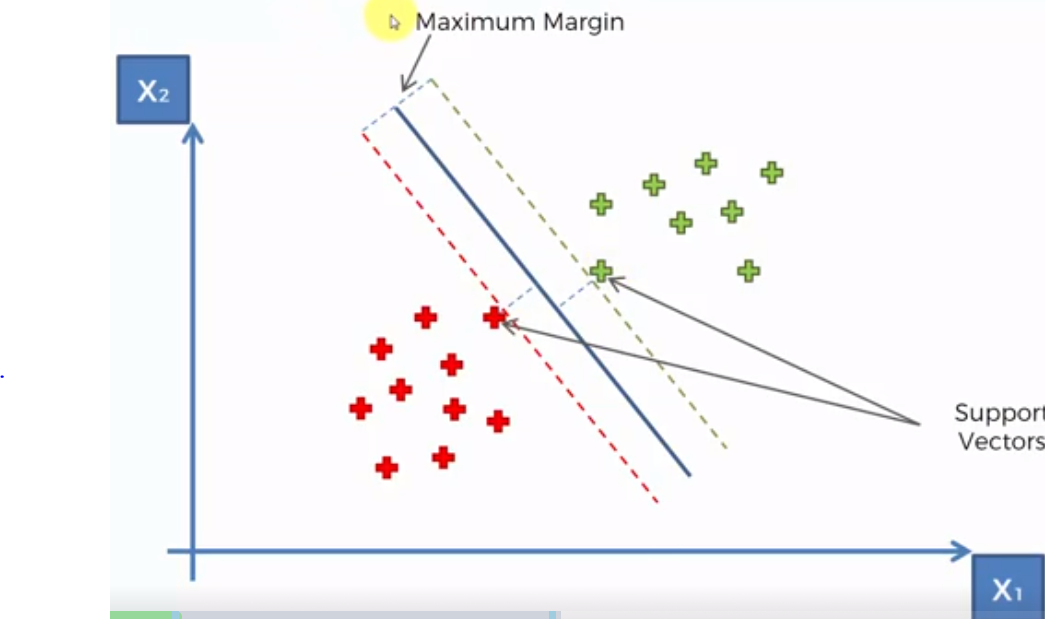
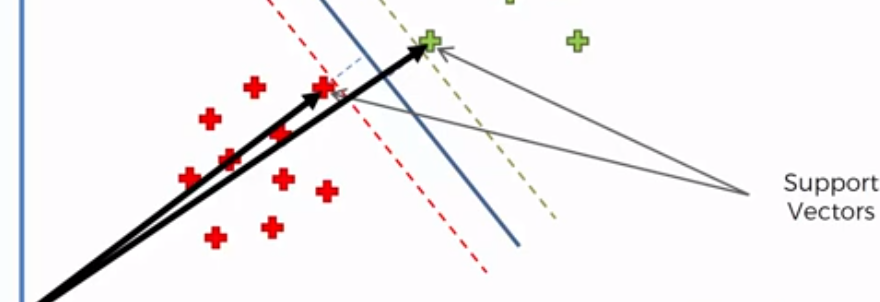
***SVM***

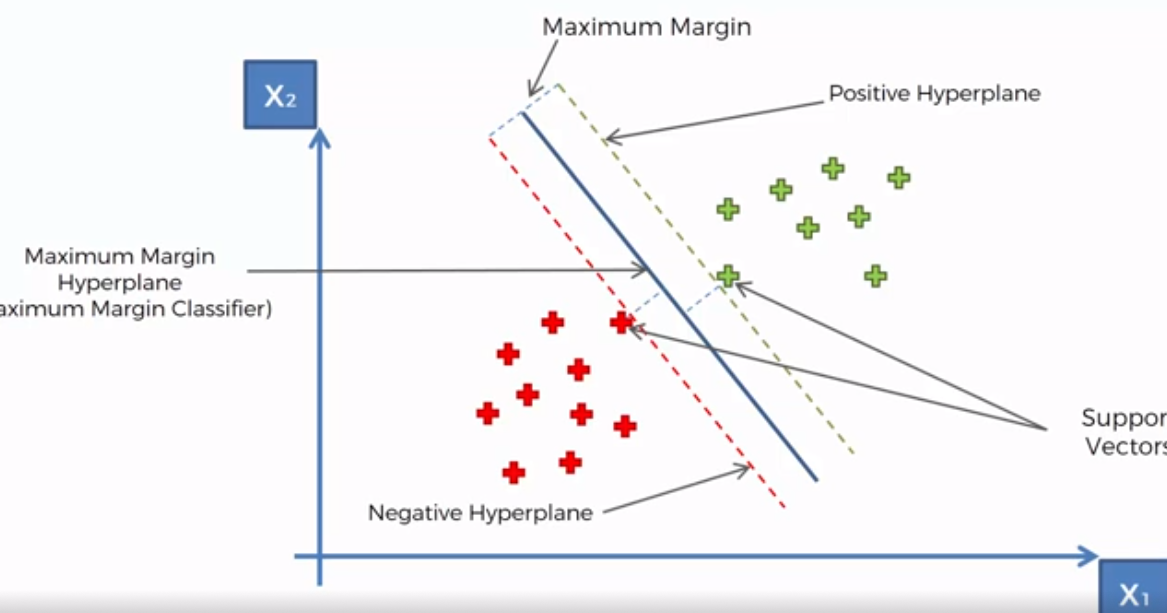
* Tries to find the best decision boundary for classification



* Notice our best-line has the **maximum margin** such that the sum of the distance between those equidistant red and blue points is the maximum it can possibly be
* These 2 lines = the **support vectors** + they are the only points that matter for the algorithm
* The algorithm wouldn’t change if we got rid of all other points outside the support vectors
* They’re **vectors =** b/c in multidimensional space, we do not really have “points”, just in this case we do b/c we only have 2 dimensions so we can plot them.



* Also have the **maximum margin hyperplane/classifier** (the line in the middle) =
* Classifier line in 2D, hyperplane in multidimensional space
* As well as the **positive + negative hyperplanes**



* Working w/ a linearly-separable dataset where we can put a hyperplane/line to separate categories with the maximum margin
* SVM’s are popular + different to other ML algorithms
* Ex: in trying to classify apples vs. oranges, most ML algorithms will find the most “apple-y” apple + most “orange-y” oranges + capture these features.
* These DP’s would be the best definition of apples + oranges the algorithm can find + the algorithm would use these definitions to classify new DP’s as apples or oranges
* SVM would try to find oranges that are most like apples + apples that are most like oranges
* The SVM then classifies these as the support vectors = the most extreme cases, near the boundary