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

# ОТЧЕТ

**по практической работе №5**

**по дисциплине «Машинное обучение»**

Студент гр. 1310 _____ Комаров Д. Е.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2025

**Задание 1**

**Постановка задачи**

Дан набор значений 2, 4, 10, 12, 3, 20, 30, 11, 25. Предположим количество кластеров k = 3, и выбраны начальные средние значения $m_1 = 2$, $m_2 = 4$, $m_3 = 6$. Покажите, какие кластеры будут сформированы после первой итерации алгоритма k-средних, и рассчитайте новые значения центров кластеров для следующей итерации.

**Код программы**

```
import numpy as np

D=[2, 4, 10, 12, 3, 20, 30, 11, 25]
startM=[2, 4, 6]

def kMeans(D,startM,iters):
    M=startM
    for i in range(iters):
        clust=[[] for j in range(len(M))]
        for d in D:
            delt=(d-m)**2 for m in M]
            clust[delt.index(min(delt))].append(d)
        M=[float(np.mean(cl)) for cl in clust]
    return (clust,M)

clust,M=kMeans(D,startM,1)
print(clust)
print(M)
```

**Результат выполнения**

В результате выполнения программы после первой итерации алгоритма были сформированы следующие кластеры: [2, 3], [4], [10, 12, 20, 30, 11, 25]. Новые значения центров кластеров для следующей итерации: [2.5, 4.0, 18.0].

**Задание 2**

**Постановка задачи**

Набор точек x и вероятности из принадлежности к кластерам C1 и C2 представлен в таблице 1.

Таблица 1 – Набор точек для 2 задания

| $x$ | $P(C_1\|x)$ | $P(C_2\|x)$ |
|---|---|---|
| 2 | 0.9 | 0.1 |
| 3 | 0.8 | 0.1 |
| 7 | 0.3 | 0.7 |
| 9 | 0.1 | 0.9 |
| 2 | 0.9 | 0.1 |
| 1 | 0.8 | 0.2 |

Необходимо выполнить следующие задания:

1. Найди оценку максимального правдоподобия для средних μ1 и μ2;

2. Предположим, что μ1 = 2, μ2 = 7 и σ1 = σ2 = 1. Необходимо найти вероятности принадлежности точки $x = 5$ к кластерам C1 и C2. Априорные вероятности каждого кластера P(C1) = P(C2) = 0.5 и P($x$ = 5) = 0.029.

**Код программы**

```
import math
X=[
    (2,[0.9,0.1]),
    (3,[0.8,0.1]),
    (7,[0.3,0.7]),
    (9,[0.1,0.9]),
    (2,[0.9,0.1]),
    (1,[0.8,0.2]),
    ]
k=2
M=[sum(x[0]*x[1][kit]for x in X)/sum([x[1][kit] for x in X]) for kit in
range(k)]
print(M)

x=5
M=[2,7]
std=[1,1]
C=[0.5,0.5]
f=[(1/(math.sqrt(2*math.pi)*std[kit]))*math.exp(-pow(x-
M[kit],2)/(2*std[kit]**2)) for kit in range(k)]
P=[f[kit]*C[kit]/sum(f[kk]*C[kk]for kk in range(k))for kit in range(k)]
print(P)
```

**Результат выполнения**

1) μ1=2.58; μ2=6.62;

2) P(C1 | $x$) = 0.08; P(C2 | $x$) = 0.92;

**Задание 3**

**Постановка задачи**

Категориальные данные размерности 5 представлены в таблице 2.

Таблица 2 – Категориальные данные для 3 задания

| Point | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ |
|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 0 | 1 | 1 | 0 |
| $\mathbf{x}_2$ | 1 | 1 | 0 | 1 | 0 |
| $\mathbf{x}_3$ | 0 | 0 | 1 | 1 | 0 |
| $\mathbf{x}_4$ | 0 | 1 | 0 | 1 | 0 |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 |
| $\mathbf{x}_6$ | 0 | 1 | 1 | 0 | 0 |

Близость двух наблюдений определяется через количество совпадений и несовпадений значений признаков. Допустим, что $n_{11}$ количество признаков одновременной равных 1 для наблюдений $x_i$ и $x_j$, и $n_{10}$ количество признаков равных 1 для наблюдения $x_i$ и в то же время равных 0 для наблюдения $x_j$. По аналогии определяются значения $n_{01}$ и $n_{00}$.

Определим следующие метрики.

Коэффициент простого совпадения

$$SMC(\mathbf{x}_i, \mathbf{x}_j) = \frac{n_{11}+n_{00}}{n_{11}+n_{10}+n_{01}+n_{00}}.$$

Коэффициент Жаккара

$$JC(\mathbf{x}_i, \mathbf{x}_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}}.$$

Коэффициент Рассела и Рао

$$RC(\mathbf{x}_i, \mathbf{x}_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}+n_{00}}.$$

Необходимо построить дендограммы полученные после иерархической кластеризации при следующих параметрах:

- метод одиночной связи с метрикой RC;
- метод полной связи с метрикой SMC;
- невзвешенный центроидный метод с метрикой JC.

**Код программы**

```
import networkx as nx
import matplotlib.pyplot as plt
from scipy.spatial.distance import squareform
from scipy.cluster.hierarchy import dendrogram, linkage

X=[
    [1,0,1,1,0],
    [1,1,0,1,0],
    [0,0,1,1,0],
    [0,1,0,1,0],
    [1,0,1,0,1],
    [0,1,1,0,0]
    ]
points = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6']

N11=[[sum([1 if x1[i] and x2[i] else 0 for i in range(len(x1))])for x2 in X]for
x1 in X ]
N10=[[sum([1 if x1[i] and (not x2[i]) else 0 for i in range(len(x1))])for x2 in
X]for x1 in X ]
N01=[[sum([1 if (not x1[i]) and x2[i] else 0 for i in range(len(x1))])for x2 in
X]for x1 in X ]
```

```
N00=[[sum([1 if (not x1[i]) and (not x2[i]) else 0 for i in range(len(x1))])for
x2 in X]for x1 in X ]

SMC=[[1-(N11[i][j]+N00[i][j])/(N11[i][j]+N10[i][j]+N01[i][j]+N00[i][j])for   j
in range(len(X))]for i in range(len(X))]
JC=[[1-N11[i][j]/(N11[i][j]+N10[i][j]+N01[i][j])for j in range(len(X))]for i in
range(len(X))]
RC=[[0 if i==j else(1-N11[i][j]/(N11[i][j]+N10[i][j]+N01[i][j]+N00[i][j]))for
j in range(len(X))]for i in range(len(X))]

linkage_rc = linkage(squareform(RC), method='single')
dendrogram(linkage_rc, labels=points)
plt.show()

linkage_smc = linkage(squareform(SMC), method='complete')
dendrogram(linkage_smc, labels=points)
plt.show()

linkage_jc = linkage(squareform(JC), method='centroid')
dendrogram(linkage_jc, labels=points)
plt.show()
```

**Результат выполнения**

1) Дендограмма, построенная при помощи метода одиночной связи с метрикой RC представлена на рисунке 1.

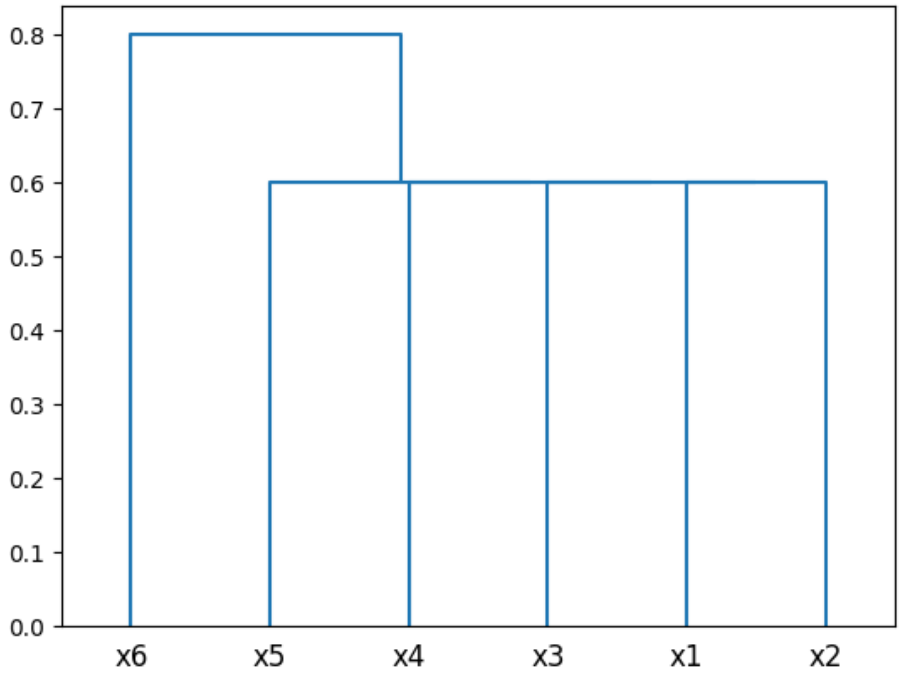


Рисунок 1 – Дендограмма, построенная при помощи метода одиночной связи с метрикой RC

2) Дендограмма, построенная при помощи метода полной связи с метрикой SMC представлена на рисунке 2.

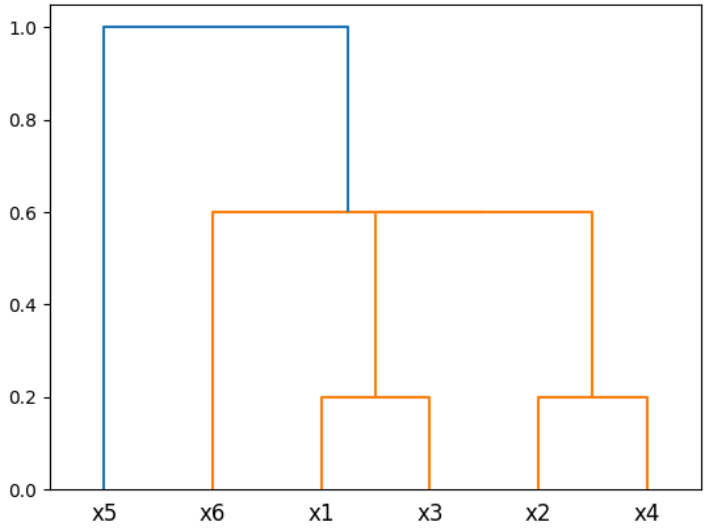


Рисунок 2 – Дендограмма, построенная при помощи метода полной связи с метрикой SMC

3) Дендограмма, построенная при помощи невзвешенного центроидного метода с метрикой JC представлена на рисунке 3.

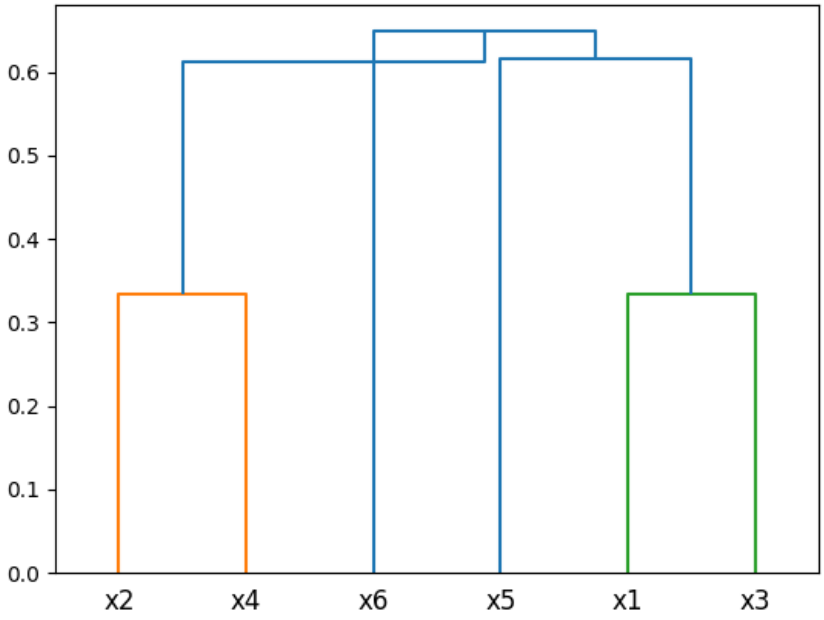


Рисунок 3 – Дендограмма, построенная при помощи невзвешенного центроидного метода с метрикой JC