

Машинное обучение: При свете ранней зари

Предыстория

Иван получил необычный запрос от потенциального клиента: классифицировать набор 2D-данных по трем классам, обозначенным разными цветами.

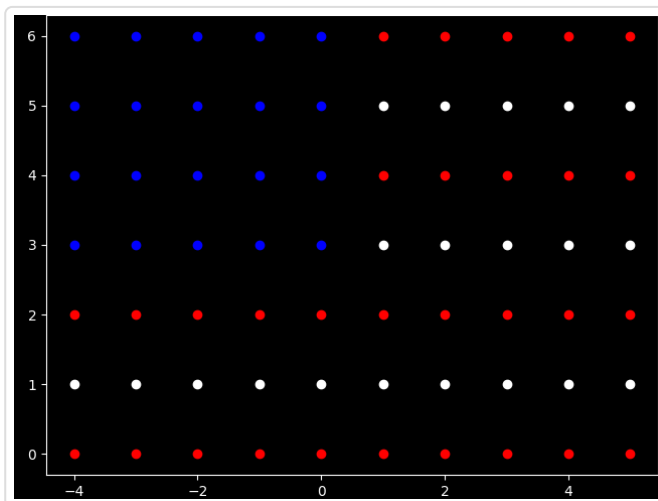
Открыв файл, Иван уставился на экран. Точки данных образовывали необычный узор, который что-то ему напоминал...

Поразмыслив немного, он улыбнулся. Из всех задач классификации, с которыми он сталкивался, эта была, безусловно, самой креативной.

У клиента были конкретные требования: во-первых, определить, какие классические алгоритмы машинного обучения теоретически могут обеспечить идеальную классификацию без преобразования координат. Во-вторых, найти минимальный набор математических условий, которые могли бы однозначно разделить все три класса.

Задача

У вас есть двумерный набор данных, представленный на рисунке:



Уточним, что ось x расположена горизонтально, а ось y — вертикально. Все точки расположены в целочисленных координатах. Их цвет обозначает класс — есть белый класс, красный класс и синий класс.

Задание {1} Предполагая, что любые преобразования координат запрещены, какой из нижеперечисленных классификаторов теоретически может научиться полностью разделить эти три класса (предсказывать метки обучающего набора со 100% точностью)?

- Классификатор K ближайших соседей
- Классификатор опорных векторов
- Классификатор дерева решений
- Кластеризация K -средних
- Многослойный перцептрон

Задание {2} Для целей этой задачи мы определяем условие как математическое выражение, состоящее только из координат x и y , а также операторов $\{+, -, *, /, \% \text{ (модуль)}, <, \leq, =, >, \geq, \text{оператор степени}\}$, которое может быть оценено как истинное или ложное. Например, $2x - y/3 > 4$ или $x^2 > -(y-x)^3$.

Составьте список таких условий, чтобы ответ «да»/«нет» на каждое условие однозначно определял метку, и реализуйте их на Python.

Формат

Первая часть должна быть представлена в виде списка ответов «да»/«нет», соответствующих классификаторам.

Для второй части предоставьте файл Python, содержащий список условий, представленный лямбда-функциями:

```
[lambda x, y: bool_expression_1, ..., lambda x, y: bool_expression_x]
```

Неправильным, но синтаксически допустимым вариантом будет

```
[lambda x, y: 2*x - y / 3 > 4, ..., lambda x, y: x**2 > -(y-x)**3]
```

Каждое из булевых выражений может содержать только `x, y, +, -, *, /, //, %, **, <, <=, ==, >=, >` и числовые константы.

Важно: каждое выражение должно быть одним сравнением (например, `x + y > 5`). Нельзя комбинировать несколько сравнений с арифметическими операторами (например, `(a > b) * (c < d)` не допускается), так как это приведет к созданию промежуточных булевых значений.

Подсказки

Если вы не знаете, как работают некоторые или все эти классификаторы, не беспокойтесь. Вы можете найти информацию о них в Интернете. Вот полезное видео, в котором объясняется каждый из них (и не только), которое может стать хорошей отправной точкой: <https://www.youtube.com/watch?v=E0Hmnixke2g>

Оценивание

- 25% за правильное выполнение задания {1},
- 75% за правильное выполнение задания {2}.

Задание {1}

- количество правильных ответов / 5

Задание {2}

- Для x количества условий начисляется $1,776^{2-x}$.
- Неправильный формат ИЛИ неправильная классификация ИЛИ ошибка python - 0 баллов.