UNIVERSITY OF ABERDEEN

CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Support Vector Machines

Data Mining & Visualisation

Lecture 14

2025

Today…

- Support Vector Machines
- Kernel functions

Intuition

Intuition

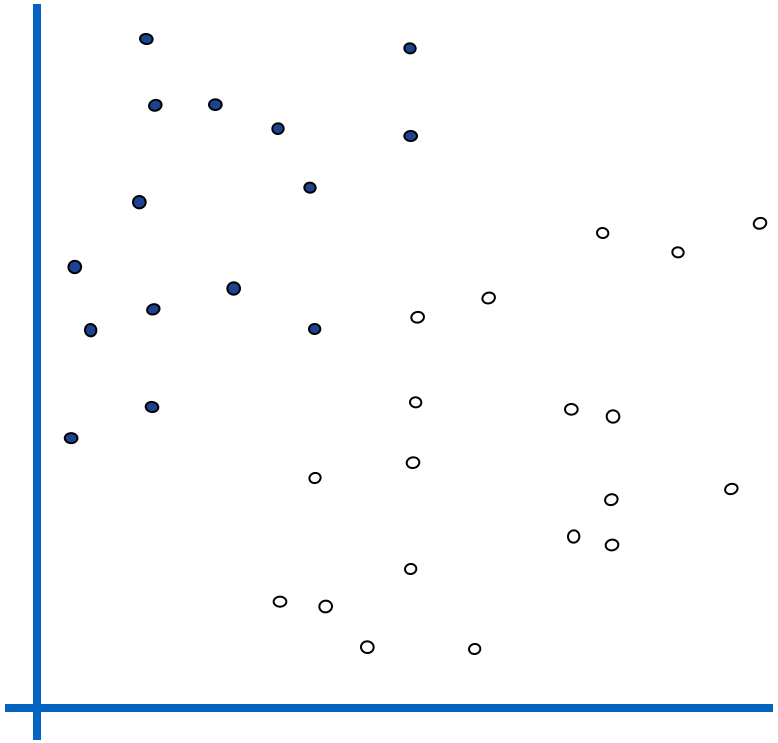
Let's say we have a dataset with three variables:

- A binary variable (our DV)
- Two continuous variables (our IVs)

And we want to train a linear model to act as a **discriminative classifier** for this data.

Intuition

- denotes +1
- denotes -1

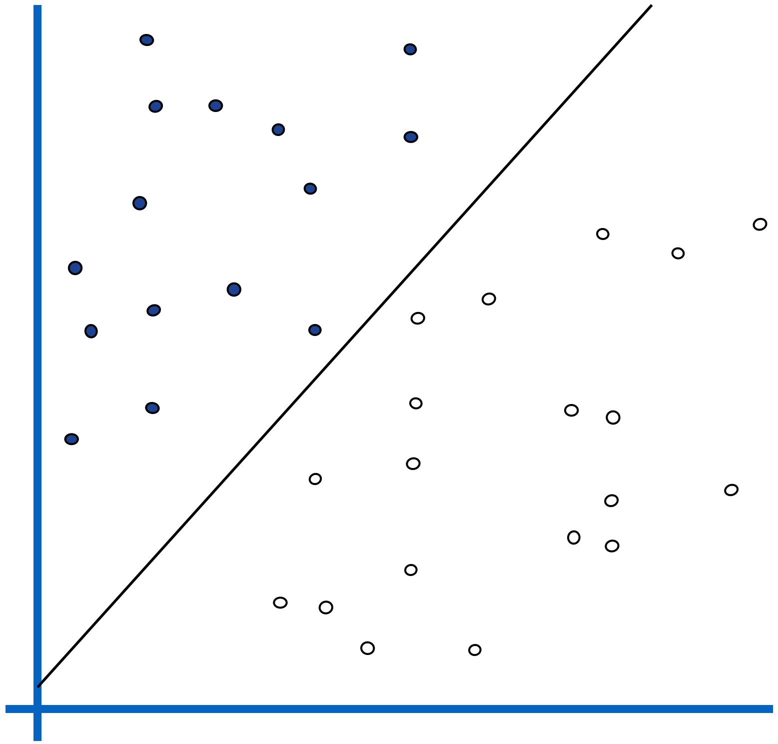


How would you linearly separate this data?

Intuition

• denotes +1

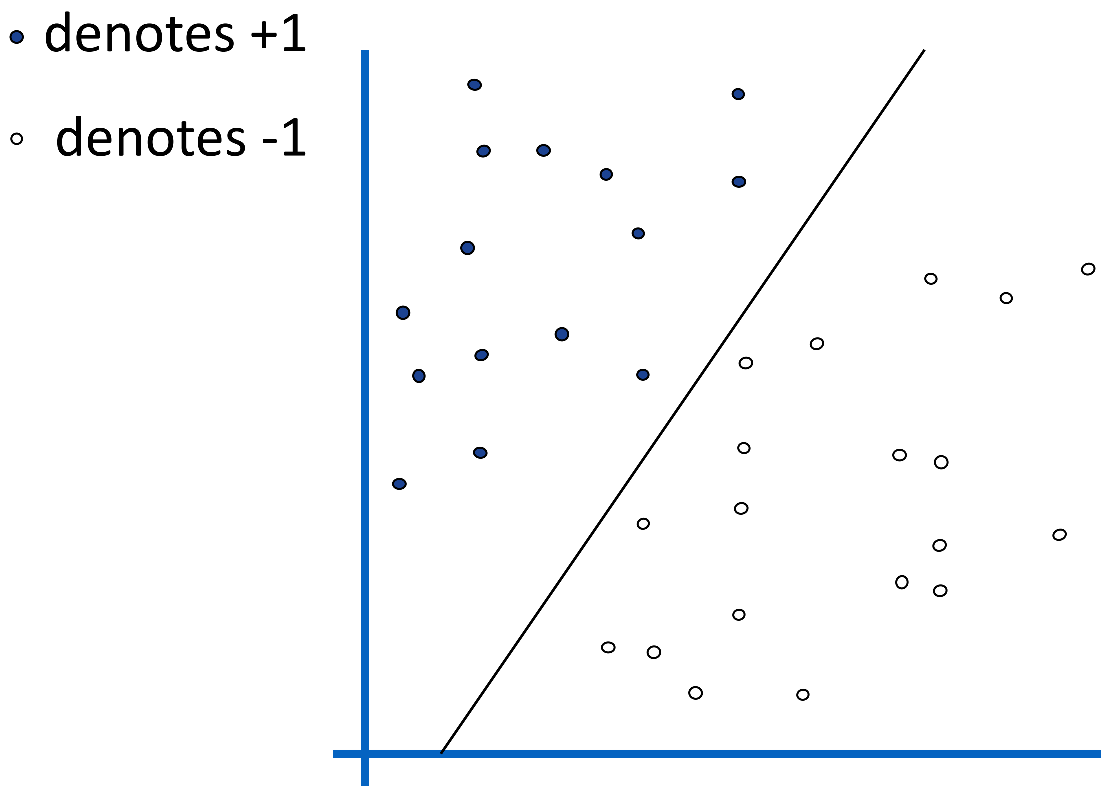
∘ denotes -1



How would you linearly separate this data?

Here?

Intuition

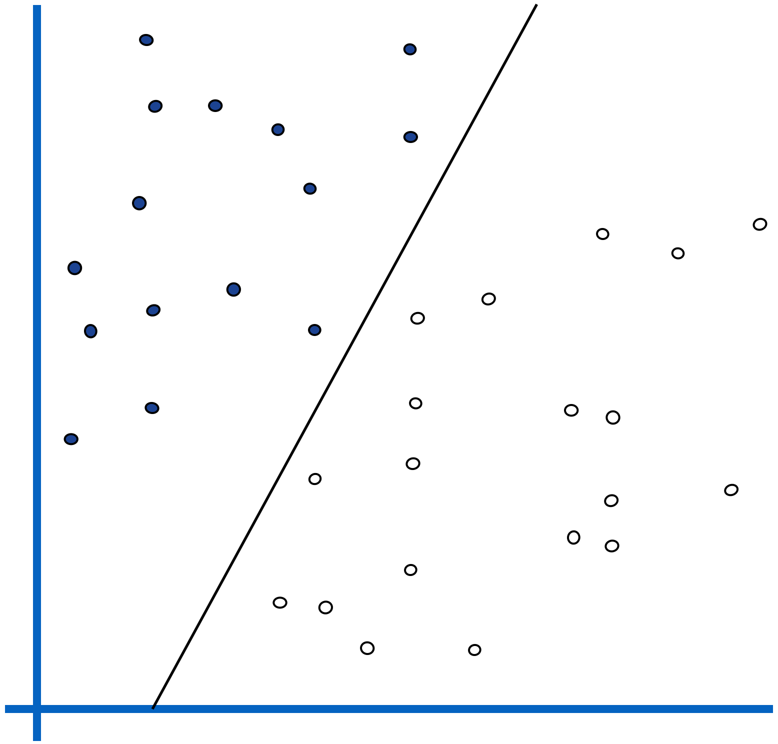


How would you linearly separate this data?

Here?

Intuition

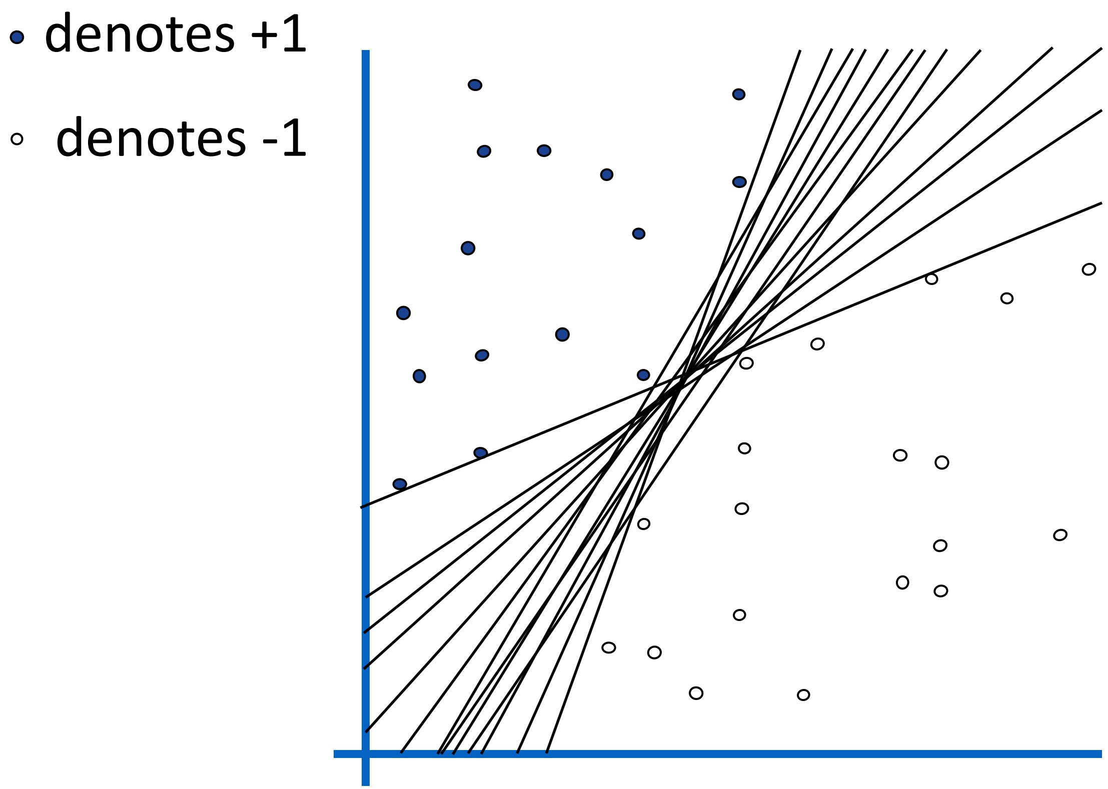
- denotes +1
- denotes -1



How would you linearly separate this data?

Here?

Intuition

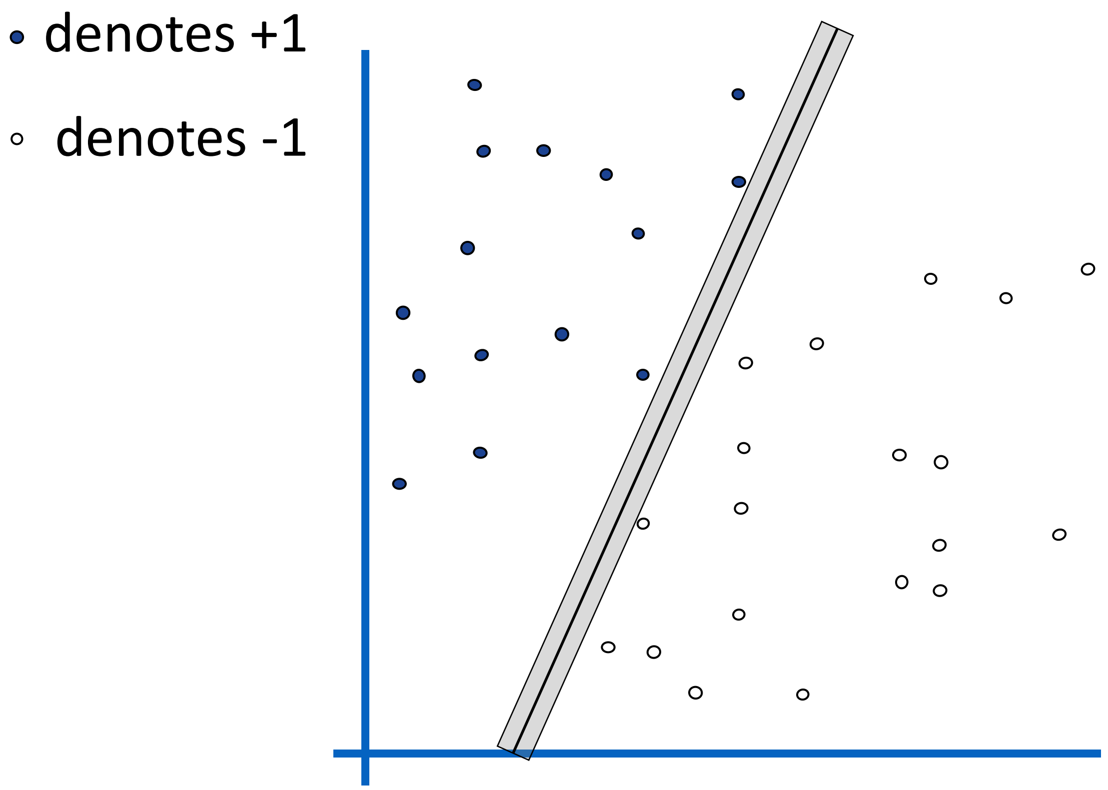


There are many possibilities for how we could linearly separate the data.

All of these are valid.

But which one is **best**?

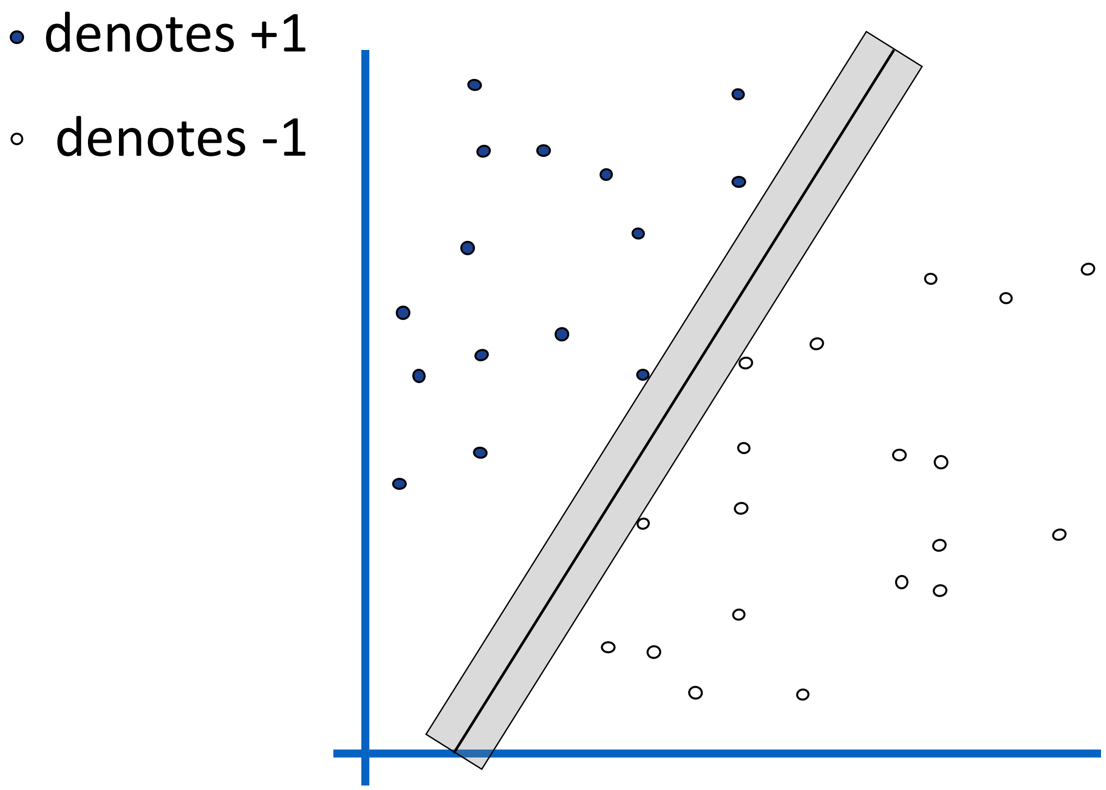
Intuition



One way to answer this question is by defining the **margin** of a linear classifier.

I.e., a *boundary* that separates the datapoints, rather than just a line.

Intuition



We can then look for the classifier that has the **widest margin**, and use that.

This is the core intuition of a Linear Support Vector Machine (LSVM).

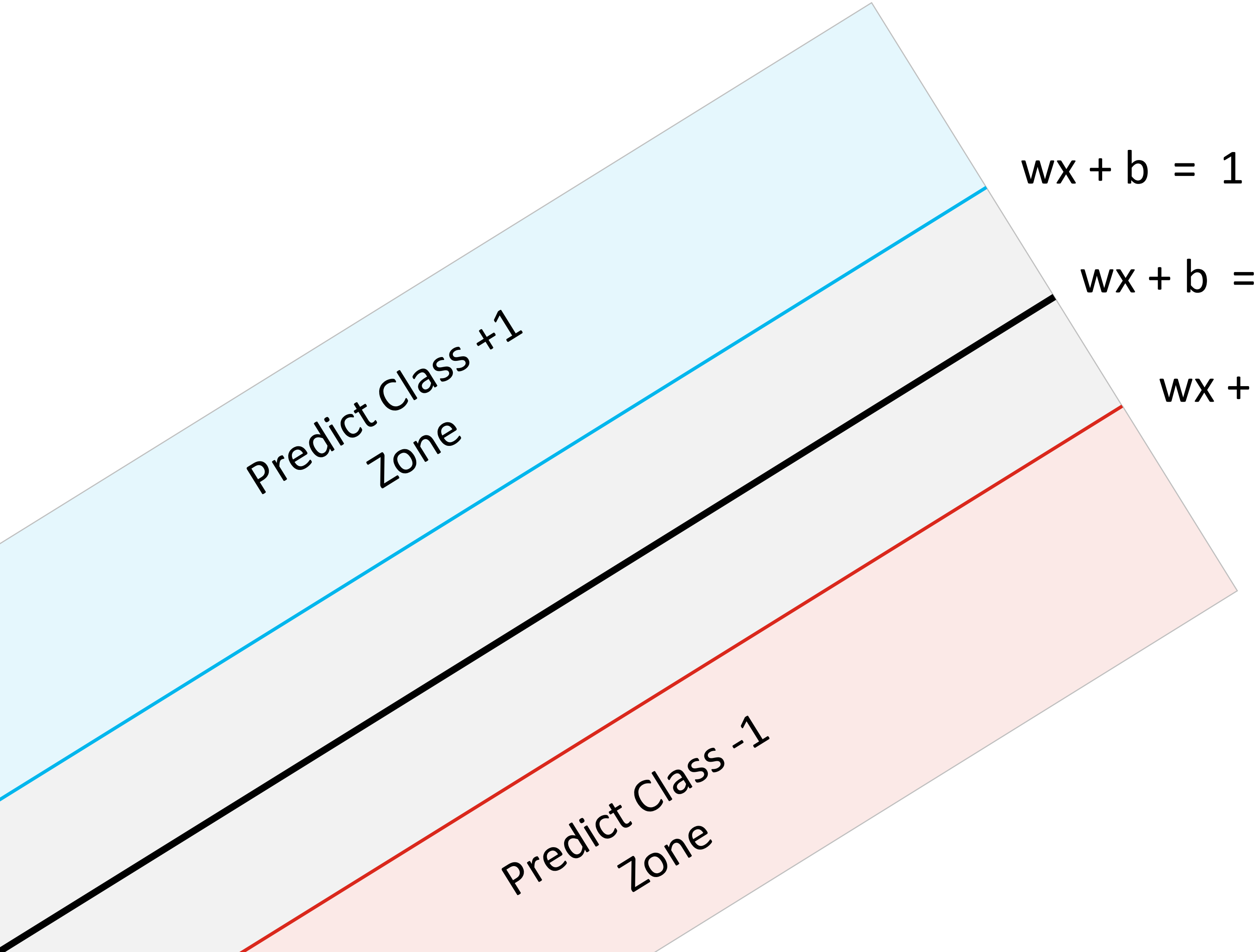
Support Vector Machines

Support Vector Machines

A Support Vector Machine (SVM) is a supervised learning algorithm that classifies data by finding the optimal boundary for separation.

It does this by *maximizing* the decision boundary, thereby maximizing the distance between each class.

Support Vector Machines

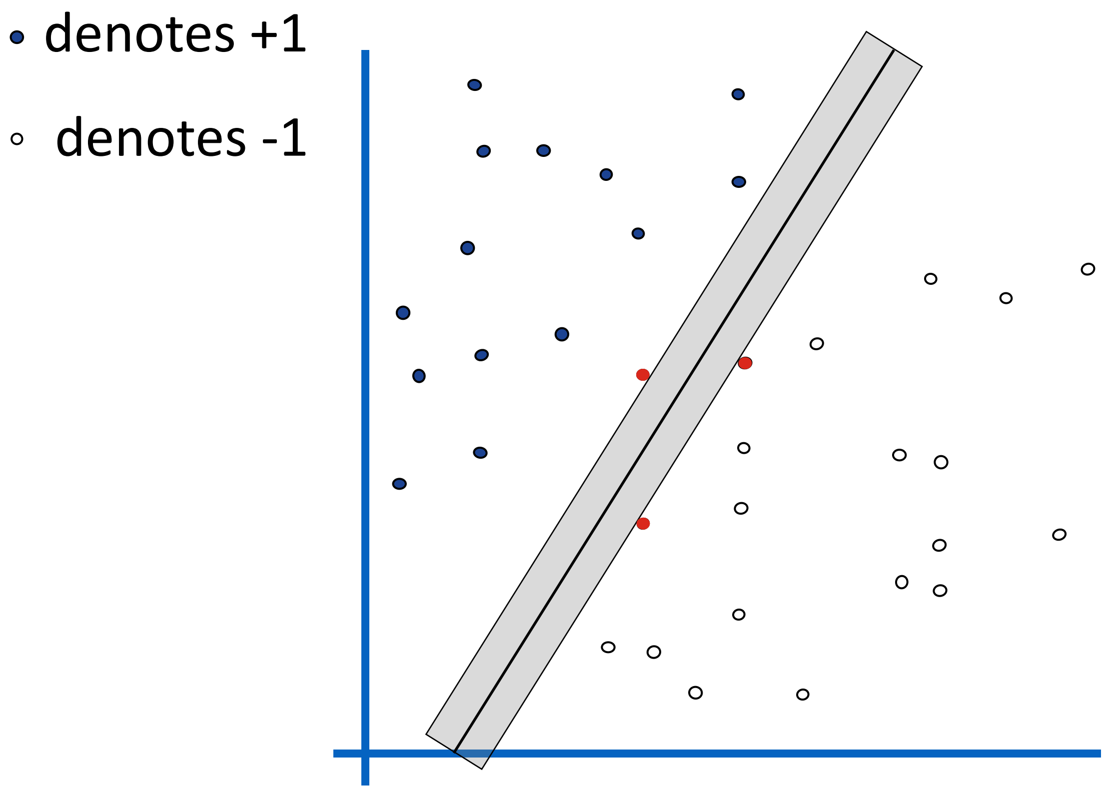


$wx + b = 1$ (positive margin boundary)

$wx + b = 0$ (decision boundary)

$wx + b = -1$ (negative margin boundary)

SVMs



The term ‘Support Vector’ is given to any datapoint on which the boundary lies.

Here, we have three support vectors.

Why Maximise the Boundary?

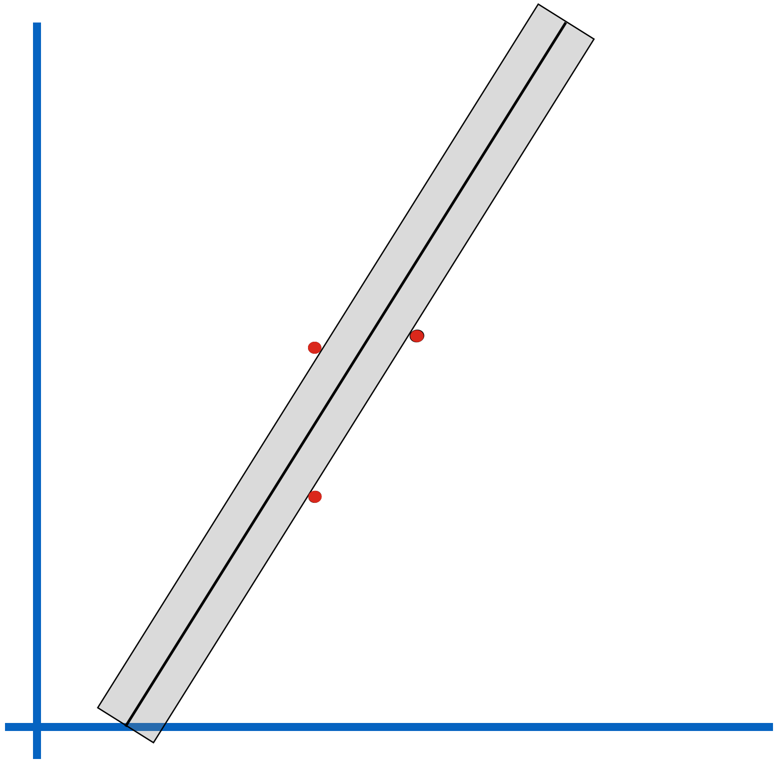
There are several reasons to maximise the boundary.

For one, a wider margin means that our linear separator is further away from any points of uncertainty, giving us more confidence in its classifications.

Second, because of this, our separator will typically generalise better to unseen data.

Why Maximise the Boundary?

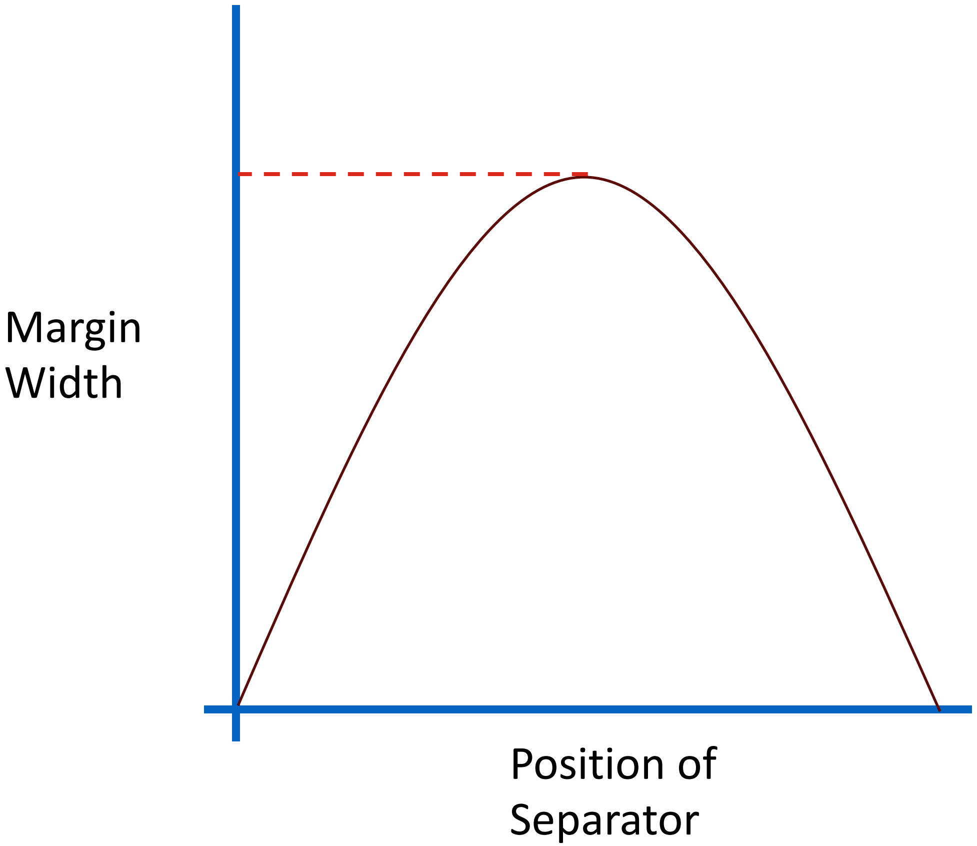
- denotes +1
- denotes -1



Third, our model is more robust and stable, given that there are fewer datapoints on which our margin depends.

In other words, the model is more robust to noise, and immune to the removal of any non-Support Vectors.

Why Maximise the Boundary?



And, fourth, maximising the margin is a convex optimization problem.

So when we maximise the margin, we're guaranteed to find an optimal solution, free of local minima.

Soft and Hard Margin SVMs

Soft and Hard Margin SVMs

There are two approaches that we can take to calculating the margin of our SVM:

- Hard Margin Classification
- Soft Margin Classification

There are advantages and disadvantages to each.

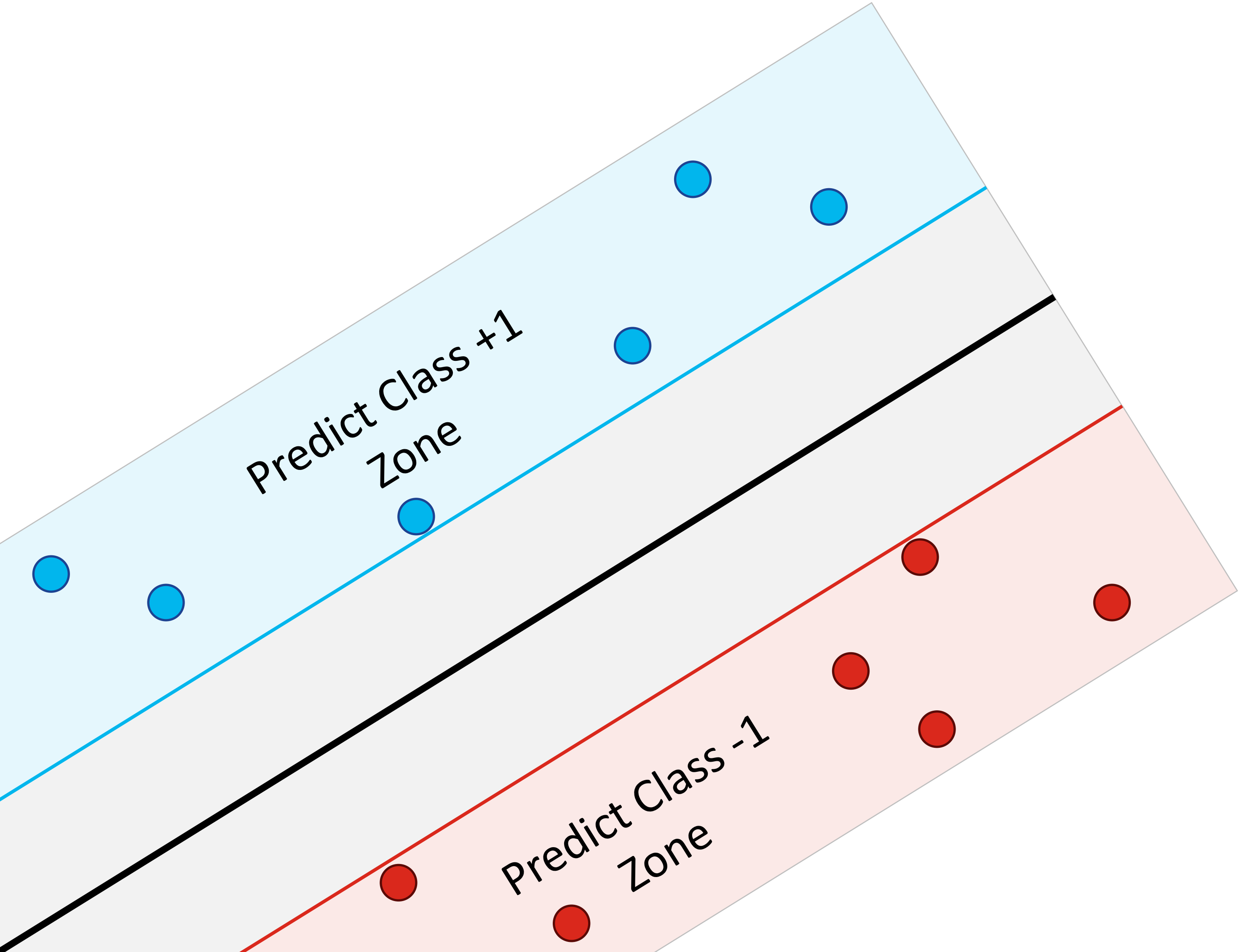
Hard Margin SVMs

The approach that we have covered so far relate to the **hard margin** approach.

For hard margin classification, our datapoints will be perfectly separated by our margin.

In other words, no data points will be allowed to fall within our margin.

Hard Margin SVMs



Hard Margin SVMs – Advantages

Some of the **advantages** of hard margin SVMs are:

Simplicity: Hard margin SVMs ensure that the classes are perfectly separated by the margin.

Computational Efficiency: It is simple and mathematically efficient to find the optimal margin for hard margin SVMs.

Hard Margin SVMs – Disadvantages

Some of the **disadvantages** of hard margin SVMs are:

Sensitivity to Outliers: A single datapoint can significantly affect the position of the decision boundary.

Not suitable for non-linear data: If data cannot be linearly separated, then hard margin SVMs will fail.

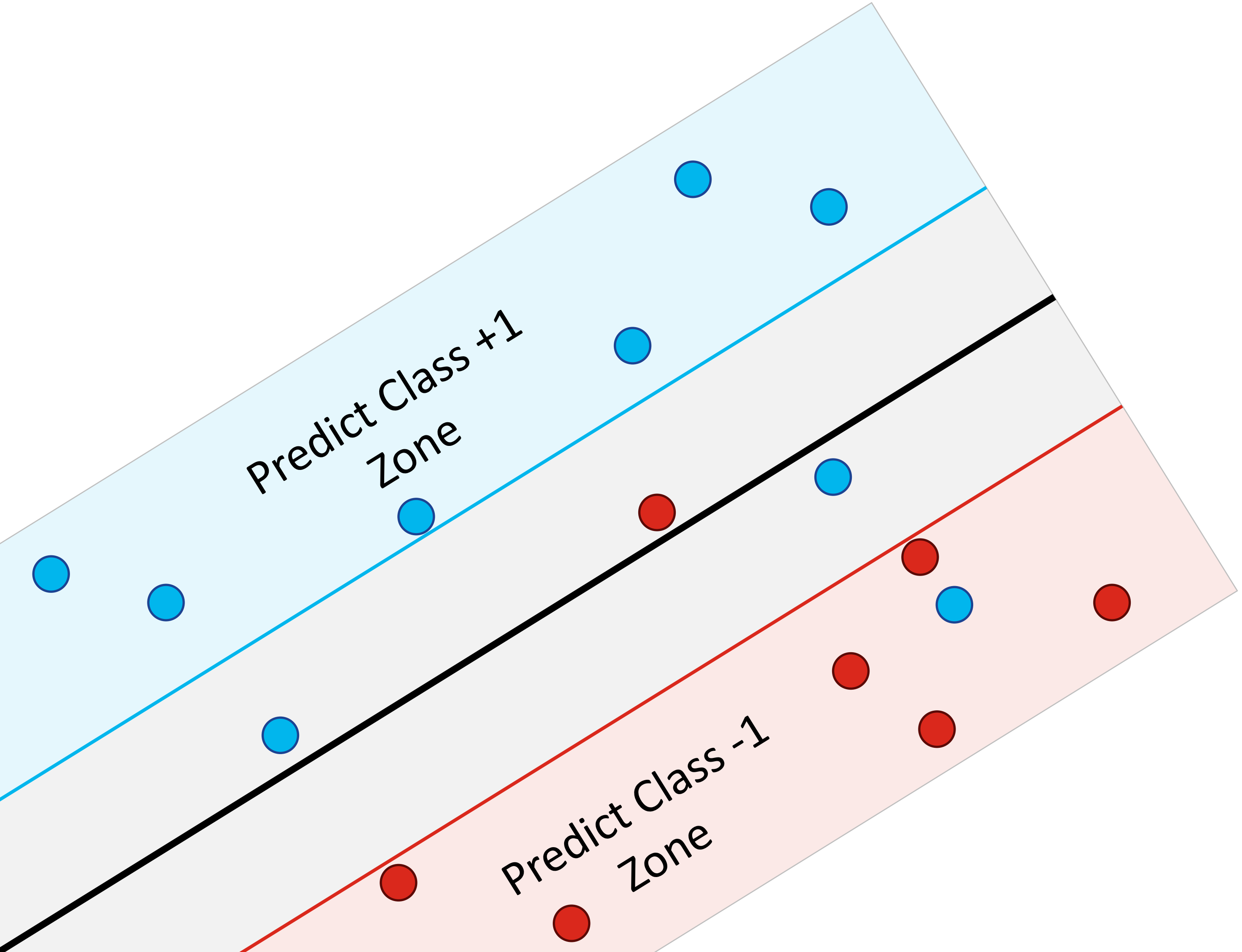
Soft Margin SVMs

The **soft margin** approach allows our SVM to be more flexible in terms of allowing the misclassification of some datapoints.

It allows some datapoints to fall within the margin, or even within the wrong side of the classification zone.

By doing so, it allows for a wider margin, at the cost of some (hopefully small number of) misclassifications.

Soft Margin SVMs



Soft Margin SVMs – Advantages

Some of the **advantages** of soft margin SVMs are:

Robustness to outliers: Soft margin SVMs are far more robust to outliers and noise than hard margin SVMs.

Works with non-linear data: Since some misclassifications are allowed, soft margin SVMs can work with data that cannot be linearly separated.

Soft Margin SVMs – Disadvantages

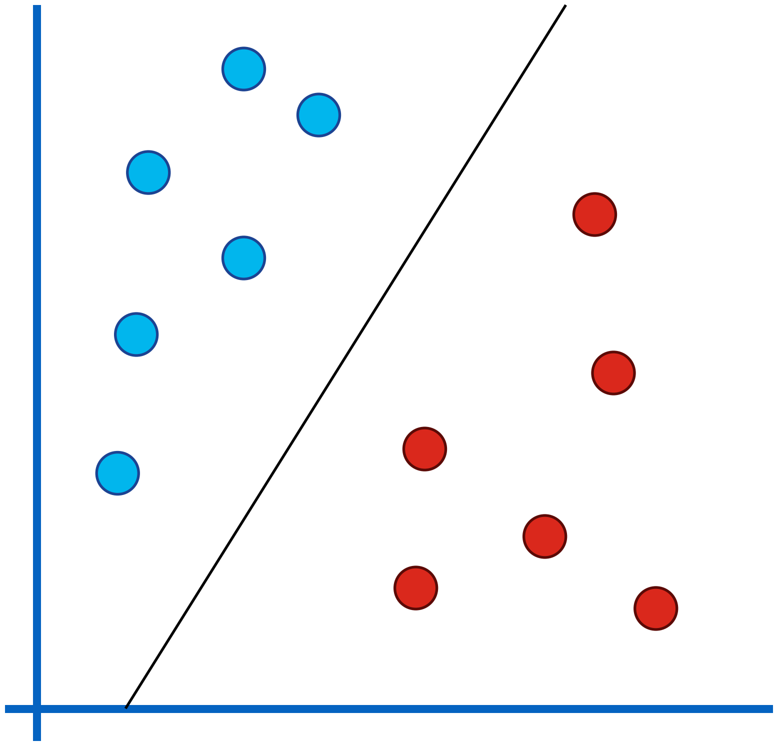
Some of the **disadvantages** of soft margin SVMs are:

Parameter tuning: More parameters, which may require careful tuning to find the best values.

More prone to overfitting: If the soft margin SVM allows too many misclassifications, this can lead to the model overfitting.

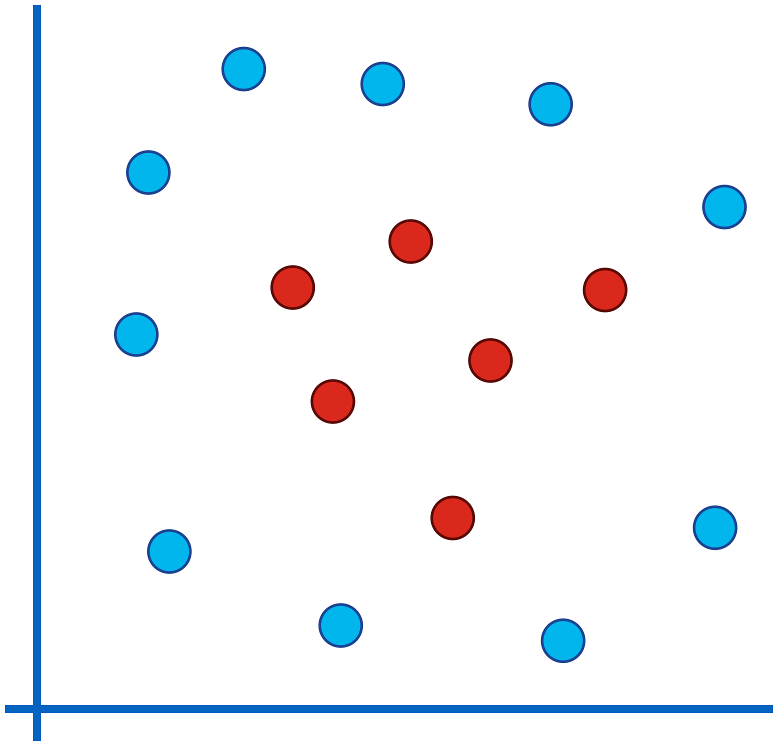
Linearly Separating Inseparable Data

Linearly Separating Inseparable Data



So far, our examples have been relatively easy to separate linearly.

Linearly Separating Inseparable Data

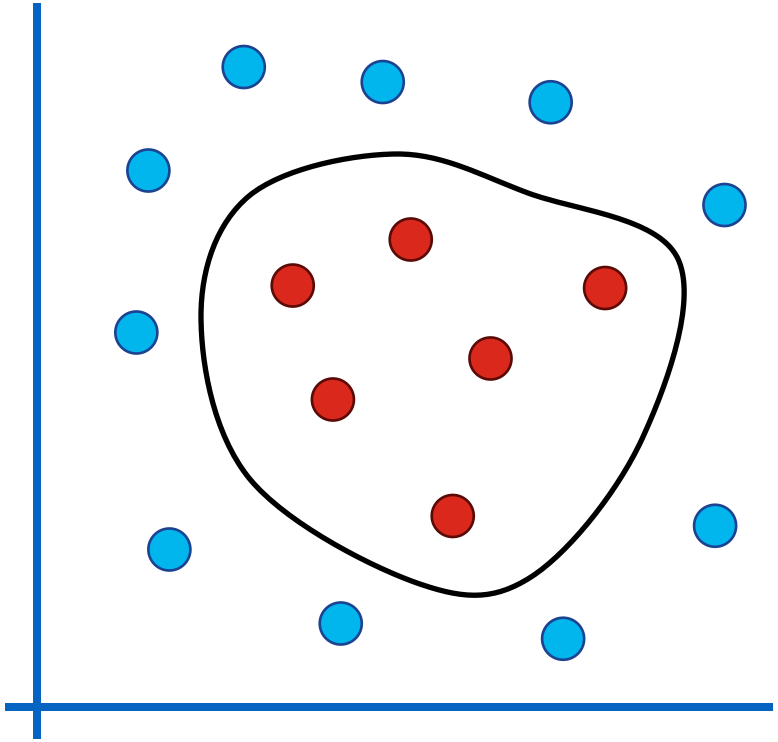


So far, our examples have been relatively easy to separate linearly.

But what if this wasn't the case?

How do you linearly separate this?

Linearly Separating Inseparable Data



One potential answer?

But we said ***linearly*** separate, and that separator is not a straight line!

So how is this a valid answer?

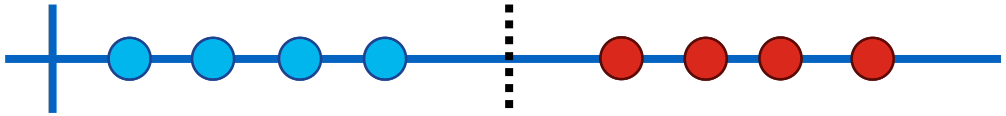
The 'Kernel Trick'

What we saw from the previous answer was an illustration of what is known as the 'kernel trick'.

In order to linearly separate our data (which did not appear to be linearly separable), we transformed the data to a higher dimension, where it *could* be linearly separated.

To illustrate this more clearly, let's move from 2 dimensions (IVs) to one.

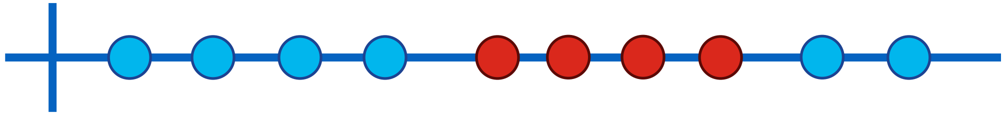
The 'Kernel Trick'



Here, we have one IV.

In this case, it is easy to linearly separate the data.

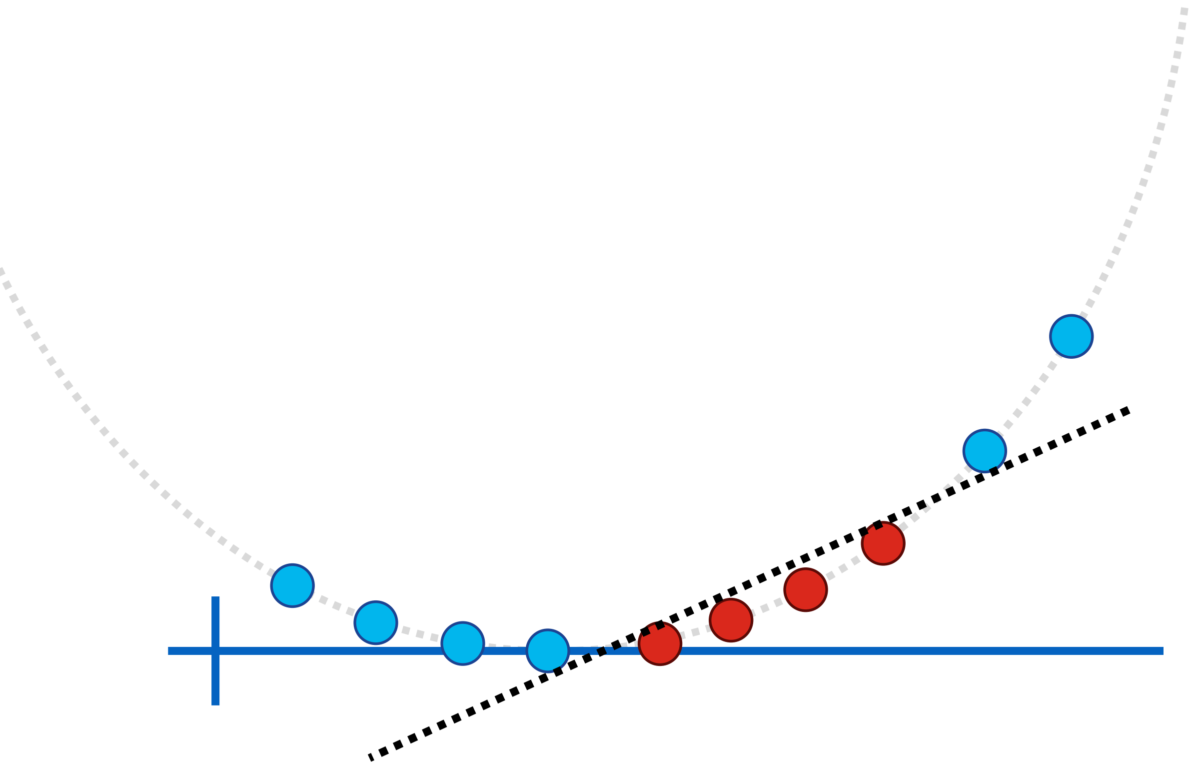
The ‘Kernel Trick’



In this case, there is no way to linearly separate the data using only one dimension.

Therefore, let’s move our data to a higher dimension.

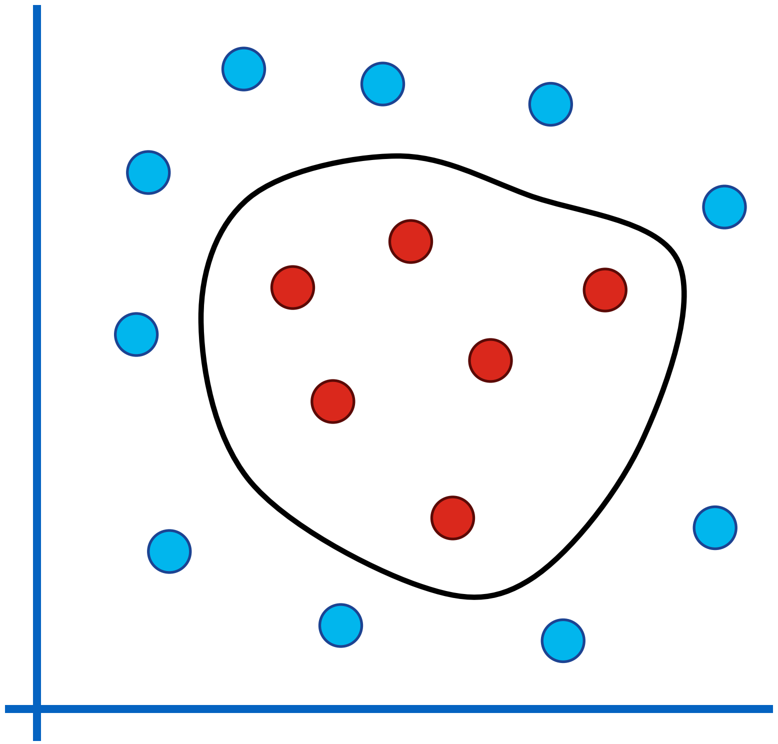
The 'Kernel Trick'



Let's say we apply some polynomial function to our data.

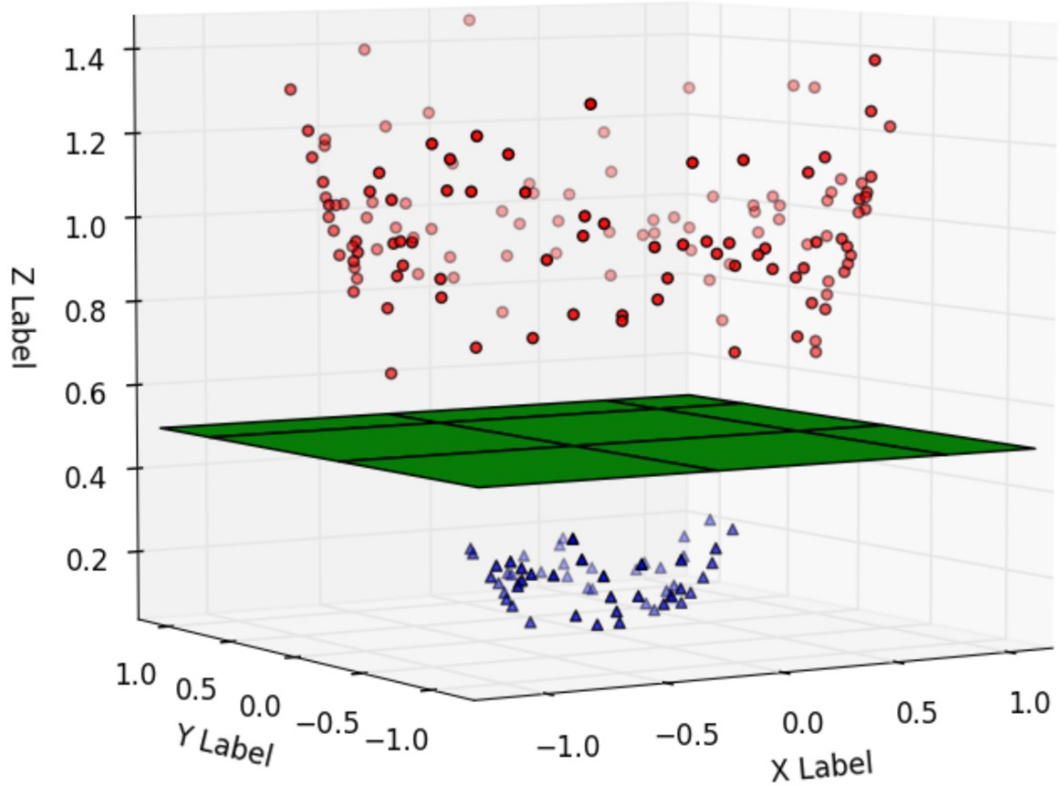
Now we can linearly separate the classes.

The 'Kernel Trick'



So returning back to our previous answer, our data might go from something like this…

The ‘Kernel Trick’



So returning back to our previous answer, our data might go from something like this…

… to something like this.

Linearly separable in a third dimension.

Image Source: https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

SVM Kernels

There are a number of different types of kernels that are often used.

These include the linear kernel, the polynomial kernel, the radial basis function (RBF) A.K.A. the gaussian kernel, and the sigmoid kernel.

Each of these will often be more or less suitable, given the shape and the characteristics of your data.

Terminology

Note that when we're talking about separating in high-dimensional space, we often use the term *hyperplane* to refer to our separator.

A hyperplane is defined as a subspace of one dimension less than the input space. In other words, if our data are represented in n dimensions, we can use a hyperplane to separate the data in n-1 dimensions.

SVMs in the Real World

Popularity of SVMs

Nowadays, advances in deep learning have meant that SVMs are mostly out of fashion.

However, for a long time, SVMs were the ‘gold standard’ for dealing with complex, high-dimensional data, such as within computer vision – typically outperforming other ML algorithms, including neural networks.

Despite their current status, SVMs still have a place in some contexts.

Other Types of SVMs

SVMs can also be adapted to a wide range of different contexts and scenarios. These include:

- SVM Regression: Works with regression problems
- Multiclass SVM: For non-binary classification problems
- One-Class SVM: Used for anomaly or outlier detection

Interpretability

Naturally, as we start to deal with more complex datasets, and high-dimensional separators, interpretability becomes more of a challenge.

As such, SVMs tend to be used more for prediction, where it often excels compared to the simpler, more interpretable models that we have discussed so far.