17.5.1

Overview of Support Vector Machines

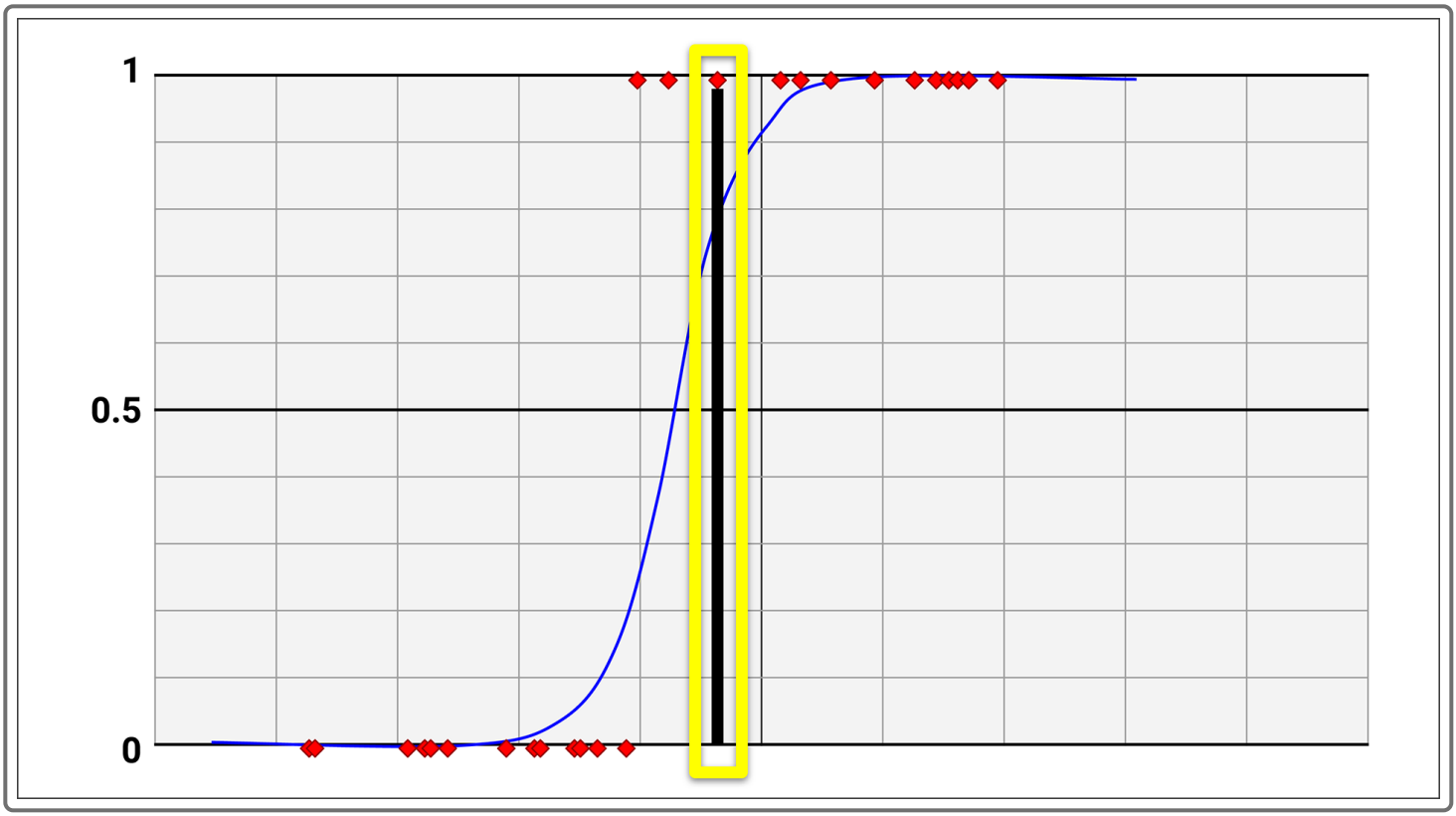
You've been working hard to understand supervised learning from all angles: theory, practice, and statistics. You're beginning to reap the rewards of your hard work as you hit your stride.

Now that you're becoming comfortable with using logistic regression and evaluating its results, Jill suggests that you learn about another powerful classification model: support vector machines. Although the name is possibly a little intimidating, you'll be able to bring much of your previous knowledge into using a support vector machine in practice.

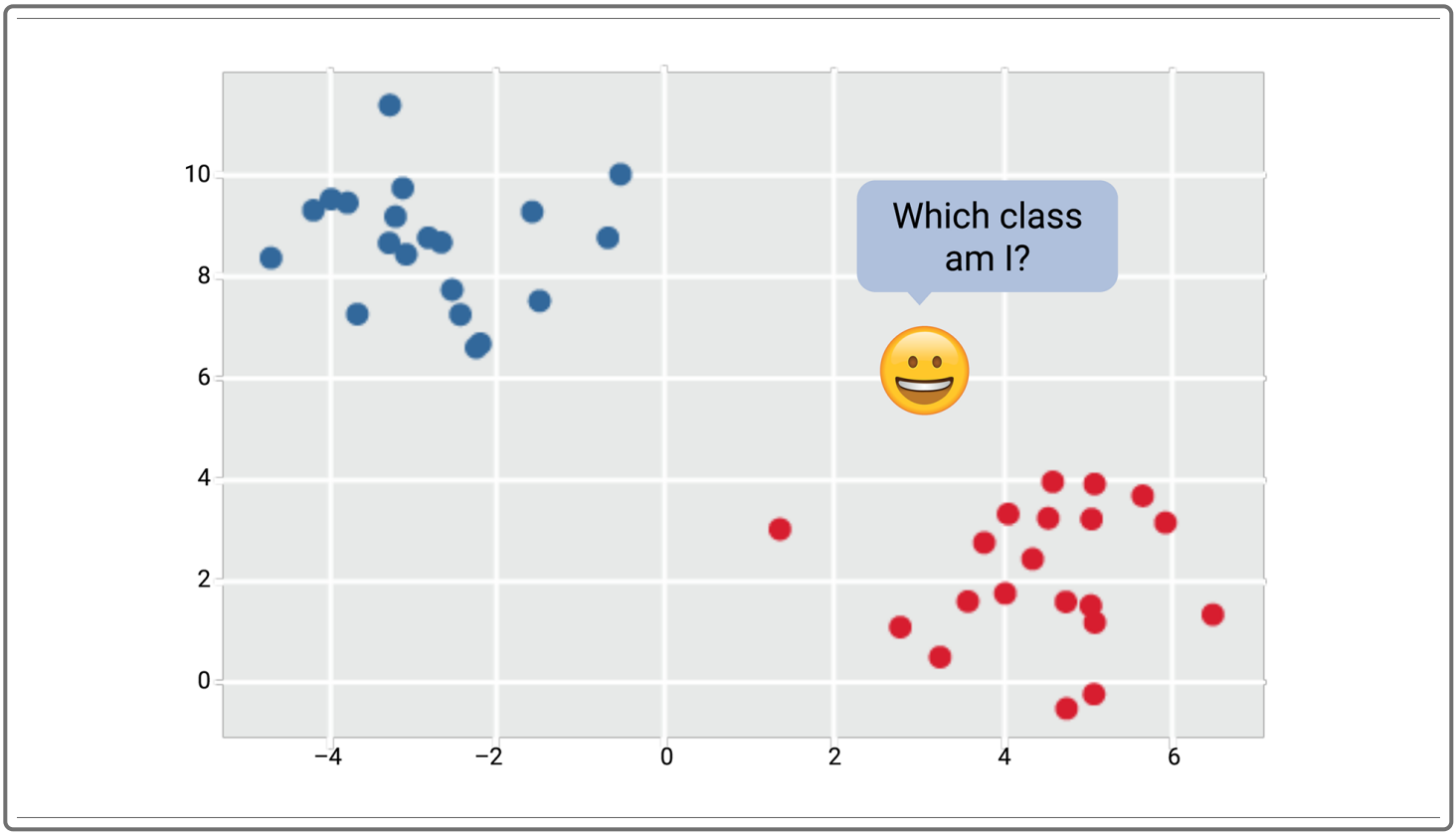
Support vector machine (SVM), like logistic regression, is a binary classifier: It can categorize samples into one of two categories (for example, yes or no).

To understand support vector machines, let's revisit logistic regression first. A logistic regression model evaluates the probability of an occurrence. For example, the model would take features into account (for example, an applicant's income and credit score) and decide whether to approve the application.

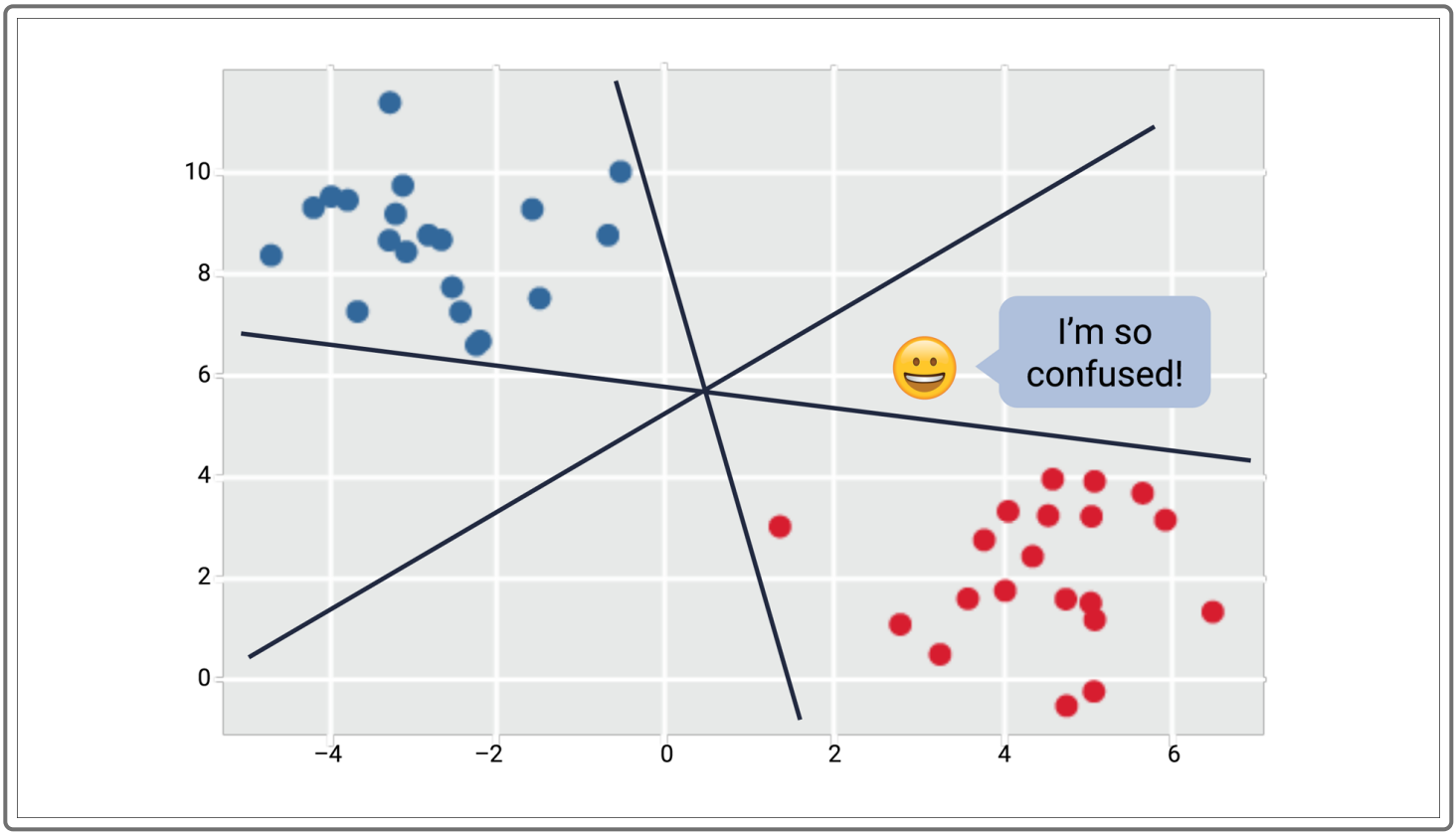
The outcome is binary because the only possible options are to approve or to deny the loan application: If the probability is higher than 0.5, the application is classified as approved, or if the probability is less than that, the application is classified as denied. There is a strict cutoff line that divides one classification from the other:



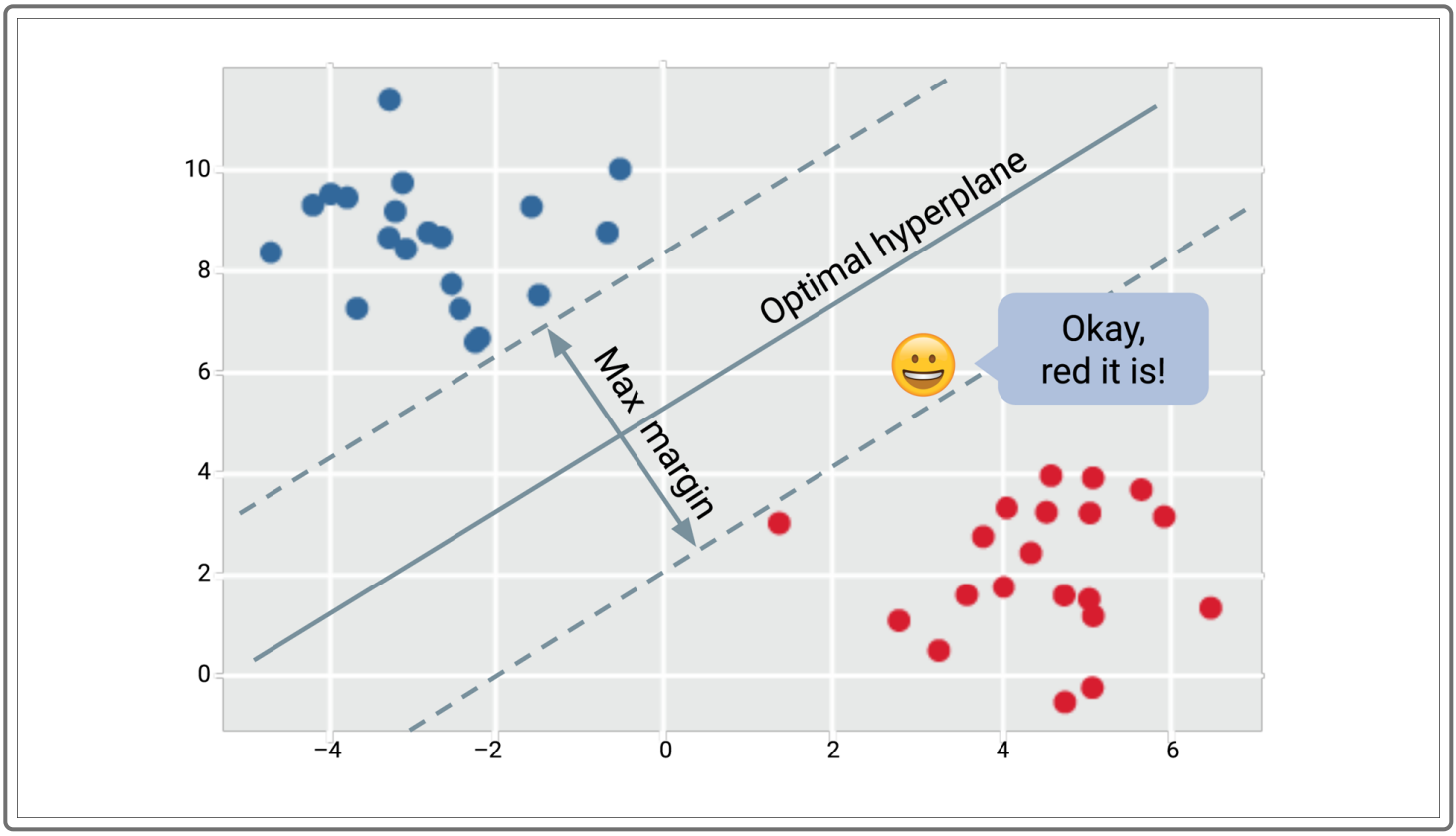
SVM also categorizes the target variable into one of two classes (for example, approved or denied). However, it differs from logistic regression in several ways. As a linear classifier, the goal of SVM is to find a line that separates the data into two classes:



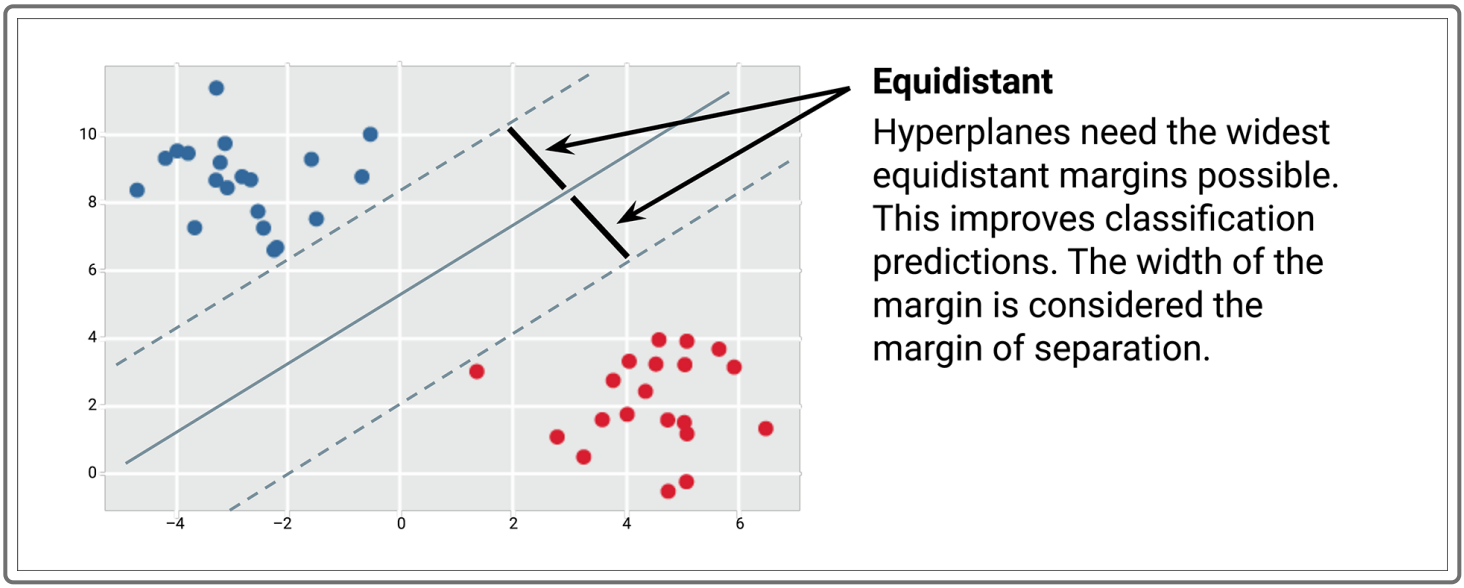
However, there may be many different ways to draw the boundary line, as shown in the diagram below. Which boundary to choose isn't always clear from visual inspection, and choosing the wrong boundary can affect the performance of the model:



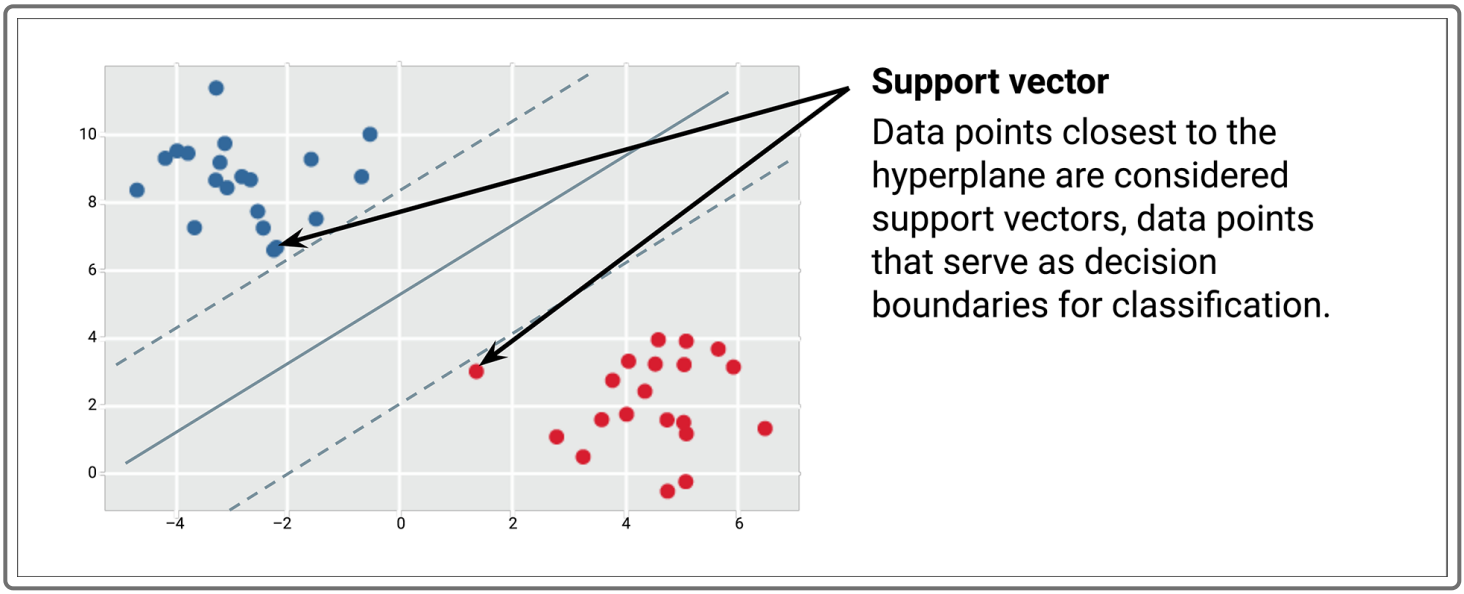
In a two-dimensional grid, as shown below, SVM draws a line at the edge of each class, and attempts to maximize the distance between them. It does so by separating the data points with the largest possible margins:



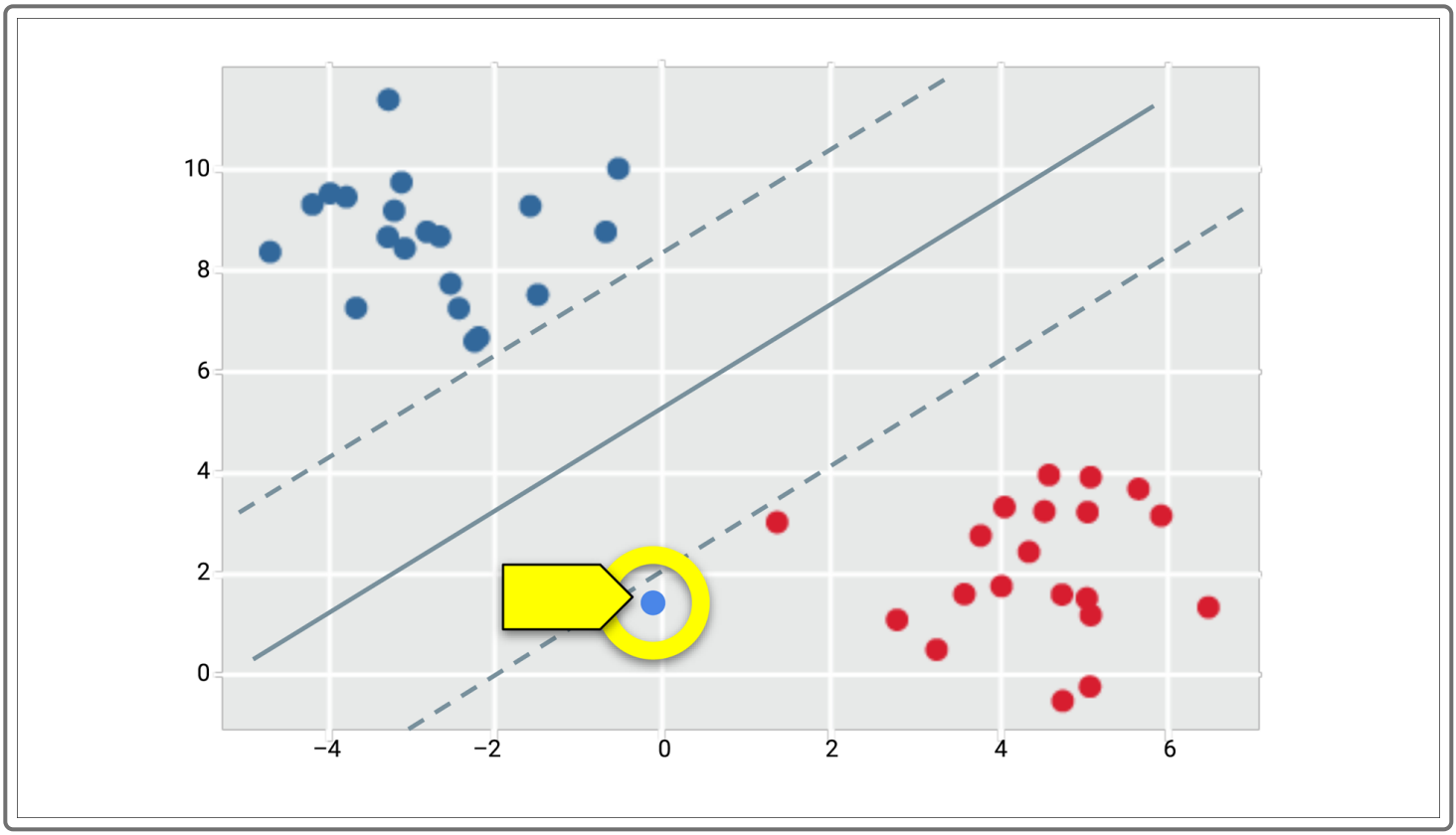
A hyperplane is the line exactly between the two margins (i.e., equidistant from both margins). Again, the SVM's goal is to find the hyperplane with the widest possible margins (i.e., the largest margin of separation between the two classes):



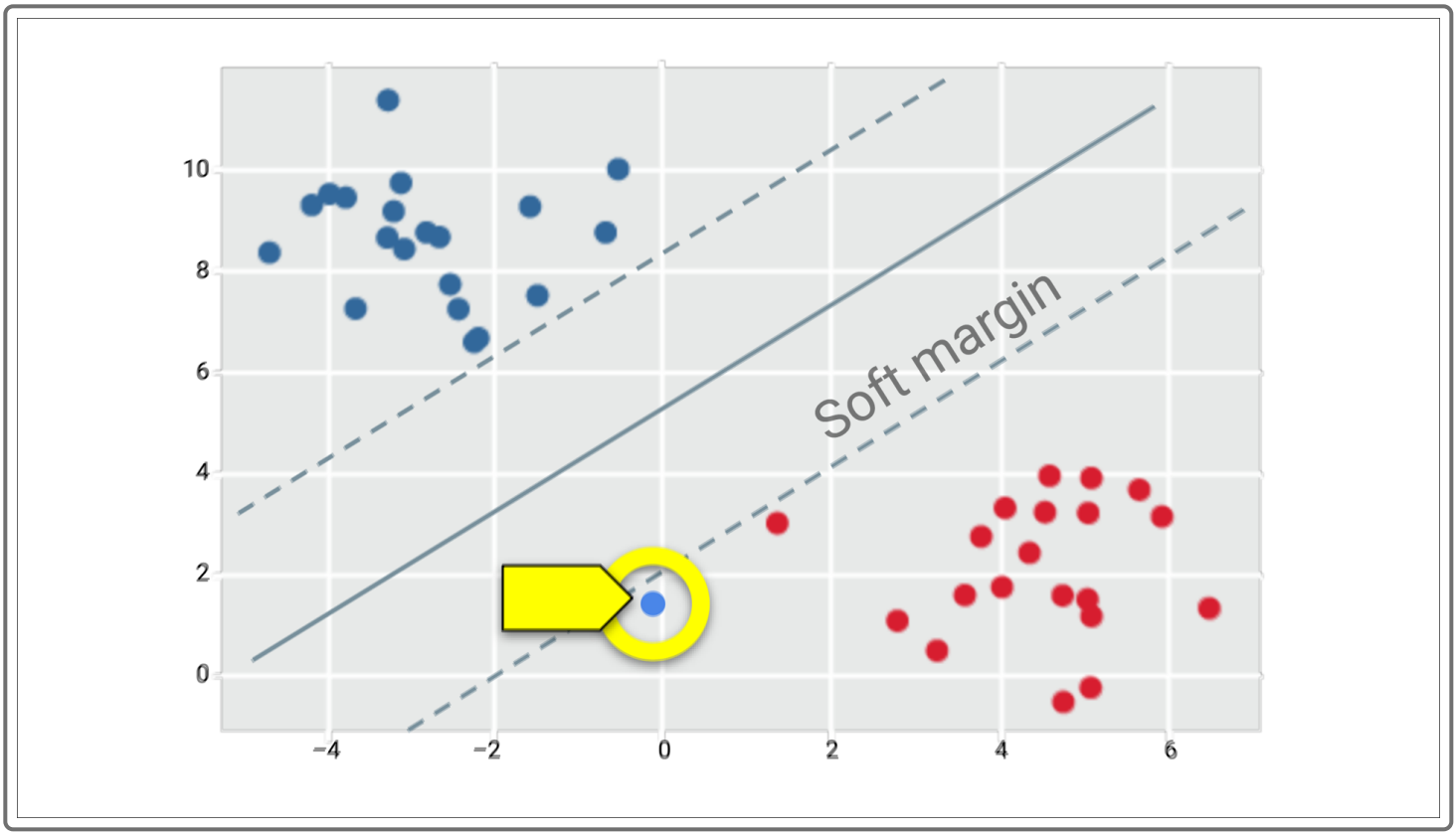
Support vectors are defined as the data points closest to the hyperplane:



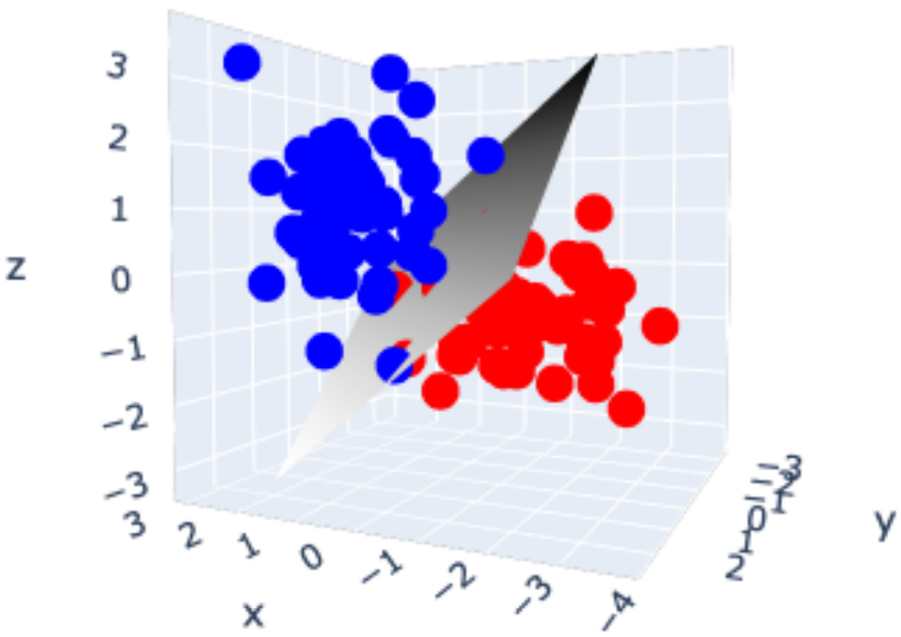
Real-life data, however, can be messy and will often not yield such a clean line of separation. Imagine that a data point belonging to the blue class were found closer to the cluster of data points that belong to the red class. In this case, would the hyperplane have to be relocated? Would the support vectors have to be redefined?



SVMs can accommodate such outliers by using soft margins. A soft margin allows SVM to make allowances for outliers that cross the hyperplane while maintaining support vectors and hyperplane to maximize the overall separation of the two classes:



Up to this point, we have visualized using SVM in datasets with two features. A dataset with three features (e.g., age, education, income) and a target with two classes (e.g., approval or denial of a loan application) would be visualized as a 3D space, with a hyperplane separating the two classes:



What are support vectors?

- ○ A line exactly between the two margins that is placed at an equal distance from both margins.
- ○ Data points closest to the margin of separation.
- ○ A line that separates the data into two classes

Check Answer

Finish ▶

To summarize, SVM works by separating the two classes in a dataset with the widest possible margins. The margins, however, are soft and can make exceptions for outliers. This stands in contrast to the logistic regression model. In logistic regression, any data point whose probability of belonging to one class exceeds the cutoff point belongs to that class; all other data points belong to the other class.