**Support Vector Machine**

1. <https://pythonprogramming.net/support-vector-machine-intro-machine-learning-tutorial/>
2. <https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html>
3. <https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8>

Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing classification, regression, and even outlier detection.

The objective of the Support Vector Machine is to find the best splitting boundary between data. In two dimensional space, you can think of this like the best fit line that divides your dataset. With a Support Vector Machine, we're dealing in vector space, thus the separating line is actually a separating hyperplane. **The best separating hyperplane is defined as the hyperplane that contains the "widest" margin between support vectors.** The hyperplane may also be referred to as a **decision boundary**. The easiest way to convey this is through images.

* We will start with the above data. We noted in the past that the most common intuition is that you would classify a new data point based on what it is closest to or the proximity, which is what the [K Nearest Neighbors algorithm](https://pythonprogramming.net/k-nearest-neighbors-intro-machine-learning-tutorial/) does for us.
* The main issue with this objective is that, per data point, you have to compare it to every single data point to get the distances, thus the algorithm just doesn't scale well, despite being fairly reliable accuracy-wise.
* What the Support Vector Machine aims to do is, one time, generate the "best fit" line (but actually a plane, and even more specifically a hyperplane!) that best divides the data. Once this hyperplane is discovered, we refer to it as a decision boundary. We do this, because, this is the boundary between being one class or another.
* Once we calculate this decision boundary, we never need to do it again, unless of course, we are re-training the dataset. Thus, this algorithm is going to scale, unlike the KNN classifier.

**The curiosity is, of course, how do we actually figure out that best dividing hyperplane?**

* The LSVM algorithm will select a line that not only separates the two classes but stays as far away from the closest samples as possible.