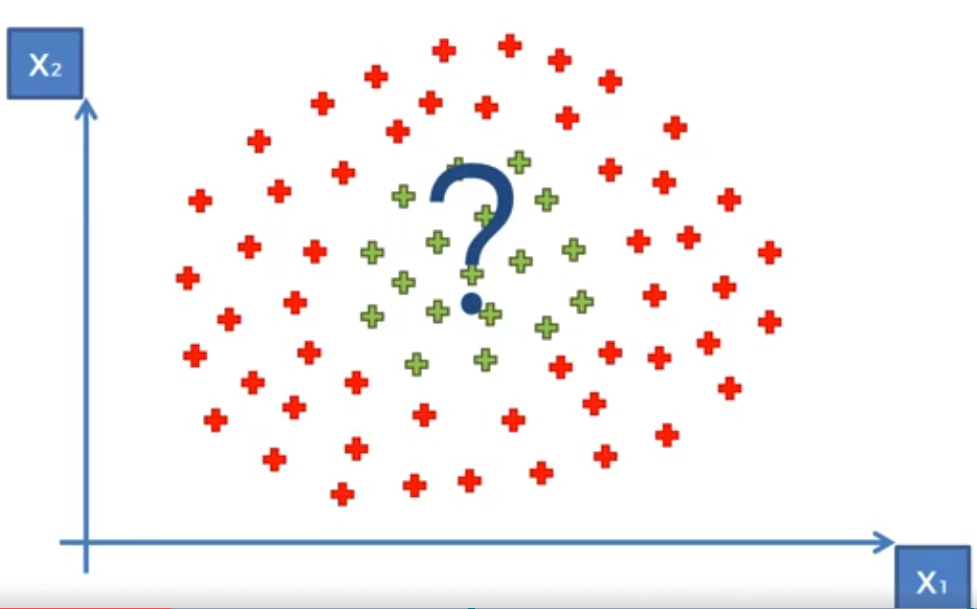
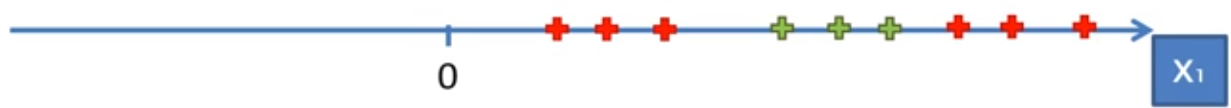
***Kernel SVM***

* What happens if data isn’t linearly separable?



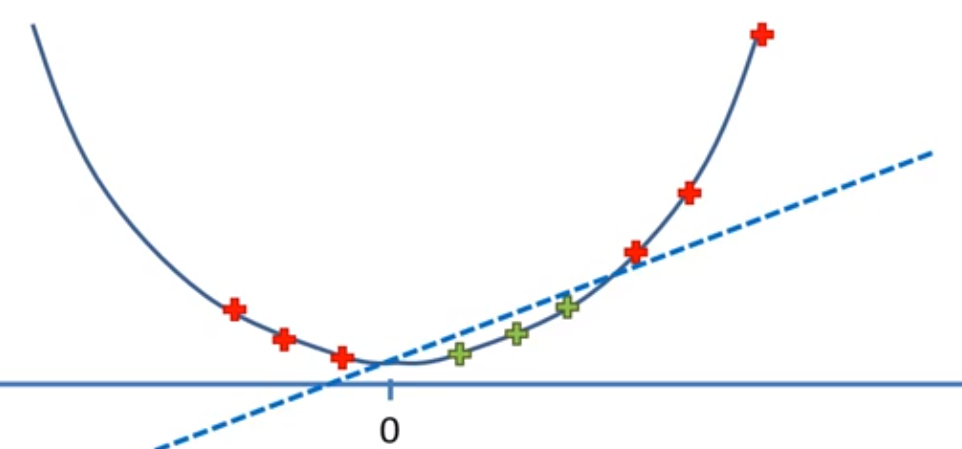
* 1D space, non-linearly separable dataset:



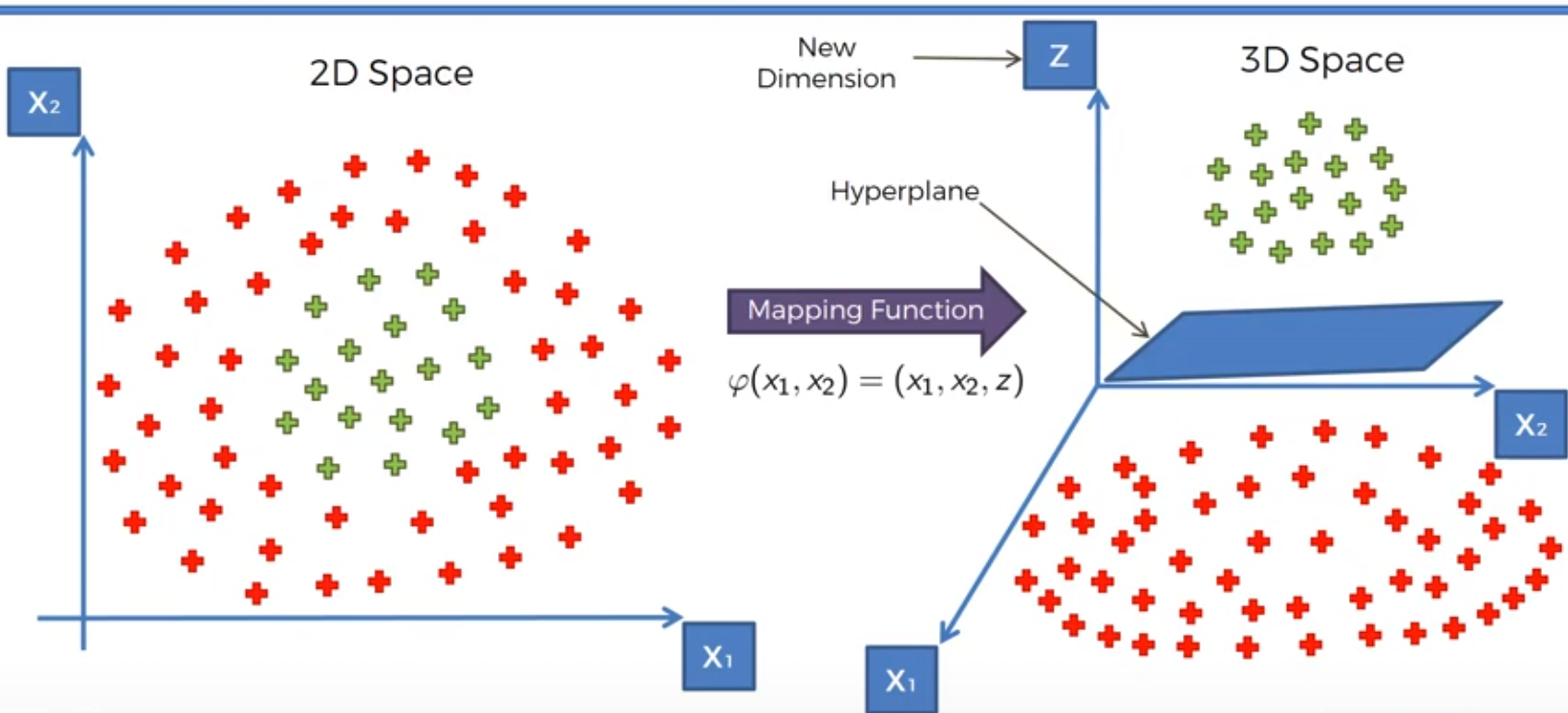
* Want to **increase dimensionality** of the space to make the dataset linearly separable in a higher-dimensional space
* 1st step: Start to build a **mapping function** 🡪 say, f = x – 5
* This move all DP’s to the left 5 spots



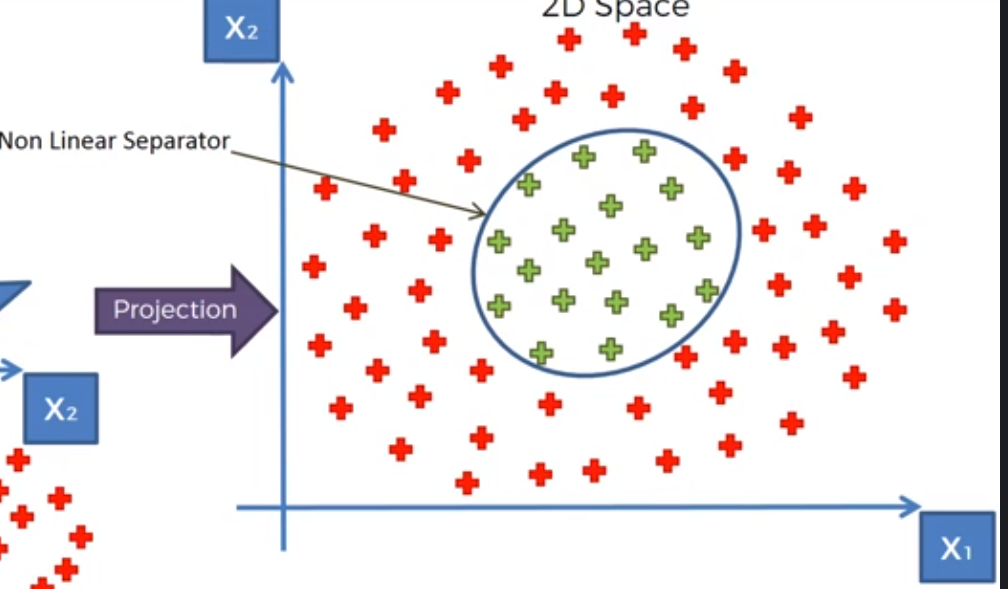
* Then square it all f = (x-5)^2 = 2 dimensional 🡪 see that data is now linearly separable:



* Then we project this back into 1D space
* Apply same principle to a 2D space to somehow map it to a 3D space:
* Then see that the data is linearly separable by a **hyperplane**



* Then we project this hyperplane back into a 2D space to see our non-linear classifier



* Possible issue 🡪 mapping to higher-dimensional spaces = computationally expensive, especially w/ higher dataset
* Therefore, this approach isn’t the most efficient
* Going to use the **kernel trick** to deal w/ this to get similar results w/out mapping to higher dimensions