

DBSCAN

```
In [2]: import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
```

```
In [12]: train=[]
for i in range(200):
    train.append(np.random.randint(low=1, high=100, size=2))
train = pd.DataFrame(train)
train.columns = ['col1', 'col2']
train.head()
```

```
Out[12]:
```

	col1	col2
0	72	29
1	79	80
2	51	45
3	50	60
4	31	2

```
In [13]: min_points=int(input('enter minimum number of points'))
r=float(input('enter radius'))
enter minimum number of points9
enter radius9
```

```
In [14]: def distance(x1,x2):
    sum1=0
    for i in range(len(x1)):
        sum1+=(x1[i]-x2[i])**2
    return math.sqrt(sum1)
```

```
In [15]: p=train.values
main=list(range(len(n)))
```

```
In [16]: z=[]
#radius=9
for k in range(len(main)):
    k=main[0]
    index=[]
    for i in range(len(p)):
        if distance(p[k],p[i])<r:
            index.append(i)
    for i in index:
        for j in range(len(p)):
            if distance(p[i],p[j])<r:
                if j not in index:
                    index.append(j)
    z.append(index)
    index=set(index)
    main=list(set(main)-index)
    main = main.copy()
    if len(main)==0:
        break
```

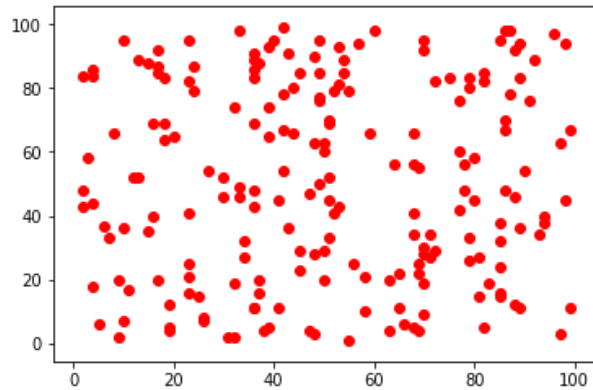
```
In [17]: len(z)
```

```
Out[17]: 14
```

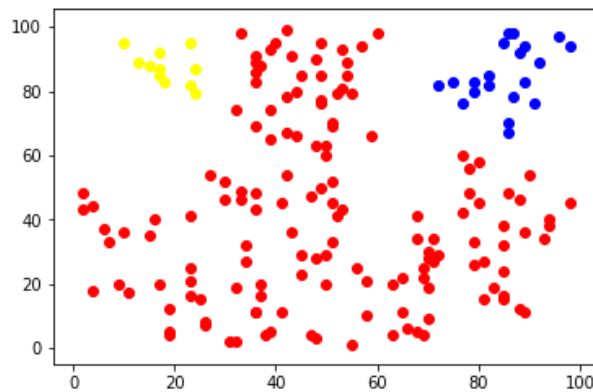
remove outlier

```
In [18]: cluster=[]
         for i in z:
             if len(i)>min_ponints:
                 cluster.append(i)
```

```
In [19]: for i in range(len(train)):
         x=train.iloc[i:i+1,0:1].values
         y=train.iloc[i:i+1,1:2].values
         plt.scatter(x[0], y[0], c='r', label='data')
         plt.show()
```



```
In [20]: color=['red','blue','yellow','green','cyan','purple','magenta','burlywood','cha
         count=0
         for i in cluster:
             for j in i:
                 x=train.iloc[j:j+1,0:1].values
                 y=train.iloc[j:j+1,1:2].values
                 x=x[0]
                 y=y[0]
                 plt.scatter(x[0], y[0], color=color[count])
                 count+=1
         plt.show()
```



In []:

In []: