFrame(out)
print(df1)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
        c=[]
```

```
d=[]
g=[]
e=[]
h=[]
for i in range(len(minr)):
    d.append(min(newr[i]))
x=min(d)
for i in range(len(minr)):
    if(min(newr[i])==x):
        h.append(i)
for i in range(len(h)):
    for j in range(len(minr)):
        if(minr[h[i]][j]==x):
            g.append((h[i],j))
f=[]
ab=len(g)/2
for i in range(int(ab)):
    k=(g[i][1],g[i][0])
    del g[g.index(k)]
lar=[]
for i in range(int(ab)):
    lar.append(max(g[i][0],g[i][1]))
lar.sort(reverse=True)
for i in range(int(ab)):
    for j in range(int(ab)):
        if(g[j][1]==lar[i]):
            e.append(g[j])
for i in range(len(e)):
    for j in range(len(minr)):
        if(j!=e[i][0]):
```

```
            y=max(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
        else:
            y=min(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
    minr[e[i][0]]=f
    l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
    f=[]
ds=pd.DataFrame(minr)
for i in range(len(e)):
    for j in range(len(minr)):
        if(j!=e[i][0]):
            minr[j][e[i][0]]=max(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
        else:
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
for i in range(len(e)):
    del minr[e[i][1]]
    del l[e[i][1]]
print(l)
df=pd.DataFrame(minr)
print(df)
n=n-1
s=s+1
```

**SCREENSHOT WITH OUTPUT:**

```
import pandas as pd
print("\n")
print("Input Distance Matrix")
print("\n")
df=pd.DataFrame(out)
print(df)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
        c=[]
    d=[]
    g=[]
    e=[]
    h=[]
    for i in range(len(minr)):
        d.append(min(newr[i]))
    x=min(d)
    for i in range(len(minr)):
        if(min(newr[i])==x):
            h.append(i)
```

```
for i in range(len(h)):
    for j in range(len(minr)):
        if(minr[h[i]][j]==x):
            g.append((h[i],j))
f=[]
ab=len(g)/2
for i in range(int(ab)):
    k=(g[i][1],g[i][0])
    del g[g.index(k)]
lar=[]
for i in range(int(ab)):
    lar.append(max(g[i][0],g[i][1]))
lar.sort(reverse=True)
for i in range(int(ab)):
    for j in range(int(ab)):
        if(g[j][1]==lar[i]):
            e.append(g[j])
for i in range(len(e)):
    for j in range(len(minr)):
        if(j!=e[i][0]):
            y=max(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
        else:
            y=min(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
    minr[e[i][0]]=f
    l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
    f=[]
ds=pd.DataFrame(minr)
```

```
for i in range(len(e)):
     for j in range(len(minr)):
        if(j!=e[i][0]):
            minr[j][e[i][0]]=max(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
        else:
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
for i in range(len(e)):
    del minr[e[i][1]]
    del l[e[i][1]]
print(l)
df=pd.DataFrame(minr)
print(df)
n=n-1
s=s+1
```

Input Distance Matrix

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.00 | 5.00 | 8.40 | 3.60 | 7.07 | 7.21 | 8.06 | 2.23 |
| 1 | 5.00 | 0.00 | 6.08 | 4.24 | 5.00 | 4.12 | 3.16 | 4.47 |
| 2 | 8.40 | 6.08 | 0.00 | 5.00 | 1.41 | 2.00 | 7.28 | 6.40 |
| 3 | 3.60 | 4.24 | 5.00 | 0.00 | 3.60 | 4.12 | 7.21 | 1.41 |
| 4 | 7.07 | 5.00 | 1.41 | 3.60 | 0.00 | 1.41 | 6.70 | 5.00 |
| 5 | 7.21 | 4.12 | 2.00 | 4.12 | 1.41 | 0.00 | 5.38 | 5.38 |
| 6 | 8.06 | 3.16 | 7.28 | 7.21 | 6.70 | 5.38 | 0.00 | 7.61 |
| 7 | 2.23 | 4.47 | 6.40 | 1.41 | 5.00 | 5.38 | 7.61 | 0.00 |

Iteration 1

['1/', '2/', '3/5/6/', '4/8/', '7/']

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0.00 | 5.00 | 8.40 | 3.60 | 8.06 |
| 1 | 5.00 | 0.00 | 6.08 | 4.47 | 3.16 |
| 2 | 8.40 | 6.08 | 0.00 | 6.40 | 7.28 |
| 3 | 3.60 | 4.47 | 6.40 | 0.00 | 7.61 |
| 4 | 8.06 | 3.16 | 7.28 | 7.61 | 0.00 |

```
Iteration  2


['1/', '2/7/', '3/5/6/', '4/8/']
      0     1     2     3
0  0.00  8.06  8.40  3.60
1  8.06  0.00  7.28  7.61
2  8.40  7.28  0.00  6.40
3  3.60  7.61  6.40  0.00


Iteration  3


['1/4/8/', '2/7/', '3/5/6/']
      0     1     2
0  0.00  8.06  8.40
1  8.06  0.00  7.28
2  8.40  7.28  0.00


Iteration  4


['1/4/8/', '2/7/3/5/6/']
     0    1
0  0.0  8.4
1  8.4  0.0
```

**3.AVERAGE LINK CLUSTERING**

**CODE:**

```
n=8
l=['1/','2/','3/','4/','5/','6/','7/','8/']
out=[[0.0, 5.0, 8.4, 3.6, 7.07, 7.21, 8.06, 2.23], [5.0, 0.0, 6.08, 4.24, 5.0, 4.12, 3.16, 4.47], [8.4, 6.08, 0.0, 5.0, 1.41, 2.0, 7.28, 6.4],
[3.6, 4.24, 5.0, 0.0, 3.6, 4.12, 7.21, 1.41],
[7.07, 5.0, 1.41, 3.6, 0.0, 1.41, 6.7, 5.0],
[7.21, 4.12, 2.0, 4.12, 1.41, 0.0, 5.38, 5.38],
[8.06, 3.16, 7.28, 7.21, 6.7, 5.38, 0.0, 7.61],
[2.23, 4.47, 6.4, 1.41, 5.0, 5.38, 7.61, 0.0]]
import pandas as pd
from statistics import mean
print("\n")
print("Input Distance Matrix")
print("\n")
df1=pd.DataFrame(out)
print(df1)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
```

```
    c=[]
d=[]
g=[]
e=[]
h=[]
for i in range(len(minr)):
    d.append(min(newr[i]))
x=min(d)
for i in range(len(minr)):
    if(min(newr[i])==x):
        h.append(i)
for i in range(len(h)):
    for j in range(len(minr)):
        if(minr[h[i]][j]==x):
            g.append((h[i],j))
f=[]
ab=len(g)/2
for i in range(int(ab)):
    k=(g[i][1],g[i][0])
    del g[g.index(k)]
lar=[]
for i in range(int(ab)):
    lar.append(max(g[i][0],g[i][1]))
lar.sort(reverse=True)
for i in range(int(ab)):
    for j in range(int(ab)):
        if(g[j][1]==lar[i]):
            e.append(g[j])
for i in range(len(e)):
    for j in range(len(minr)):
```

```
        if(j!=e[i][0]):
            cd=(minr[e[i][0]][j],minr[e[i][1]][j])
            y=mean(cd)
            f.append(y)
        else:
            y=min(minr[e[i][0]][j],minr[e[i][1]][j])
            f.append(y)
    minr[e[i][0]]=f
    l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
    f=[]
ds=pd.DataFrame(minr)
for i in range(len(e)):
     for j in range(len(minr)):
        if(j!=e[i][0]):
            ef=(minr[j][e[i][0]],minr[j][e[i][1]])
            minr[j][e[i][0]]=mean(ef)
            del minr[j][e[i][1]]
        else:
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
for i in range(len(e)):
    del minr[e[i][1]]
    del l[e[i][1]]
print(l)
df=pd.DataFrame(minr)
print(df)
n=n-1
s=s+1
```

## SCREENSHOT WITH OUTPUT:

```
import pandas as pd
from statistics import mean
print("\n")
print("Input Distance Matrix")
print("\n")
df1=pd.DataFrame(out)
print(df1)
minr=out
s=1
while(len(minr[0])!=2):
    print("\n")
    print("Iteration ",s)
    print("\n")
    c=[]
    newr=[]
    for i in range(len(minr)):
        for j in range(len(minr)):
            if(minr[i][j]!=0):
                c.append(minr[i][j])
        newr.append(c)
        c=[]
    d=[]
    g=[]
    e=[]
    h=[]
    for i in range(len(minr)):
        d.append(min(newr[i]))
    x=min(d)
    for i in range(len(minr)):
        if(min(newr[i])==x):
            h.append(i)
```

```
 for i in range(len(h)):
     for j in range(len(minr)):
         if(minr[h[i]][j]==x):
             g.append((h[i],j))
 f=[]
 ab=len(g)/2
 for i in range(int(ab)):
     k=(g[i][1],g[i][0])
     del g[g.index(k)]
 lar=[]
 for i in range(int(ab)):
     lar.append(max(g[i][0],g[i][1]))
 lar.sort(reverse=True)
 for i in range(int(ab)):
     for j in range(int(ab)):
         if(g[j][1]==lar[i]):
             e.append(g[j])
 for i in range(len(e)):
     for j in range(len(minr)):
         if(j!=e[i][0]):
             cd=(minr[e[i][0]][j],minr[e[i][1]][j])
             y=mean(cd)
             f.append(y)
         else:
             y=min(minr[e[i][0]][j],minr[e[i][1]][j])
             f.append(y)
     minr[e[i][0]]=f
     l[e[i][0]]=l[e[i][0]]+l[e[i][1]]
     f=[]
 ds=pd.DataFrame(minr)
```

```
for i in range(len(e)):
     for j in range(len(minr)):
        if(j!=e[i][0]):
            ef=(minr[j][e[i][0]],minr[j][e[i][1]])
            minr[j][e[i][0]]=mean(ef)
            del minr[j][e[i][1]]
        else:
            minr[j][e[i][0]]=min(minr[j][e[i][0]],minr[j][e[i][1]])
            del minr[j][e[i][1]]
for i in range(len(e)):
    del minr[e[i][1]]
    del l[e[i][1]]
print(l)
df=pd.DataFrame(minr)
print(df)
n=n-1
s=s+1
```

```
Input Distance Matrix


      0     1     2     3     4     5     6     7
0  0.00  5.00  8.40  3.60  7.07  7.21  8.06  2.23
1  5.00  0.00  6.08  4.24  5.00  4.12  3.16  4.47
2  8.40  6.08  0.00  5.00  1.41  2.00  7.28  6.40
3  3.60  4.24  5.00  0.00  3.60  4.12  7.21  1.41
4  7.07  5.00  1.41  3.60  0.00  1.41  6.70  5.00
5  7.21  4.12  2.00  4.12  1.41  0.00  5.38  5.38
6  8.06  3.16  7.28  7.21  6.70  5.38  0.00  7.61
7  2.23  4.47  6.40  1.41  5.00  5.38  7.61  0.00


Iteration  1


['1/', '2/', '3/5/6/', '4/8/', '7/']
       0      1       2       3     4
0  0.000  5.000  7.7700  2.9150  8.06
1  5.000  0.000  5.3200  4.3550  3.16
2  7.770  5.320  0.0000  5.1125  6.66
3  2.915  4.355  5.1125  0.0000  7.41
4  8.060  3.160  6.6600  7.4100  0.00
```

```
Iteration  2


['1/4/8/', '2/', '3/5/6/', '7/']
         0       1        2      3
0  0.00000  4.6775  6.44125  7.735
1  4.67750  0.0000  5.32000  3.160
2  6.44125  5.3200  0.00000  6.660
3  7.73500  3.1600  6.66000  0.000


Iteration  3


['1/4/8/', '2/7/', '3/5/6/']
         0        1        2
0  0.00000  6.20625  6.44125
1  6.20625  0.00000  5.99000
2  6.44125  5.99000  0.00000


Iteration  4


['1/4/8/', '2/7/3/5/6/']
         0        1
0  0.00000  6.32375
1  6.32375  0.00000
```

**4.K NEAREST NEIGHBOUR**

**CODE:**

```
import pandas as pd
import math
import statistics
from statistics import mode
df=pd.read_excel("D://VIT/Datasets//Knndataset.xlsx")
del df['Unnamed: 4']
df
from sklearn.preprocessing import LabelEncoder
df['Genderlabel']=LabelEncoder().fit_transform(df.Gender)
df.head()
k=int(input("Enter number of clusters K : "))
a=23
b=0
distance=[]
for i in range(len(df)):
    distance.append(math.sqrt((df['Age'][i]-x)**2+(df['Genderlabel'][i]-y)**2))
print(distance)
df['Distance']=distance
df
sortdist=distance
sortdist.sort()
output=[]
newop=[]
distinct={}
for i in range(3):
    for j in range(len(df)):
        if(df['Distance'][j]==sortdist[i]):
            output.append(df.Sport[j])
```

```
for i in range(k):
    newop.append(output[i])
newop
mode(newop)
```

**SCREENSHOT WITH OUTPUT:**

```
: import pandas as pd
  import math
  import statistics
  from statistics import mode
```

```
: df=pd.read_excel("D://VIT/Datasets//Knndataset.xlsx")
  del df['Unnamed: 4']
```

```
: df
```

:

| | Name | Age | Gender | Sport |
|---|---|---|---|---|
| **0** | ajay | 32 | M | football |
| **1** | mark | 40 | M | neither |
| **2** | sara | 16 | F | cricket |
| **3** | zaira | 34 | F | cricket |
| **4** | sachin | 55 | M | neither |
| **5** | rahul | 40 | M | cricket |
| **6** | pooja | 20 | F | neither |
| **7** | smith | 15 | M | cricket |
| **8** | laxmi | 55 | F | football |
| **9** | michael | 15 | M | football |

```
from sklearn.preprocessing import LabelEncoder
df['Genderlabel']=LabelEncoder().fit_transform(df.Gender)
df.head()
```

| | Name | Age | Gender | Sport | Genderlabel |
|---|---|---|---|---|---|
| 0 | ajay | 32 | M | football | 1 |
| 1 | mark | 40 | M | neither | 1 |
| 2 | sara | 16 | F | cricket | 0 |
| 3 | zaira | 34 | F | cricket | 0 |
| 4 | sachin | 55 | M | neither | 1 |

```
k=int(input("Enter number of clusters K : "))
a=23
b=0
```

```
Enter number of clusters K : 3
```

```
distance=[]
for i in range(len(df)):
    distance.append(math.sqrt((df['Age'][i]-x)**2+(df['Genderlabel'][i]-y)**2))
```

```
print(distance)
```

```
[27.018512172212592, 35.014282800023196, 11.0, 29.0, 50.00999900019995, 35.014282800023196, 15.0, 10.04987562112089, 50.0, 10.0
4987562112089]
```

```
df['Distance']=distance
df
```

| | Name | Age | Gender | Sport | Genderlabel | Distance |
|---|---|---|---|---|---|---|
| 0 | ajay | 32 | M | football | 1 | 27.018512 |
| 1 | mark | 40 | M | neither | 1 | 35.014283 |
| 2 | sara | 16 | F | cricket | 0 | 11.000000 |
| 3 | zaira | 34 | F | cricket | 0 | 29.000000 |
| 4 | sachin | 55 | M | neither | 1 | 50.009999 |
| 5 | rahul | 40 | M | cricket | 1 | 35.014283 |
| 6 | pooja | 20 | F | neither | 0 | 15.000000 |
| 7 | smith | 15 | M | cricket | 1 | 10.049876 |
| 8 | laxmi | 55 | F | football | 0 | 50.000000 |
| 9 | michael | 15 | M | football | 1 | 10.049876 |

```
sortdist=distance
sortdist.sort()
```

```
output=[]
newop=[]
distinct={}
for i in range(3):
    for j in range(len(df)):
        if(df['Distance'][j]==sortdist[i]):
            output.append(df.Sport[j])
for i in range(k):
    newop.append(output[i])
```

```
newop
```

```
['cricket', 'football', 'cricket']
```

```
mode(newop)
```

```
'cricket'
```