Support Vector Machines

SVM is essentially a classification algorithm but can be used for regression problems as well

The main idea behind SVM is to find a way to separate two or more classes in the best possible manner. Once the way is decided, we can use it to predict future data

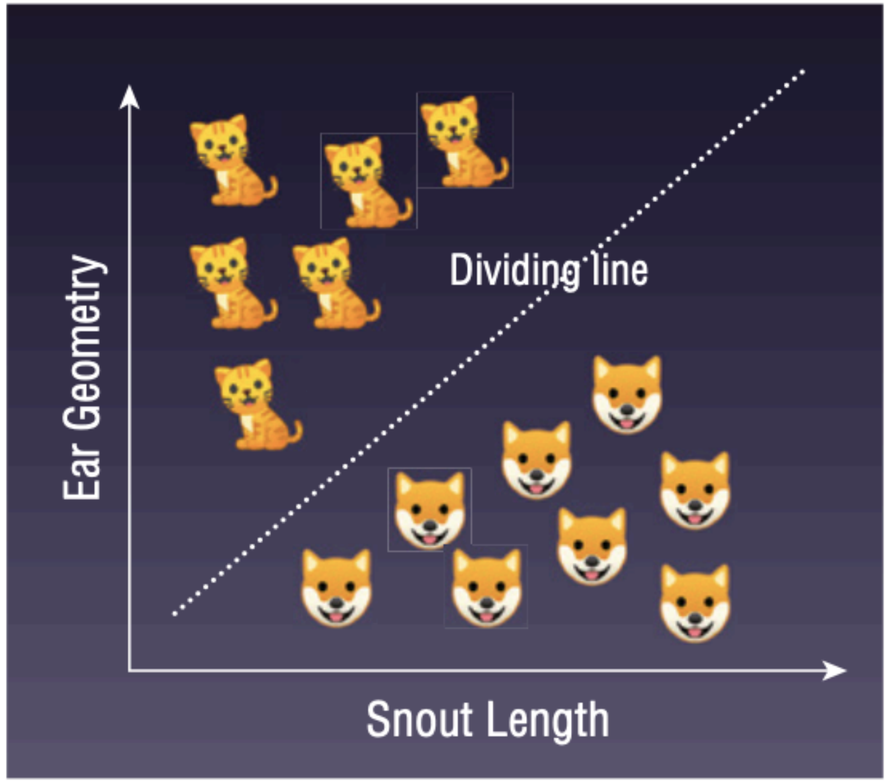


Figure 8.1: Using SVM to separate two classes of animals

For example, given the length and the ear geometry of a new unknown animal, you can now use this dividing line as a classifier to predict whether the animal is a cat or a dog

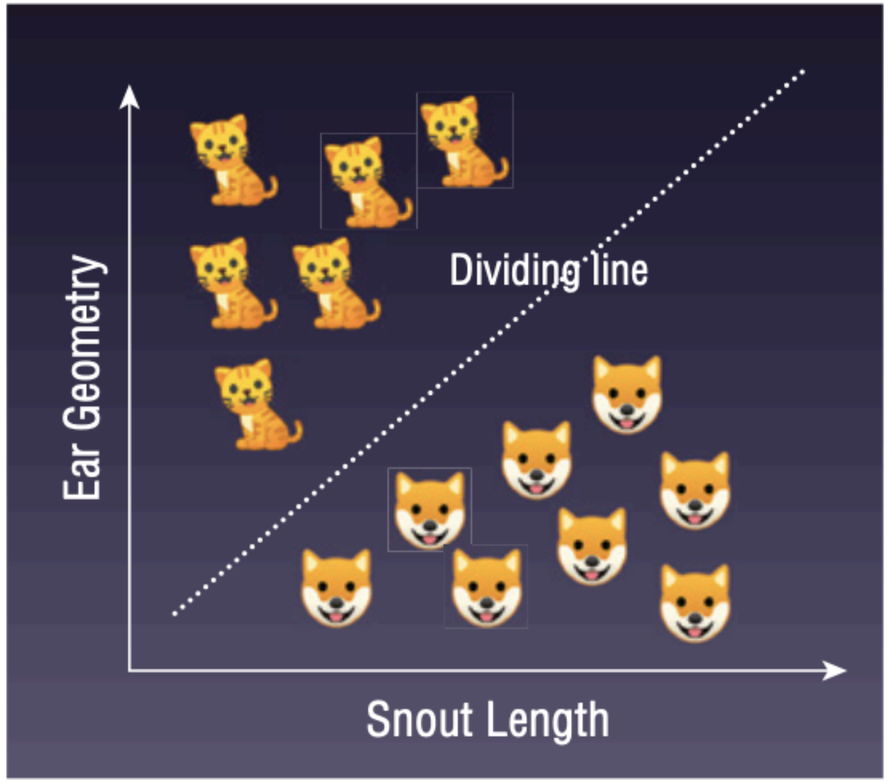


Figure 8.1: Using SVM to separate two classes of animals

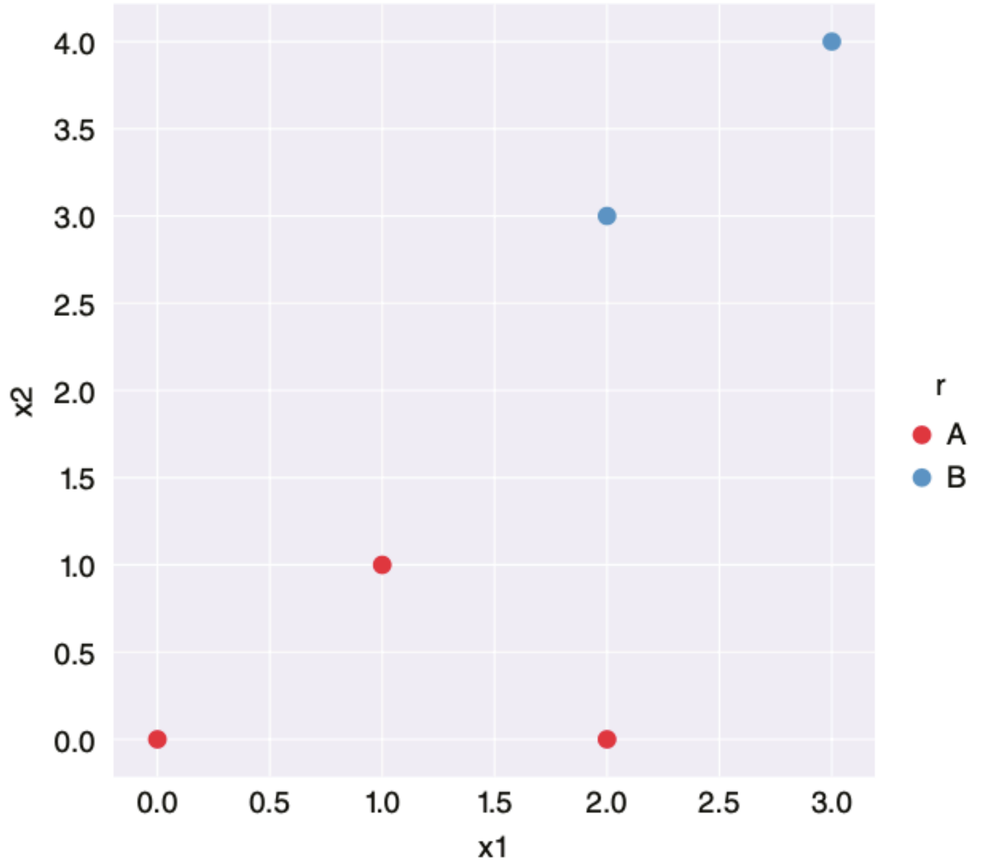


Figure 8.2: A set of points that can be separated using SVM

Consider the set of points in the figure and now visually think of a straight line dividing the points into two groups

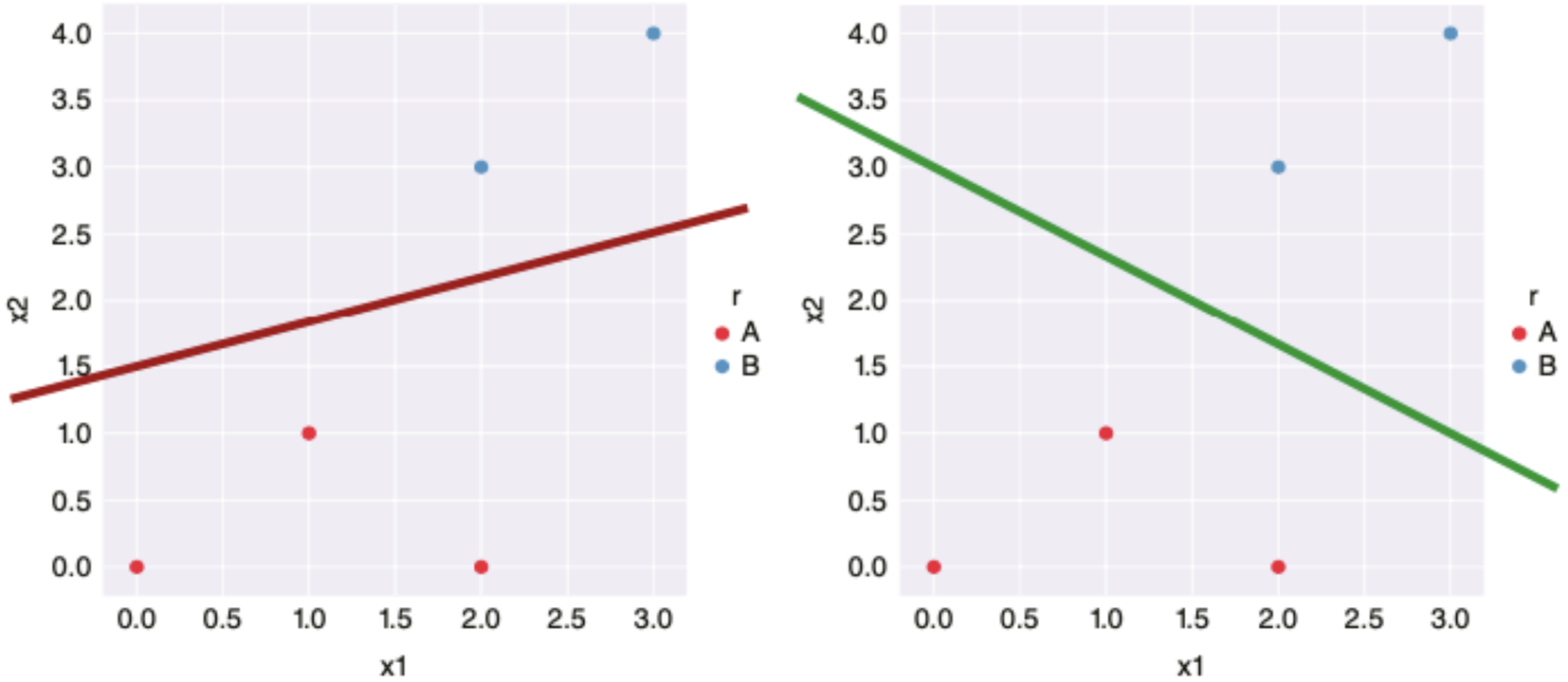


Figure 8.3: Two possible ways to split the points into two classes

Though both lines separate the points into two distinct groups, which is the right one?

For SVM, the better option is the one with the widest margins to separate the two groups

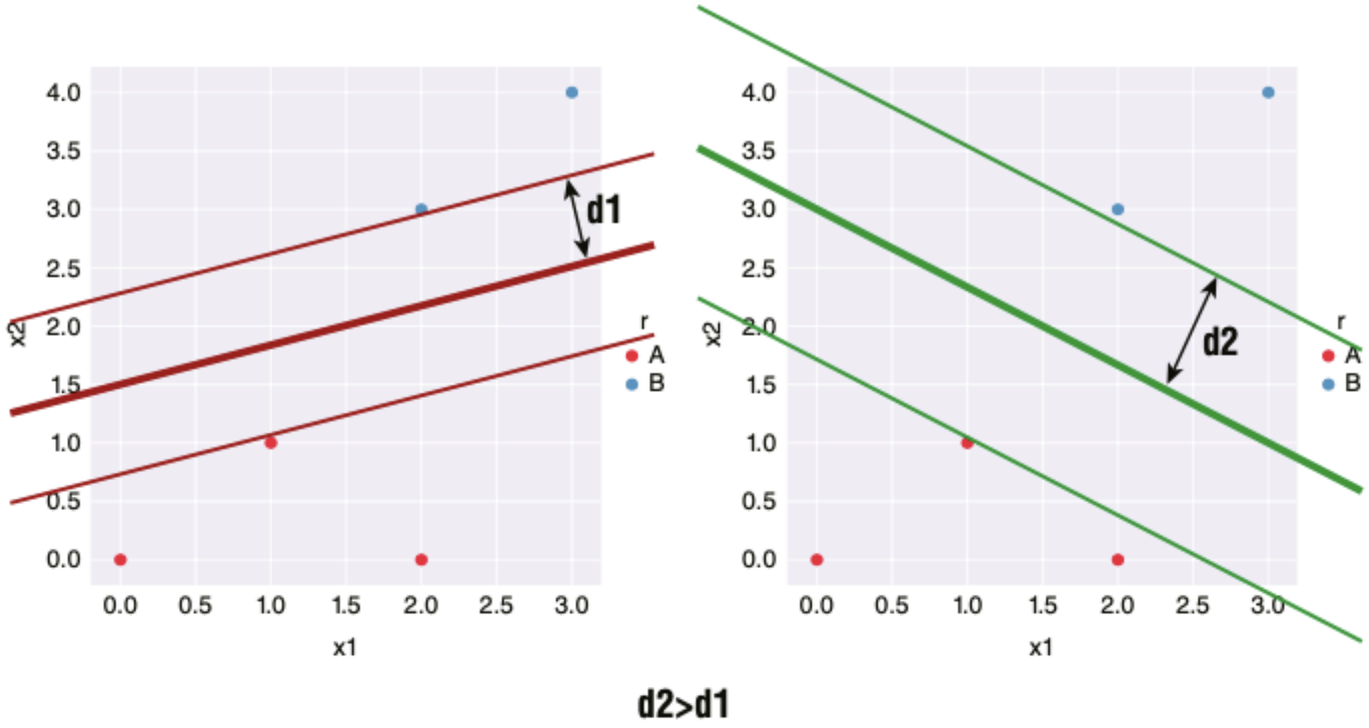


Figure 8.4: SVM seeks to split the two classes with the widest margin

The separating line is called a hyperplane. We use the term hyperplane and not simply line because in SVM we typically deal with more than two dimensions.

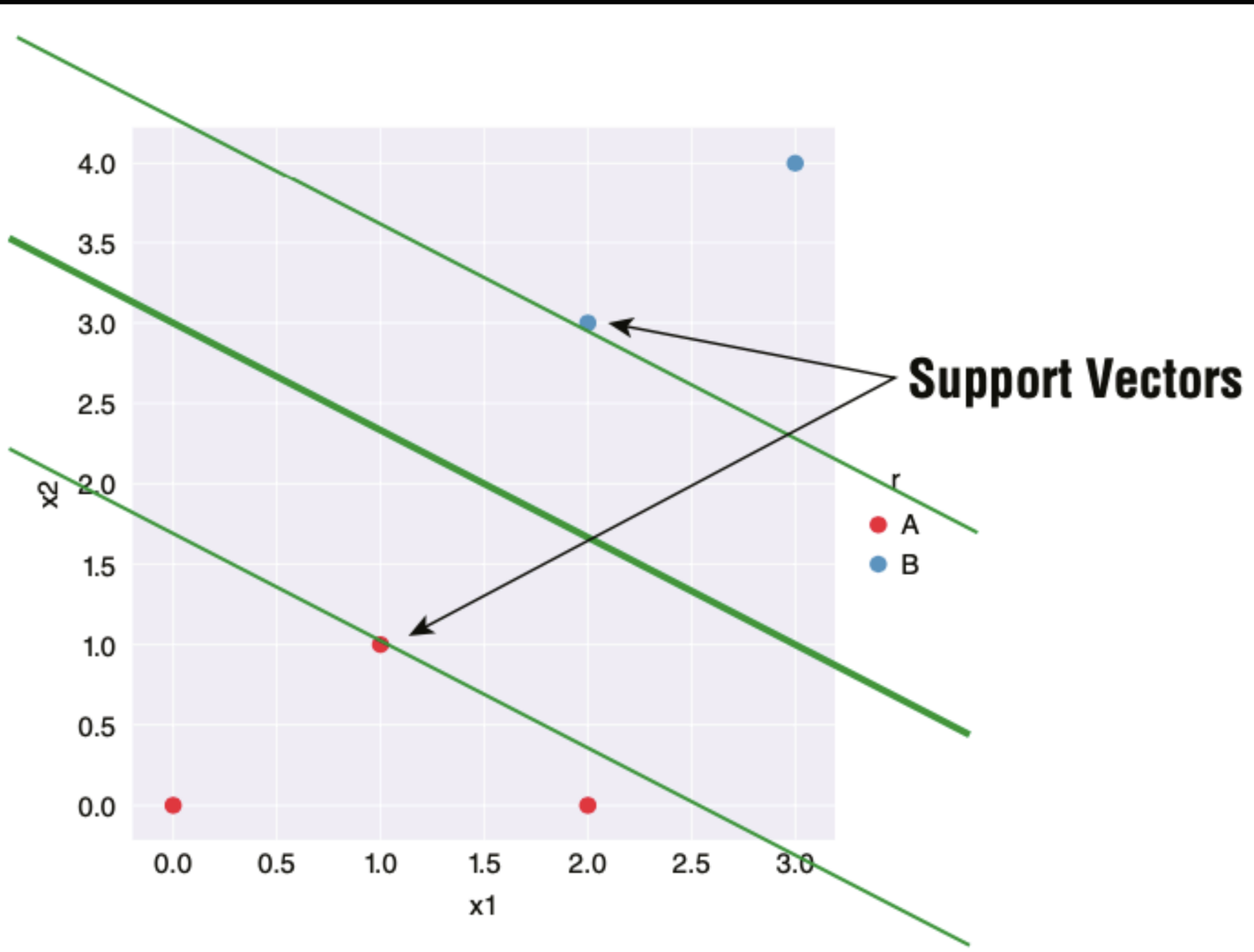
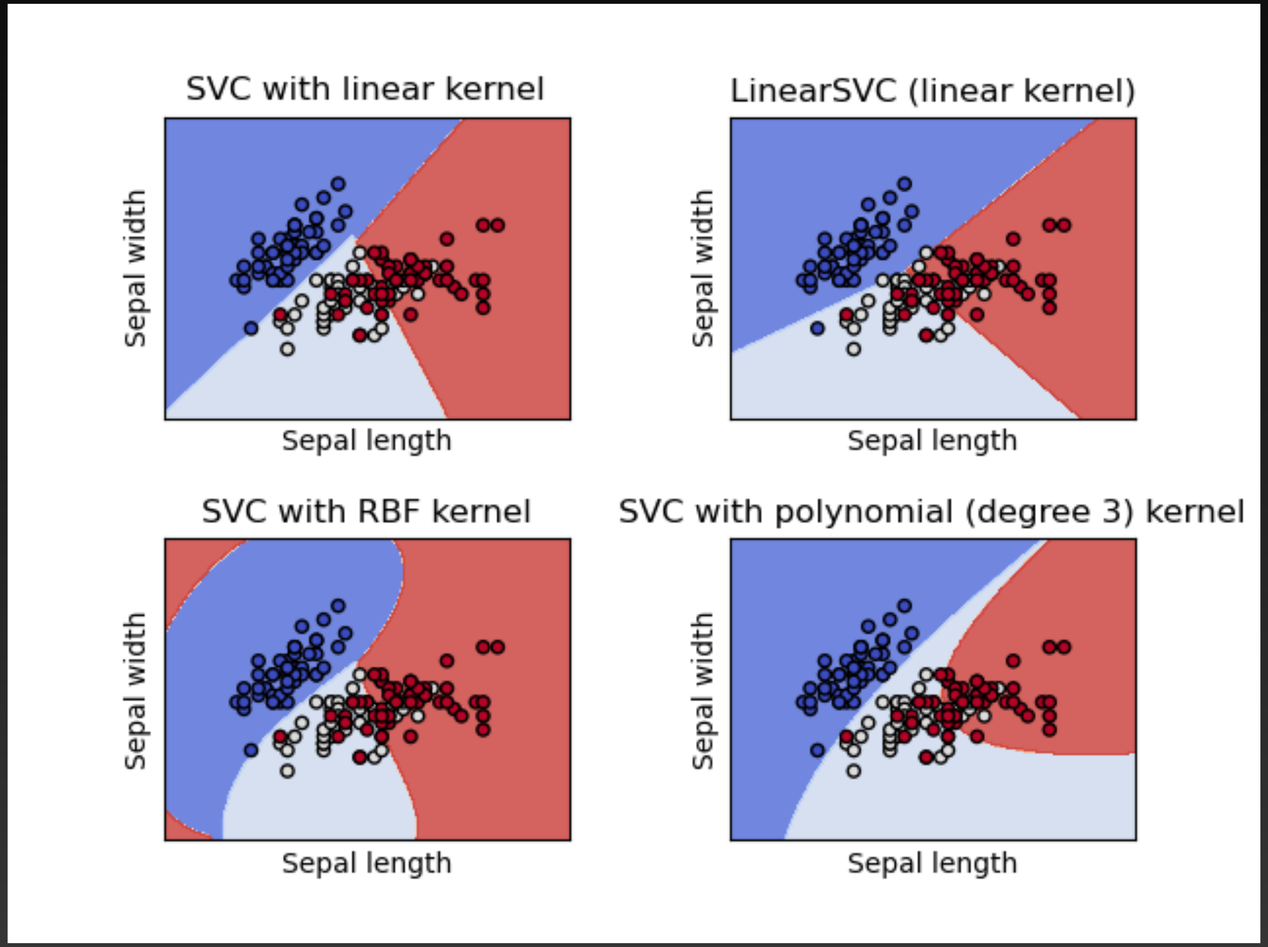


Figure 8.5: Support vectors are points that lie on the margins

Types of SVM Kernels:



Hyperparameter Tuning

C is the penalty parameter of the error term. It controls the tradeoff between the smooth decision boundary and classifying the training points correctly. For example, if the value of C is high, then the SVM algorithm will seek to ensure that all points are classified correctly. The downside to this is that it may result in a narrower margin.

In contrast, a lower C will aim for the widest margin possible, but will result in some points being classified incorrectly.

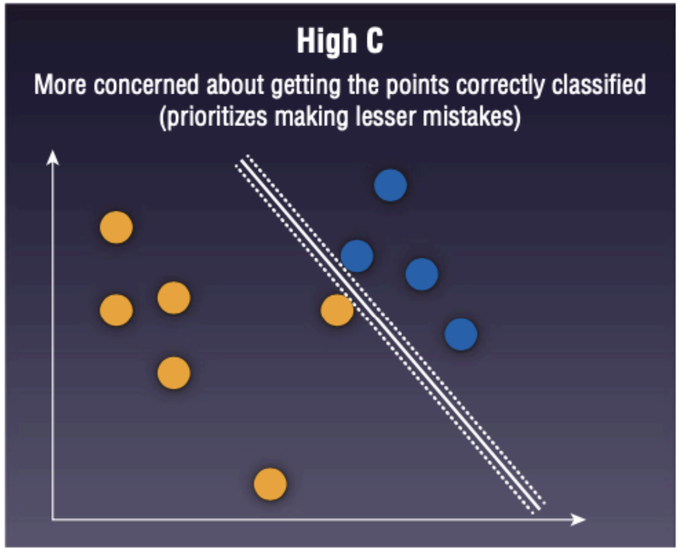


Figure 8.20: A high C focuses more on getting the points correctly classified

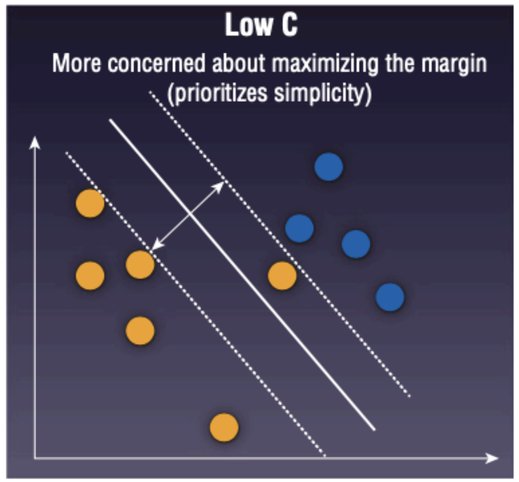


Figure 8.21: A low C aims for the widest margin, but may classify some points incorrectly

Gamma defines how far the influence of a single training example reaches

The higher the value of the gamma, the more it will try to fit to the training data and might result in overfitting

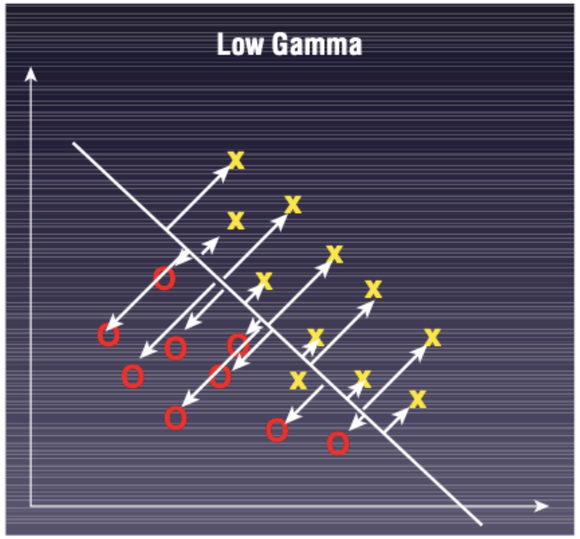


Figure 8.25: A low Gamma value allows every point to have equal reach

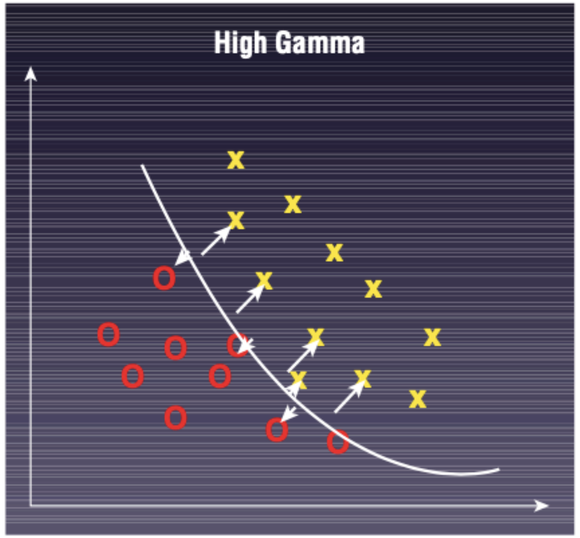


Figure 8.26: A high Gamma value focuses more on points close to the boundary