

Легкая задача

Условие:

Вы являетесь разработчиком базовой системы машинного обучения, которая должна упростить классификацию объектов. У вас есть набор данных с результатами экспериментов, сохранённый в файле **data_easy.txt**. В этом файле каждая строка содержит два числа: первое число x — это целочисленное значение признака (например, масса объекта), второе число y — это метка класса (0 или 1). Найдите оптимальный порог t , который позволяет разделить объекты двух классов, минимизируя количество ошибок классификации, и посчитайте, сколько объектов было ошибочно классифицировано при использовании оптимального порога t . Выведите в ответ два числа: найденный порог t (если таких несколько, выведите порог с минимальным индексом) и количество ошибок классификации.

Объект считается ошибочно классифицированным, если:

- $x < t$, а $y = 1$, или
- $x \geq t$, а $y = 0$.

Пример содержимого файла **data_easy.txt**:

```
1 0
2 0
3 0
4 1
5 1
6 1
```

Пример вывода:

```
4
0
```

Решение:

1. Сначала считываем файл **data_easy.txt** и записываем пары чисел (x, y) в список. Для этого используем генератор списков, где каждое число приводится к типу `int`.
2. Создаём переменные для хранения минимального количества ошибок (`min_errors`) и оптимального порога t (`best_t`), изначально задав им бесконечное значение (`float('inf')`) и `None` соответственно.
3. Для каждого значения x из данных считаем его возможным порогом t .
4. Проходим по всем данным и подсчитываем количество ошибок при текущем пороге t по условию (ошибка, если $x < t$, а $y = 1$, или $x \geq t$, а $y = 0$.)
5. Если количество ошибок для текущего порога меньше текущего минимального значения, обновляем минимальные ошибки и оптимальный порог.
6. Выводим найденный порог t и минимальное количество ошибок в требуемой форме.

```

1 with open('data_easy.txt', 'r') as file:
2     data = [tuple(map(int, line.split())) for line in file]
3
4 min_errors = float('inf')
5 best_t = None
6
7 for t in [point[0] for point in data]:
8     errors = sum(1 for x, y in data if (x < t and y == 1) or (x >= t and y == 0))
9     # Условие обновления: строго меньшее количество ошибок или минимальный индекс при равных ошибках
10    if errors < min_errors or (errors == min_errors and (best_t is None or t < best_t)):
11        min_errors = errors
12        best_t = t
13
14 print(best_t, min_errors)

```

Листинг кода:

Лёгкий уровень

with open('data_easy.txt', 'r') as file:

data = [tuple(map(int, line.split())) for line in file]

min_errors = float('inf')

best_t = None

for t in [point[0] for point in data]:

errors = sum(1 for x, y in data if (x < t and y == 1) or (x >= t and y == 0))

if errors < min_errors or (errors == min_errors and (best_t is None or t < best_t)):

min_errors = errors

best_t = t

print(best_t, min_errors)

Ответ для файла:

4

0