

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks

Imagine you have a dataset with different points, and you want to draw a line to separate these points into different categories. But you want to draw the line in such a way that it maximizes the distance between the line and the nearest points of each category.

Imagine you have two categories of points, let's say cats and dogs.

Each point in your dataset represents a pet, and it has some features like size and weight.

SVM tries to find the best line (or hyperplane) that separates the cats from the dogs, in a way that the distance between the line and the closest cat and dog points is maximized.

These closest points, called support vectors, play a crucial role in determining the line. They are the data points that are closest to the line and influence its position.

SVM can handle not only straight lines but also curved or more complex decision boundaries.

It does this by using a mathematical technique called the kernel trick. The kernel trick transforms the original features into a higher-dimensional space, where the points might become more easily separable.

Once the line or decision boundary is found, you can use it to classify new, unseen data points.

For example, if you have a new pet and want to determine if it's a cat or a dog, you can check which side of the line it falls on.

Load the Iris dataset:

Split the data into training and testing sets: Split the dataset into two parts:

one for training the SVM model and another for evaluating its performance.

Create an SVM model: Instantiate an SVM model using the SVC class from scikit-learn.

Train the SVM model: Fit the SVM model to the training data using the fit() function.

Make predictions: Use the trained model to make predictions on the test dataset