Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИУК «Информатика и управление»**
КАФЕДРА **ИУК4 «Программное обеспечение ЭВМ,**
информационные технологии»

ДОМАШНЯЯ РАБОТА №1

«Применение байесовского подхода принятия решений»

ДИСЦИПЛИНА: «Программные системы распознавания и обработки информации»

Выполнил: студент гр. ИУК4-31М ______________ (Сафронов Н.С,)
(подпись) (Ф.И.О.)

Проверил: ______________ (Гагарин Ю.Е.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2025

Цель:

Формирование практических навыков использования байесовского подхода принятия решений в случае классификации двух классов, условные плотности вероятностей которых соответствуют одномерному и двумерному нормальному закону распределения.

Задачи:

1. Определение границы разделения классов;

2. Построение графиков условных плотностей вероятностей и границ разделения классов;

3. Использование формулы Байеса для определения апостериорных вероятностей;

4. Определение вероятности ошибки классификации.

Задание

1. Использование байесовского классификатора для двух классов.

Для первого класса ω_1, из определенного диапазона, задать случайным образом 50 значений признака x.

Для второго класса ω_2, из определенного диапазона, задать случайным образом 70 значений признака x.

Предполагая, что условные плотности вероятности $P(x|\omega_1)$ и $P(x|\omega_2)$ соответствуют нормальному закону распределения $P(x|\omega) = \frac{1}{\sqrt{w\pi}\,\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$,

определить значения параметров $(\mu.\sigma)$ для двух классов.

Найти границу разделения классов.

Построить графики условных плотностей вероятностей и границы разделения классов.

Задать значение признака X и предполагая, что априорные вероятности равны $P(\omega_1) = P(\omega_2) = \frac{1}{2}$, определить апостериорные вероятности по формуле Байеса

$$P(\omega_1|x) = \frac{P(X|\omega_1)P(\omega_1)}{\sum_{j=1}^{2} P(X|w_j)P(\omega_j)} \ .$$

Определить к какому классу относится значение признака X.

Рассчитать вероятность ошибки классификации.

2. Классификация двух классов по двум признакам.

Для первого класса ω_1, из определенного диапазона, задать случайным образом 50 значений признаков (x_1, x_2).

Для второго класса ω_2, из определенного диапазона, задать случайным образом 70 значений признаков (x_1, x_2).

Предполагая, что условные плотности вероятности $P(x|\omega_1)$ и $P(x|\omega_2)$ соответствуют двумерному нормальному закону распределения

$$P(x|\omega) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{d}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right]$$

и для случая, когда $\Sigma_i = \sigma^2 I$ определить значения параметров μ, σ для двух классов.

Определить границу разделения классов.

Построить графики условных плотностей вероятностей и границы разделения классов.

Задать значения признаков (X_1, X_2) и предполагая, что априорные вероятности равны $P(\omega_1) = P(\omega_2) = \frac{1}{2}$, определить апостериорные вероятности по формуле Байеса

$$P(\omega_1|x) = \frac{P(X|\omega_1)P(\omega_1)}{\sum_{j=1}^{2} P(X|w_j)P(\omega_j)} \ .$$

Определить к какому классу относится значение признака (X_1, X_2).

Результаты выполнения работы

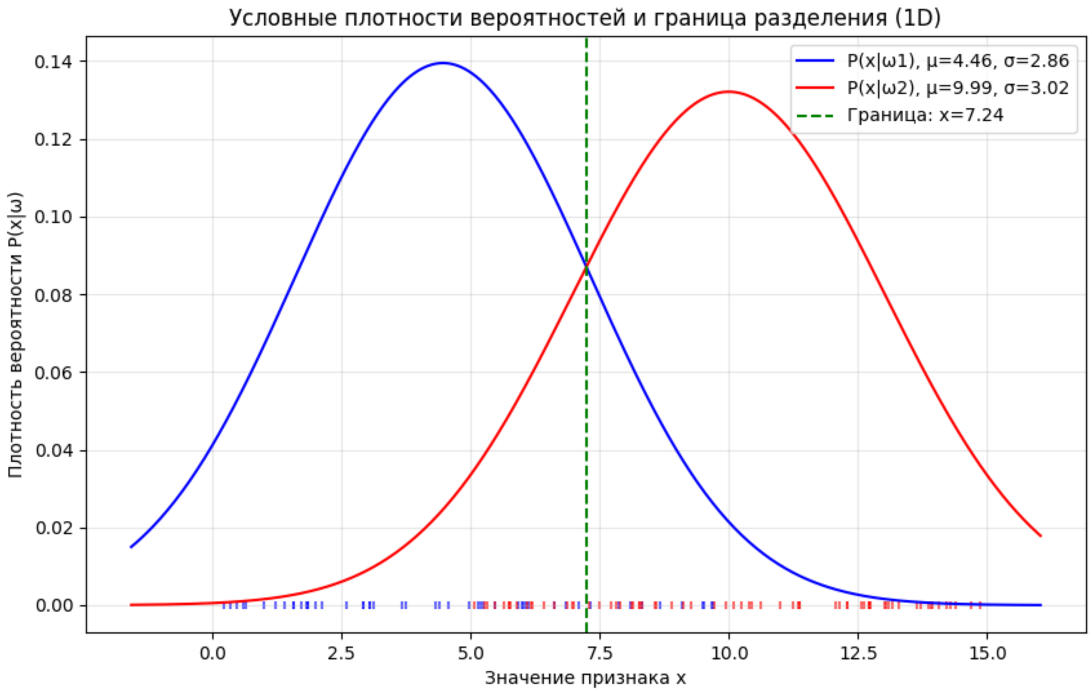

Рисунок 1 – Условные плотности вероятностей и граница разделения для классов

```
Класс ω1: μ = 4.459, σ = 2.860
Класс ω2: μ = 9.994, σ = 3.020
Граница разделения классов: x = 7.235

Анализ точки X = 7.5
Апостериорная вероятность P(ω1|X) = 0.4576
Апостериорная вероятность P(ω2|X) = 0.5424
Точка X = 7.5 относится к классу ω2

Вероятность ошибки классификации P(error) = 0.1732
```

Рисунок 2 – Значения (μ, σ) классов, граница распределения, вероятности для точки X и ошибка классификации

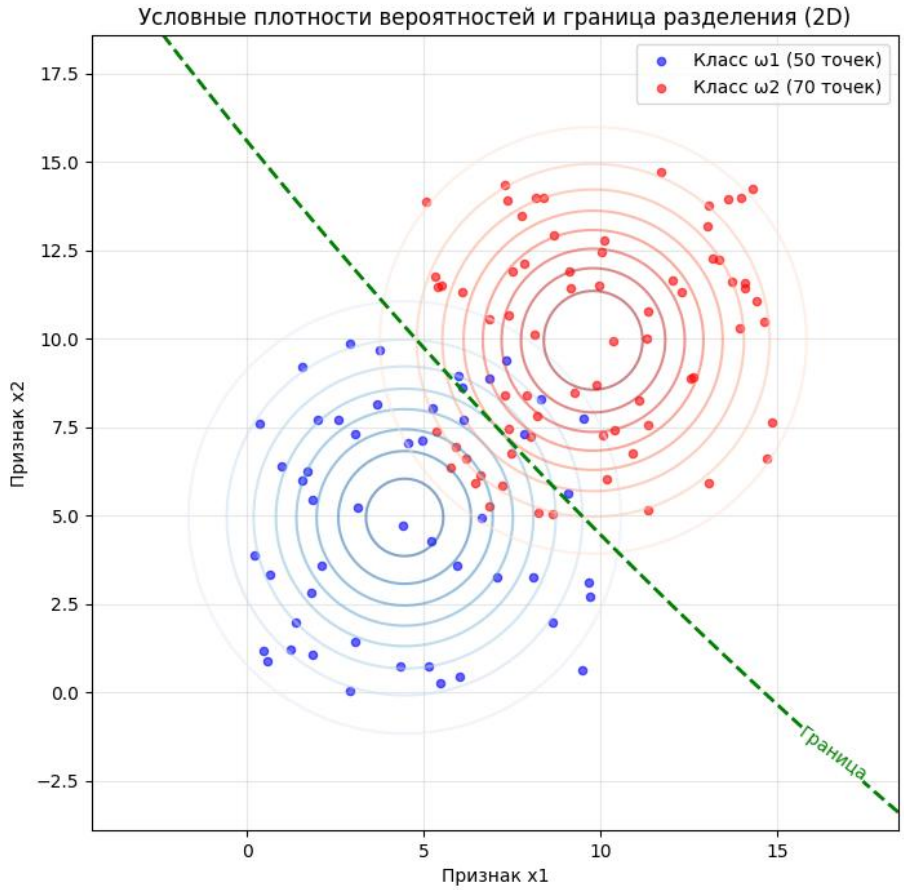

Рисунок 3 – Условные плотности вероятностей и граница разделения для классов

```
Класс ω1: μ = [4.459, 4.944], σ = 2.950
Класс ω2: μ = [9.794, 9.956], σ = 2.879

Анализ точки X = (7.5, 7.5)
Апостериорная вероятность P(ω1|X) = 0.4319
Апостериорная вероятность P(ω2|X) = 0.5681
Точка X = (7.5, 7.5) относится к классу ω2
```

Рисунок 4 – Значения $(\mu, \sigma\ )$ классов, граница распределения, вероятности для точки X_1, X_2 и ошибка классификации

Вывод: в результате выполнения домашней работы были сформированы практические навыки использования байесовского подхода принятия решений в случае классификации двух классов, условные плотности вероятностей которых соответствуют одномерному и двумерному нормальному закону распределения.

Листинг программы

Задание 1

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

np.random.seed(42)

x1 = np.random.uniform(0, 10, 50)
x2 = np.random.uniform(5, 15, 70)

mu1, sigma1 = np.mean(x1), np.std(x1)
mu2, sigma2 = np.mean(x2), np.std(x2)

print(f"Класс ω1: μ = {mu1:.3f}, σ = {sigma1:.3f}")
print(f"Класс ω2: μ = {mu2:.3f}, σ = {sigma2:.3f}")

def find_boundary(mu1: float, sigma1: float, mu2: float, sigma2: float,
search_range: tuple[float, float]) -> float:
    """
    Нахождение границы разделения классов.
    При равных априорных вероятностях P(ω1)=P(ω2)=0.5:
    Граница - это точка, где P(x|ω1) * P(ω1) = P(x|ω2) * P(ω2), т.е. P(x|ω1) =
P(x|ω2)
    """
    x_vals = np.linspace(search_range[0], search_range[1], 1000)
    pdf1 = norm.pdf(x_vals, mu1, sigma1)
    pdf2 = norm.pdf(x_vals, mu2, sigma2)
    diff = pdf1 - pdf2
    idx = np.argmin(np.abs(diff))
    return x_vals[idx]

search_start = min(mu1, mu2) - 2*max(sigma1, sigma2)
search_end = max(mu1, mu2) + 2*max(sigma1, sigma2)
boundary = find_boundary(mu1, sigma1, mu2, sigma2, (search_start, search_end))
print(f"Граница разделения классов: x = {boundary:.3f}")

x_plot = np.linspace(search_start, search_end, 500)
pdf1_plot = norm.pdf(x_plot, mu1, sigma1)
```

```
pdf2_plot = norm.pdf(x_plot, mu2, sigma2)

plt.figure(figsize=(10, 6))
plt.plot(x_plot, pdf1_plot, label=f'P(x|ω1), μ={mu1:.2f}, σ={sigma1:.2f}', color='blue')
plt.plot(x_plot, pdf2_plot, label=f'P(x|ω2), μ={mu2:.2f}, σ={sigma2:.2f}', color='red')
plt.axvline(x=boundary, color='green', linestyle='--', label=f'Граница: x={boundary:.2f}')
plt.scatter(x1, [0]*len(x1), alpha=0.6, color='blue', s=20, marker='|')
plt.scatter(x2, [0]*len(x2), alpha=0.6, color='red', s=20, marker='|')
plt.title('Условные плотности вероятностей и граница разделения (1D)')
plt.xlabel('Значение признака x')
plt.ylabel('Плотность вероятности P(x|ω)')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()

X = 7.5
print(f"\nАнализ точки X = {X}")

P_w1 = 0.5
P_w2 = 0.5

P_X_w1 = norm.pdf(X, mu1, sigma1)
P_X_w2 = norm.pdf(X, mu2, sigma2)

P_X = P_X_w1 * P_w1 + P_X_w2 * P_w2

P_w1_X = (P_X_w1 * P_w1) / P_X
P_w2_X = (P_X_w2 * P_w2) / P_X

print(f"Апостериорная вероятность P(ω1|X) = {P_w1_X:.4f}")
print(f"Апостериорная вероятность P(ω2|X) = {P_w2_X:.4f}")

predicted_class = "ω1" if P_w1_X > P_w2_X else "ω2"
print(f"Точка X = {X} относится к классу {predicted_class}")

P_error_w1 = 1 - norm.cdf(boundary, mu1, sigma1)
P_error_w2 = norm.cdf(boundary, mu2, sigma2)
```

```
P_error = P_w1 * P_error_w1 + P_w2 * P_error_w2

print(f"\nВероятность ошибки классификации P(error) = {P_error:.4f}")
```

Задание 2

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

np.random.seed(42)

x1_class1 = np.random.uniform(0, 10, 50)
x2_class1 = np.random.uniform(0, 10, 50)
data_class1 = np.column_stack((x1_class1, x2_class1))

x1_class2 = np.random.uniform(5, 15, 70)
x2_class2 = np.random.uniform(5, 15, 70)
data_class2 = np.column_stack((x1_class2, x2_class2))

mu1_2d = np.mean(data_class1, axis=0)
mu2_2d = np.mean(data_class2, axis=0)

sigma1_2d = np.sqrt(np.mean(np.var(data_class1, axis=0)))
sigma2_2d = np.sqrt(np.mean(np.var(data_class2, axis=0)))

cov1 = np.array([[sigma1_2d**2, 0],
                 [0, sigma1_2d**2]])
cov2 = np.array([[sigma2_2d**2, 0],
                 [0, sigma2_2d**2]])

print(f"Класс ω1: μ = [{mu1_2d[0]:.3f}, {mu1_2d[1]:.3f}], σ = {sigma1_2d:.3f}")
print(f"Класс ω2: μ = [{mu2_2d[0]:.3f}, {mu2_2d[1]:.3f}], σ = {sigma2_2d:.3f}")

x_min,     x_max     =     min(mu1_2d[0]-3*sigma1_2d,     mu2_2d[0]-3*sigma2_2d),
max(mu1_2d[0]+3*sigma1_2d, mu2_2d[0]+3*sigma2_2d)
```

```
y_min, y_max = min(mu1_2d[1]-3*sigma1_2d, mu2_2d[1]-3*sigma2_2d), max(mu1_2d[1]+3*sigma1_2d, mu2_2d[1]+3*sigma2_2d)

xx, yy = np.meshgrid(np.linspace(x_min, x_max, 200),
                     np.linspace(y_min, y_max, 200))
pos = np.dstack((xx, yy))

rv1 = multivariate_normal(mu1_2d, cov1)
rv2 = multivariate_normal(mu2_2d, cov2)

pdf1_2d = rv1.pdf(pos)
pdf2_2d = rv2.pdf(pos)

fig, ax = plt.subplots(1, 1, figsize=(10, 8))

levels = np.linspace(0, max(np.max(pdf1_2d), np.max(pdf2_2d)), 10)
cs1 = ax.contour(xx, yy, pdf1_2d, levels=levels, cmap='Blues', alpha=0.5)
cs2 = ax.contour(xx, yy, pdf2_2d, levels=levels, cmap='Reds', alpha=0.5)

diff = pdf1_2d - pdf2_2d
boundary_contour = ax.contour(xx, yy, diff, levels=[0], colors='green', linewidths=2, linestyles='--')
plt.clabel(boundary_contour, inline=True, fontsize=10, fmt='Граница')

ax.scatter(data_class1[:, 0], data_class1[:, 1], c='blue', s=20, alpha=0.6, label='Класс ω1 (50 точек)')
ax.scatter(data_class2[:, 0], data_class2[:, 1], c='red', s=20, alpha=0.6, label='Класс ω2 (70 точек)')

ax.set_title('Условные плотности вероятностей и граница разделения (2D)')
ax.set_xlabel('Признак x1')
ax.set_ylabel('Признак x2')
ax.legend()
ax.grid(True, alpha=0.3)
ax.set_aspect('equal', 'box')
plt.show()

X_2d = np.array([7.5, 7.5])
print(f"\nАнализ точки X = ({X_2d[0]}, {X_2d[1]})")
```

```python
P_w1 = 0.5
P_w2 = 0.5

P_X_w1 = rv1.pdf(X_2d)
P_X_w2 = rv2.pdf(X_2d)

P_X = P_X_w1 * P_w1 + P_X_w2 * P_w2

P_w1_X = (P_X_w1 * P_w1) / P_X
P_w2_X = (P_X_w2 * P_w2) / P_X

print(f"Апостериорная вероятность P(ω1|X) = {P_w1_X:.4f}")
print(f"Апостериорная вероятность P(ω2|X) = {P_w2_X:.4f}")

predicted_class_2d = "ω1" if P_w1_X > P_w2_X else "ω2"
print(f"Точка X = ({X_2d[0]}, {X_2d[1]}) относится к классу {predicted_class_2d}")
```