**Support Vector Machine**

**(SVR)**

Chart, scatter chart

Description automatically generated

* Let’s say we have two sets of data points, red and green. And we would like to separate them so that they can be distinguished easily, using the border or the decision line, in the future when we start adding new points?
* That basically is the point of the classification – when we add new points in the future which haven’t been classified yet, it will be easy to decide about where they fall.
* One of the ways is to draw a horizontal line dividing them – anything to the right will be green, and anything to the left will be red. And when a new point falls on this space, we will know right away which category it belongs to.
* However, there’s another way – we can draw a vertical line.
* Or we can draw one or multiple diagonal lines as show below which will achieve the same result that will separate our points into two classes. But at the same time, they all in the future will have different consequences.

Chart, scatter chart

Description automatically generated

* So, when we add new points, depending on where that point will either fall into the green zone or the red zone. Thus, we want to find an optimal solution.
* And that’s what SVMs are all about – they are about finding the best line, or the best decision boundary which will help us separate our space into classes.

**Working of SVM –**

* The decision boundary line is searched through the maximum margin.

Chart, scatter chart

Description automatically generated

* So basically, it’s the line that separates these two classes of points and at the same time it has the maximum margin.
* Maximum Margin – The solid line is drawn equidistant from the most extreme points of the red and green classes. That’s margin.
* The sum of those two distances should be maximized for this line to be the result of SVM.
* And the extreme two points are called support vectors.

Chart, scatter chart

Description automatically generated

* The above-mentioned points are supporting the whole algorithm.
* Even if we get rid of all the rest of the points, nothing will change. So, the other points don’t contribute to the results of the algorithm.
* Only the extreme points are supporting the algorithm, that’s why they’re called ‘Support Vector’s.
* Support Vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.
* You can call them supporting points, but, they are called ‘Vectors’ and this is why: In a multi-dimensional space, when you have more than just two variables, each point is no longer a point because you cannot visualize it on a 2D plan or even a 3D space.
* Therefore, each of the points that we see here is a vector in a multi-dimensional space.
* So, the more general term for the points that we see here are vectors.
* And since we have vectors supporting the whole algorithm, it’s called Support Vector Machine.

**Additional Information about The Algorithm:**

1. Maximum Margin Hyperplane (Maximum Margin Classifier) – It’s the solid line in the middle. In a 2D space it’s a classifier. But in a 2D space it is a hyperplane. All the other lines plotted before the maximum hyperplane are also hyperplanes, but the one plotted at last is the maximum hyperplane, as it covers the maximum distance as compared to all the other lines.
2. Boundary Line – Two parallel lines drawn to the sides of the Support Vector (or the Maximum Margin Hyperplane), with error threshold value Epsilon are called Boundary Lines.
3. Positive Hyperplane – The boundary line (green dotted line) situated at the positive side is called (next to the maximum margin hyperplane) is called a positive hyperplane.
4. Negative Hyperplane – The boundary line (red dotted line) situated at the negative side (next to the maximum margin hyperplane) is called a negative hyperplane.

So, the essence, the intuitive part of the Support Vector Machine is that we are working with a linearly separable dataset where it’s given to us that we can put a line through our chart which will separate the two categories, and then we are just searching the one with the maximum margin.

**What’s so special about SVMs, and why are they different to other Machine Learning Algorithms?**

* Imagine you are trying to teach a machine how to distinguish between apples and oranges where you are giving the machine some test data where you are training it to analyze, and then you give it a fruit which will either be a fruit – apple or orange – and it will have to analyze and output whether it’s an apple or orange.
* Now in our case, let’s say we have apples on the right and oranges on the left. So, what predominant machine learning algorithms would do is they would look at most apple-ly apples and most orangey oranges – they would look at the most stock standard apples and the most stock standard common types of oranges. In our case, those would be the apples on the extreme hearts of their sides, far away from the Maximum Margin Line and far away from each other.
* The predominant Machine Learning Algorithms would try to learn what an apple is from the apples that are very much like apples and what an orange is from the oranges that are very much like oranges. And based on that, it would be able to come up with some predictions and classifying for new data elements which you would give it.
* In the case of Support Vector Machine, it is different. Instead of looking at the most stock standard fruits, it will look at the apples that are very much like an orange. Also, it would look at oranges which are not stock standard oranges, but at the orange that will look more like an apple than anything else.
* So those two vectors – the apple that looks most like an orange, and the orange that looks most like an apple – are the support vectors for the SVM.
* And in the figure, it can also be seen that the vectors are very close to the boundary and to each other.
* You can also look at SVM at a very risky and extreme kind of an algorithm because it looks at the very extreme case, very close to the boundary, and it uses that to construct its analysis.
* And that itself makes SVM very special, very different kind of an algorithm compared to most of the other algorithms. That’s why at times it preforms much better than non-support vector Machine Learning algorithms.