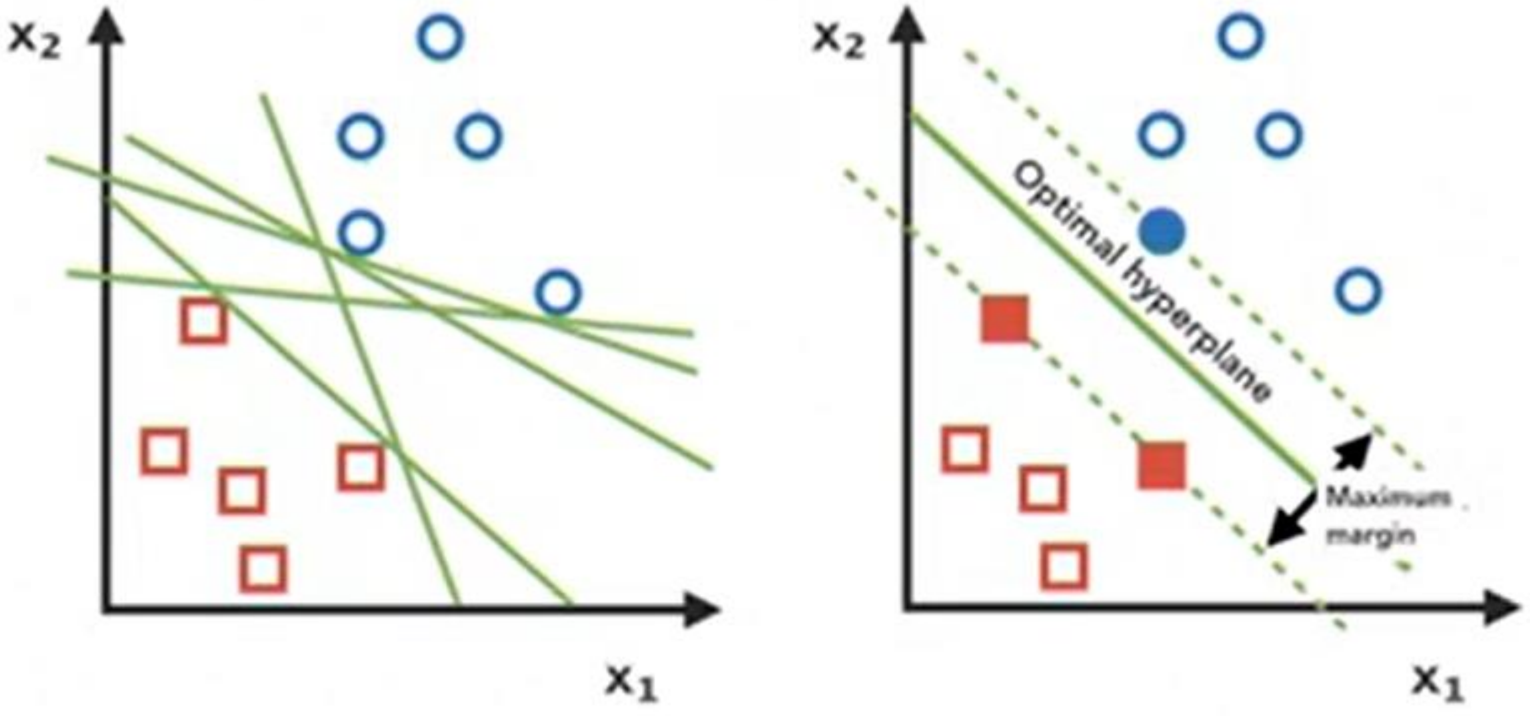
6.05 SVMs

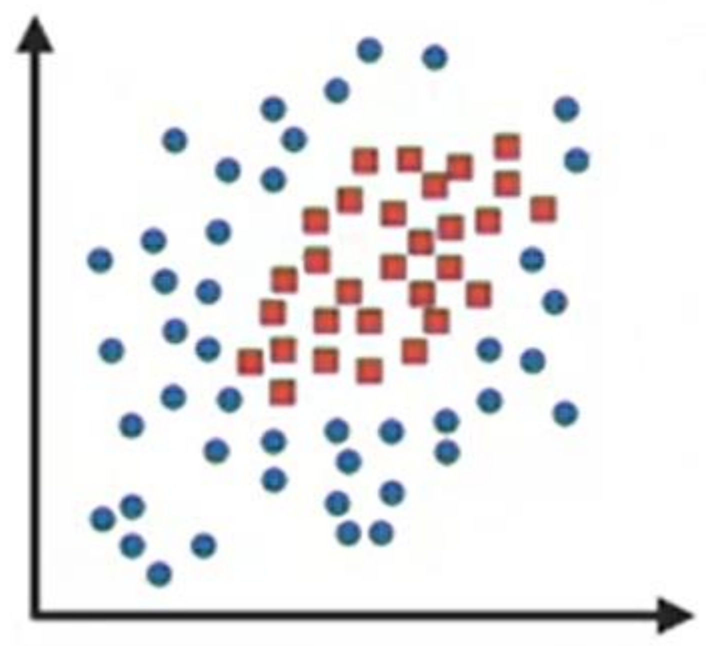
Support Vector Machines (SVMs)

- Classifier that finds an optimal hyperplane that maximises margin between 2 classes

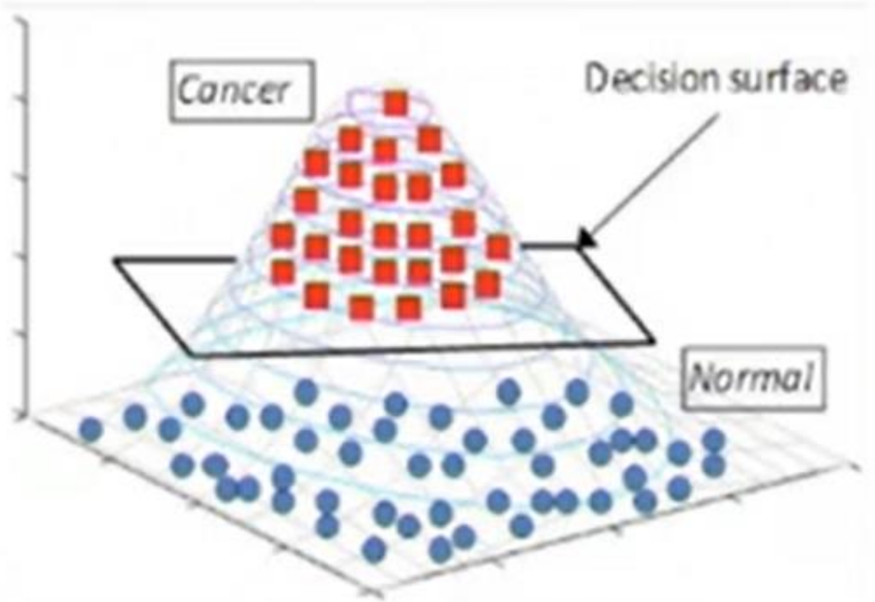


Kernel Trick

- Transforms data that is not linearly separable in n-dimensional space to a higher dimension where it is linearly separable



Original: Data is not linearly separable in 2D space



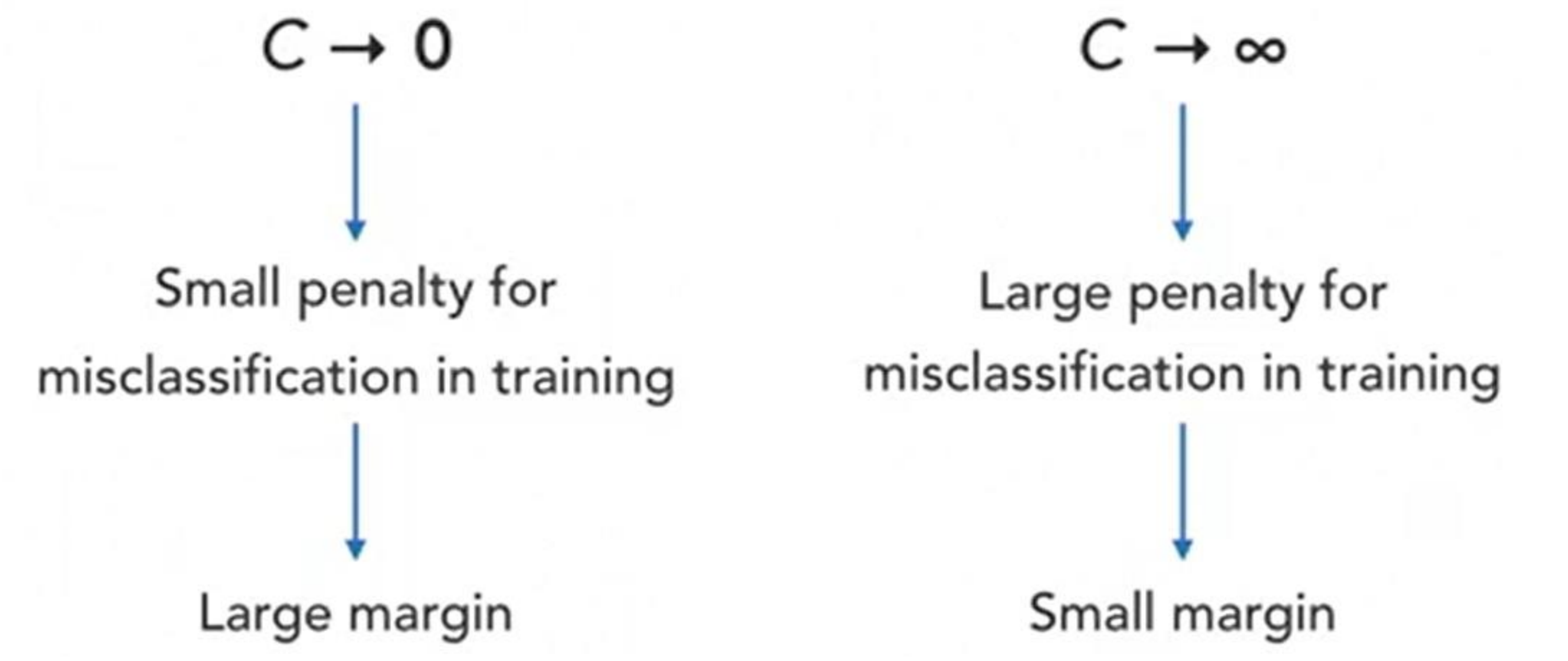
New: Data is now linearly separable in 3D space

How does SVM increase "dimensionality" of the data set e.g. to move from 2D to 3D space?

- SVM leverages a polynomial kernel function to transform data into a higher-dimensional space, where it becomes easier to separate points of diff classes.
- The kernel function computes the similarity between any two data points in the higher-dimensional space.
- The polynomial kernel function is defined as follows:
  - k(x, y) = (x*y + 1)^d
- where x and y are data points in original space, and d is degree of the polynomial.
- For example, suppose we have a 2D data set with the following points:
  - x1 = [1, 2]
  - x2 = [3, 4]
- If we use a polynomial kernel with degree 2, then the similarity between x1 and x2 will be computed as follows:
  - k(x1, x2) = (x1*x2 + 1)^2 = (1*3 + 2*4 + 1)^2 = 25

C Hyperparameter

- Penalty term that determines how closely model fits to training data

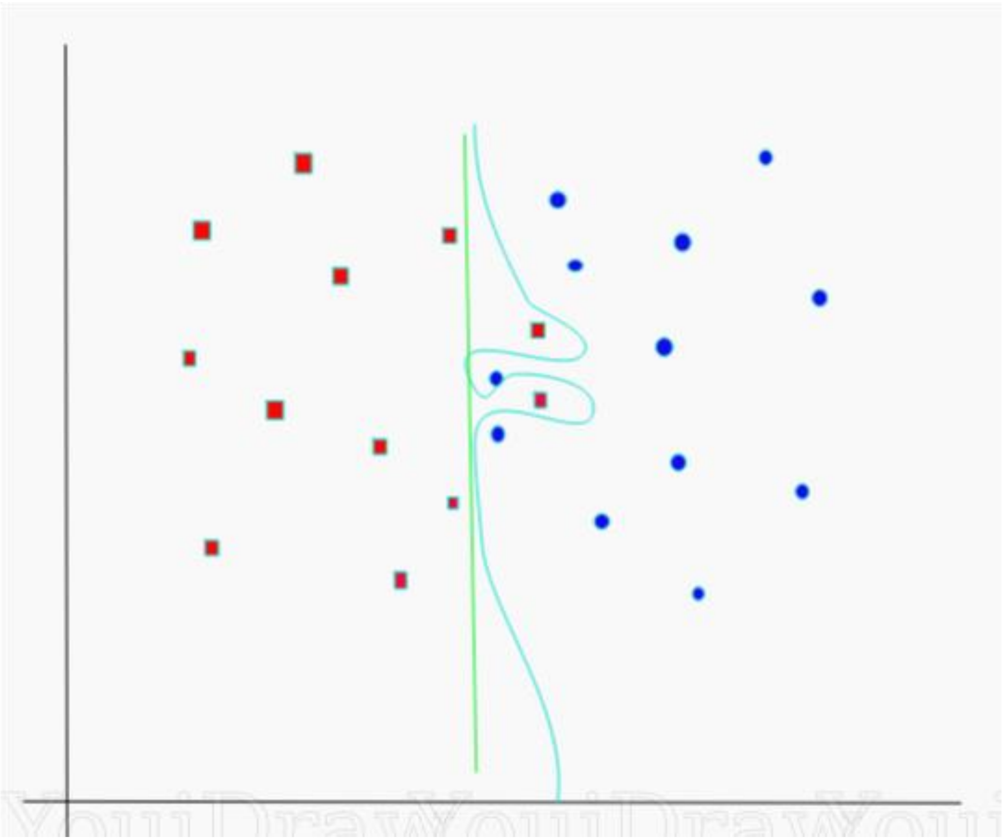


C Hyperparameter

- It controls the trade off between smooth decision boundary and classifying training points correctly.

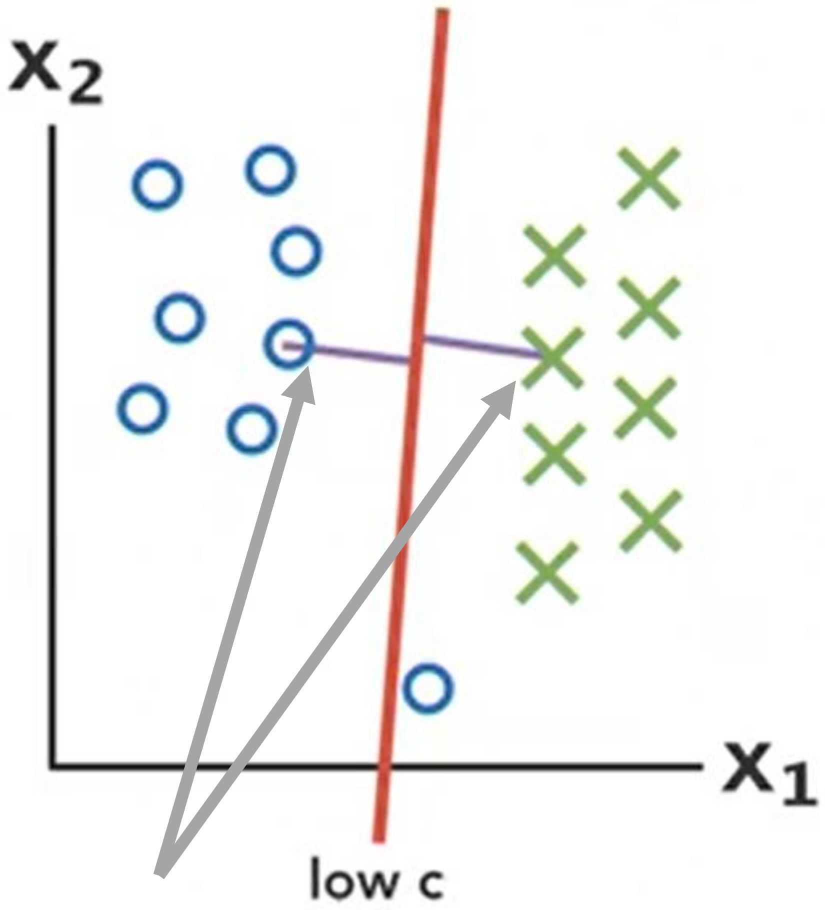
- A large value of c means you will get more training points correctly.

C Hyperparameter

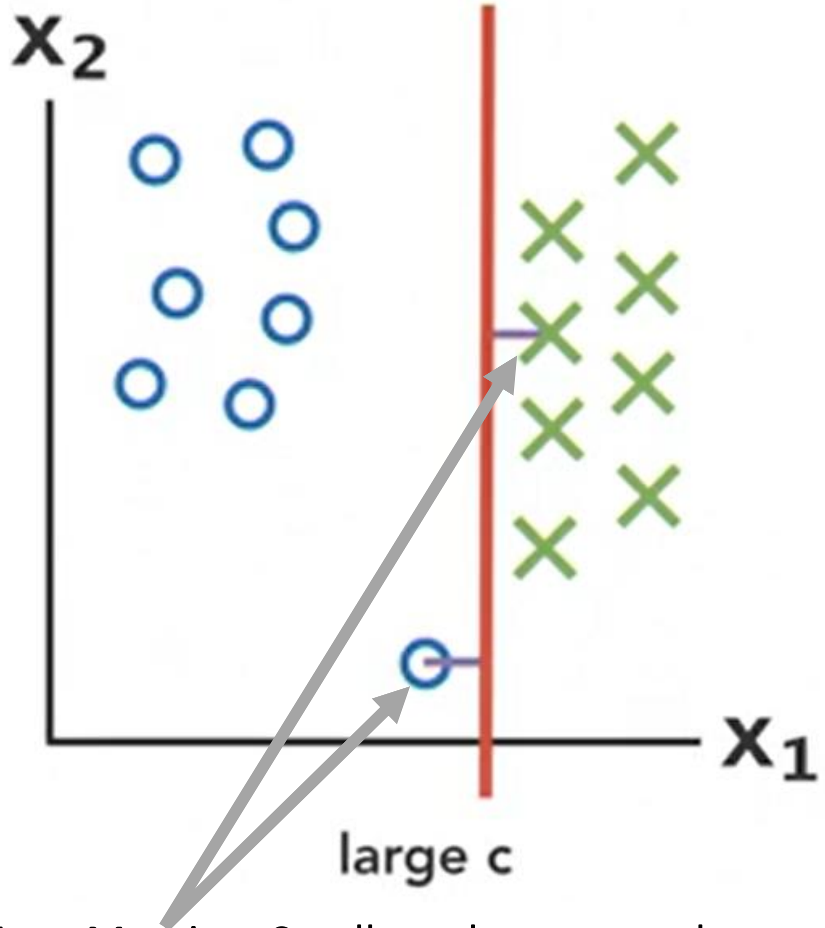


Consider an example as shown in the figure above. There are a number of decision boundaries that we can draw for this dataset. Consider a straight (green colored) decision boundary which is quite simple but it comes at the cost of a few points being misclassified. These misclassified points are called outliers. We can also make something that is considerably more wiggly(sky blue colored decision boundary) but where we get potentially all of the training points correct. Of course the trade off having something that is very intricate, very complicated like this is that chances are it is not going to generalize quite as well to our test set. So something that is simple, more straight maybe actually the better choice if you look at the accuracy. Large value of c means you will get more intricate decision curves trying to fit in all the points. Figuring out how much you want to have a smooth decision boundary vs one that gets things correct is part of artistry of machine learning. So try different values of c for your dataset to get the perfectly balanced curve and avoid over fitting.

How does value of C impact SVM model?



High Margin = Large gap between values and line



Low Margin = Small gap between values and line

SVMs Applications

- When to use?
  - Binary target variable
  - Feature-to-row ratio is very high
  - Very complex relationships
  - Lots of outliers

- When not to use?
  - Feature-to-row ratio is very low
  - Significance of predictors need to be known
  - Looking for a quick model that trains and predicts fast