МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по практической работе №6

по дисциплине «Машинное обучение»

Студент гр. 1310 _____ Комаров Д. Е.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2025

Задание 1

Постановка задачи

Дан набор данных, представленный на рисунке 1.

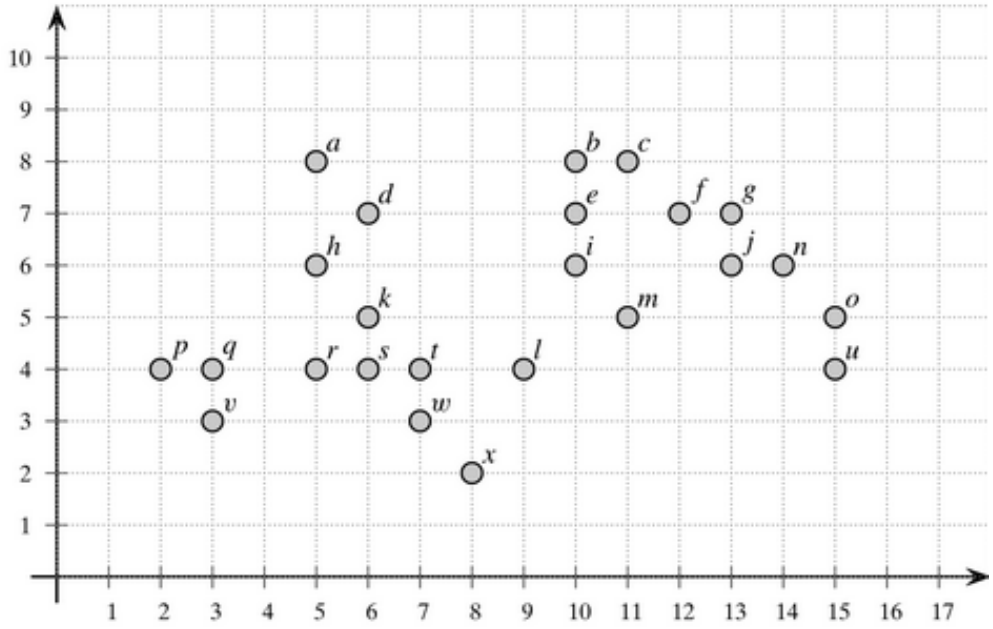

Рисунок 1 – Набор данных для 1 задания

Допустим, что используется Евклидово расстояние, $\epsilon = 2$ и minPts = 3. Необходимо:

1) выписать список всех основных точек;

2) показать, является ли точка *a* прямо достижимой из точки *d*;

3) показать, является ли точка *o* достижимой по плотности из точки *i* и если нет, то показать на какой точке цепочка построения пути оборвалась;

4) показать кластеры, полученные алгоритмом DBSCAN и выпавшие точки.

Код программы

```
import numpy as np
import math
from sklearn import cluster
D=[
    [5, 8],
    [10, 8],
    [11, 8],
    [6, 7],
    [10, 7],
    [12, 7],
    [13, 7],
    [5, 6],
    [10, 6],
    [13, 6],
```

```
    [6, 5],
    [9, 4],
    [11, 5],
    [14, 6],
    [15, 5],
    [2, 4],
    [3, 4],
    [5, 4],
    [6, 4],
    [7, 4],
    [15, 4],
    [3, 3],
    [7, 3],
    [8, 2],
]

eps=2
minPts=3
dist=[[math.sqrt((d[0]-dd[0])**2+(d[1]-dd[1])**2) for dd in D] for d in D]
neibs=[sum([1 if ds<=eps and ds>0 else 0 for ds in dst])for dst in dist]
core=[chr(ord('a')+i) for i in  range(len(neibs)) if neibs[i]>=minPts ]
print(core)
clust=cluster.DBSCAN(eps=eps,min_samples=minPts,metric="euclidean").fit(D)
for i in range(len(set(clust.labels_))):
        print(i,":",[chr(ord('a')+j)   for   j   in   range(len(neibs))   if
clust.labels_[j]==i ])
```

Результат выполнения

1) Основные точки: ['b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'n', 'q', 'r', 's', 't', 'w'].

2) Точка *a* является прямо достижимой из *d*, так как расстояние между ними меньше 2.

3) Точка *o* является достижимой по плотности из *i* по цепочке *i->e->b->c->f->j->n->o.*

4) В результате работы алгоритма было получено 2 кластера: ['a', 'd', 'h', 'k', 'l', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x'] и ['b', 'c', 'e', 'f', 'g', 'i', 'j', 'm', 'n', 'o', 'u']. Выпавших точек получено не было.

Задание 2

Постановка задачи

Даны следующие метрики:

1. $L_{\infty}(\mathbf{x},\mathbf{y}) = \max_{i=1}^{d}\{|x_i - y_i|\}$,
2. $L_{\frac{1}{2}}(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{a}|x_i - y_i|^{\frac{1}{2}}\right)^2$,
3. $L_{\min}(\mathbf{x},\mathbf{y}) = \min_{i=1}^{d}\{|x_i - y_i|\}$,

4. $$L_{pow}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} 2^{i-1}(x_i - y_i)^2\right)^{1/2}$$.

Данные представлены на рисунке 2.

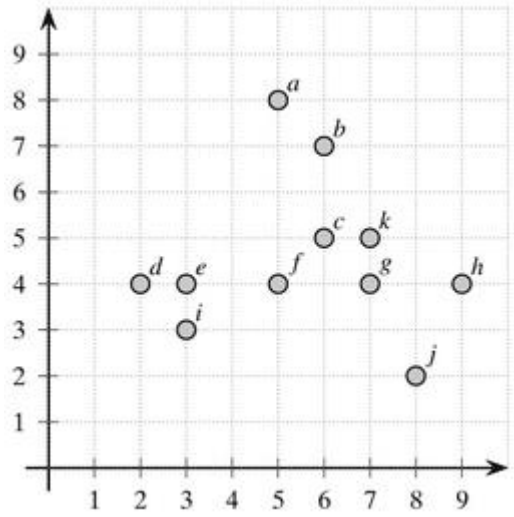

Рисунок 2 – Набор данных для 2 задания

Используя метод DBSCAN необходимо провести кластеризацию при следующих параметрах:

1) ϵ = 2 и minPts = 5 и метрика (1),

2) ϵ = 4 и minPts = 3 и метрика (2),

3) ϵ = 1 и minPts = 6 и метрика (3),

4) ϵ = 4 и minPts = 6 и метрика (4).

Для всех случаев построить кластеры и отобразить основные точки, достижимые по плотности точки и выпавшие точки.

Код программы

```
import math
from sklearn import cluster

D = [
    [5, 8],
    [6, 7],
    [6, 5],
    [2, 4],
    [3, 4],
    [5, 4],
    [7, 4],
    [9, 4],
    [3, 3],
    [8, 2],
    [7, 5]
]
```

```
def Linf(x,y):
    return max(abs(x[0]-y[0]),abs(x[1]-y[1]))
def Lh(x,y):
    return math.pow(math.sqrt(abs(x[0]-y[0]))+math.sqrt(abs(x[1]-y[1])),2)
def Lmin(x,y):
    return min(abs(x[0]-y[0]),abs(x[1]-y[1]))
def Lpow(x,y):
    return math.sqrt((x[0]-y[0])**2+2*((x[1]-y[1])**2))

eps=2
min_samples=5
metric=Linf
neibs=[sum([1 if metric(d,dd)<=eps and dd!=d else 0 for dd in D]) for d in D]
core=[chr(ord('a')+i) for i in  range(len(neibs)) if neibs[i]>=min_samples ]
print("Core:",core)
clust=cluster.DBSCAN(eps=eps,min_samples=min_samples,metric=metric).fit(D)
bord=[]
for i in range(len(clust.labels_)):
    if clust.labels_[i]>=0 and chr(ord('a')+i) not in core:
        bord.append(chr(ord('a')+i))
print("Border:",bord)
for i in [it+min(clust.labels_) for it in range(len(set(clust.labels_)))]:
    print(i,":",[chr(ord('a')+j)    for    j    in    range(len(neibs))    if
clust.labels_[j]==i ])


eps=4
min_samples=3
metric=Lh
neibs=[sum([1 if metric(d,dd)<=eps and dd!=d else 0 for dd in D]) for d in D]
core=[chr(ord('a')+i) for i in  range(len(neibs)) if neibs[i]>=min_samples ]
print("Core:",core)
clust=cluster.DBSCAN(eps=eps,min_samples=min_samples,metric=metric).fit(D)
bord=[]
for i in range(len(clust.labels_)):
    if clust.labels_[i]>=0 and chr(ord('a')+i) not in core:
        bord.append(chr(ord('a')+i))
print("Border:",bord)
for i in [it+min(clust.labels_) for it in range(len(set(clust.labels_)))]:
    print(i,":",[chr(ord('a')+j)    for    j    in    range(len(neibs))    if
clust.labels_[j]==i ])

eps=1
min_samples=6
metric=Lmin
neibs=[sum([1 if metric(d,dd)<=eps and dd!=d else 0 for dd in D]) for d in D]
core=[chr(ord('a')+i) for i in  range(len(neibs)) if neibs[i]>=min_samples ]
print("Core:",core)
clust=cluster.DBSCAN(eps=eps,min_samples=min_samples,metric=metric).fit(D)
bord=[]
for i in range(len(clust.labels_)):
    if clust.labels_[i]>=0 and chr(ord('a')+i) not in core:
        bord.append(chr(ord('a')+i))
print("Border:",bord)
for i in [it+min(clust.labels_) for it in range(len(set(clust.labels_)))]:
    print(i,":",[chr(ord('a')+j)    for    j    in    range(len(neibs))    if
clust.labels_[j]==i ])

eps=4
min_samples=6
metric=Lpow
neibs=[sum([1 if metric(d,dd)<=eps and dd!=d else 0 for dd in D]) for d in D]
core=[chr(ord('a')+i) for i in  range(len(neibs)) if neibs[i]>=min_samples ]
```

```
print("Core:",core)
clust=cluster.DBSCAN(eps=eps,min_samples=min_samples,metric=metric).fit(D)
bord=[]
for i in range(len(clust.labels_)):
    if clust.labels_[i]>=0 and chr(ord('a')+i) not in core:
        bord.append(chr(ord('a')+i))
print("Border:",bord)
for i in [it+min(clust.labels_) for it in range(len(set(clust.labels_)))]:
    print(i,":",[chr(ord('a')+j)    for    j    in    range(len(neibs))    if
clust.labels_[j]==i ])
```

Результат выполнения

1) Основные точки: ['f', 'g', 'k'], граничные точки ['b', 'c', 'e', 'h', 'i', 'j'], выпавшие точки: ['a', 'd'], единственный образовавшийся кластер: ['b', 'c', 'e', 'f', 'g', 'h', 'i', 'j', 'k'].

2) Основные точки: ['c', 'd', 'e', 'f', 'g'], граничные точки: ['a', 'b', 'h', 'i', 'k'], выпавшие точки: ['j'], единственный образовавшийся кластер: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k'].

3) Основные точки: ['c', 'd', 'e', 'f', 'g', 'h', 'i', 'k'], граничные точки: ['a', 'b', 'j'], выпавших точек нет, единственный образовавшийся кластер: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k']

4) Основные точки: ['c', 'f', 'g'], граничные точки: ['b', 'd', 'e', 'h', 'i', 'j', 'k'], выпавшие точки: ['a'], единственный образовавшийся кластер: ['b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k'].