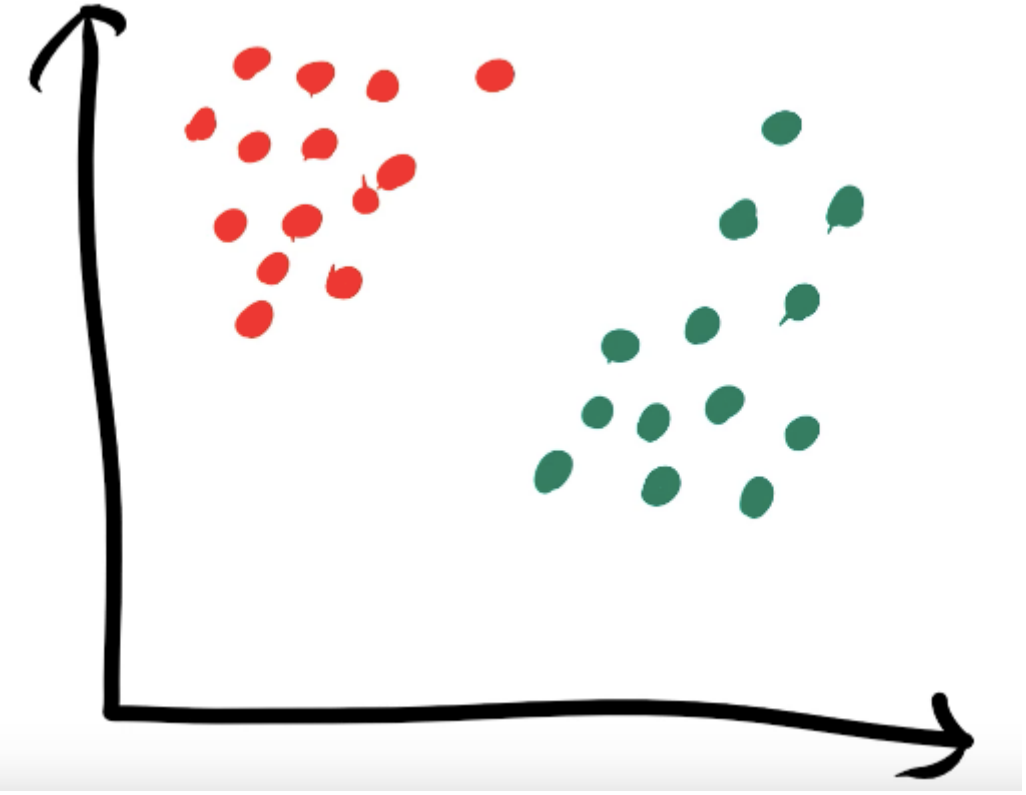
5) SVM

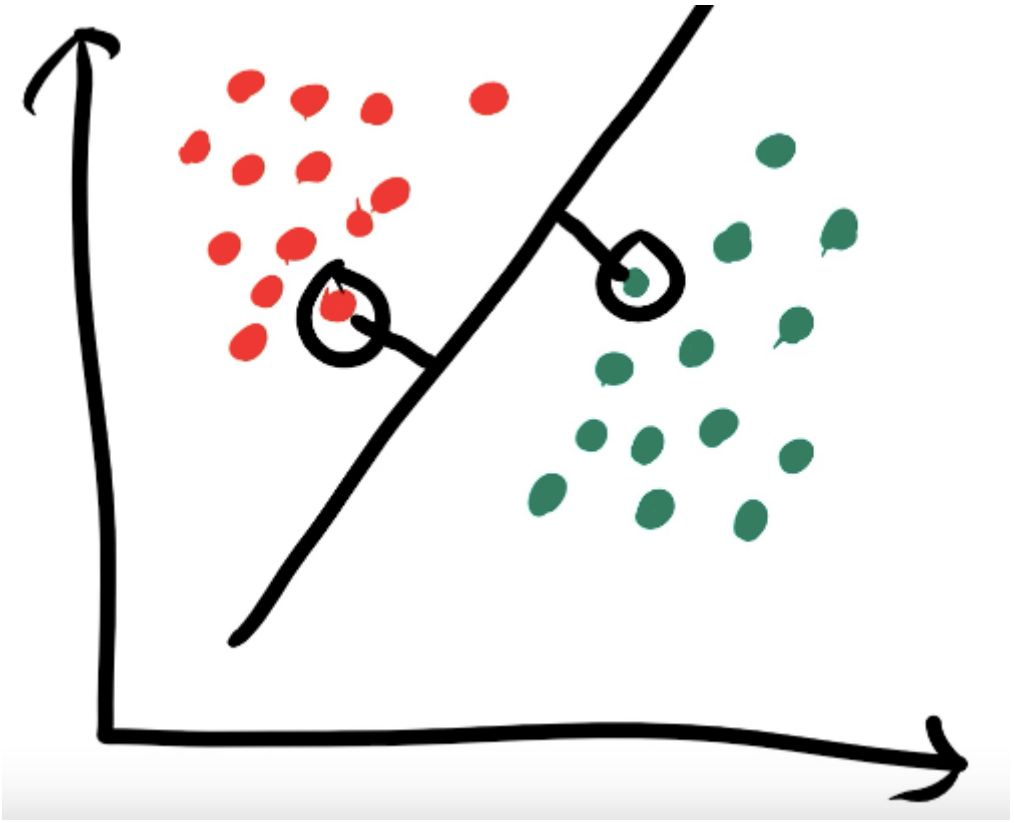
- Scaling is extremely important (https://stats.stackexchange.com/questions/65094/why-scaling-is-important-for-the-linear-svm-classification)

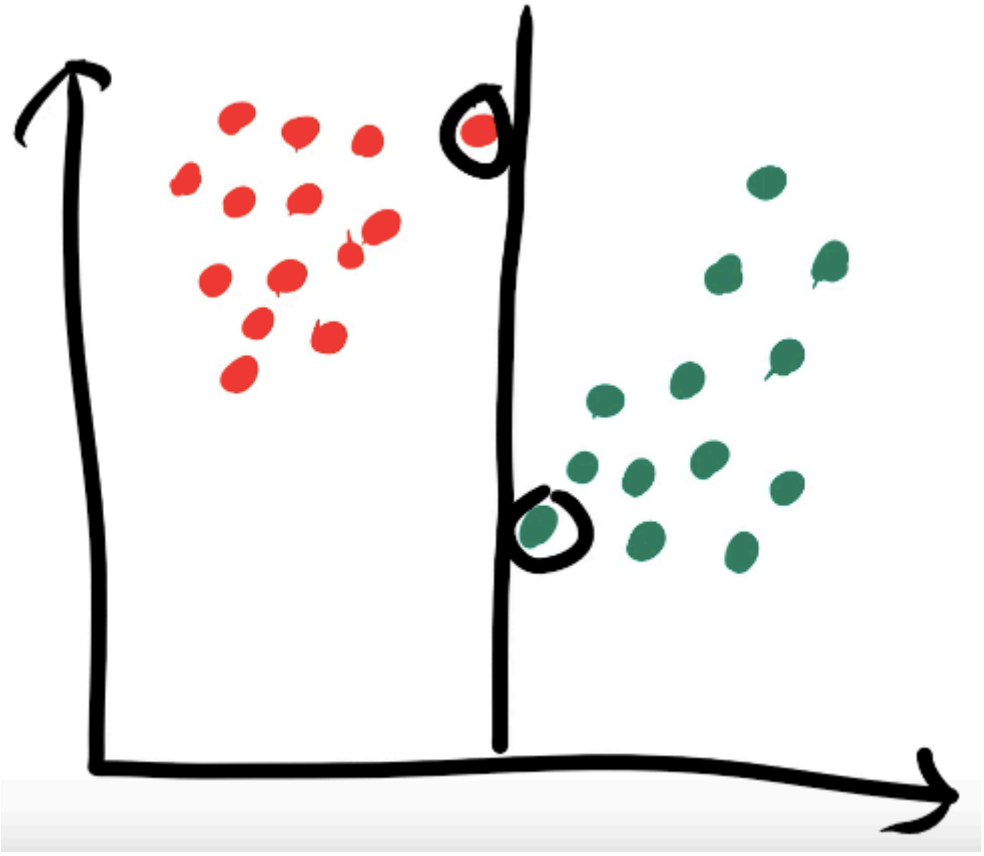
How it works

So basically we plot a "hyperplane" (ie an (n-1)-dimensional object where n is the total number of dimensions, which in most cases will be the number of features) to divide the clusters. A hyperplane should have equal distances to the closet points of different classes. So for example, in a dataset like



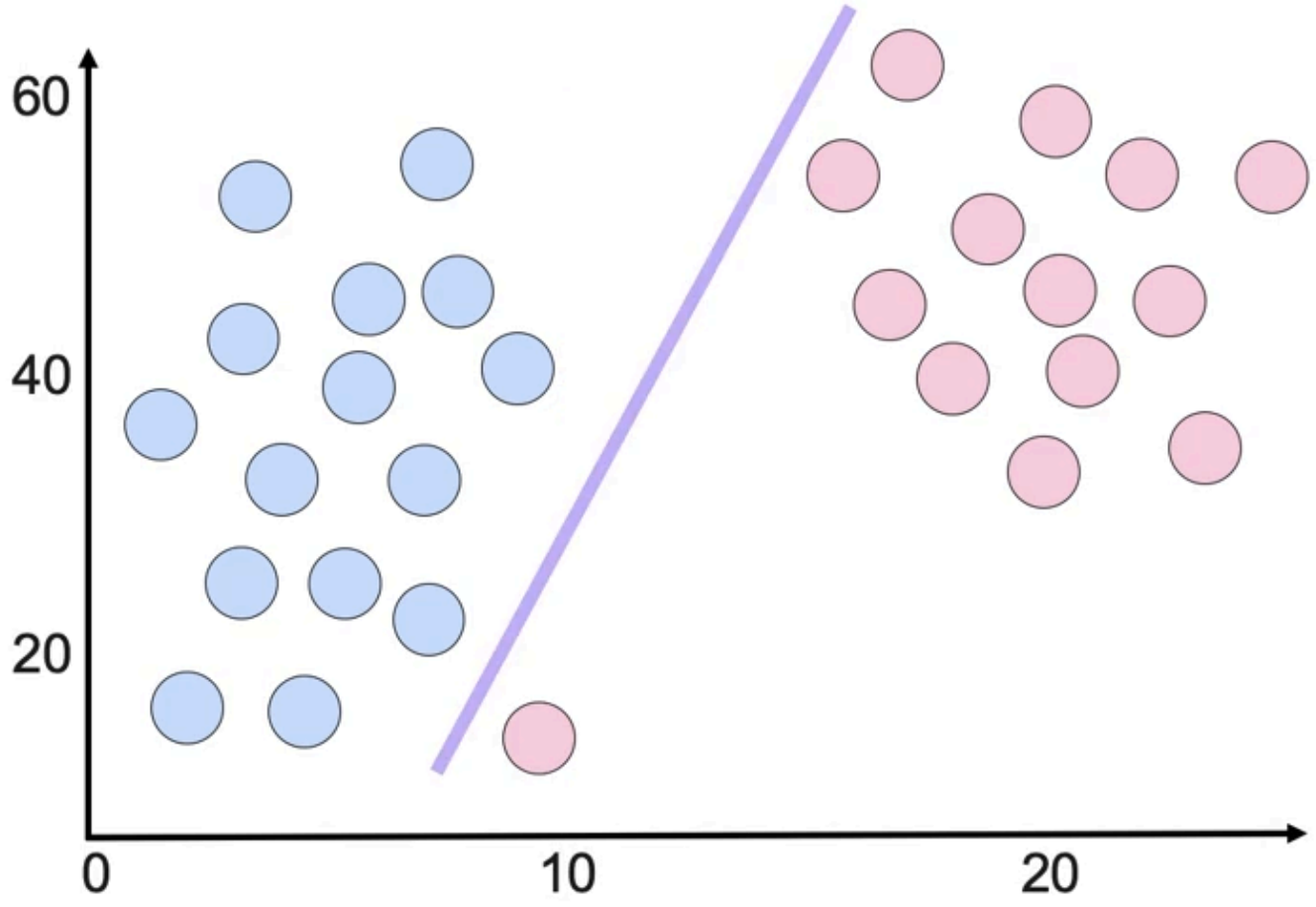
There can be the following hyperplanes:





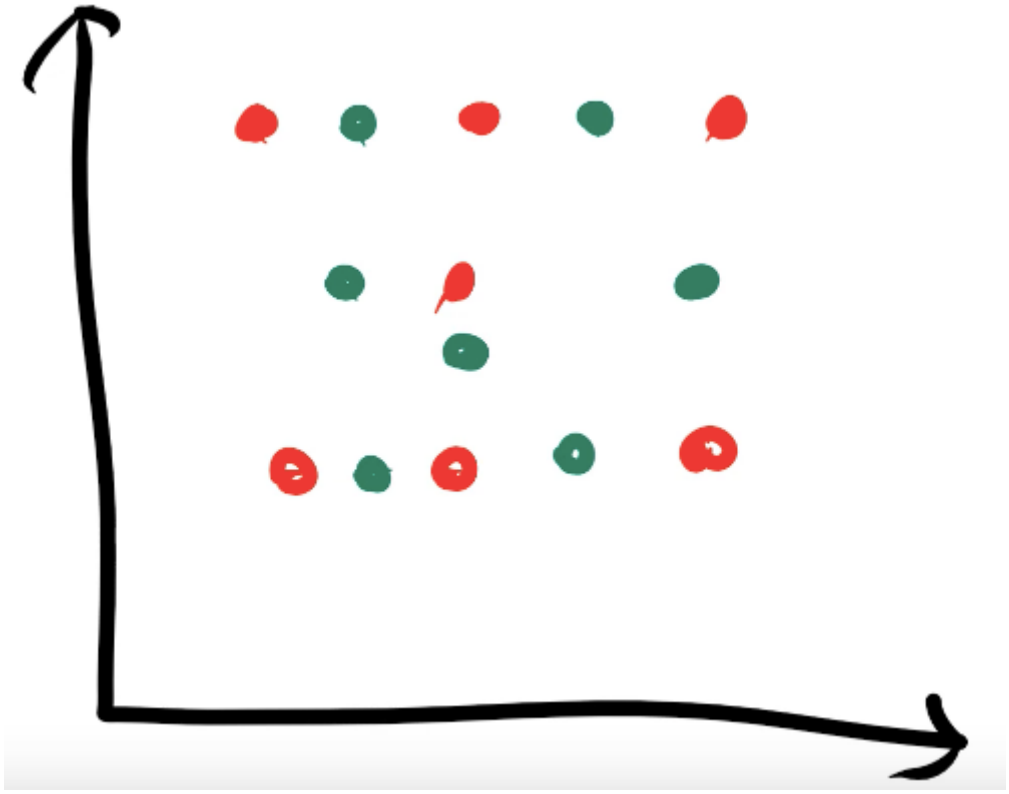
The optimal (ie best) hyperplane would be the hyperplane with the highest **margin**, ie the distance to the closet points of different classes

However, this approach can sometimes overfit when dealing with outliers. For example,

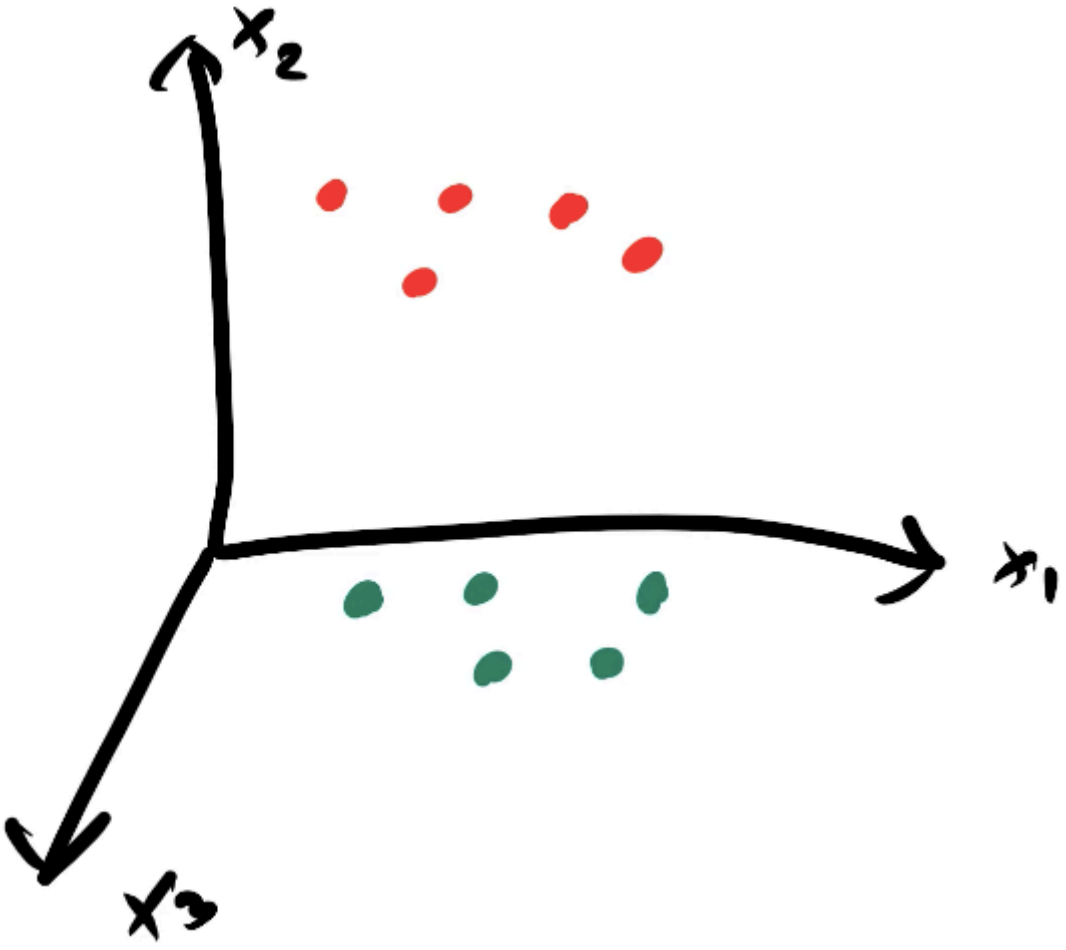


Kernels

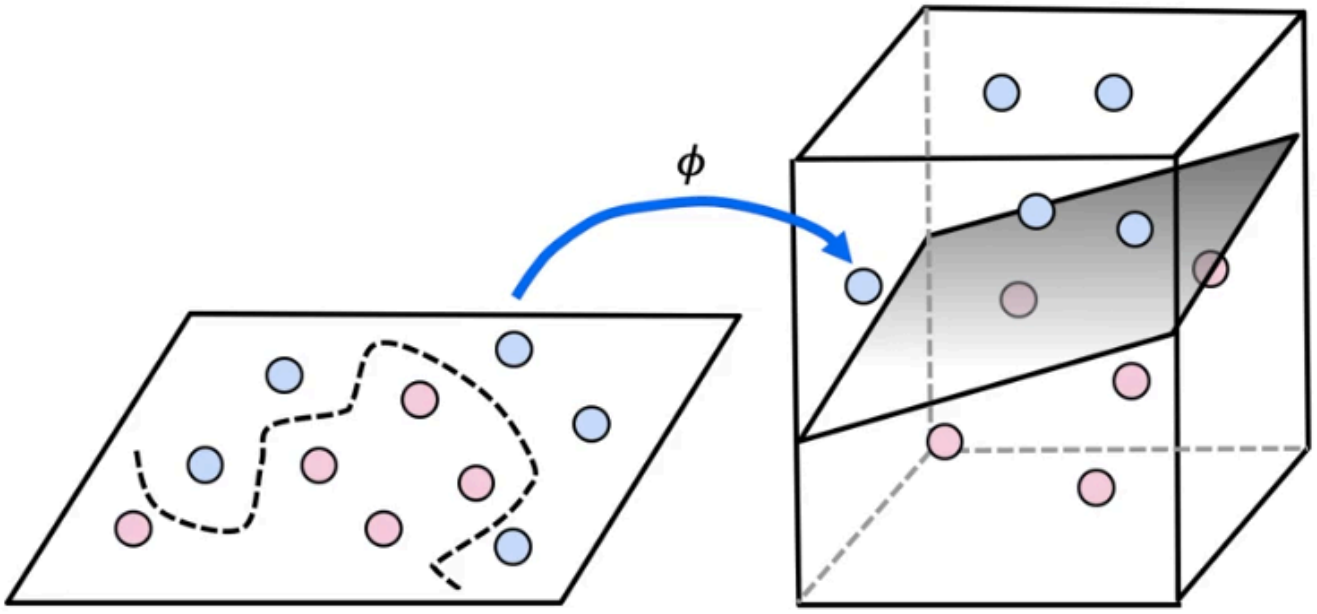
Sometimes, the data looks like



In cases like these we use a "kernel", ie essentially a function which takes in the features as inputs and adds another feature (ie the output of the function), sort of how we added polynomial features in polynomial regression in case there was any correlation between them and the label. So w can use a kernel $f(x_1,x_2)->x_3$ on the above graph so that it can be transformed into:

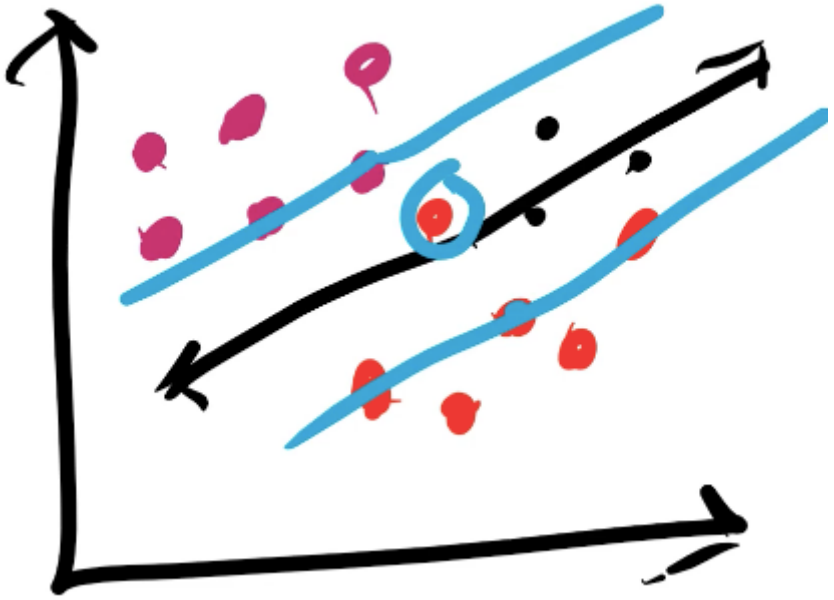


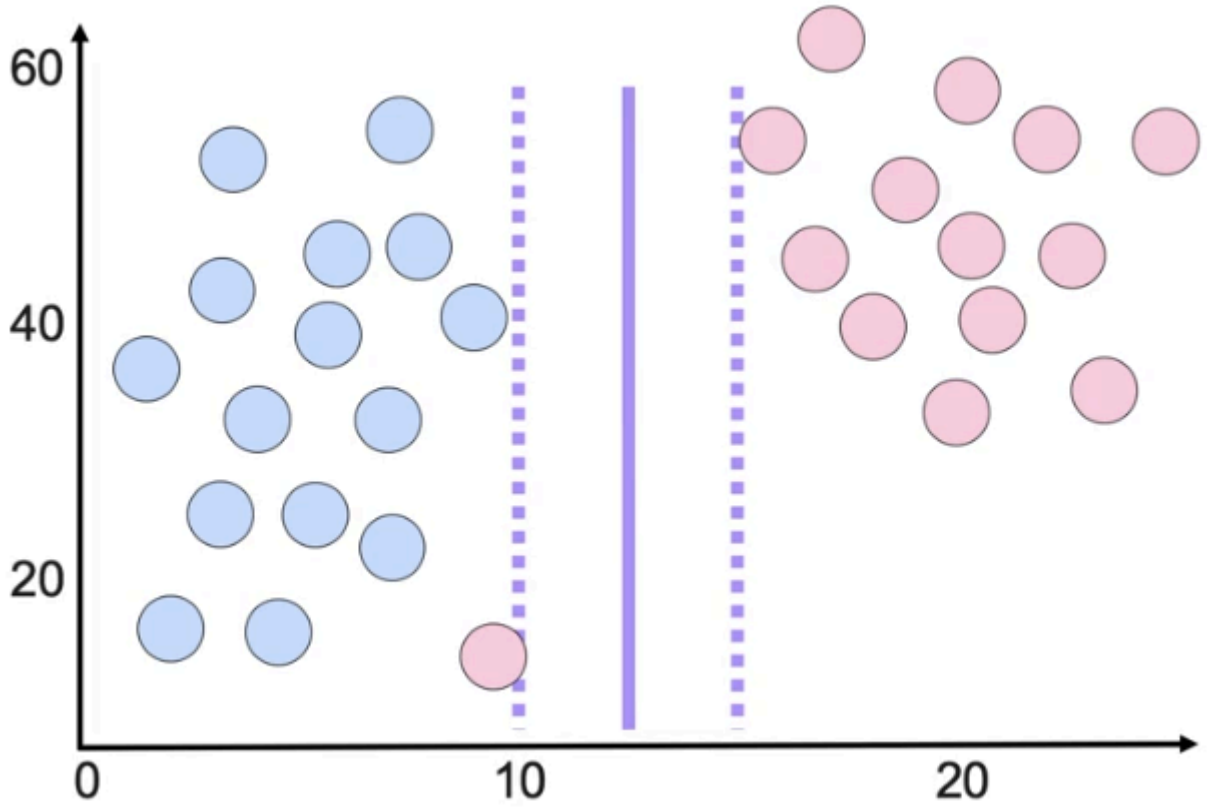
Another example:



Soft margin

A margin is basically a distance to the closet points of different classes. In real world datasets, we cannot find a hyperplane which perfectly divides the data (cause of areas on the graph where both classes collide), so we sometimes use a "soft margin", ie a margin where points of any class can exist. For eg:





Regularisation/Error function

$$J(\beta_i) = SVMCost(\beta_i) + \frac{1}{C}\sum_i \beta_i$$

Here the first term refers to amount of misclassifications of training data by our hyperplane (hinge loss?), and the second term is inversely proportional to the margin, so we gotta minimise both the terms

Hyperparamaters

`kernel`
`penalty` (Which regularisation method we want to use, ie L1, L2, or elastic net)
`c` Lower c value means more regularisation and a simpler model