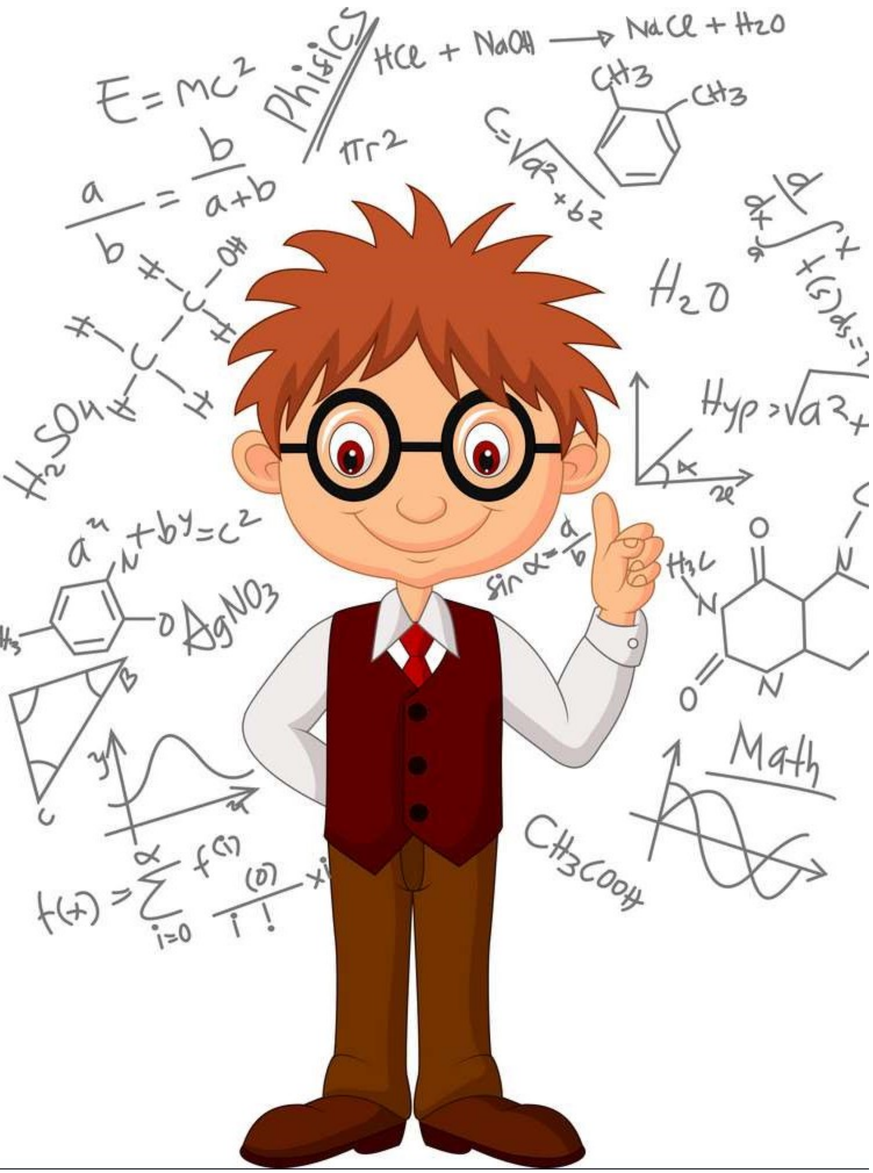
Predictive Analytics

Lecturer: Marina Iantorno

E-mail: miantorno@cct.ie

July 2022

cct | College Dublin
Computing • IT • Business



In today's class we will cover:

□ Classification vs Regression Models

□ K-NN

□ SVM

□ Practice in Python

Regression vs Classification

DIFFERENCES

Regression vs Classification

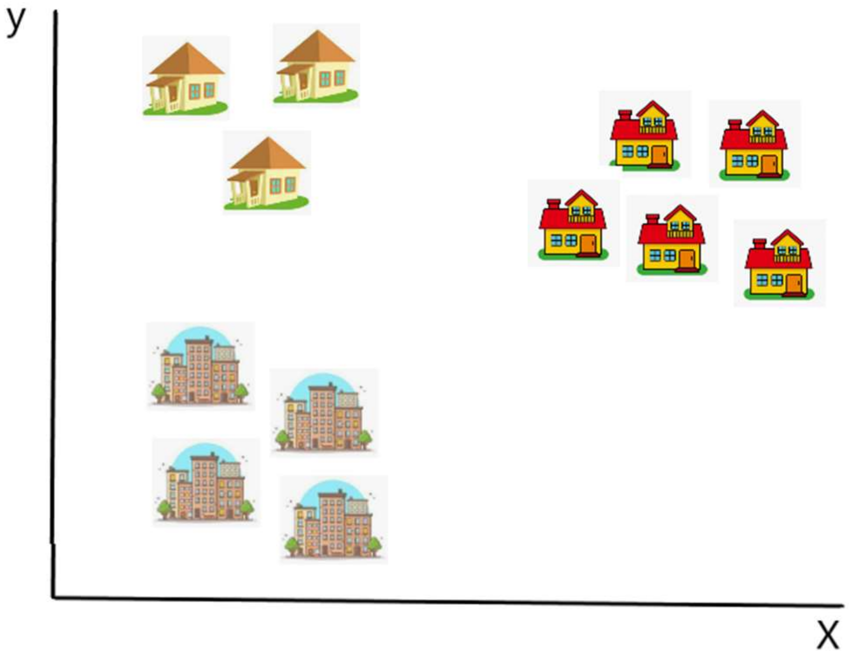
So far, we have covered a range of Regression Models, and based on our data we predicted possible outcomes on continuous variables, and this is the main difference between one model and another one. While Regression predicts on continuous variables, Classification does it on discrete variables.

Regression vs Classification

In a Regression Model, we try to predict the result of a “y” variable depending on certain features, and the result would be within a specific domain. For instance, if we want to predict the house pricing, we can do it with a Regression model, because price is a continuous variable and ultimately we will get a numeric result.

Regression vs Classification

In a Classification model, we will not get a numerical value as a result, we will get a category that in DA we call “class”. Following our example, we would see the type of house in question. For instance, we could decide whether we talk about a bungalow, a house or an apartment.



Classification

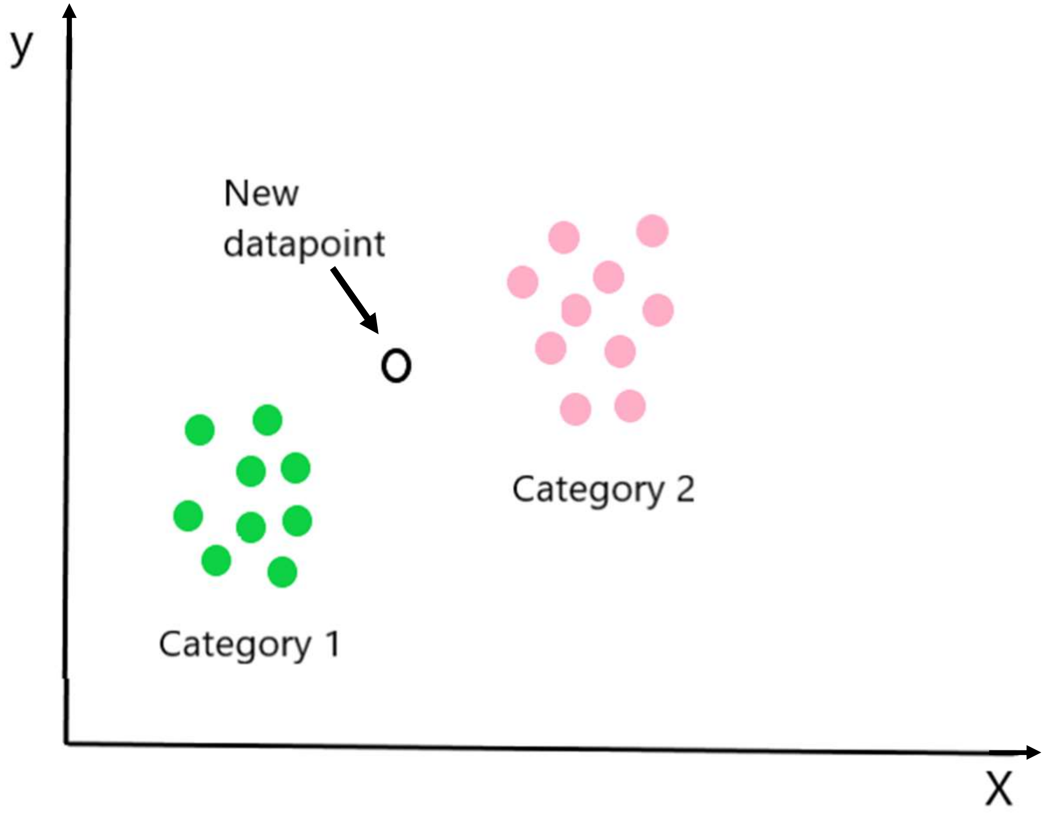
K- Nearest Neighbours (K-NN)

K-NN

The K-NN is a classification method that was developed in 1951 by Evelyn Fix and Joseph Hudges. The main goal of this classification model is to find for datapoint “k” the closest training group in the dataset.

Let’s go deeper with an example.

K-NN

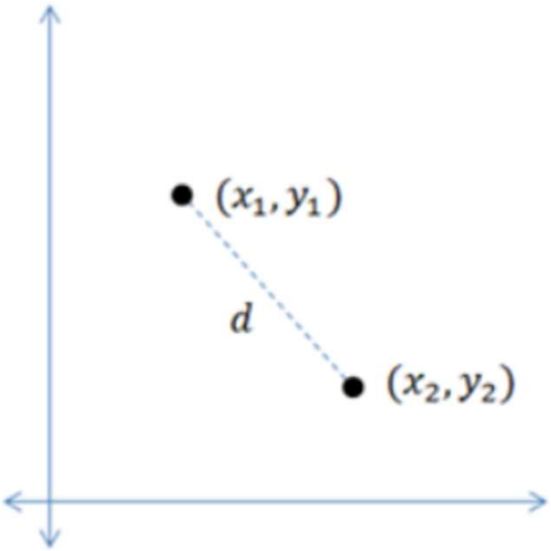


How do we decide the category of the new datapoint?

K-NN

- ➢Step 1: Choose the number of K neighbours of your algorithm. The most common value of K is 5.
- ➢Step 2: Take the K nearest neighbours of the new datapoint according to the Euclidean distance.
- ➢Step 3: Among these K neighbours, count the datapoints in each category.
- ➢Step 4: Assign the new datapoint to the category where you counted most of the neighbours.

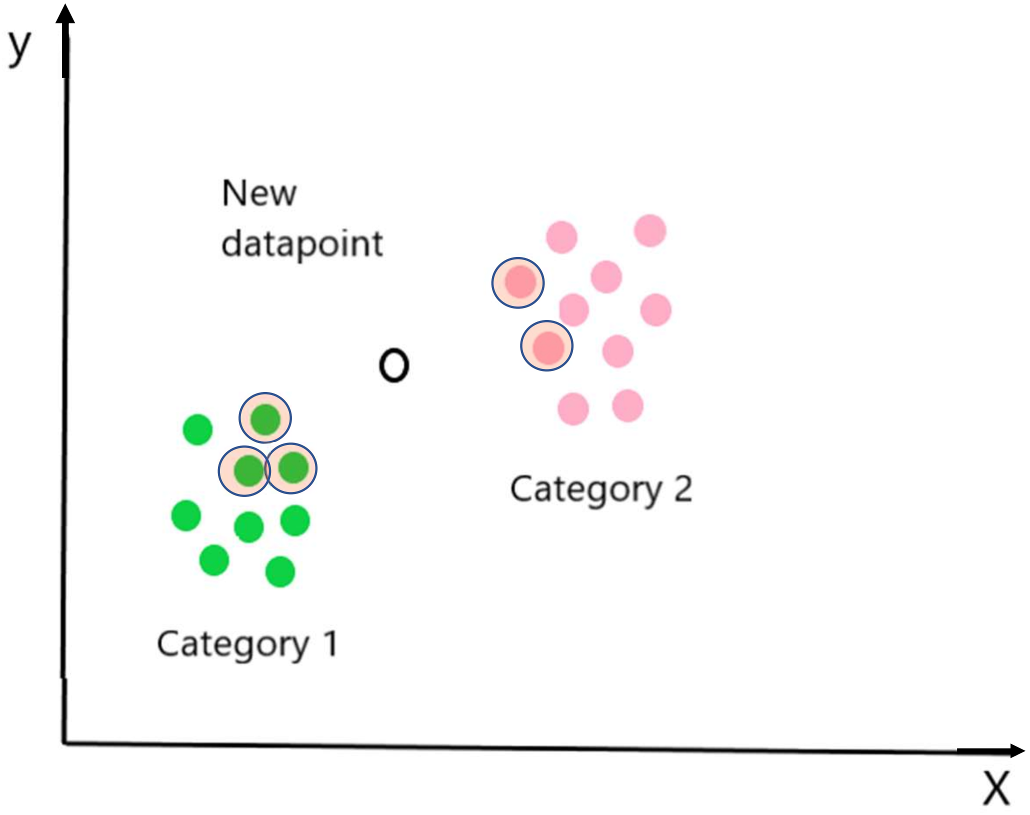
K-NN



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Geometrically, we need to follow some rules to calculate the distance. The distance between the two points is measured according to the formula.

K-NN

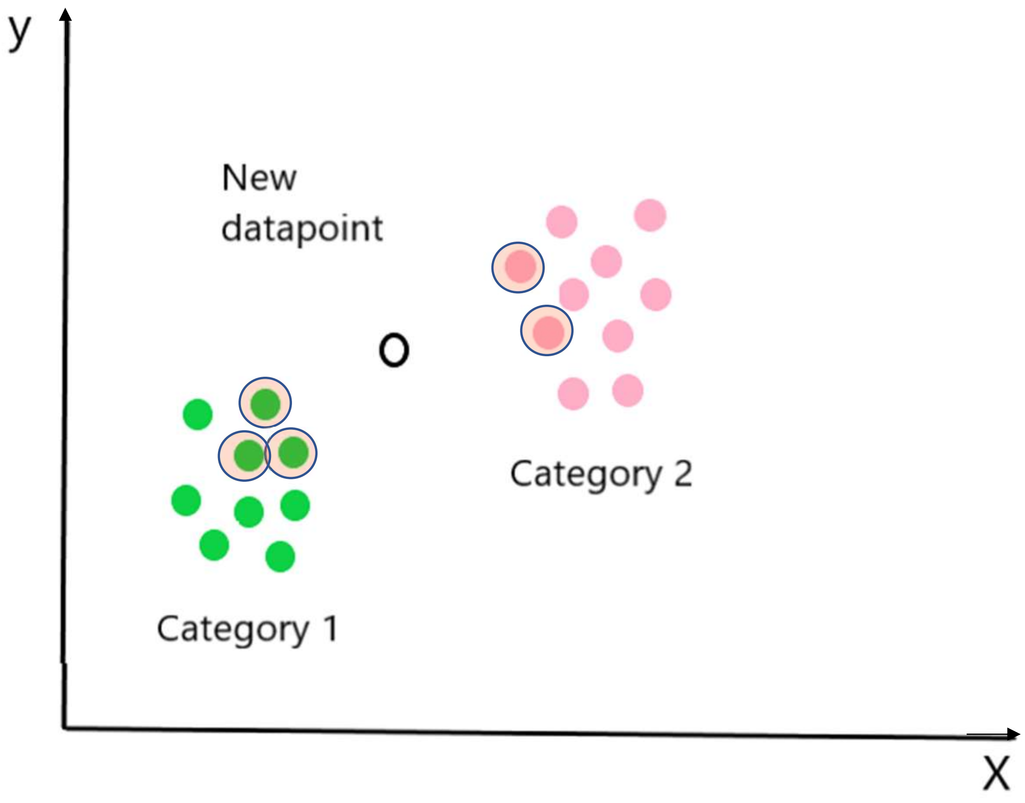


Let’s follow the steps!

Step 1: We decide k = 5

Step 2: We decide the nearest neighbours according to the Euclidean Distance and we highlight the dots.

K-NN



Let’s follow the steps!

Step 3: We count the datapoints in each category. Category 1: 3 neighbours. Category 2: 2 neighbours.

Step 4: We assign the new datapoint to the category with more neighbours, therefore we would assign it to Category 1.

K-NN

Let’s try it in Python!

Classification

Support Vector Machine

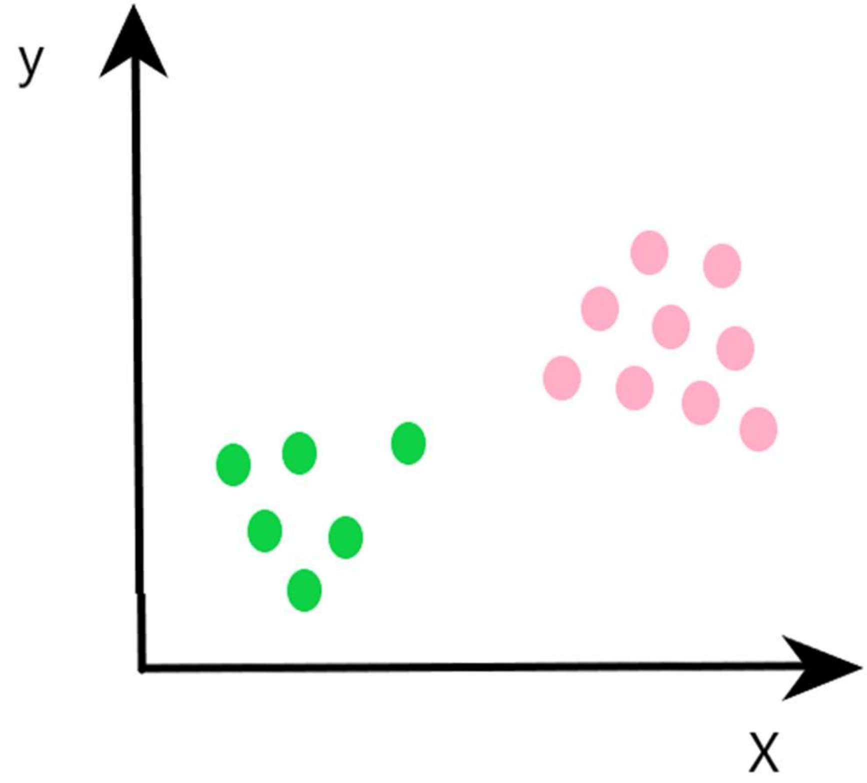
SVM

In previous classes we talked about Support Vector Machine. We used it in Regression Models as Support Vector Regression, and we know that this is not a new algorithm. The idea of this algorithm started in 70s and was finally put in practice by Vladimir Vapnik and his colleague from the AT & T Bell Laboratories in the 90s.

Let’s see how it works in classification.

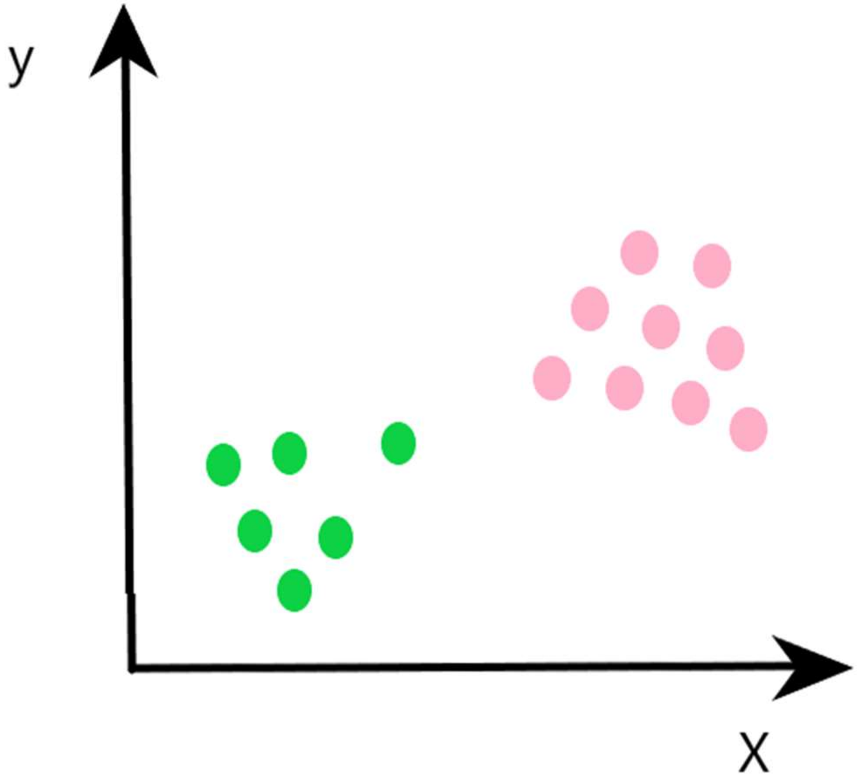
SVM

Suppose that we have two variables X and Y, and we have two categories, 1 and 2.

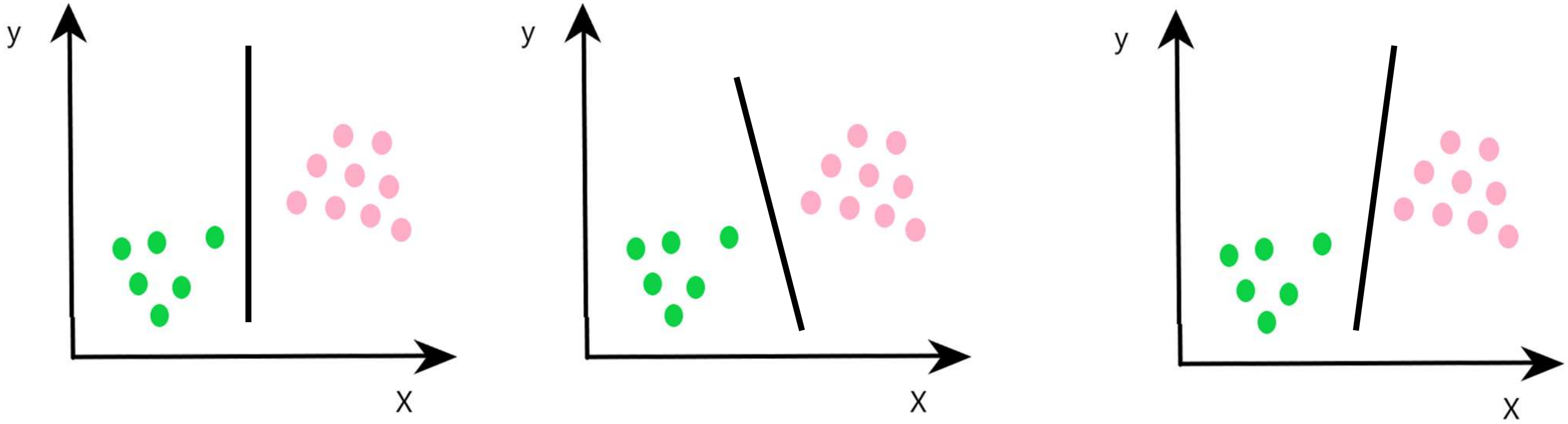


SVM

Now we want to create a line to divide the dots. We would have many different options.



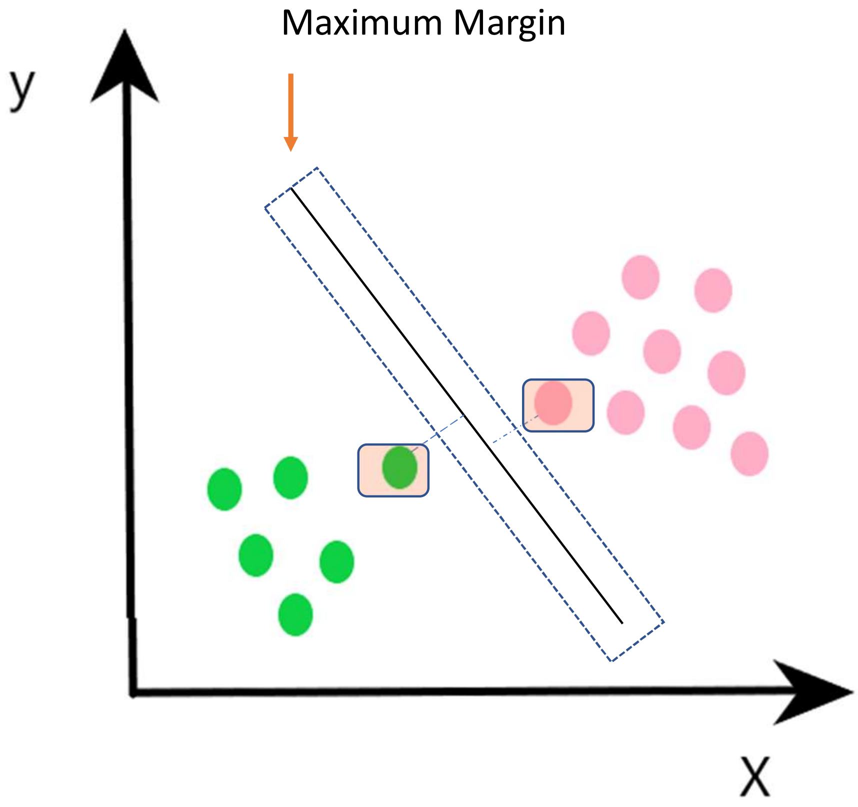
SVM



The line that we have to divide the dots will matter because when we have new datapoints, the line will determine whether to allocate the new observation in one category or another one.

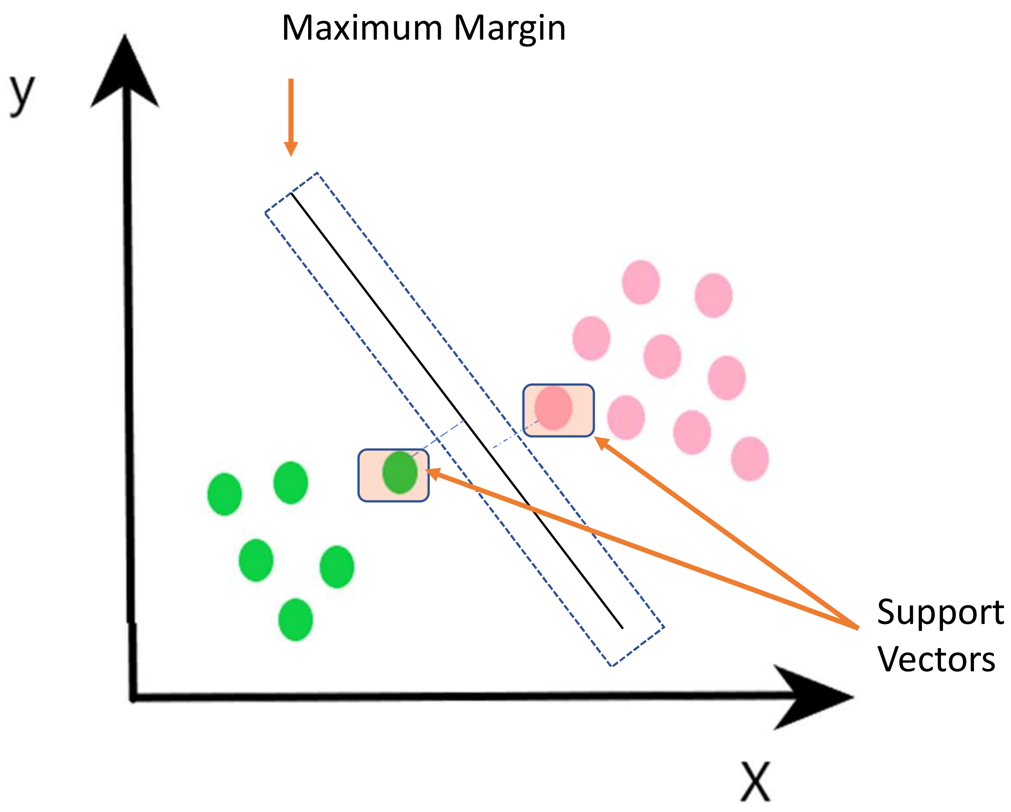
SVM

There will be a vector that will divide the categories and the closest points (see the highlight) will define where the vector is located. The distance between the dots and the line is the same.



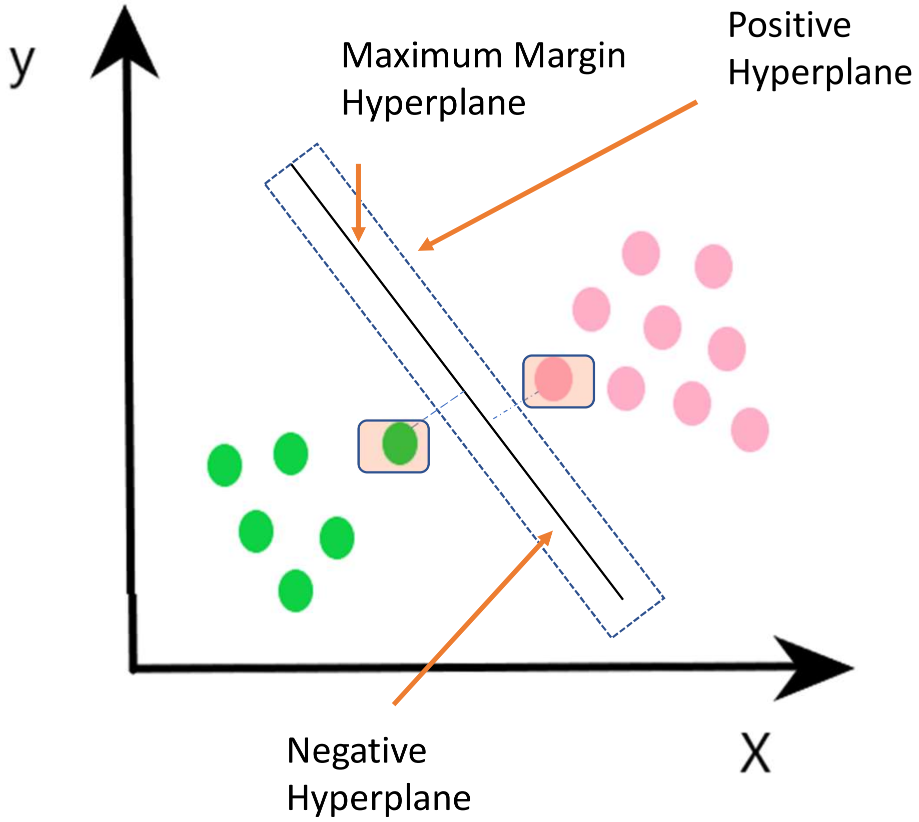
SVM

Those dots support the whole structure of it, that is why they are called the “supporting vectors”. Here we see them as points, but they are actually vectors. As the structure is supported by these two vectors, the algorithm is known as “Support Vector Machine”.



SVM

The right line of the vector is called “Positive Hyperplane” and the left line of the vector is called “Negative Hyperplane”.



SVM

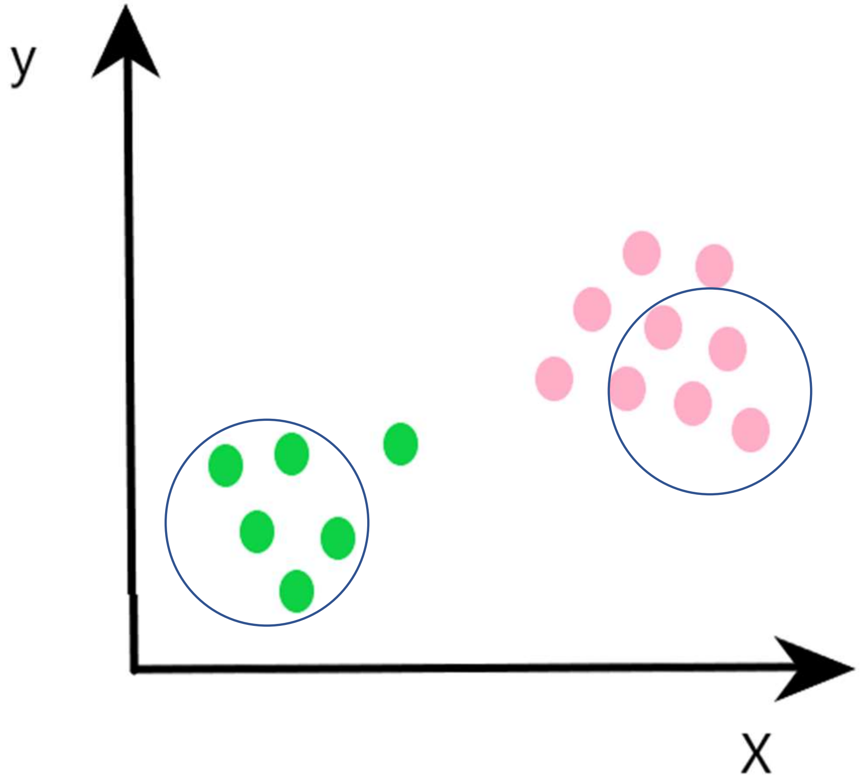
To sum up, this is a simple algorithm mathematically speaking. We receive a dataset, and the algorithm will create a vector to classify the dots.

What we do with this model, is teaching the machine how to distinguish between different elements.

SVM

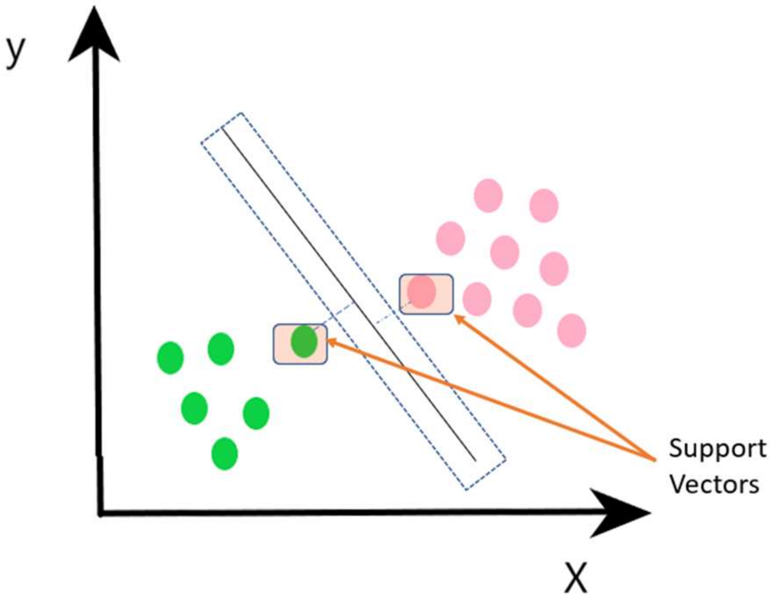
Following the original example, machine learning will teach the machine the differences between a house an apartment. The machine will analyse them, check what are the features, parameters, etc. And then, we show a property to the machine and it has to decide whether it is a house or an apartment.

SVM



Imagine that the green dots are the apartments, and the pink points are the houses. In a common machine learning process, the algorithm will identify those that really look like houses and those that really look like apartments.

SVM



What SVM does is to look at those houses that look like apartments and those apartments that look like houses. And if we have a look, the green dot close to the structure would be an apartment that looks like a house and the pink dot would be a house that looks like an apartment. This is why this algorithm is very powerful in comparison with other models.

SVM

Let’s try it in Python!

THAT'S ALL FOR TODAY

THANK YOU