

Machine Learning: By the Dawn's Early Light

Problem Story

Ivan received a peculiar request from a prospective client: classify a 2D dataset into three color-coded classes.

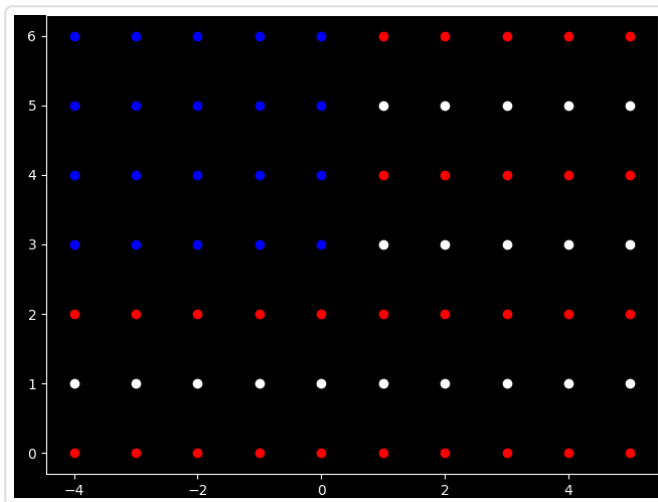
When he opened the file, Ivan stared at his screen. The data points formed an unusual pattern that looked vaguely familiar...

After a moment of contemplation, he smiled. Of all the classification problems he'd encountered, this was certainly the most creative.

The client had specific requirements: first, determine which classical ML algorithms could theoretically achieve perfect classification without any coordinate transformations. Second, find the minimal set of mathematical conditions that could unambiguously separate all three classes.

Problem

You have the 2D dataset as presented in the picture:



For clarity, the x-axis is horizontal, the y-axis is vertical. All the points are located at integer coordinates. Their color represents their class — so there is a white class, red class, and a blue class.

Task {1} Assuming that any transformations of the coordinates are forbidden, which of the classifiers below can theoretically learn to fully separate these three classes (predict the labels of the training set with 100% accuracy)?

- K Nearest Neighbours Classifier
- Support Vector Classifier
- Decision Tree Classifier
- K-Means Clustering
- Multi-Layer Perceptron

Task {2} For the purposes of this task, we define a condition to be a mathematical statement only consisting of coordinates x and y , as well as operators $\{+, -, *, /, \% \text{ (modulo)}, <, \leq, =, >, \geq, \text{power operator}\}$, that can evaluate to true or false. For example, $2x - y/3 > 4$ or $x^2 > -(y-x)^3$.

Make a list of such conditions such that answering yes/no on every condition unambiguously identifies the label and implement them in Python.

Format

The first part should be provided as list of yes/no answers corresponding to the classifiers.

For second part, provide a Python file containing a list of lambda functions representing the conditions:

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

A wrong but syntactically valid submission would be

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

Each of the Boolean expressions can include only `x, y, +, -, *, /, //, %, **, <, <=, ==, >=, >` and numerical constants.

Important: Each expression must be a single comparison (e.g., `x + y > 5`). You cannot combine multiple comparisons with arithmetic operators (e.g., `(a > b) * (c < d)` is not allowed), as this would create intermediate Boolean values.

Hints

If you don't know how some or all of these classifiers work, don't worry. You can research them online. Here is a useful video explaining each of them (and more), which can be a good starting point: <https://www.youtube.com/watch?v=E0Hmnixke2g>

Grading

- 25% for correct task {1},
- 75% for correct task {2}.

Task {1}

- number of correct answers / 5

Task {2}

- For x number of conditions, awarded is 1.776^{2-x} .
- Incorrect format OR wrong classification OR python error - 0