
Bayesian Classifier & Support Vector Machines

By
Prof(Dr) Premanand Pralhad Gahdekar

Outline

1. Introduction
2. Naïve Bayes Classifier Algorithm
3. Numericals
4. Support Vector Machines Algorithm
5. Numericals etc

Naïve Bayes Classifier Algorithm

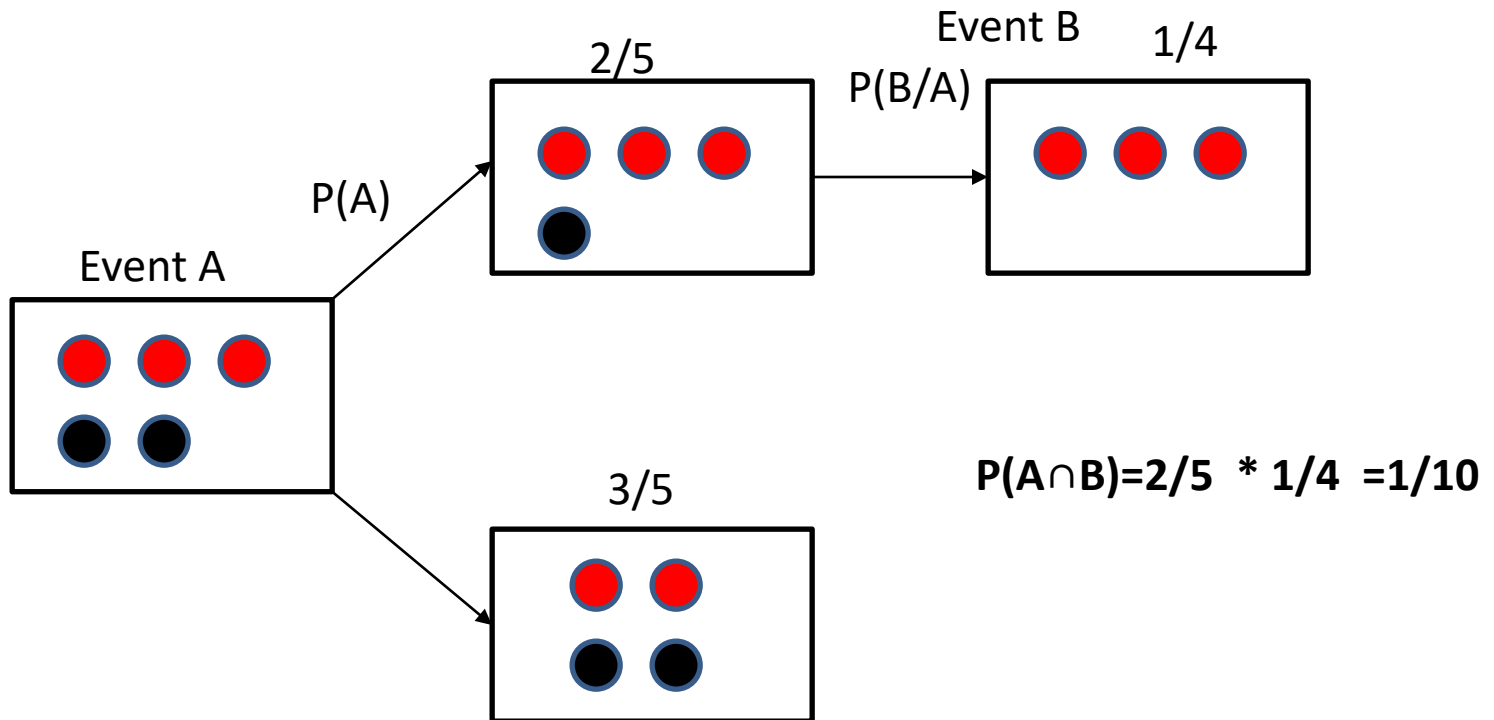
- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem**.
- It is mainly used in **text classification** that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is **one of the simple and most effective Classification algorithms** which helps in **building the fast machine learning models that can make quick predictions**.
- **It is a probabilistic classifier**, which means it predicts on the **basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**

Naïve Bayes Classifier Algorithm

- ❖ **Naïve:** It is called Naïve because it assumes that **the occurrence of a certain feature is independent of the occurrence of other features.**
- ❖ **Ex-** Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- ❖ **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes Theorem

- ❖ **Conditional Probability**
- ❖ **Independent Events**-Tossing two Coins
- ❖ **Dependent Events**-Red & Black Marbles in a Bag



Bayes Theorem Proof

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

$$= P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Bayes' Theorem

- ❖ Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- ❖ The formula for **Bayes' theorem** is given as:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

- ❖ Where,
- ❖ **P(A|B) is Posterior probability:** Probability of hypothesis A on the observed event B.
- ❖ **P(B|A) is Likelihood probability:** Probability of the evidence given that the probability of a hypothesis is true.
- **P(A) is Prior Probability:** Probability of hypothesis before observing the evidence.
- **P(B) is Marginal Probability:** Probability of Evidence.

Naïve Bayes

- What is Naïve Bayes
- Bayes Theorem and Its use
- Mathematical working of Naïve Bayes
- Step by step programming Naïve Bayes
- Prediction using Naïve Bayes
- Predictive Analytics

Naïve Bayes

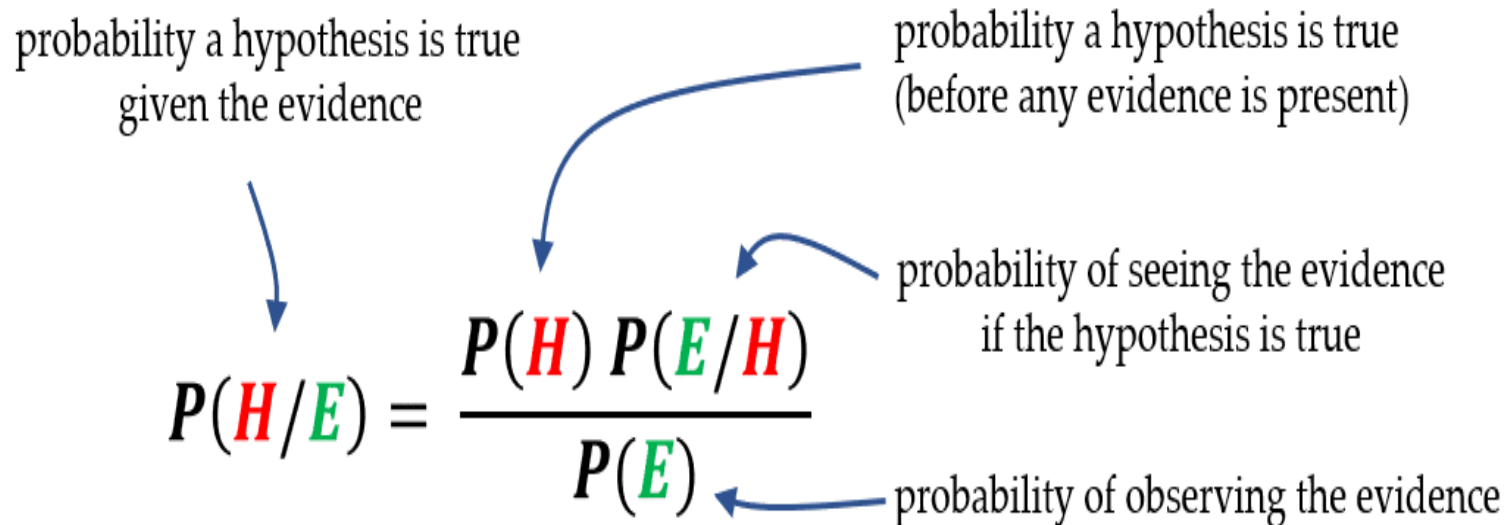
Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling.



Classification Technique

Bayes Theorem

Given a hypothesis H and Evidence E , Bayes theorem states that the relationship between the probability of the hypothesis before getting the evidence $P(H)$ and the probability of the Hypothesis after getting the evidence $P(H/E)$ is



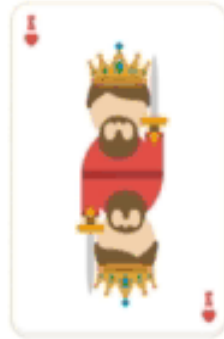
The diagram illustrates Bayes' Theorem with the following components and arrows:

- Left side:** $P(H/E)$ is labeled "probability a hypothesis is true given the evidence". An arrow points from this text to the $P(H/E)$ term.
- Right side (Numerator):**
 - $P(H)$ is labeled "probability a hypothesis is true (before any evidence is present)". An arrow points from this text to the $P(H)$ term.
 - $P(E/H)$ is labeled "probability of seeing the evidence if the hypothesis is true". An arrow points from this text to the $P(E/H)$ term.
- Right side (Denominator):**
 - $P(E)$ is labeled "probability of observing the evidence". An arrow points from this text to the $P(E)$ term.

$$P(H/E) = \frac{P(H) P(E/H)}{P(E)}$$

Bayes Theorem Example

Calculate the Probability of King card if the face card is given?



$$P(\text{King}|\text{Face}) = \frac{P(\text{Face}|\text{King}).P(\text{King})}{P(\text{Face})}$$

$$= \frac{1.(1/13)}{3/13} = 1/3$$

$$P(\text{King}) = 4/52 = 1/13$$

$$P(\text{Face}|\text{King}) = 1$$

$$P(\text{Face}) = 12/52 = 3/13$$

Naïve Bayes

Classification technique based on Bayes theorem

Assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

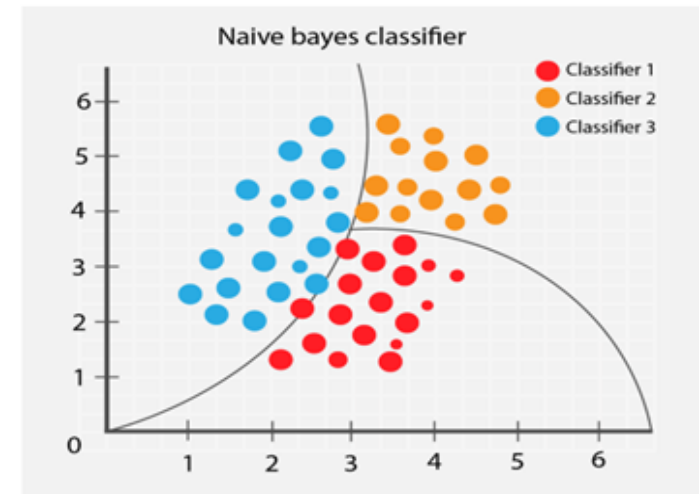
Naive Bayes

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$



Naïve Bayes

A **posterior probability**, in Bayesian statistics, is the revised or updated probability of an event occurring after taking into consideration new information.

The posterior probability is calculated by updating the prior probability using Bayes' theorem.

Likelihood

How probable is the evidence
Given that our hypothesis is true?

Prior

How probable was our hypothesis
Before observing the evidence?

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

Posterior

How probable is our Hypothesis
Given the observed evidence?
(Not directly computable)

Marginal

How probable is the new evidence
Under all possible hypothesis?

Naïve Bayesian Classifier

❖ If the features or attributes are assumed to be implemented, the resulting classifier is called **Naïve Bayesian Classifier**.

❖ Algorithm

1. Train the classifier with the training images or labelled featured data.
2. Compute the probability $P(i)$ using intuition, based on experts' opinion, or using Histogram-based estimation.
3. Compute $P(i/x)$
4. Find the maximum $P(i/x)$ and assign the unknown instances to that class.

Naïve Bayesian classifier does not work for real time datasets as it Naïve to assume that the features are independent of each other and also naïve Bayesian classification does not work for continuous data.

Naïve Bayes Working

Classification Steps

- Handling Data
- Summarizing Data
- Making a Prediction
- Making all the Prediction
- Evaluate the Accuracy
- Tying all together

Bayesian Classifier-Numerical

Naive Bayes Classifier Algo.

Fruit = { Yellow, Sweet, long }

Fruit	Yellow	Sweet	Long	Total
Orange	350	450	0	650
Banana	400	300	350	400
Others	50	100	50	150
Total	800	850	400	1200

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

$$P(\text{Yellow/orange}) = \frac{P(\text{orange/Yellow}) \cdot P(\text{Yellow})}{P(\text{orange})} = \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}} = 0.5$$

$$P(S/O) = 0.69, P(L/O) = 0$$

$$P(\text{Fruit/Orange}) = 0.53 \times 0.69 \times 0 = 0$$

$$P(\text{Fruit/Banana}) = 1 \times 0.75 \times 0.87 = 0.65$$

$$P(\text{Fruit/Others}) = 0.33 \times 0.66 \times 0.33 = 0.072$$

Bayesian Classifier-Numerical

1. Let us assume a simple dataset, as shown in Table. Let us apply the Bayesian classifier to predict (2,2).

a1	a2	class(i)
2	0	c1
0	2	c1
2	4	c2
0	2	c2
3	2	c2

Soln-Here $c1=2$ and $c2=3$ from the training set. Therefore the prior probabilities are $P(c1) = 2/5$ and $P(c2)=3/5$. The conditional probability is estimated.

$$P(a1=2/c1)=1/2; P(a1=2/c2) = 1/3$$

$$P(a2=2/c1)=1/2; P(a2=2/c2) = 2/3$$

Therefore, $P(x/c1) = P(a1=2/c1) \times P(a2=2/c1) = 1/2 \times 1/2 = 1/4$

Bayesian Classifier-Numerical

Soln-

$$\begin{aligned} P(x/c2) &= P(a1=2/c2) \times P(a2=2/c2) \\ &= 1/3 \times 2/3 = 2/9 \end{aligned}$$

$$p(x)=1$$

This is used to evaluate

$$\begin{aligned} P(c1/x) &= P(c1) \times P(x/c1)/p(x) \\ &= 2/5 \times 1/4 = 2/20=0.1 \end{aligned}$$

$$\begin{aligned} P(c2/x) &= P(c2) \times P(x/c2)/p(x) \\ &= 3/5 \times 2/9=6/45=0.13 \end{aligned}$$

Since $P(c2/x) > P(c1/x)$, the sample is predicted to be in class c2.

Bayesian Classifier-Numerical

1. Let us consider a classification problem that involves classification of an image pixel using a single feature colour into two classes-forest and non-forest. Let the prior probability of the forest class be 0.6, the feature i of colour green belonging to the forest image in the training set be 0.2, and the probability of the green pixel feature belonging to the forest in the overall population be 0.4. What is the probability that an image is a forest image given that the image contains the green colour feature?

Soln-For the two given classes, the only feature commonly available is colour. Let the feature be x . So the available information is

- (a) Prior probability of the class $P(i)$ is 0.6
- (b) Conditional Probability that the class i has $x=P(x/i)=0.2$
- (c) $P(x)=0.4$

So as per the Bayesian theorem,
$$P(i/x) = \frac{P(x/i) P(i)}{P(x)} = \frac{0.2 \times 0.6}{0.4} = 0.3$$

Bayesian Classifier-Numerical

Classify whether players will play or not based on weather condition using Naïve Bayes.

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	Yes
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Bayesian Classifier-Numerical

Frequency table		Play Game	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rain	3	2

Frequency table		Play Game	
		Yes	No
Humidity	High	4	3
	Normal	6	1

Frequency table		Play Game	
		Yes	No
Wind	Strong	3	3
	Weak	7	1

Bayesian Classifier-Numerical

Problem: If the weather is sunny, then the Player should play or not?

Likelihood Table		Play		
		Yes	No	
Outlook	Sunny	3/10	2/4	5/14
	Overcast	4/10	0/4	4/14
	Rainy	3/10	2/4	5/14
		10/14	4/14	

$P(x|c) = P(\text{Sunny} | \text{Yes}) = 3/10 = 0.3$
 $P(x) = P(\text{Sunny}) = 5/14 = 0.36$
 $P(\text{No}) = 4/14 = 0.29$
 $P(c) = P(\text{Yes}) = 10/14 = 0.71$

Likelihood of 'Yes' given Sunny is

$$P(c|x) = P(\text{Yes} | \text{Sunny}) = P(\text{Sunny} | \text{Yes}) * P(\text{Yes}) / P(\text{Sunny}) = (0.3 \times 0.71) / 0.36 = 0.591$$

Similarly Likelihood of 'No' given Sunny is

$$P(c|x) = P(\text{No} | \text{Sunny}) = P(\text{Sunny} | \text{No}) * P(\text{No}) / P(\text{Sunny}) = (0.5 \times 0.29) / 0.36 = 0.40$$

$$P(\text{Yes} | \text{Sunny}) > P(\text{No} | \text{Sunny})$$

Hence on a Sunny day, Player can play the game.

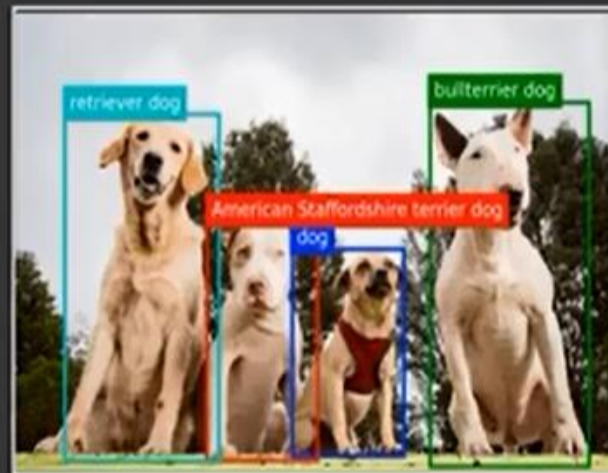
Bayesian Classifier-Applications



- National
- International
- Sports
- Media
- Travel & Lifestyle
- Stock Market
- Politics
- Finance

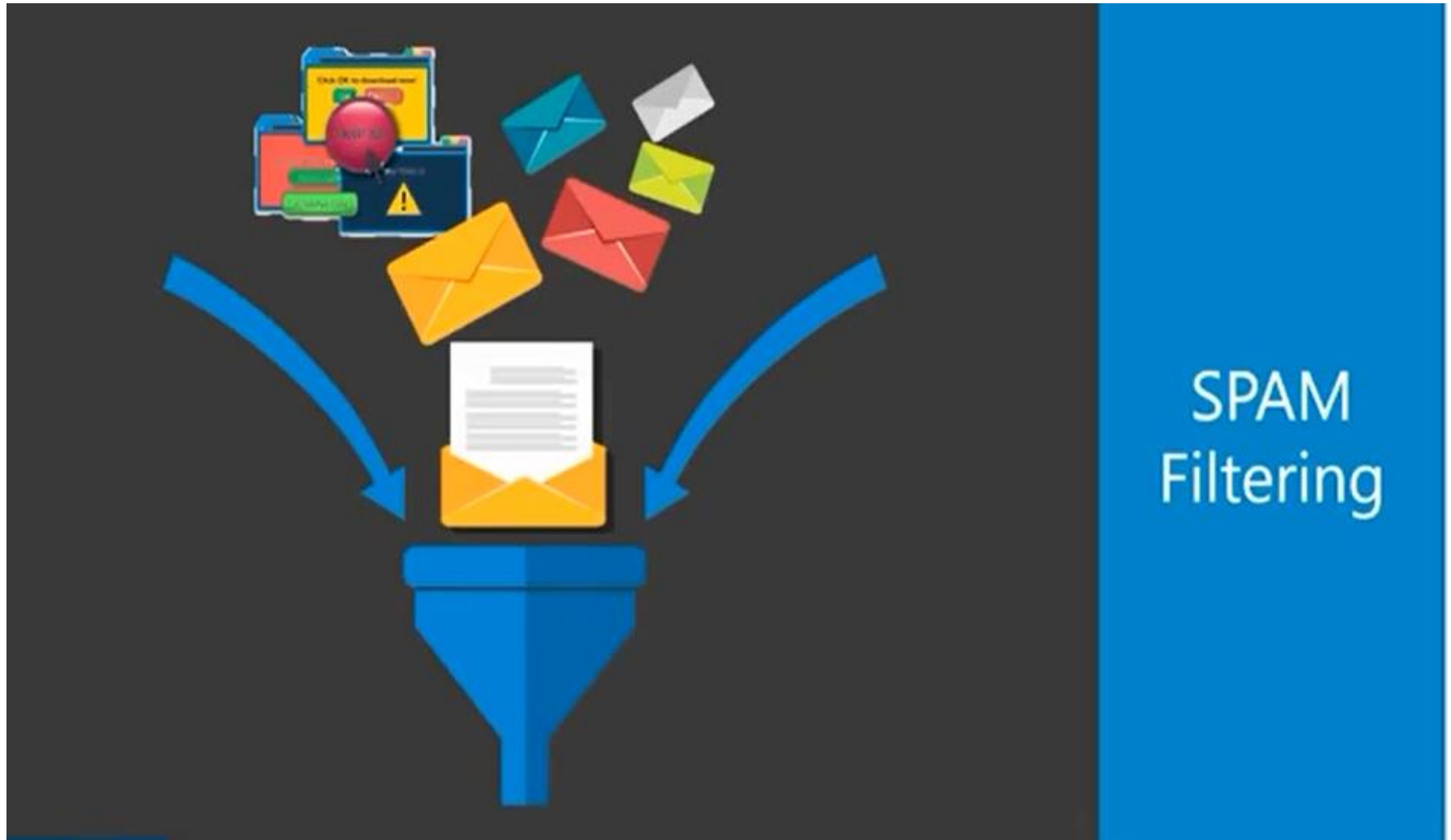
NEWS
Categorization

Bayesian Classifier-Applications



OBJECT
&
FACE
Recognition

Bayesian Classifier-Applications



Bayesian Classifier-Applications



Support Vector Machine

What is SVM

Support Vector Machine is a supervised Classification method that separates data using Hyperplanes.



Supervised machine
learning algorithm



Classification &
Regression algorithm



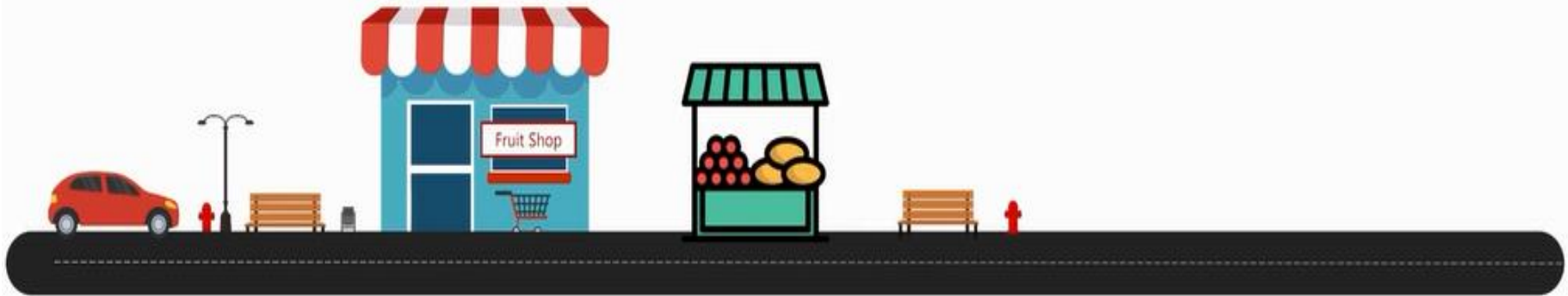
SVM kernel
functions

Support Vector Machine

- ❖ **Support Vectors**
- ❖ **Hyperplanes**
- ❖ **Marginal Distance**
- ❖ **Linear Separable**
- ❖ **Non Linear Separable**

Why Support Vector Machine

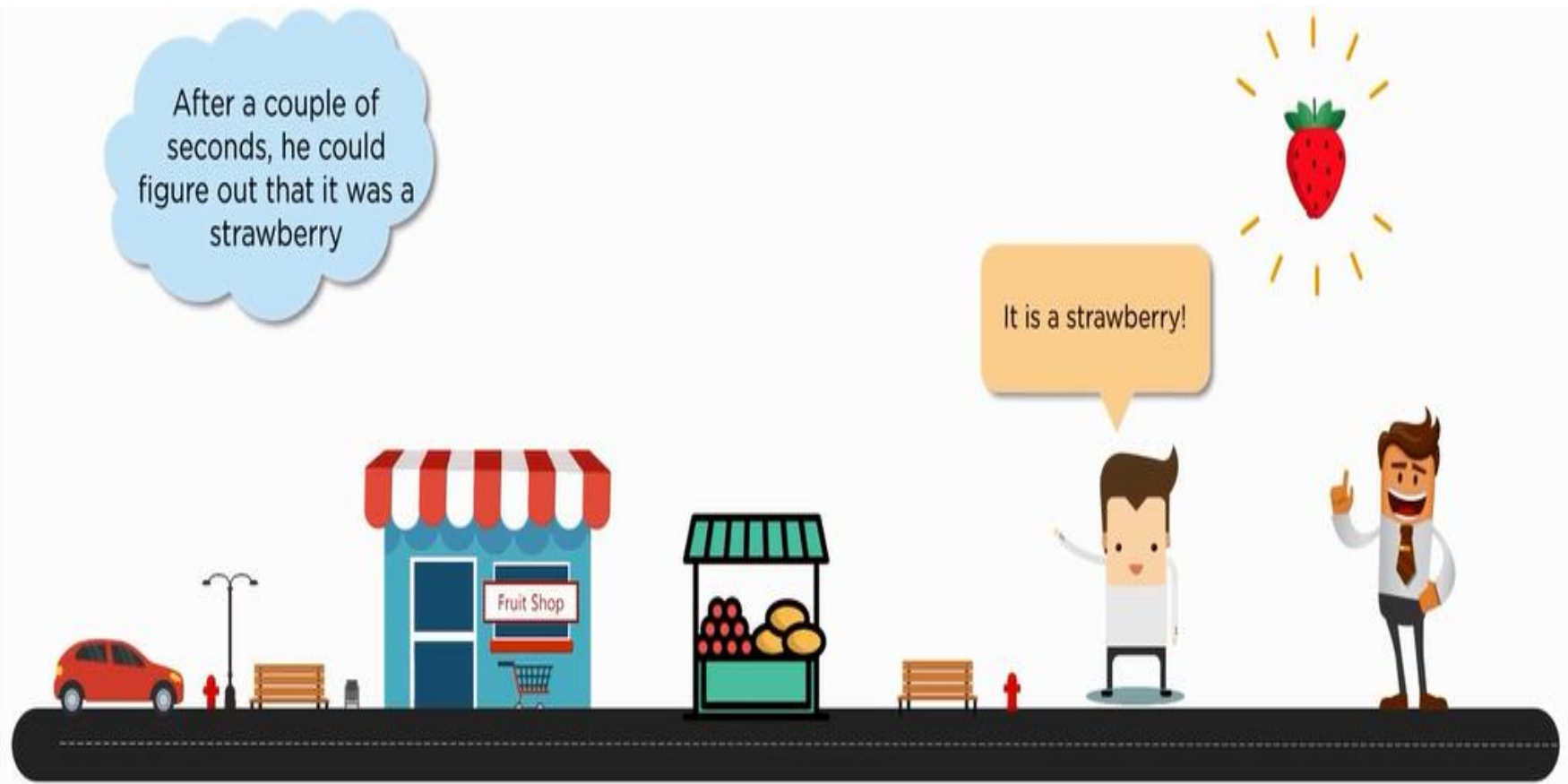
Last week, my son and I
visited a fruit shop



Why Support Vector Machine

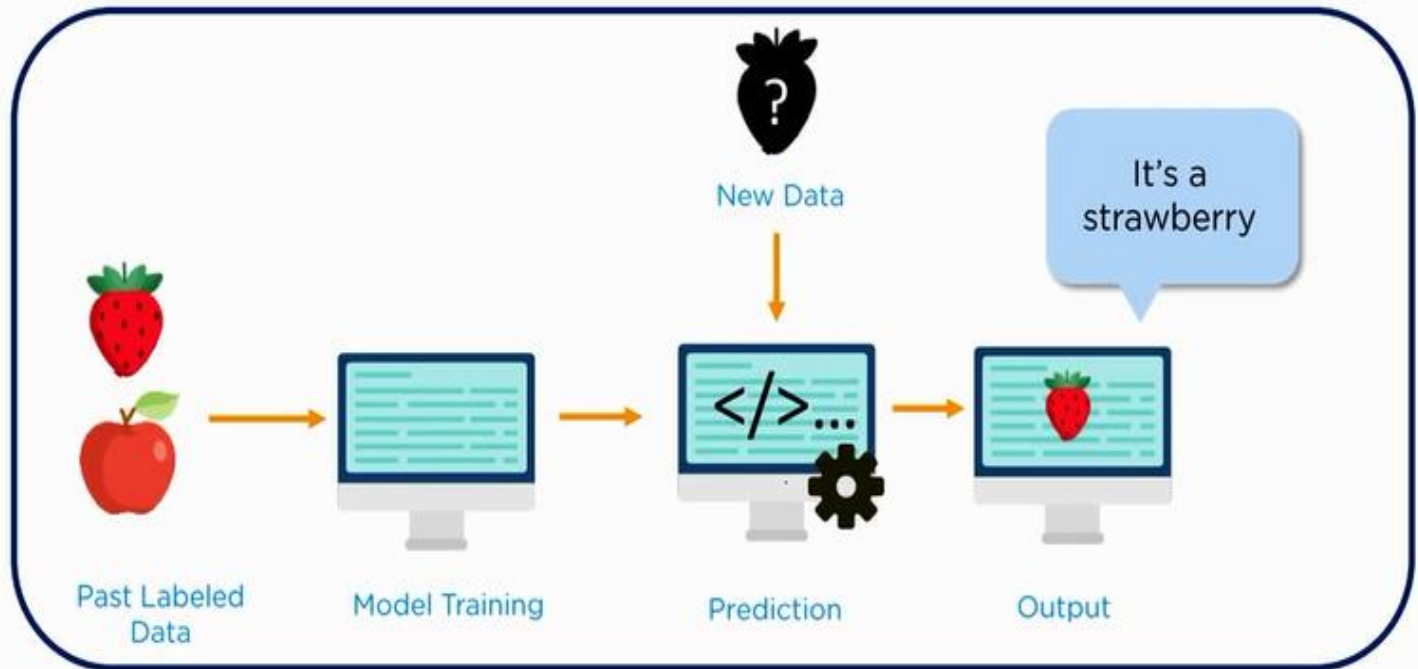


Why Support Vector Machine

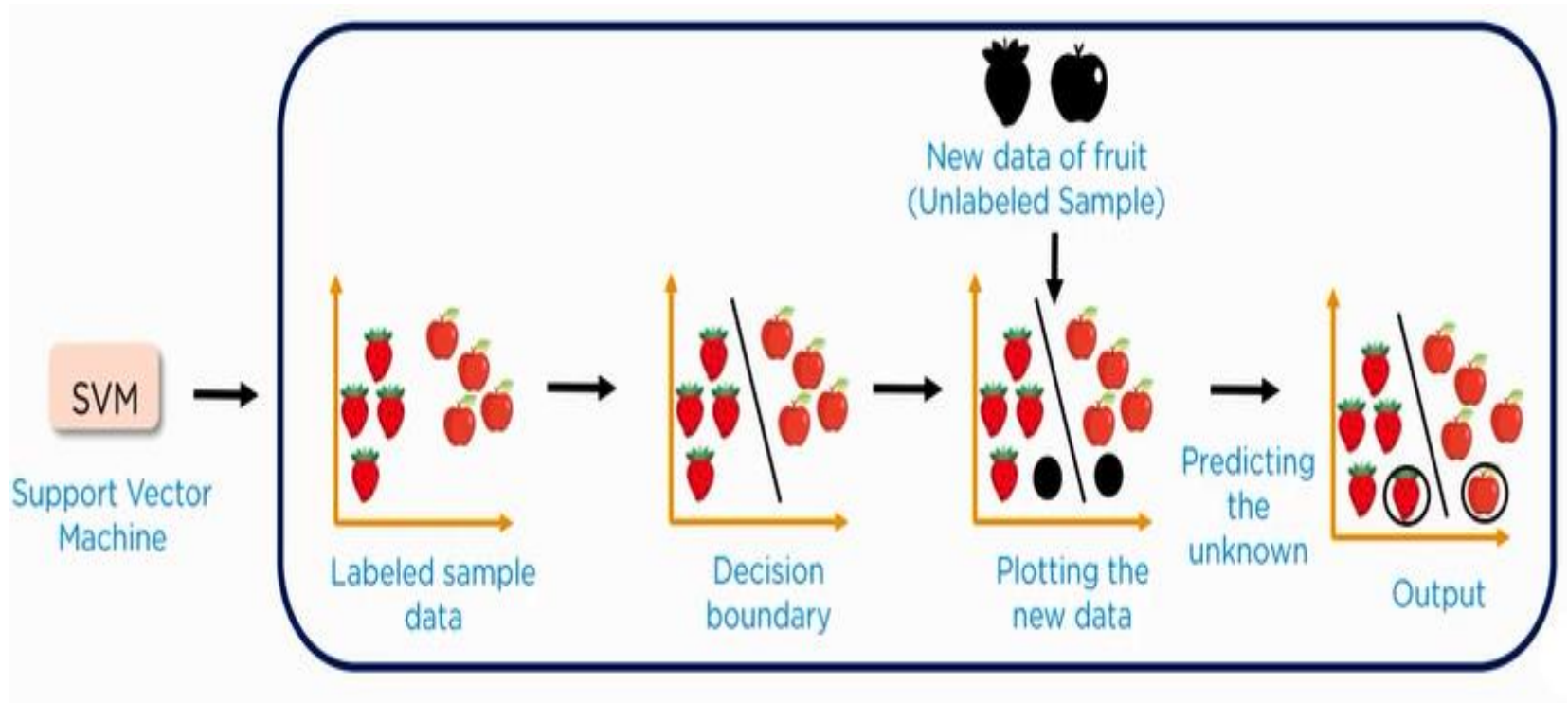


Why Support Vector Machine

Why not build a model which can predict an unknown data??

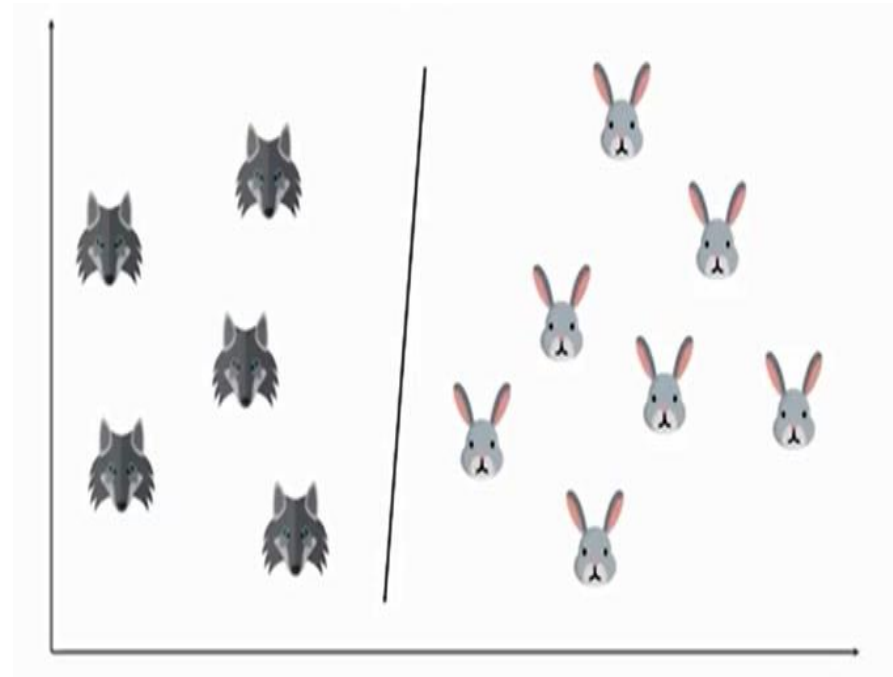
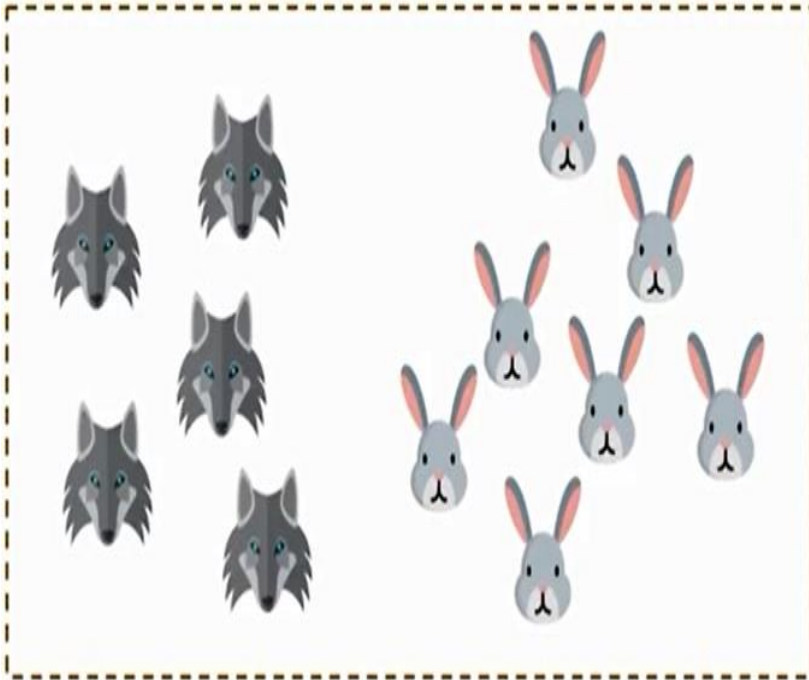


Why Support Vector Machine



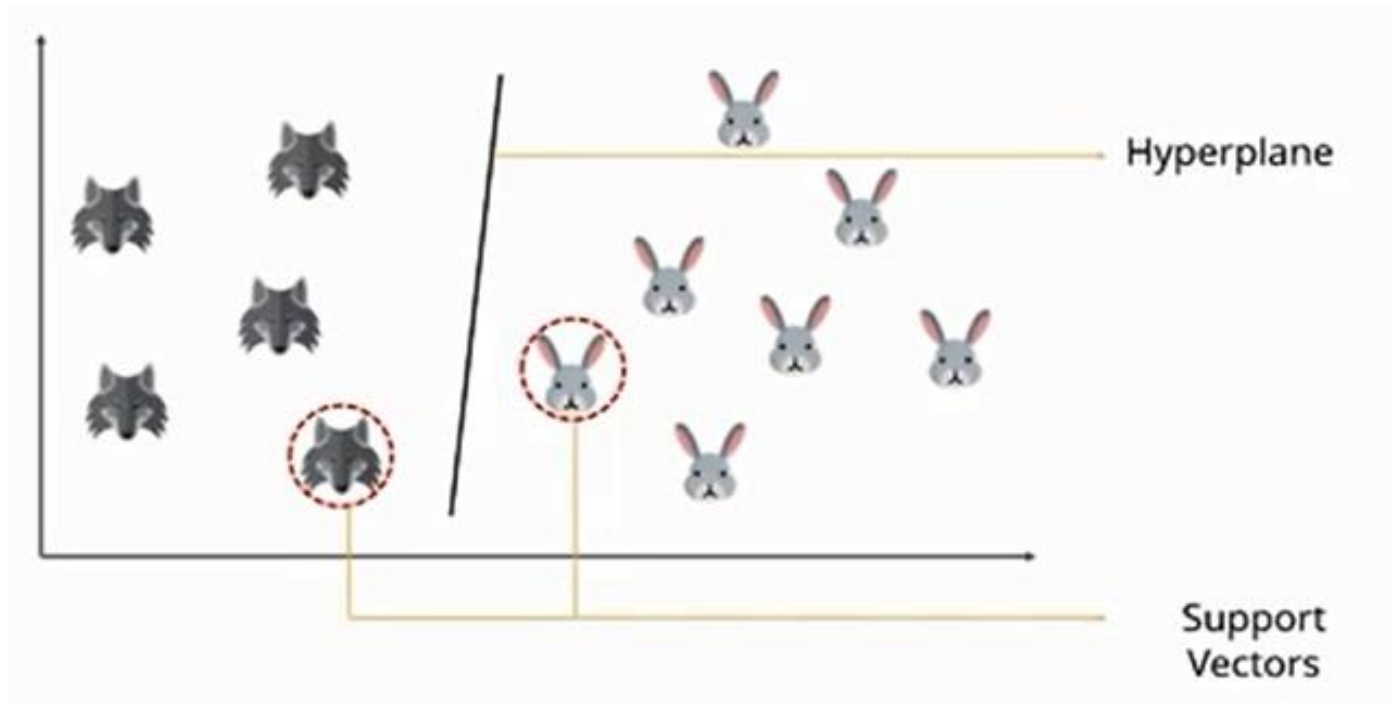
How does Support Vector Machine Work

Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.



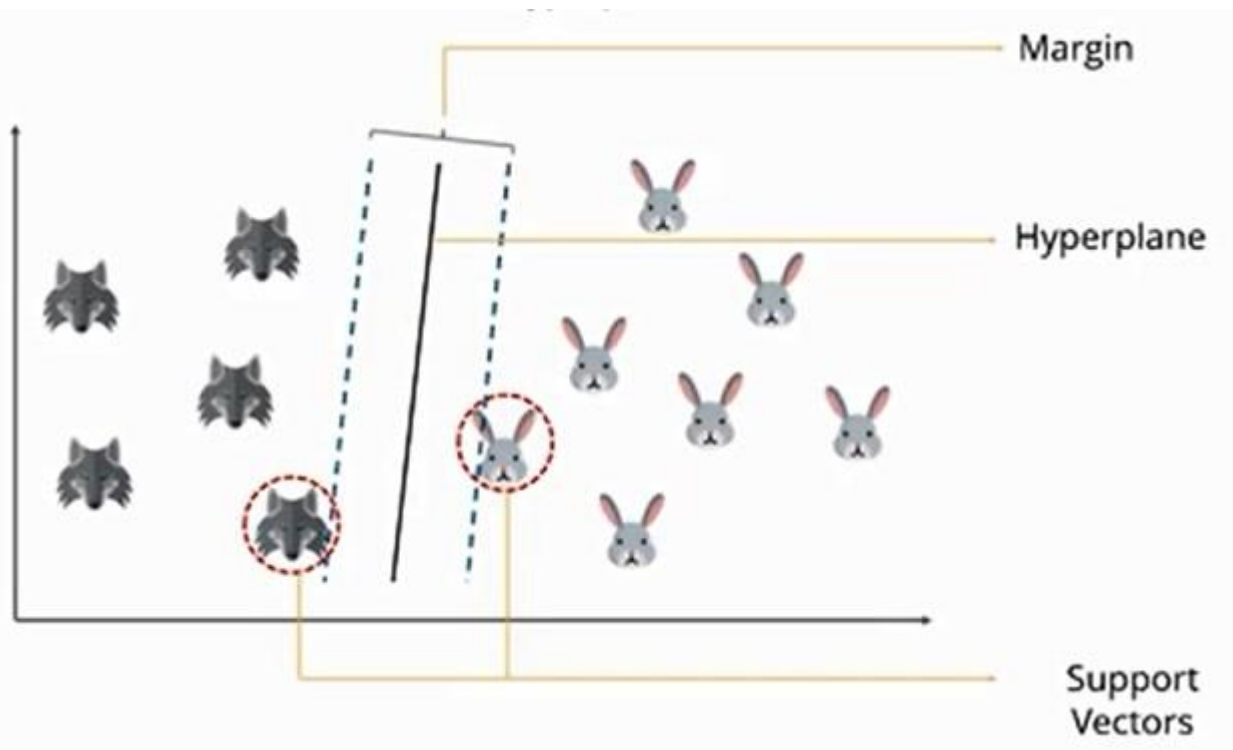
How does Support Vector Machine Work

Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.



How does Support Vector Machine Work

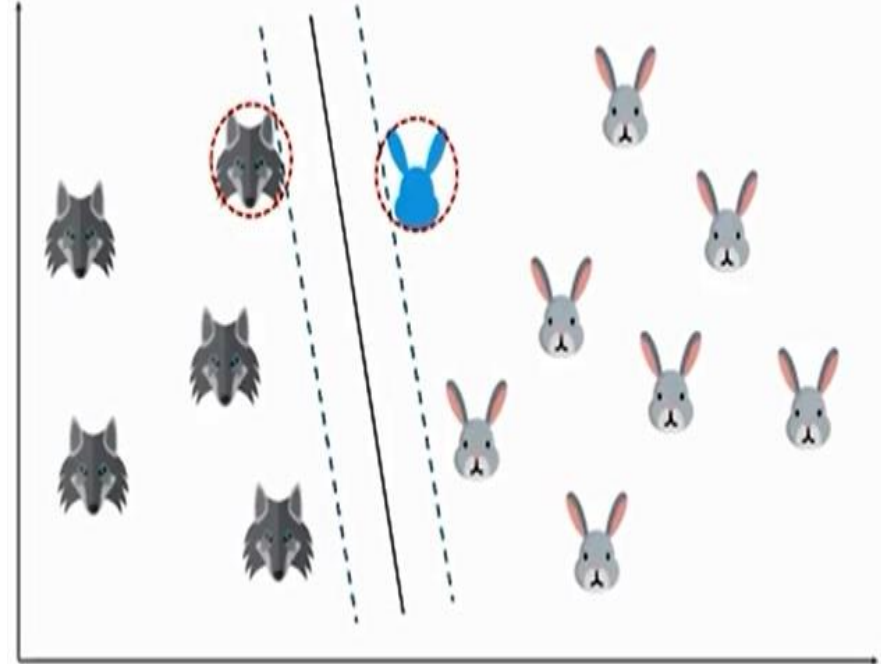
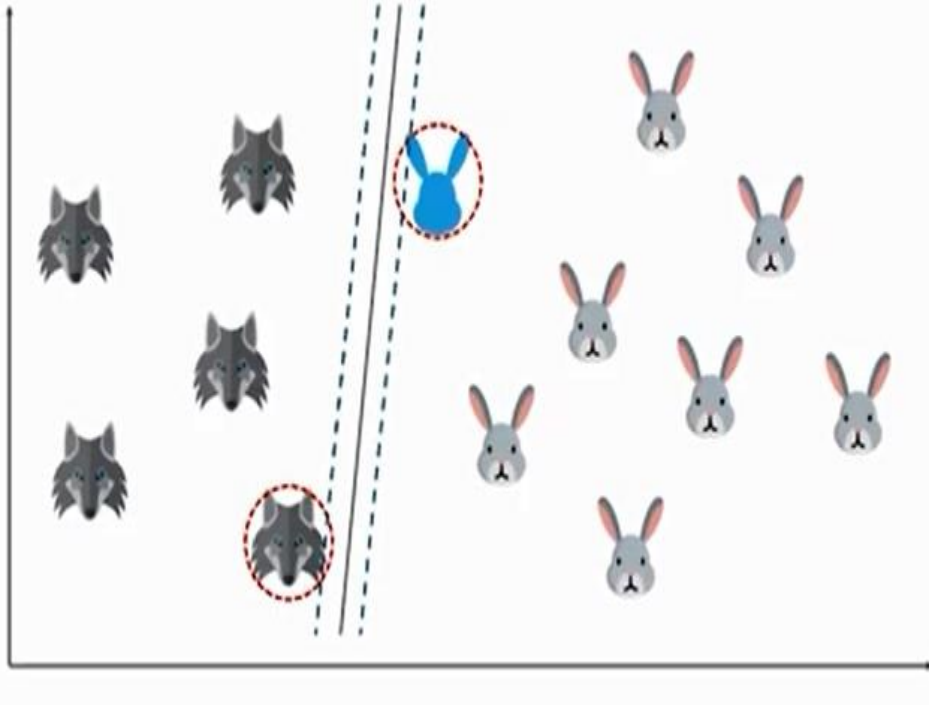
Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.



Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

How does Support Vector Machine Work

Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.

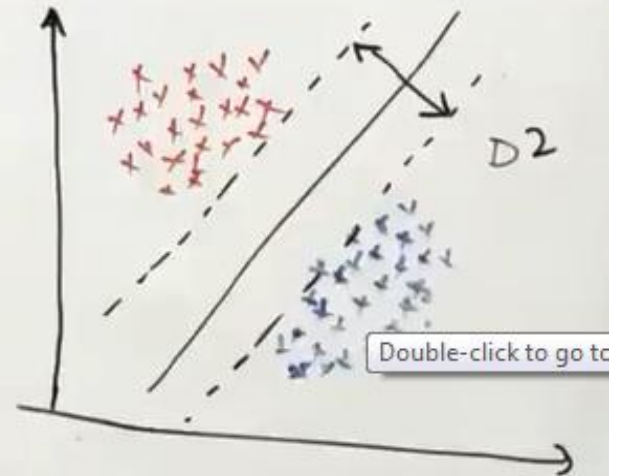
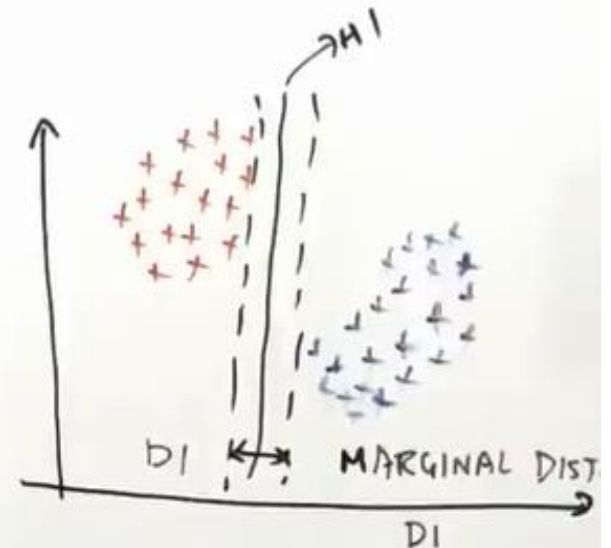


Support Vector Machine-Use Cases

SUPPORT VECTOR MACHINES



- ① Support Vectors
- ② Hyperplanes
- ③ Marginal Distance
- ④ Linear Separable
- ⑤ Non Linear Separable



Double-click to go to

Support Vector Machine-Example

Problem Statement-We have the people of different heights and widths

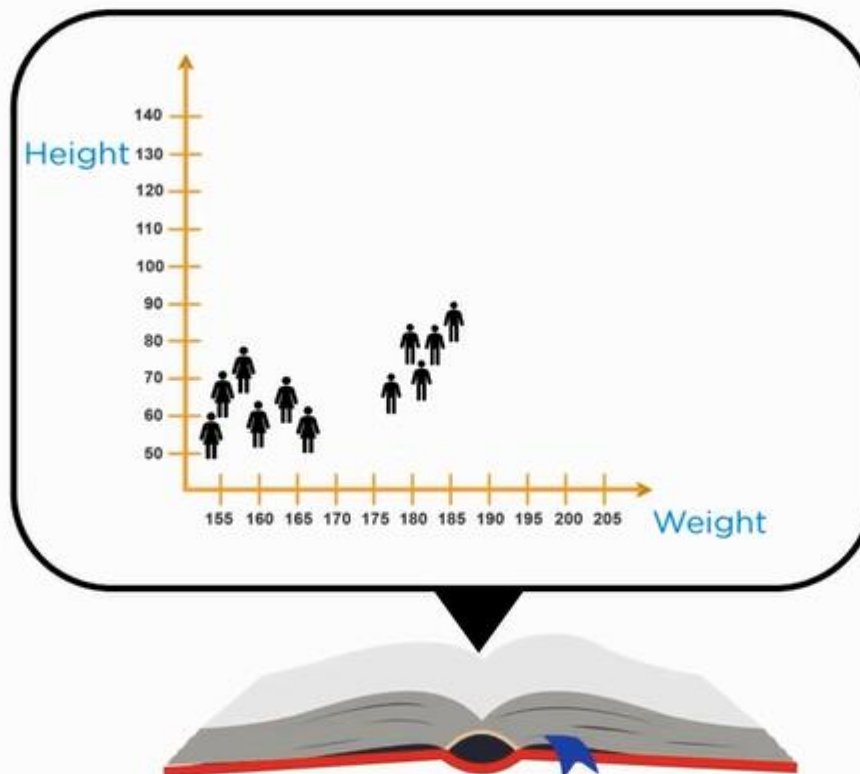
Female

Height	Weight
174	65
174	88
175	75
180	65
185	80

male

Height	Weight
179	90
180	80
183	80
187	85
182	72

Support Vector Machine-Example

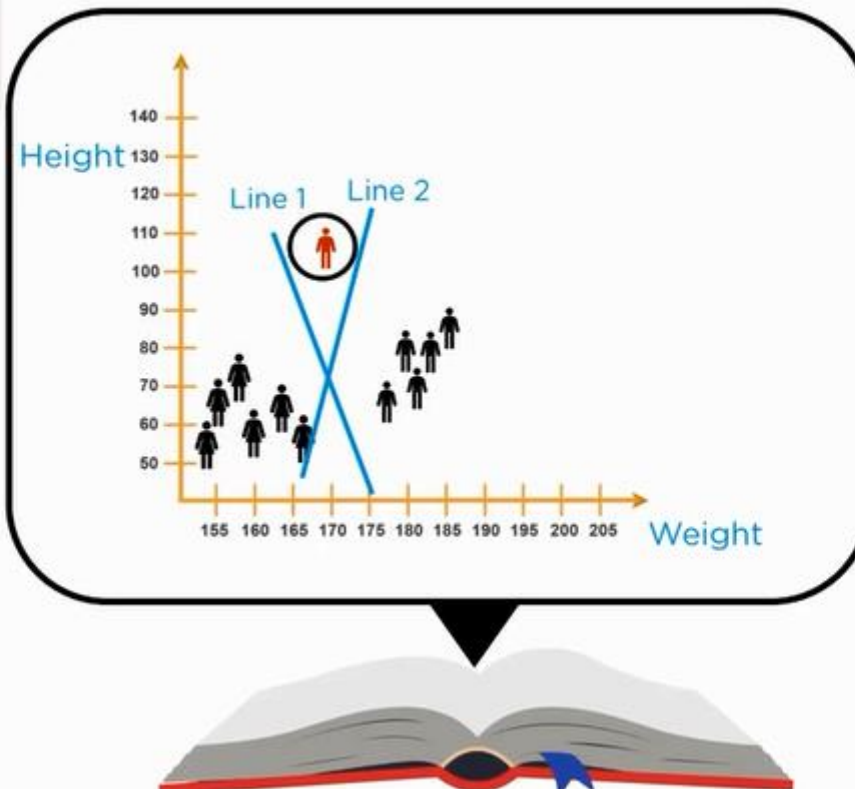


Let's add a new data point and figure out if it's a male or a female?



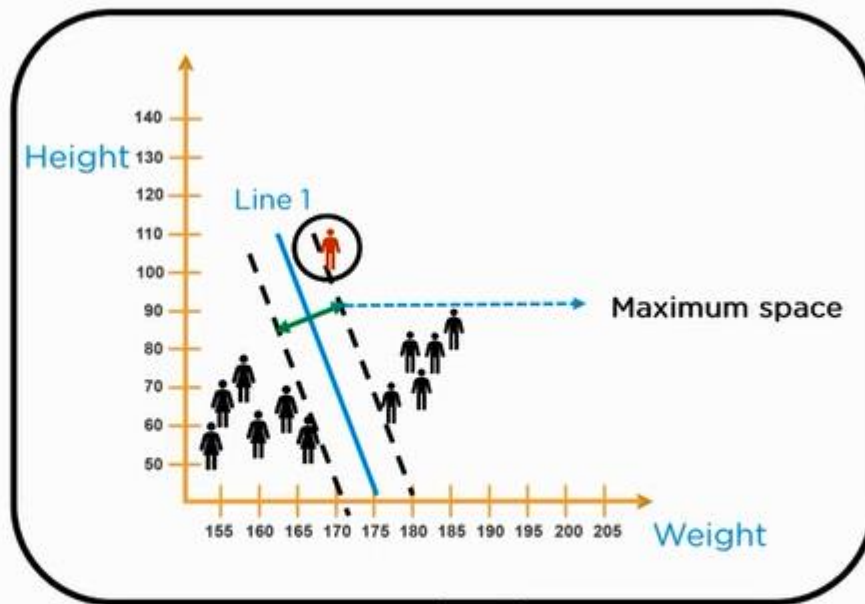
Support Vector Machine-Example

We can split our data by choosing any of these lines



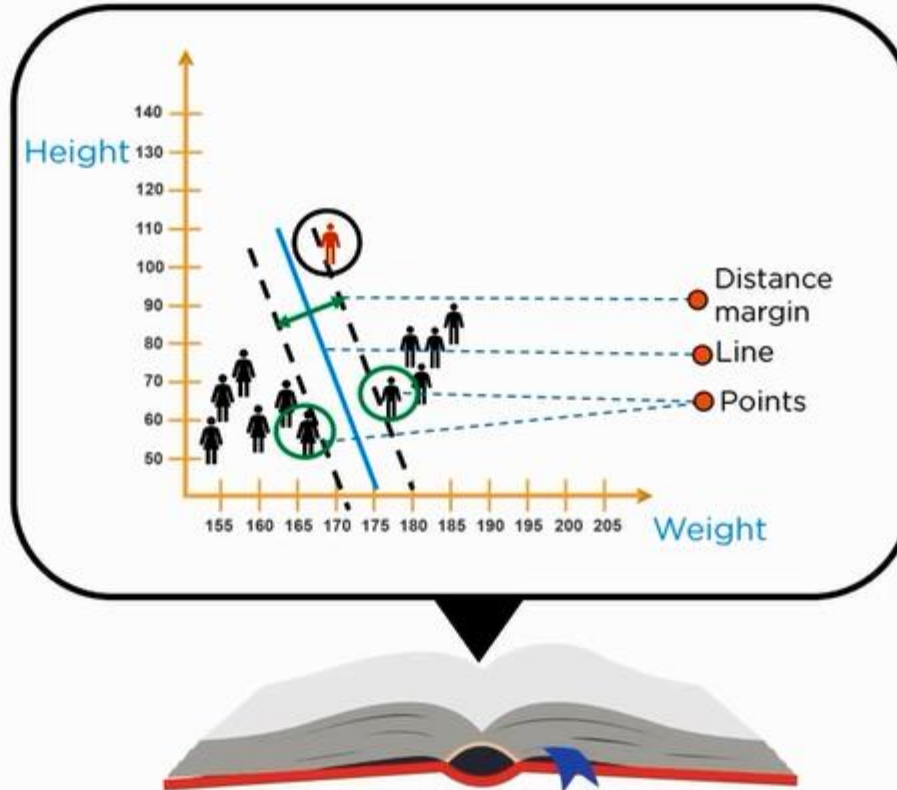
Support Vector Machine-Example

This line has the maximum space that separates the two classes



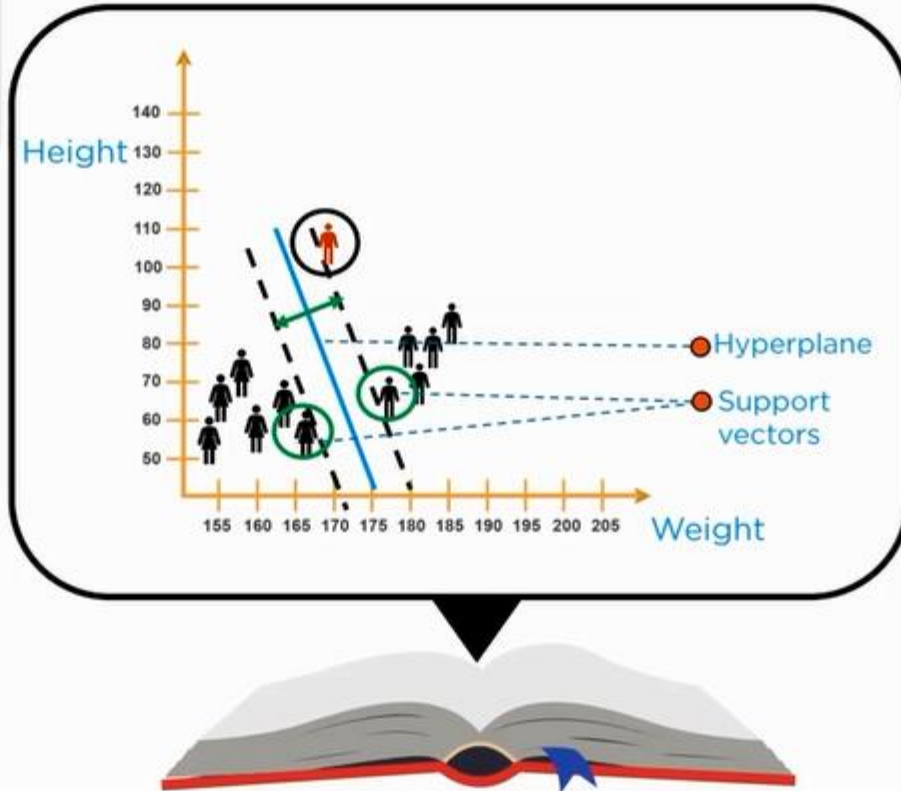
Support Vector Machine-Example

We can also say that the distance between the points and the line should be far as possible



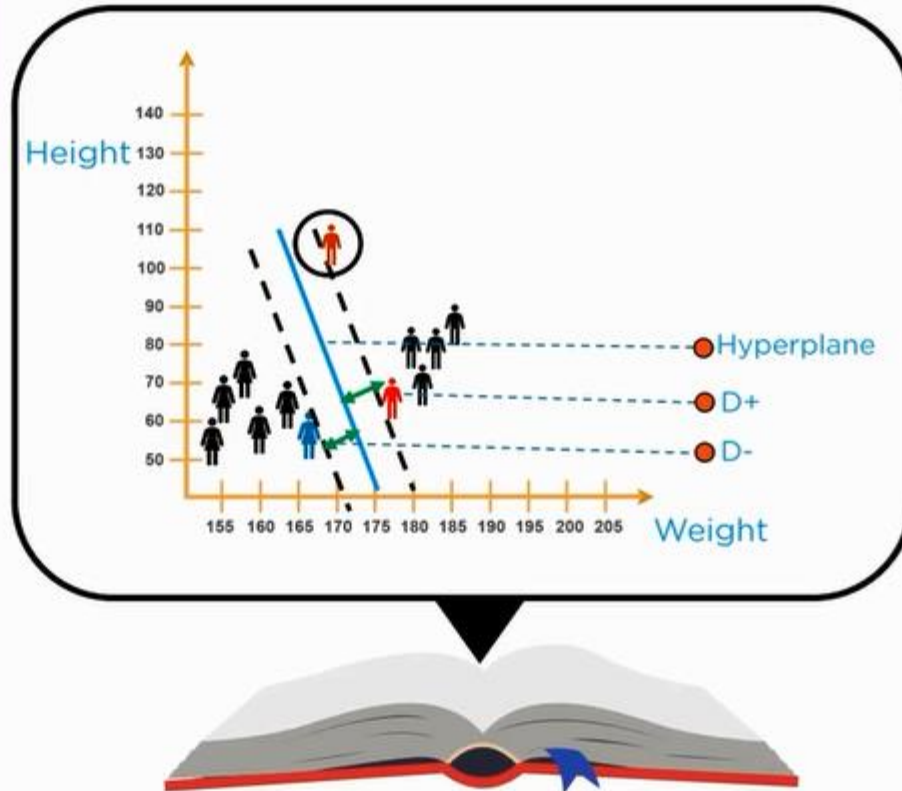
Support Vector Machine-Example

In technical terms, we can say that the distance between the support vector and the hyperplane should be as far as possible



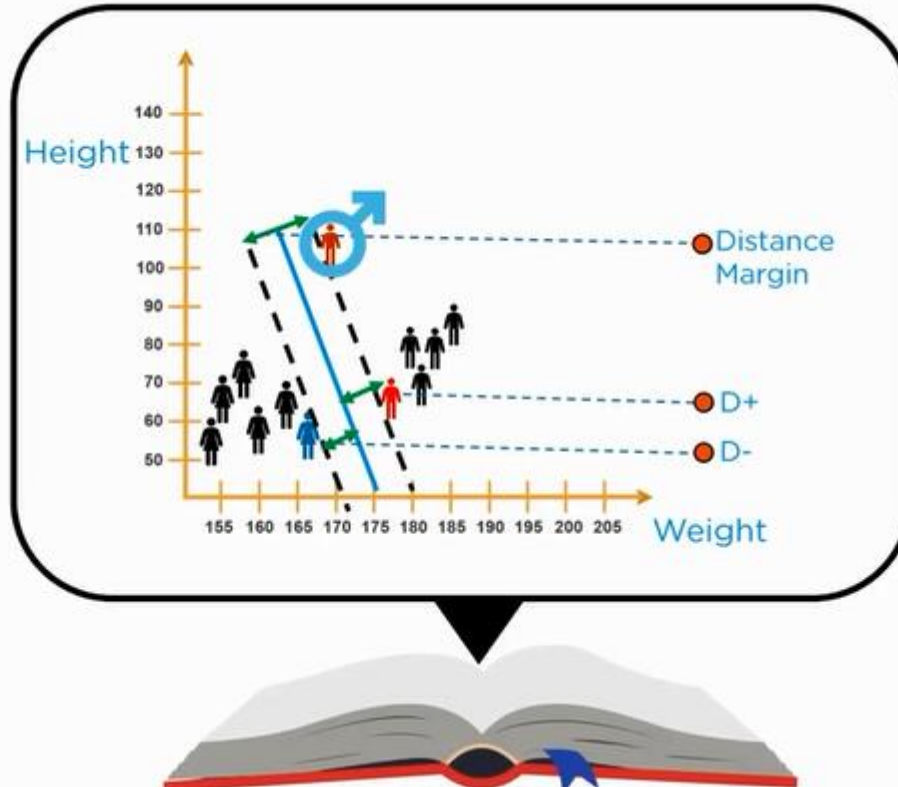
Support Vector Machine-Example

And D^- is the shortest distance to the closest negative point



Support Vector Machine-Example

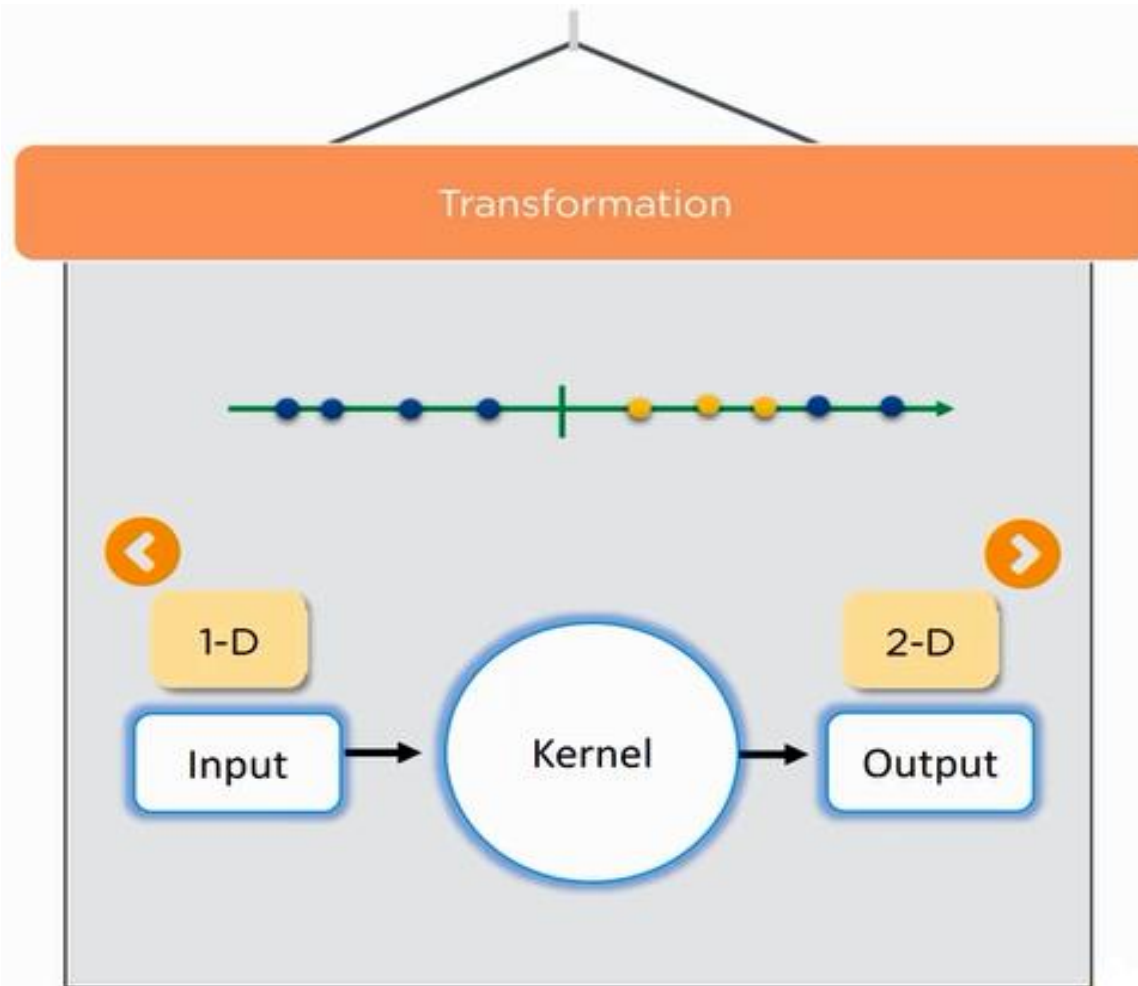
Based on the hyperplane, we can say the new data point belongs to male gender



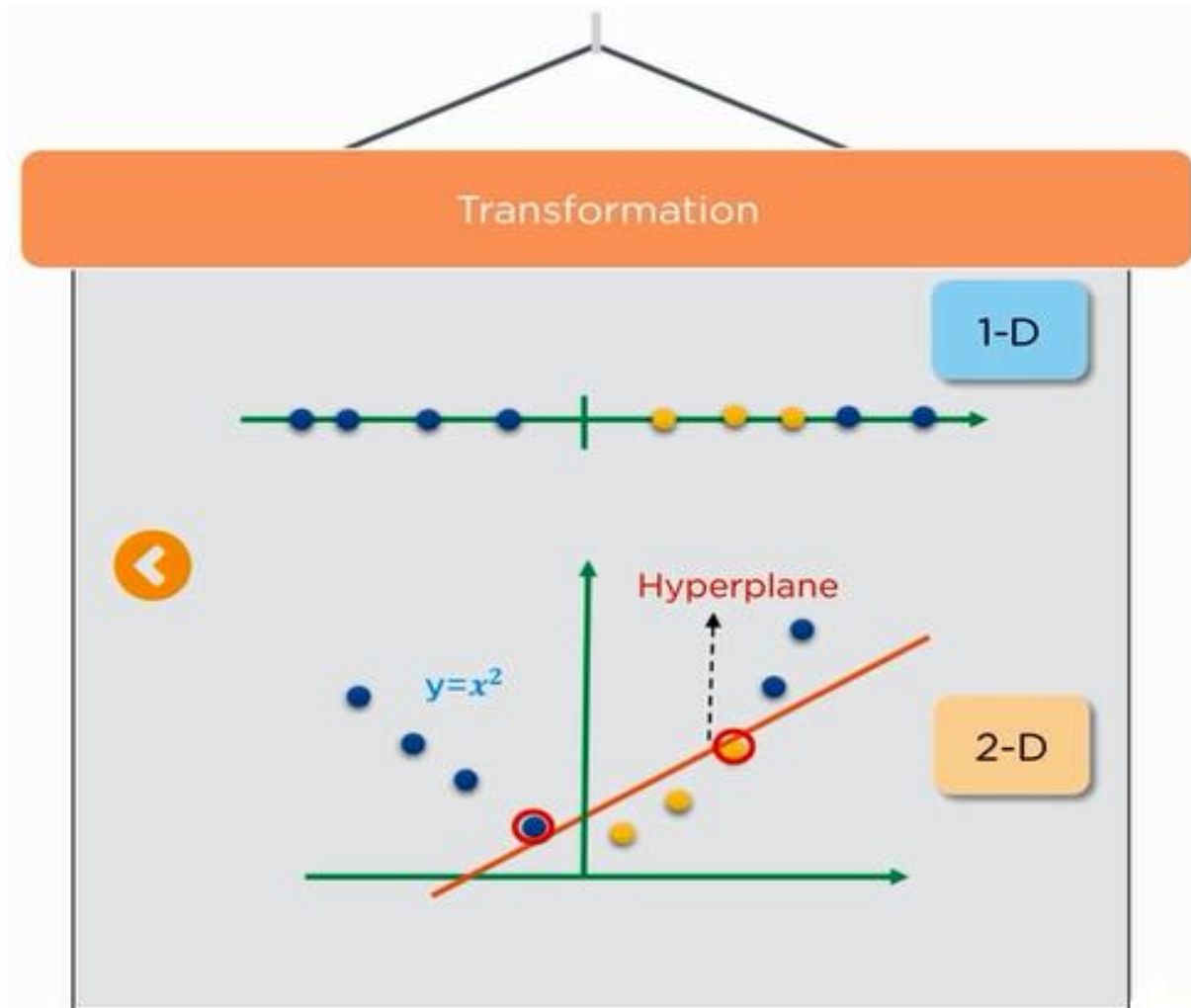
Support Vector Machine-Example



Support Vector Machine-Example

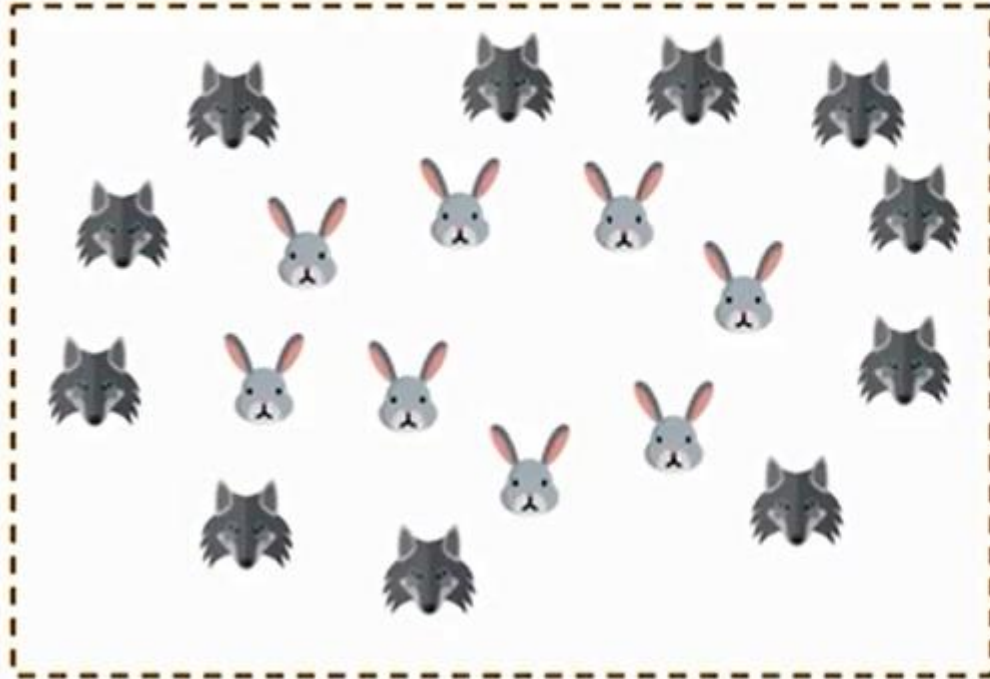


Support Vector Machine-Example

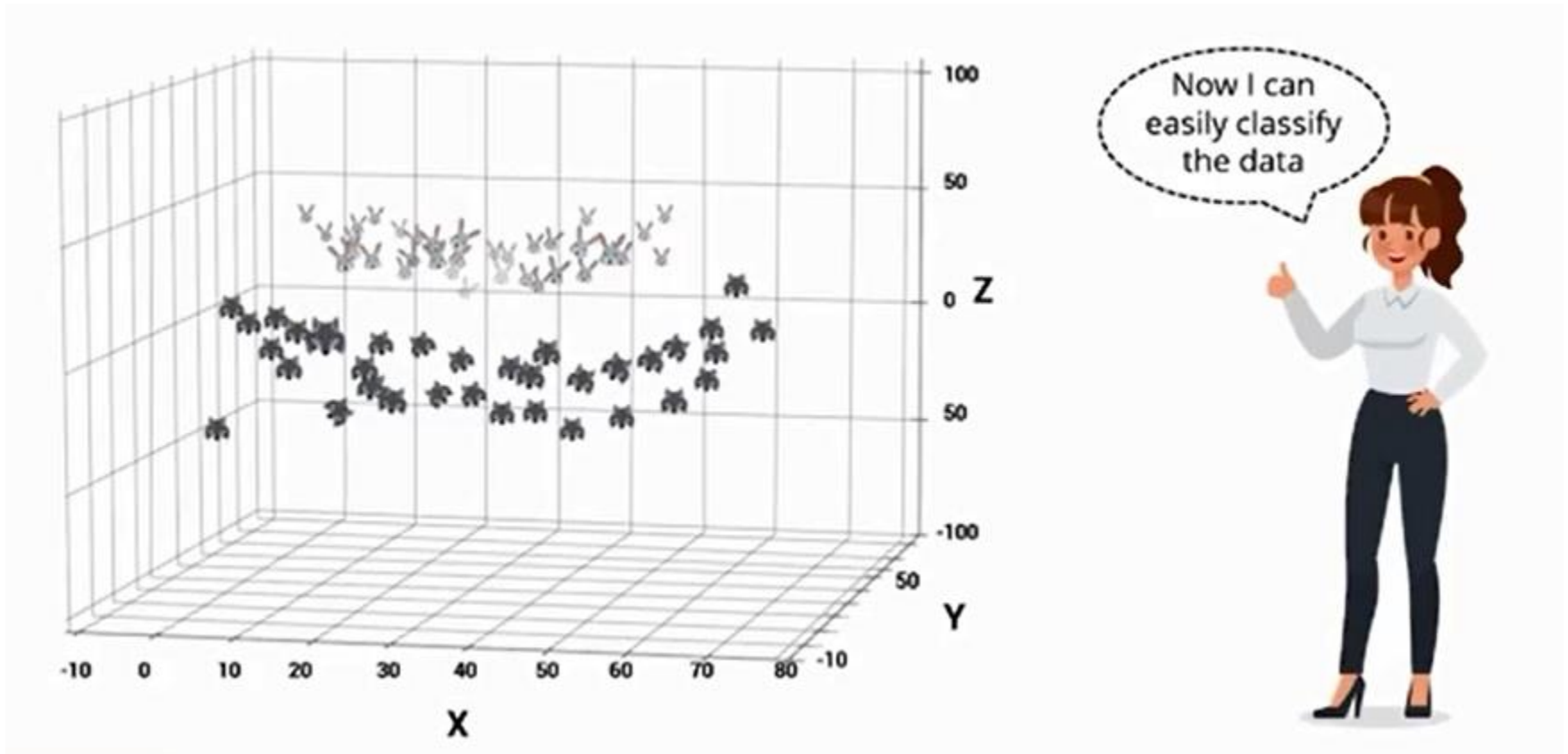


Non Linear Support Vector Machine

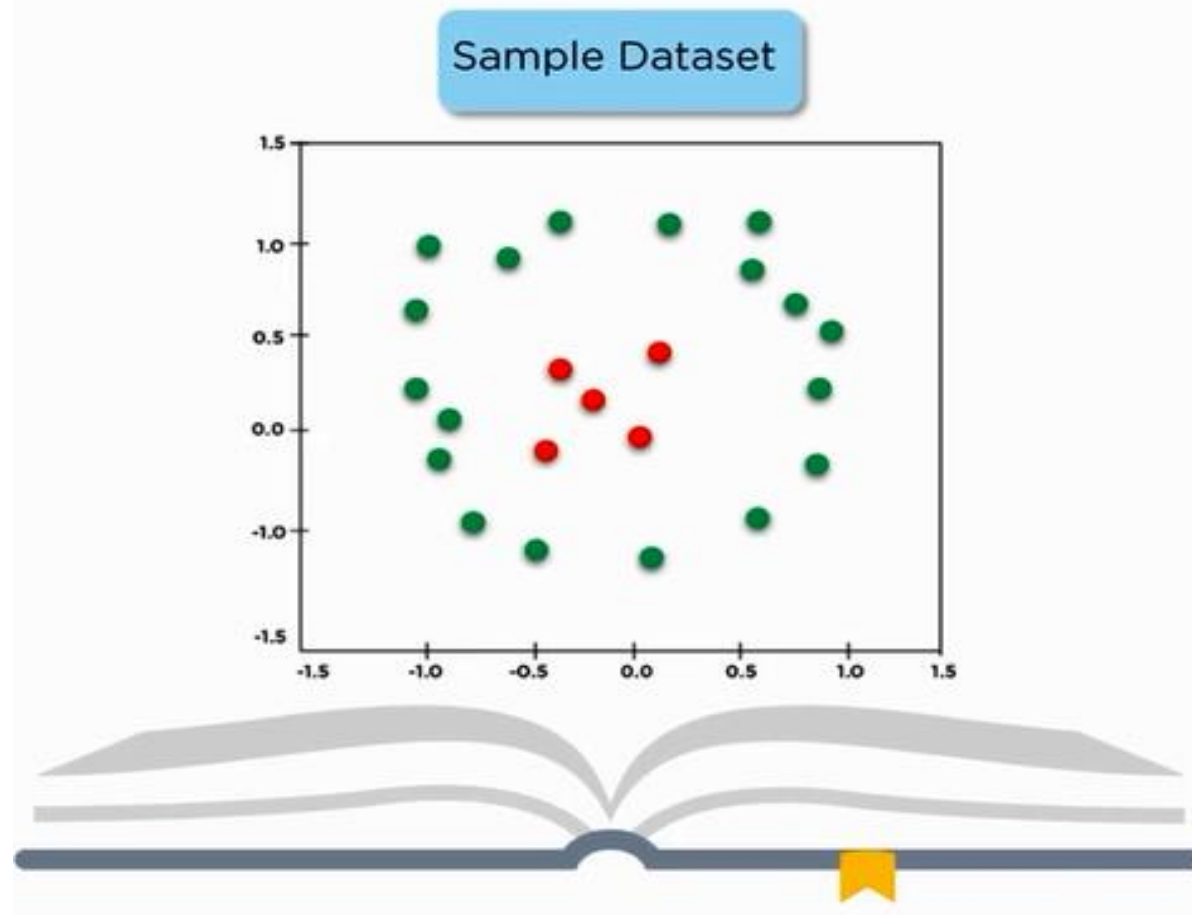
Non-linear SVM is used when the data can't be separated using a straight line



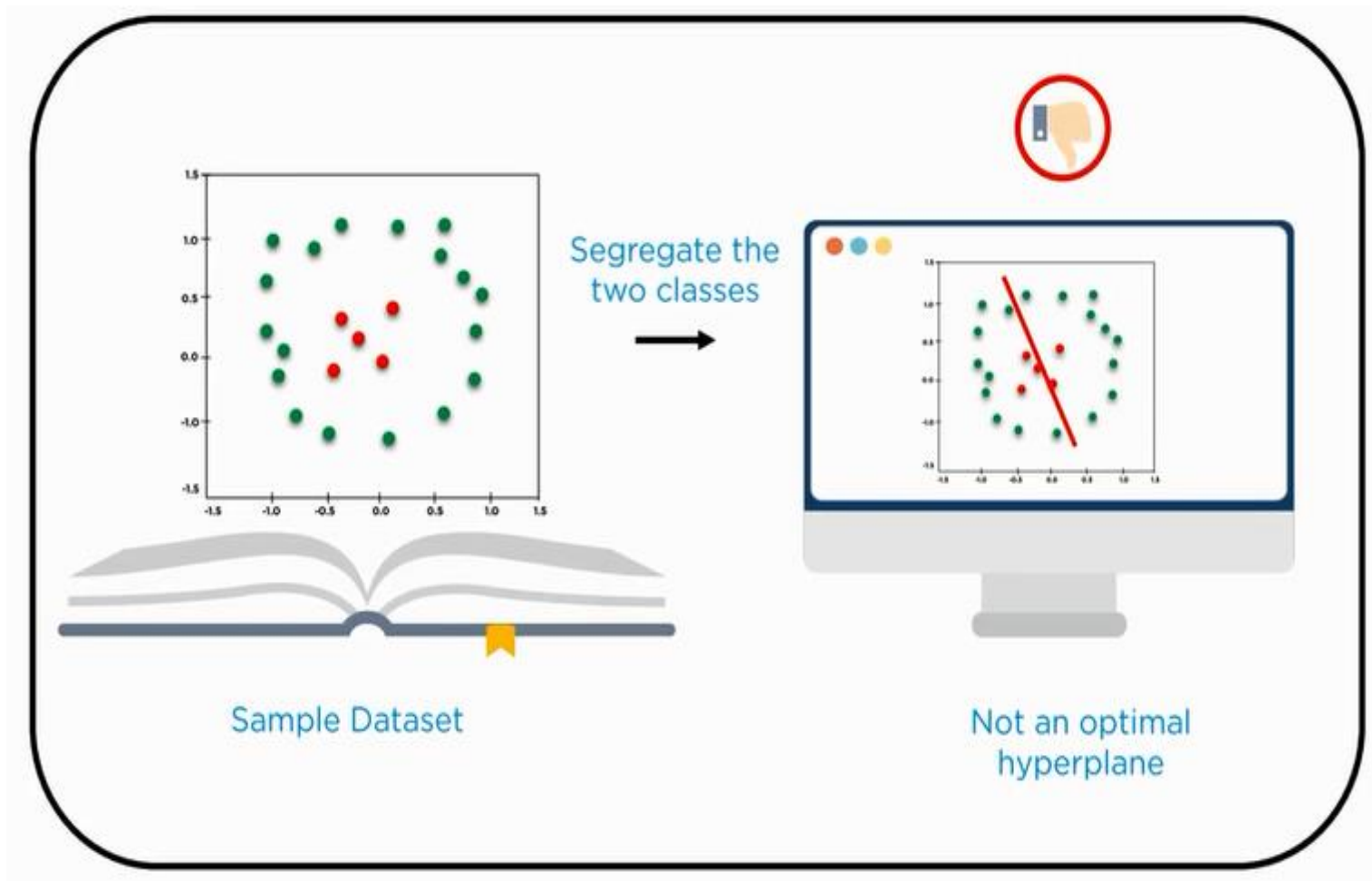
Non Linear Support Vector Machine



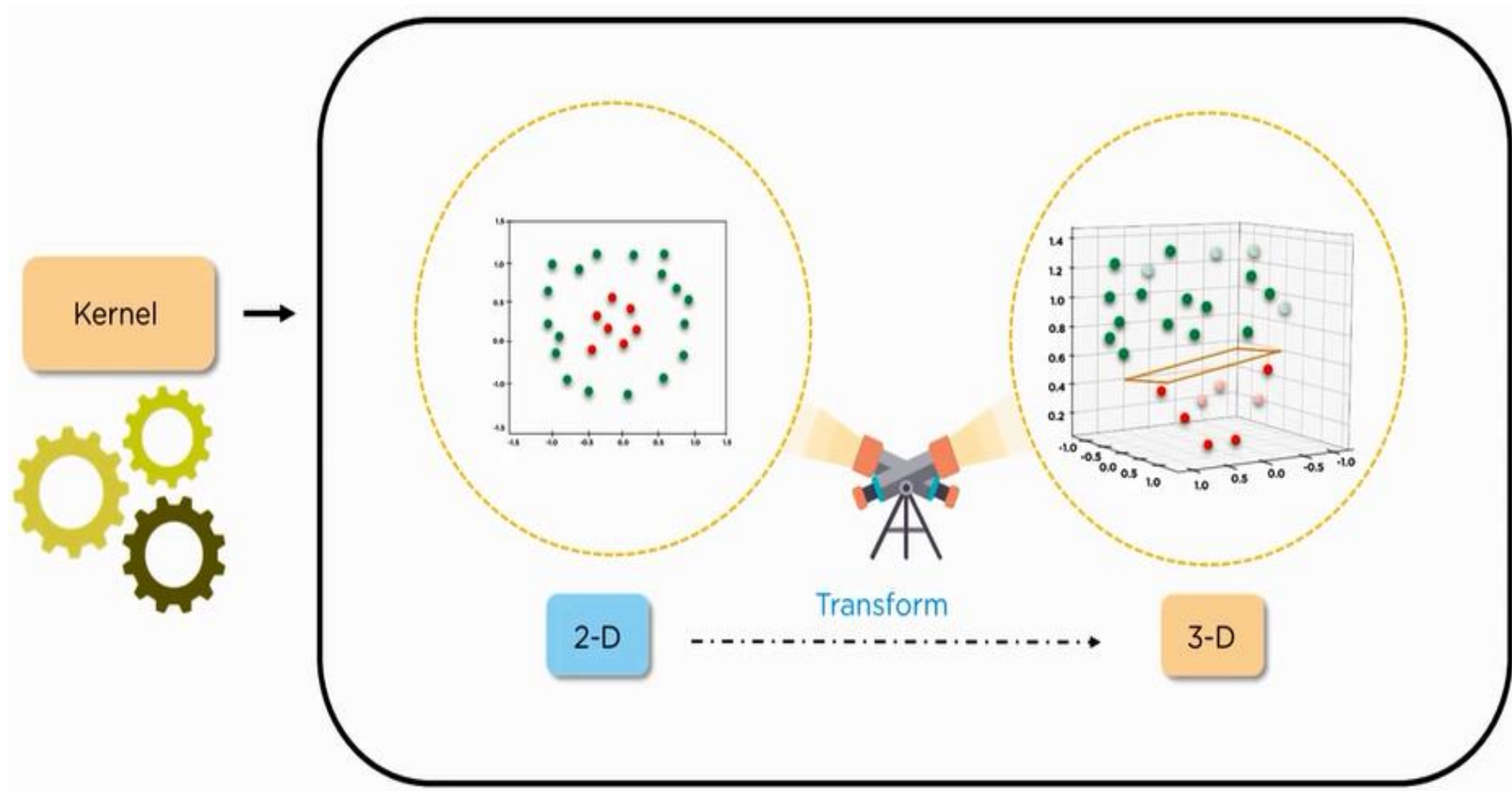
Support Vector Machine-Example



Support Vector Machine-Example



Support Vector Machine-Example



Kernels in Support Vector Machine

The most interesting feature of SVM is that it can even work with a non-linear dataset and for this, we use “Kernel Trick” which makes it easier to classifies the points.

Different Kernel functions-Polynomial kernel

Following is the formula for the polynomial kernel:

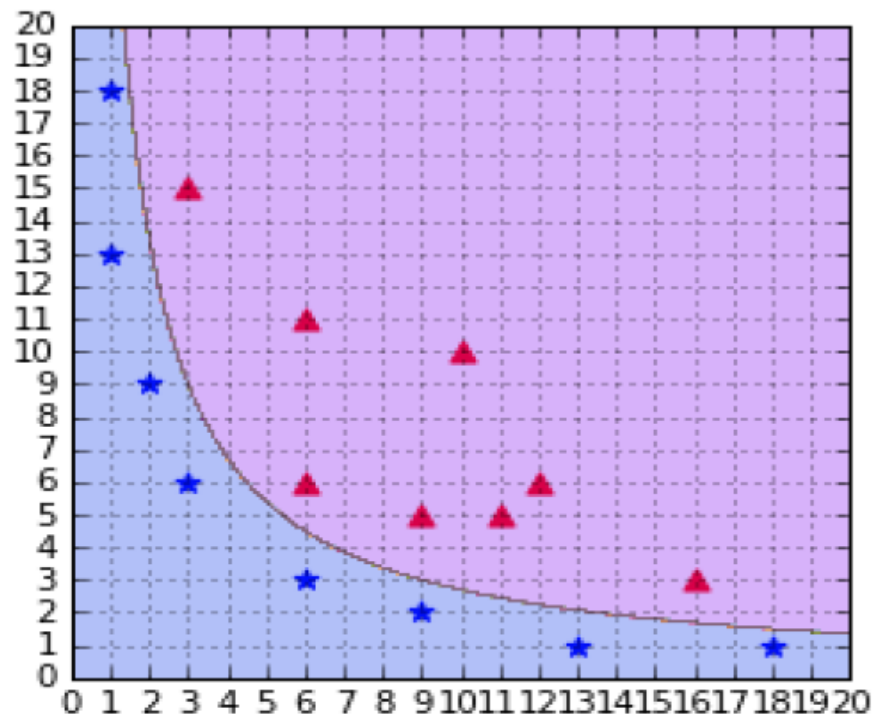
$$f(X1, X2) = (X1^T \cdot X2 + 1)^d$$

Here **d** is the **degree of the polynomial**, which we need to specify manually. Suppose we have two features X1 and X2 and output variable as Y, so using Polynomial kernel we can write it as:

$$\begin{aligned} X1^T \cdot X2 &= \begin{bmatrix} X1 \\ X2 \end{bmatrix} \cdot [X1 \quad X2] \\ &= \begin{bmatrix} X1^2 & X1 \cdot X2 \\ X1 \cdot X2 & X2^2 \end{bmatrix} \end{aligned}$$

So we basically need to find X_1^2 , X_2^2 and $X1 \cdot X2$, and now we can see that 2 dimensions got converted into 5 dimensions.

Different Kernel functions-Polynomial kernel



A SVM using a polynomial kernel is able to separate the data (degree=2)

Kernels in Support Vector Machine-RBF kernel

What it actually does is to create non-linear combinations of our features to lift your samples onto a higher-dimensional feature space.

Where we can use a linear decision boundary to separate your classes.

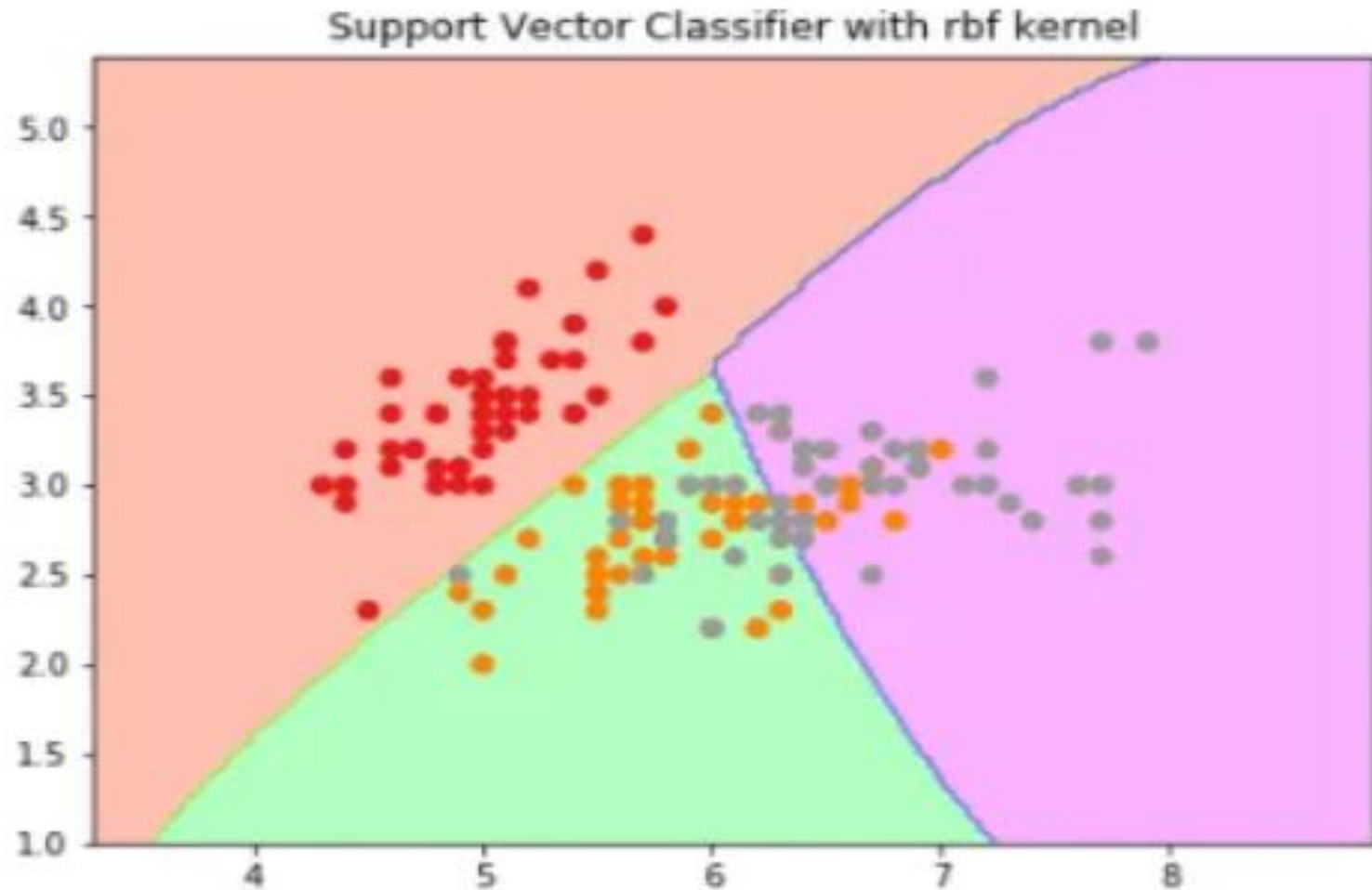
It is the most used kernel in SVM classifications, the following formula explains it mathematically

$$f(x_1, x_2) = e^{\frac{-||x_1 - x_2||^2}{2\sigma^2}}$$

where,

1. ' σ ' is the variance and our hyperparameter
2. $||x_1 - x_2||$ is the Euclidean Distance between two points x_1 and x_2

Kernels in Support Vector Machine-RBF kernel



Kernels in Support Vector Machine-Bessel function

It is mainly used for eliminating the cross term in mathematical functions.
Following is the formula of the Bessel function kernel:

$$k(x, y) = \frac{J_{v+1}(\sigma \|x - y\|)}{\|x - y\|^{-n(v+1)}}$$

Kernels in Support Vector Machine-Anova

It performs well on multidimensional regression problems. The formula for this kernel function is:

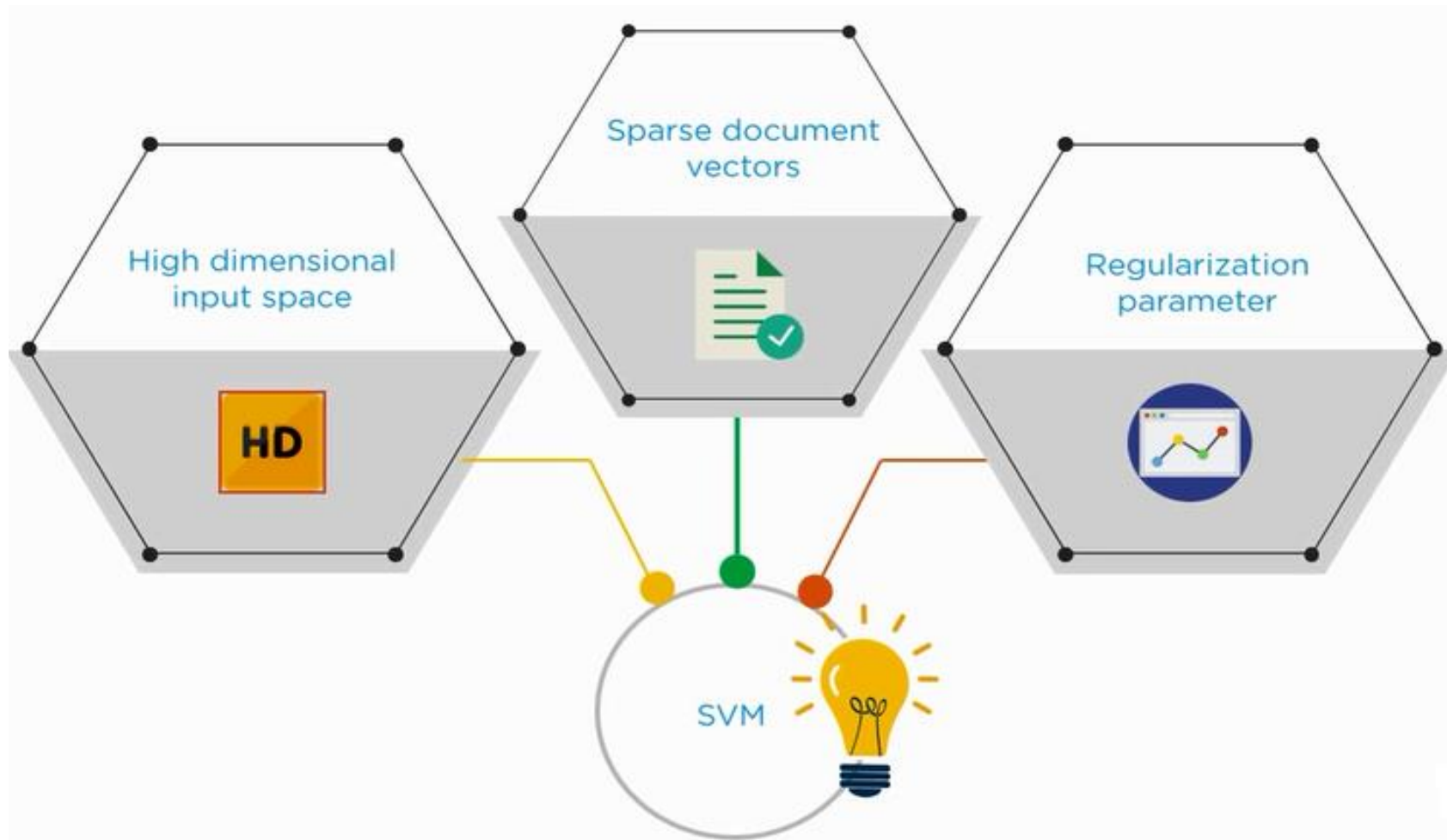
$$k(x, y) = \sum_{k=1}^n \exp(-\sigma(x^k - y^k)^2)^d$$

How to choose the right Kernel?

It is necessary to choose a good kernel function because the performance of the model depends on it.

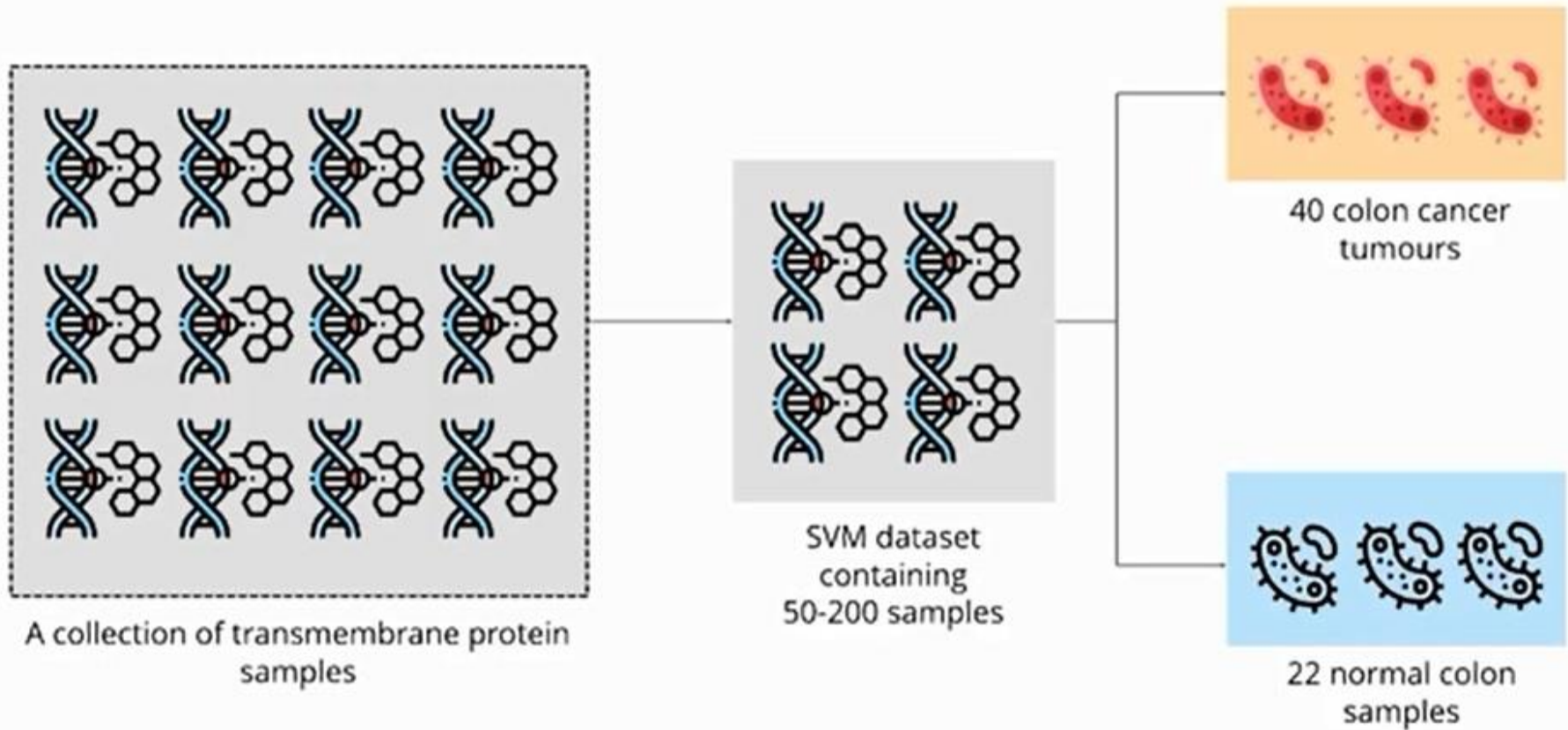
- Choosing a kernel totally depends on what kind of dataset are you working on.
- If it is linearly separable then you must opt.
- for linear kernel function since it is very easy to use and the complexity is much lower compared to other kernel functions.
- I'd recommend you start with a hypothesis that your data is linearly separable and choose a linear kernel function.
- You can then work your way up towards the more complex kernel functions. Usually, we use SVM with RBF and linear kernel function because other kernels like polynomial kernel are rarely used due to poor efficiency.

Support Vector Machine-Example



Support Vector Machine-Use Cases

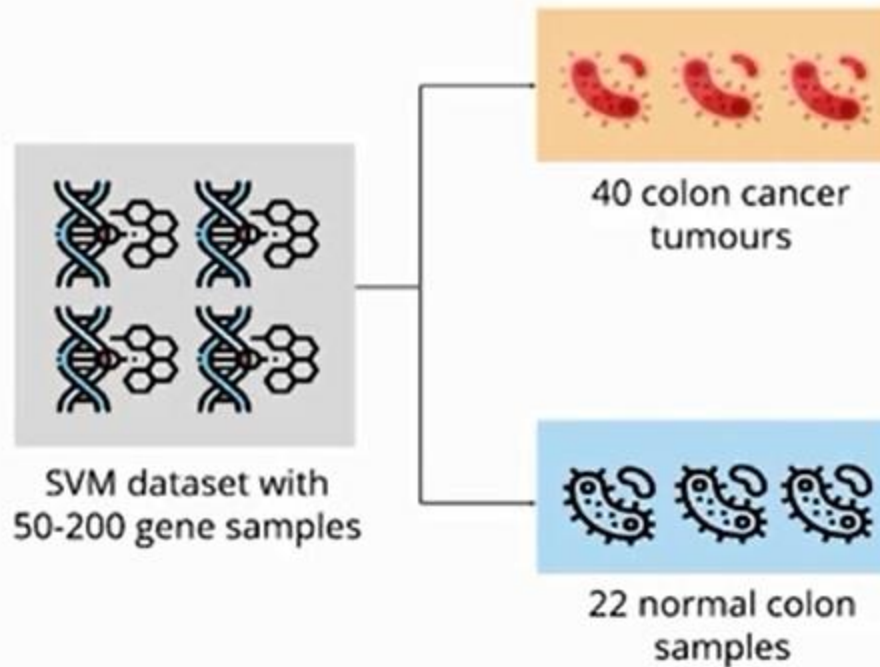
Colon Cancer Classification



Support Vector Machine-Use Cases

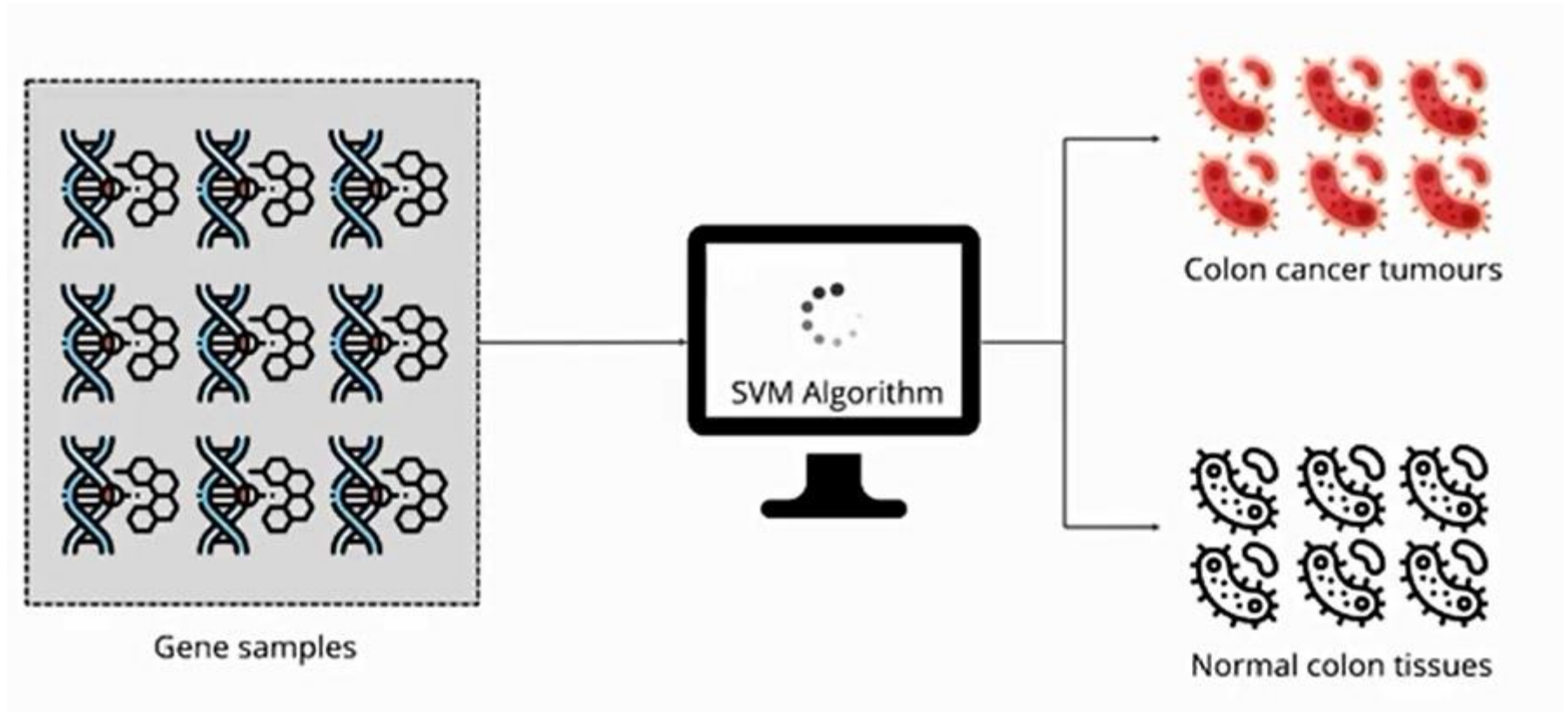
Problem Statement- To Classify gene samples based on whether they are cancerous or not

To classify gene samples based on whether they are cancerous or not



Support Vector Machine-Use Cases

Problem Statement- To Classify gene samples based on whether they are cancerous or not



Support Vector Machine Advantages

1. SVM works better when the data is Linear
2. It is more effective in high dimensions
3. With the help of the kernel trick, we can solve any complex problem
4. SVM is not sensitive to outliers
5. Can help us with Image classification

Support Vector Machine Disadvantages

1. Choosing a good kernel is not easy
2. It doesn't show good results on a big dataset
3. The SVM hyperparameters are Cost -C and gamma.

It is not that easy to fine-tune these hyper-parameters. It is hard to visualize their impact

Support Vector Machine-Applications



Face detection



Text and hypertext
categorization



Classification of
images



Bioinformatics

