Naive Bays & Support Vector Machines

By
Prof(Dr) Premanand Pralhad Gahdekar

Outline

1. Introduction
2. Bayesian Classifier
3. Support Vector Machines

Bayesian Classifiers

❖ There are three **different types of Bayesian Classifiers** .

1. Maximum Likelihood Classifier
2. Minimum Distance Classifier
3. Minimum Risk Classifier

Maximum likelihood classifier is the most popular classifier.

It requires the following information

1. P(i)- Prior Probability of the class
2. P(x/i)-Conditional Probability that class i has x. This can be calculated from the training data table.
3. P(x)-Sum of P(x/i) over the entire dataset. This information is not the probability information, but serves as a normalization factor.

Bayesian Classifiers

- **There are three types of Naive Bayes Model, which are given below:**
- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Bayesian Classifiers

- **What is Bayesian Principle?**
- As per the Bayesian Principle, one can find the inverse probability P(i/x) from P(x/i) and P(i). **The Bayes theorem can be given as**

$$P(i/x) = \frac{P(x/i)\, P(i)}{P(x)}$$

The Prior Probability of the class can be obtained from the training set.

The Prior Probability can be estimated by plotting a histogram of the image.

Maximum likelihood Classifier

- The **Bayesian rule** can be applied as the only term that is unknown is P(i/x).
- The Bayesian optimality rule states that any instance assigned to a wrong class is worse and all types of misclassifications are equally worse.
- Therefore, according to the Bayesian maximum likelihood classifier, the instance is assigned to class i for which P(i/x) is maximum.

Advantages

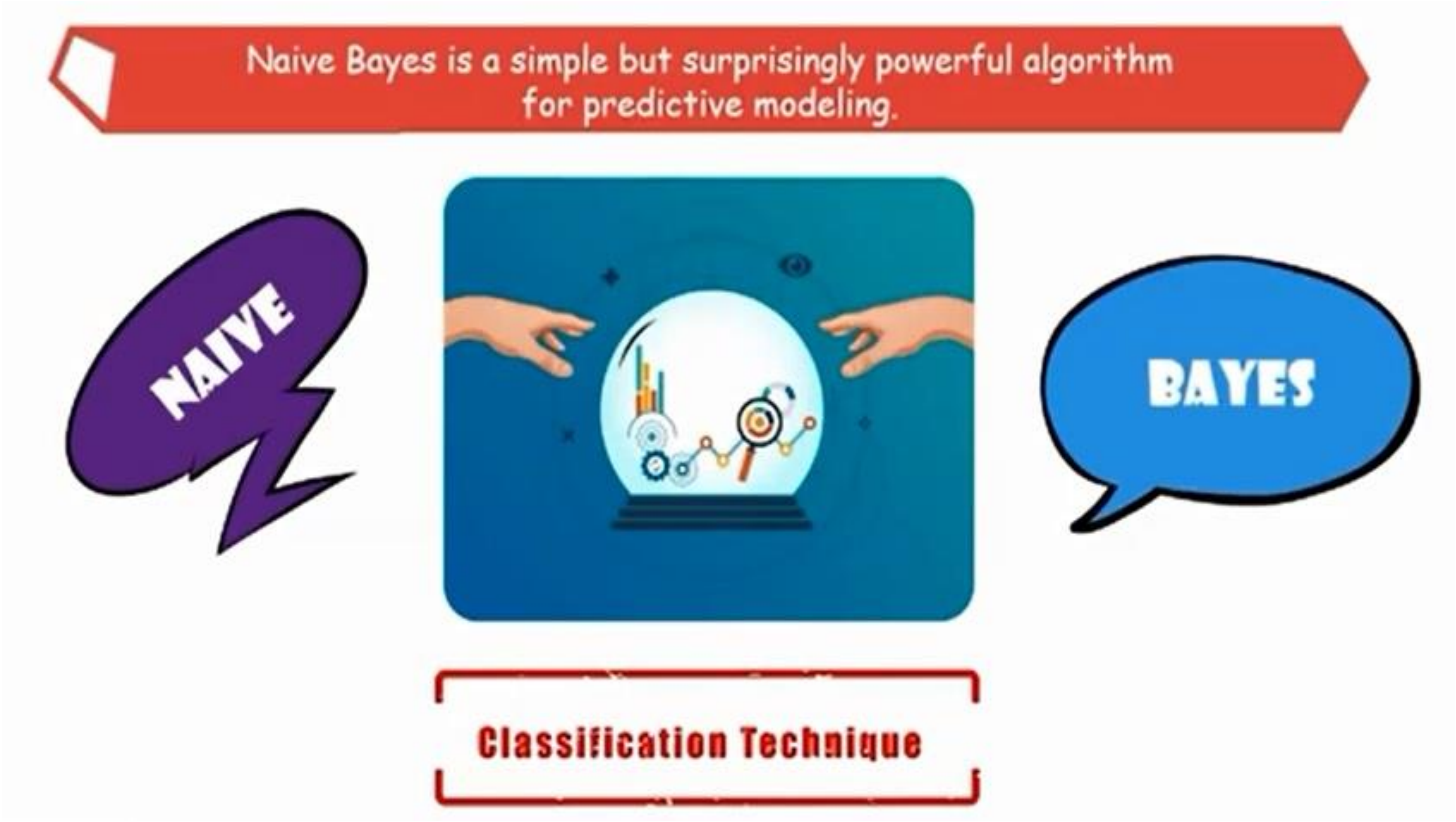
Easy to use, It requires only one scan of the training set, & not affected much by missing values.

It produces good results for datasets with simple relationships.

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.
NAIVE
BAYES
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

probability a hypothesis is true given the evidence

probability a hypothesis is true (before any evidence is present)

probability of seeing the evidence if the hypothesis is true

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

probability of observing the evidence

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.

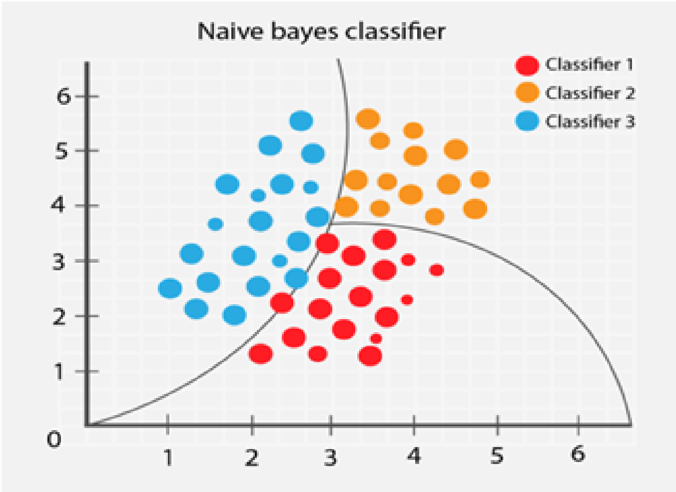
Naive Bayes

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

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



Bayes Theorem Proof

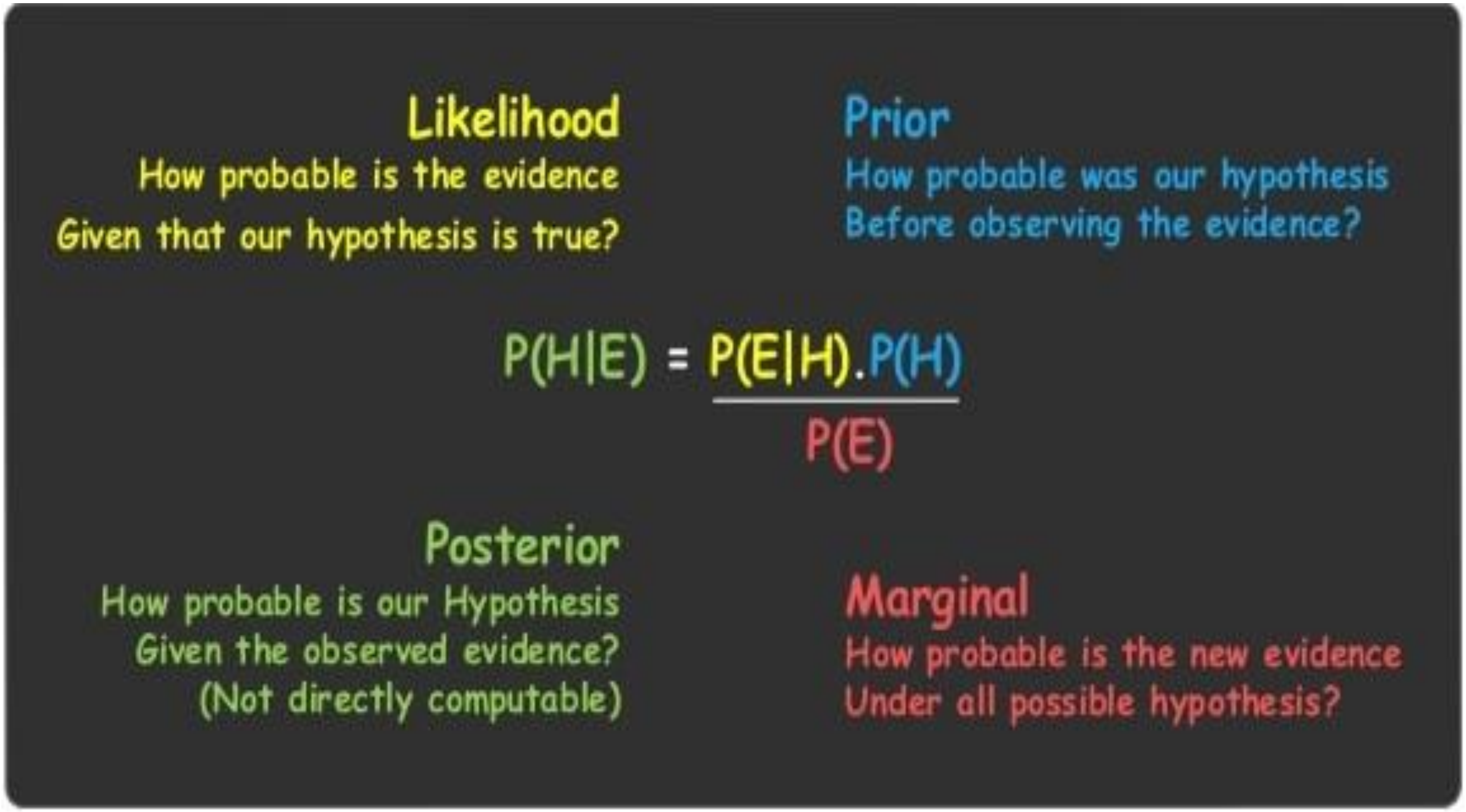
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

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

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

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

Bayes Theorem Proof



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

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) x P(a2=2/c1) =1/2 x 1/2 =1/4

Bayesian Classifier-Numerical

Soln-

$P(x/c2) = P(a1=2/c2) \times P(a2=2/c2)$

$= 1/3 \times 2/3 = 2/9$

$p(x)=1$

This is used to evaluate

$P(c1/x) = P(c1) \times P(x/c1)/p(x)$

$= 2/5 \times 1/4 = 2/20 = 0.1$

$P(c2/x) = P(c2) \times P(x/c2)/p(x)$

$= 3/5 \times 2/9 = 6/45 = 0.13$

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

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

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

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

$$= 0.5$$

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

$P(Fruit|Orange) = 0.53 \times 0.69 \times 0 = 0$

$P(Fruit|Banana) = 1 \times 0.75 \times 0.87$

$= 0.65$

$P(Fruit|Others) = 0.33 \times 0.66 \times 0.33$

$= 0.072$

Bayesian Classifier-Numerical

Example 13.6 Use Naïve Bayes classifier and classify the unknown pixel X. There are two classes of pixels # and * present in the image as shown:

$$\begin{pmatrix} \# & \# & \# & * \\ \# & X & \# & * \\ \# & \# & * & * \\ \# & \# & * & * \end{pmatrix}$$

Consider the 8-neighbourhood of X and determine the class of X.

Solution The prior probabilities are $P(\text{Pixel} = \text{'\#'}) = \dfrac{\text{Number of \# pixels}}{\text{Total number of pixels}} = \dfrac{9}{16} \cong 0.56$

$$P(\text{Pixel} = \text{'*'}) = \frac{\text{Number of * pixels}}{\text{Total number of pixels}} = \frac{6}{16} \cong 0.38$$

Given the 8-neighbourhood of X, it is possible to calculate the likelihood of X.

Likelihood of X given '#' in the 8-neighbourhood

$$= \frac{\text{Number of \# pixels in neighbourhood of } X}{\text{Total number of '\#' pixels}} = \frac{7}{9} \cong 0.78$$

Likelihood of X given '*' in the 8-neighbourhood

$$= \frac{\text{Number of * pixels in neighbourhood of } X}{\text{Total number of '*' pixels}} = \frac{1}{6} \cong 0.17$$

Now posterior probability can be calculated as

[Prior probability $P(\text{pixel} = \text{'\#'})$] × [Likelihood of X given '#' in the 8-neighbourhood]

$$= 0.56 \times 0.78 = 0.4368$$

[Prior probability $P(\text{pixel} = \text{'*'})$] × [Likelihood of X given '*' in the 8-neighbourhood]

$$= 0.38 \times 0.17 = 0.0646$$

Since 0.4368 is greater than 0.0646, the pixel X must be #.

Bayesian Classifier-Numerical

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

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

		Play Game	
		Yes	No
Humidity	High	4	3
	Normal	6	1

		Play Game	
		Yes	No
Wind	Strong	3	3
	Week	7	1

Bayesian Classifier-Numerical

$P(x|c) = P(Sunny|Yes) = 3/10 = 0.3$

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	

$P(x) = P(Sunny) = 5/14 = 0.36$

$P(c) = P(Yes) = 10/14 = 0.71$

Likelihood of 'Yes' given Sunny is

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

Similarly Likelihood of 'No' given Sunny is

$$P(c/x) = P(No/Sunny) = P(Sunny|No)* P(No) / P(Sunny) = (0.4 \times 0.36) /0.36 = 0.40$$

Bayesian Classifier-Numerical

Suppose we have a day with the following values

Outlook	=	Rain
Humidity	=	High
Wind	=	Weak
Play	=	?

Likelihood of 'Yes' on that Day = P(Outlook = Rain|Yes)*P(Humidity= High|Yes)* P(Wind= Weak|Yes)*P(Yes)

= 2/9 * 3/9 * 6/9 * 9/14 = 0.0199

Likelihood of 'No' on that Day = P(Outlook = Rain|No)*P(Humidity= High|No)* P(Wind= Weak|No)*P(No)

= 2/5 * 4/5 * 2/5 * 5/14 = 0.0166

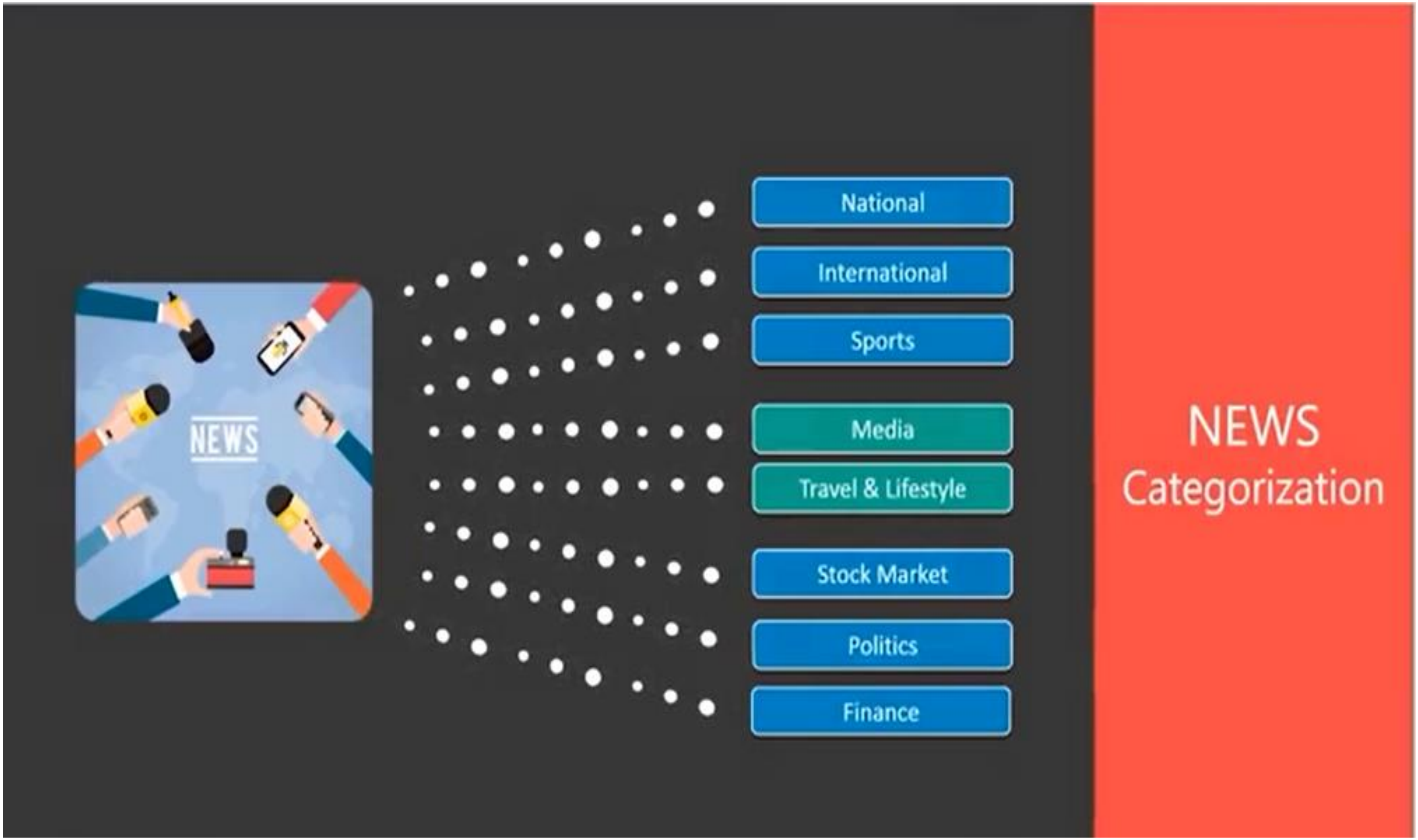
Bayesian Classifier-Numerical

$P(Yes) = 0.0199 / (0.0199 + 0.0166) = 0.55$

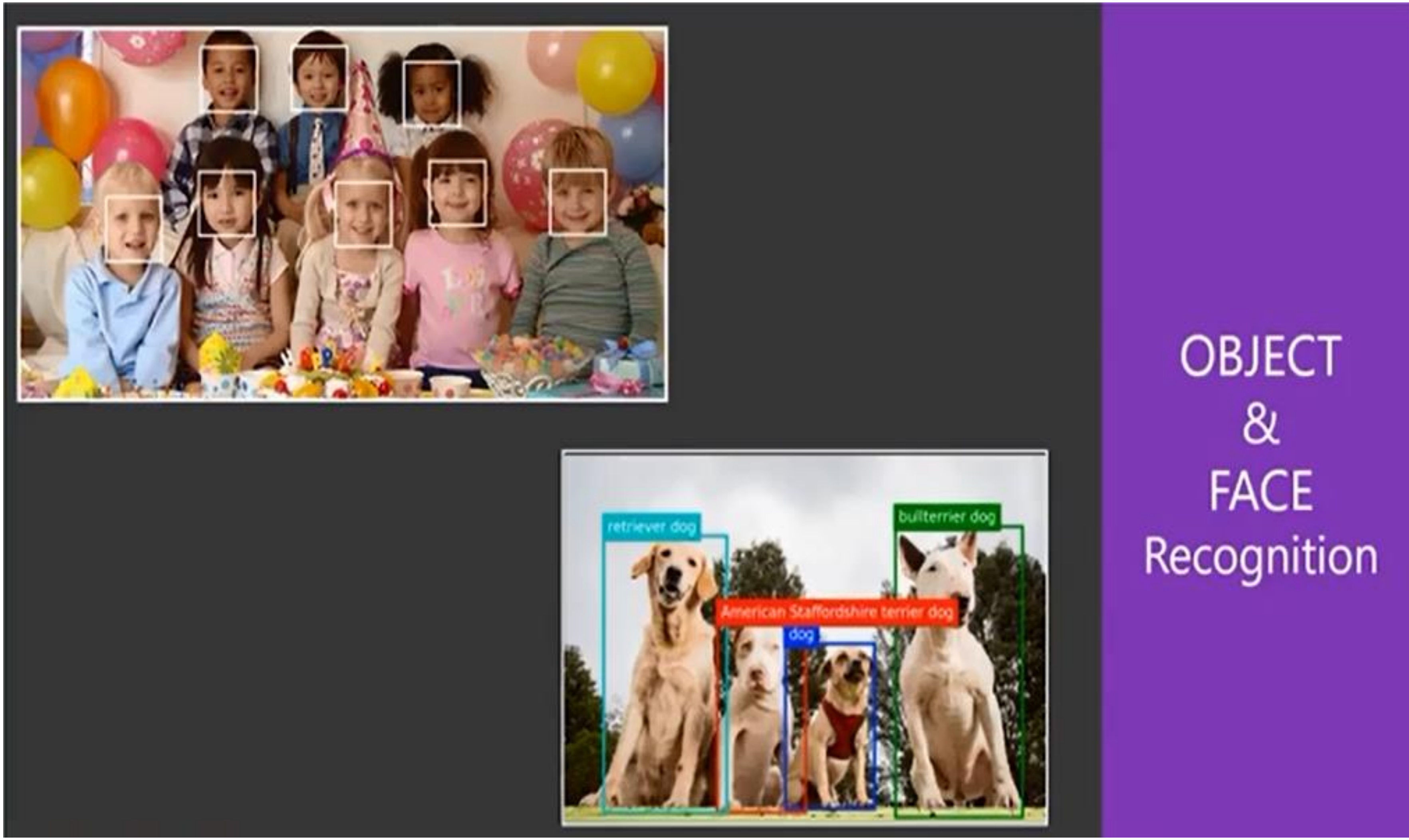
$P(No) = 0.0166 / (0.0199 + 0.0166) = 0.45$

Our model predicts that there is a 55% chance there will be game tomorrow

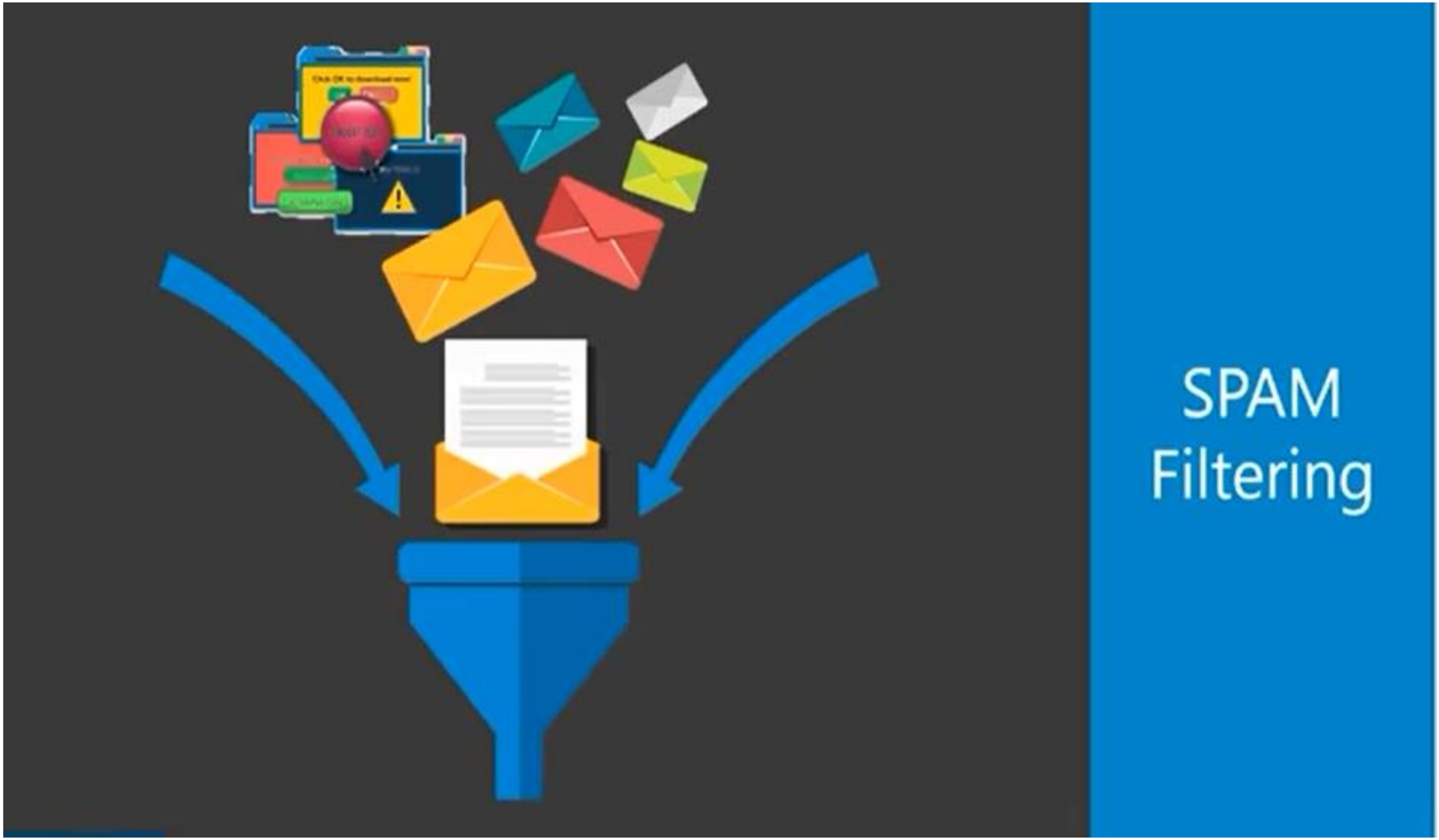
Bayesian Classifier-Applications



Bayesian Classifier-Applications



Bayesian Classifier-Applications



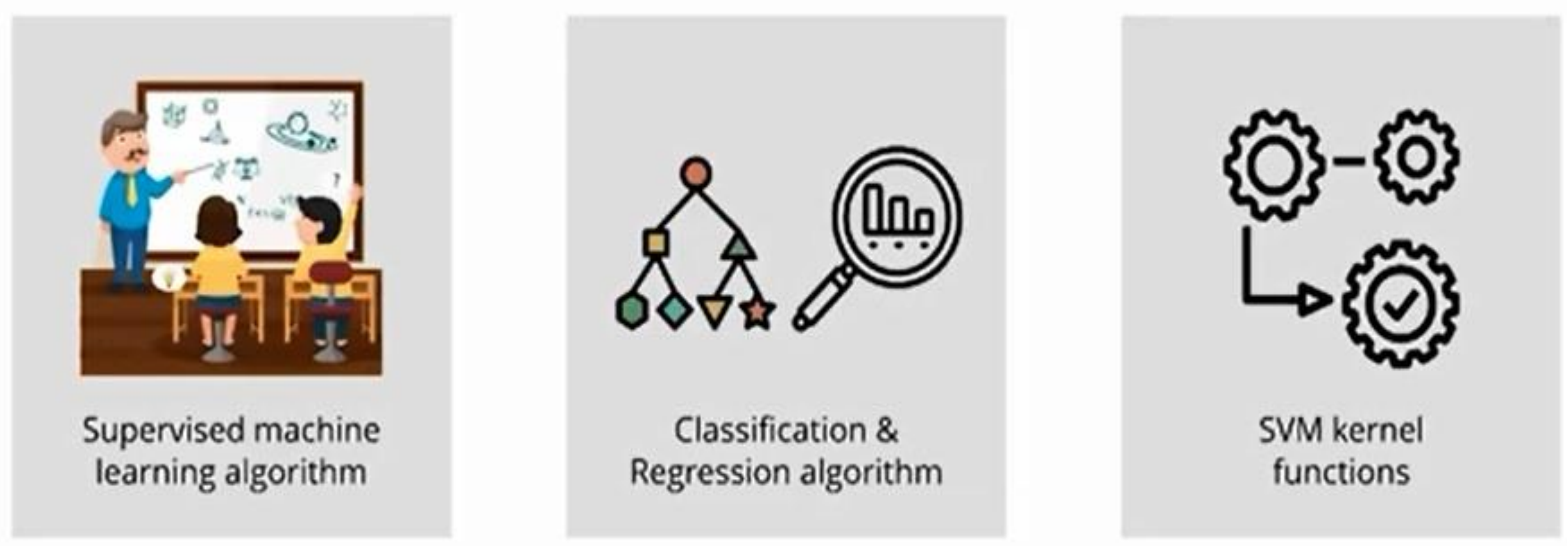
Bayesian Classifier-Applications

Bayesian Classifier-Applications

Support Vector Machine

What is SVM

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



Support Vector Machine

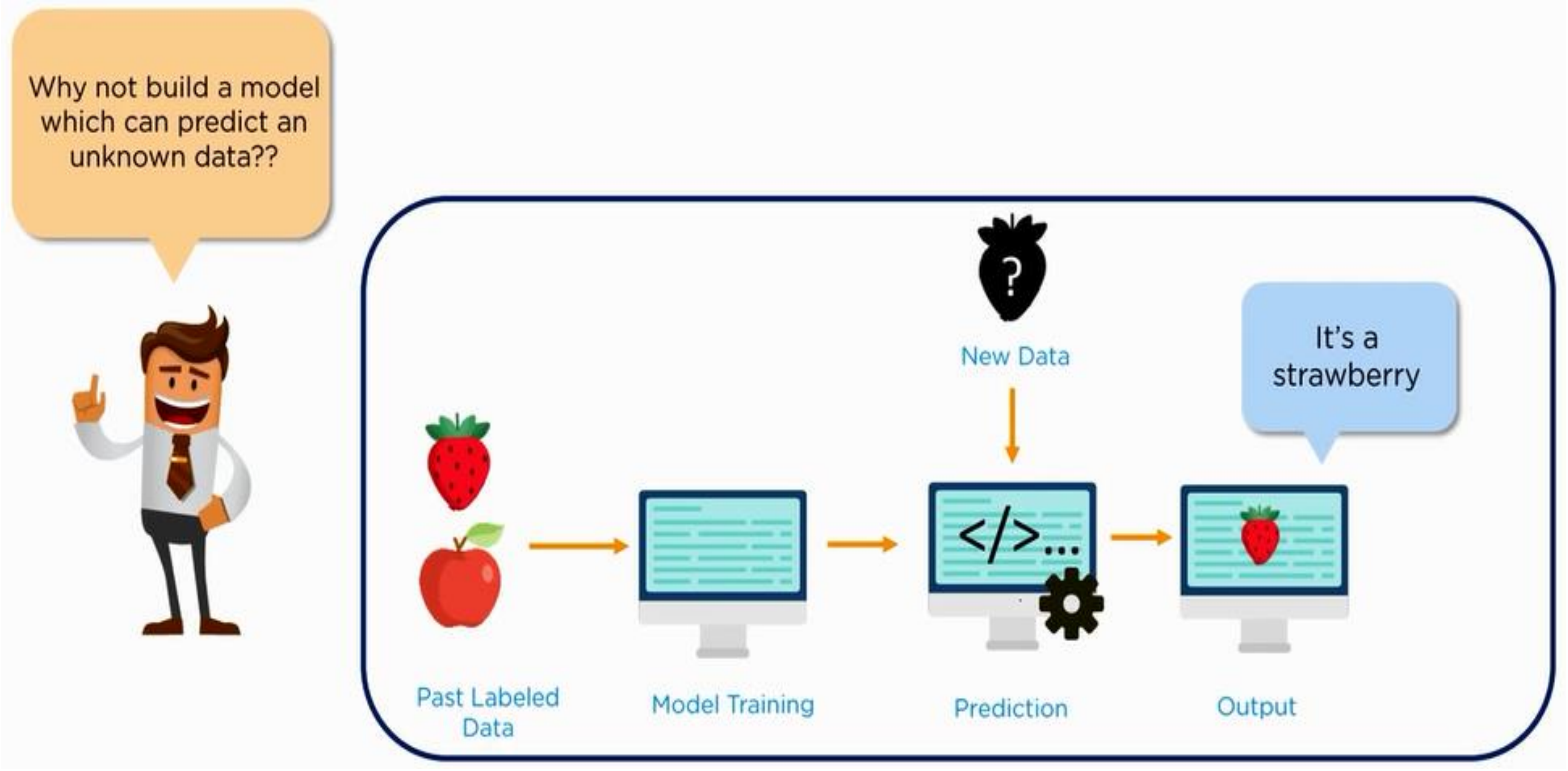
- Support Vectors
- Hyperplanes
- Marginal Distance
- Linear Separable
- Non Linear Separable

Why Support Vector Machine

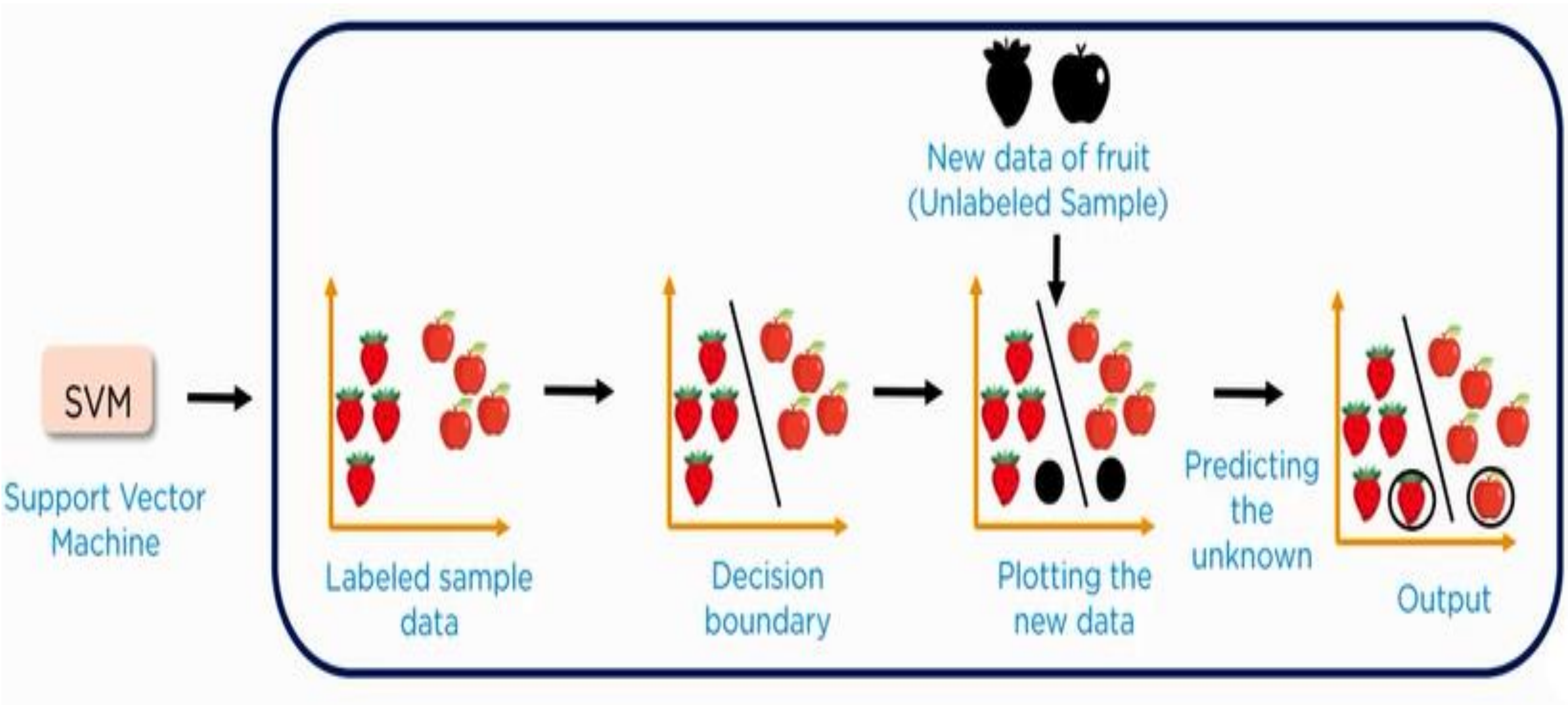
Why Support Vector Machine

Why Support Vector Machine

Why Support Vector Machine

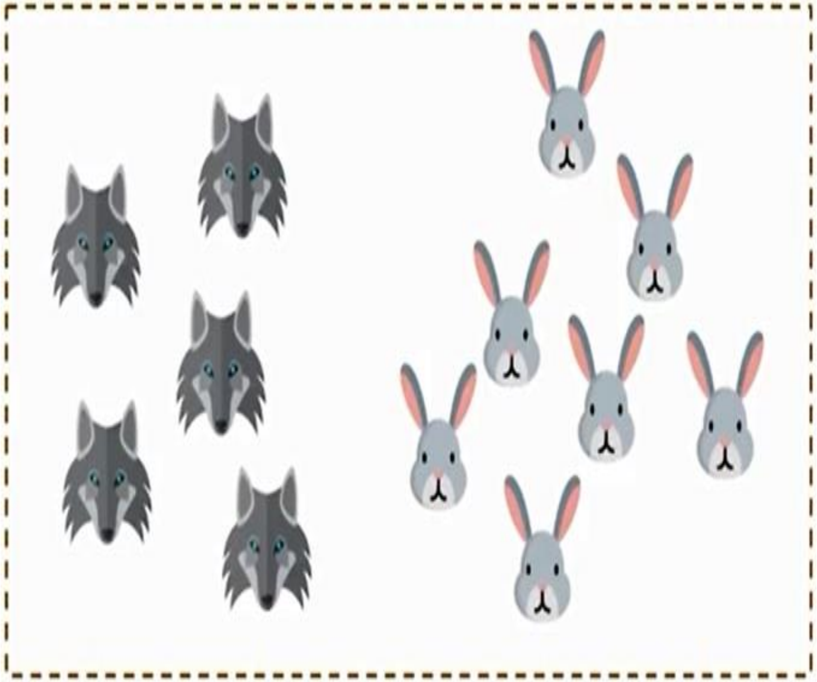


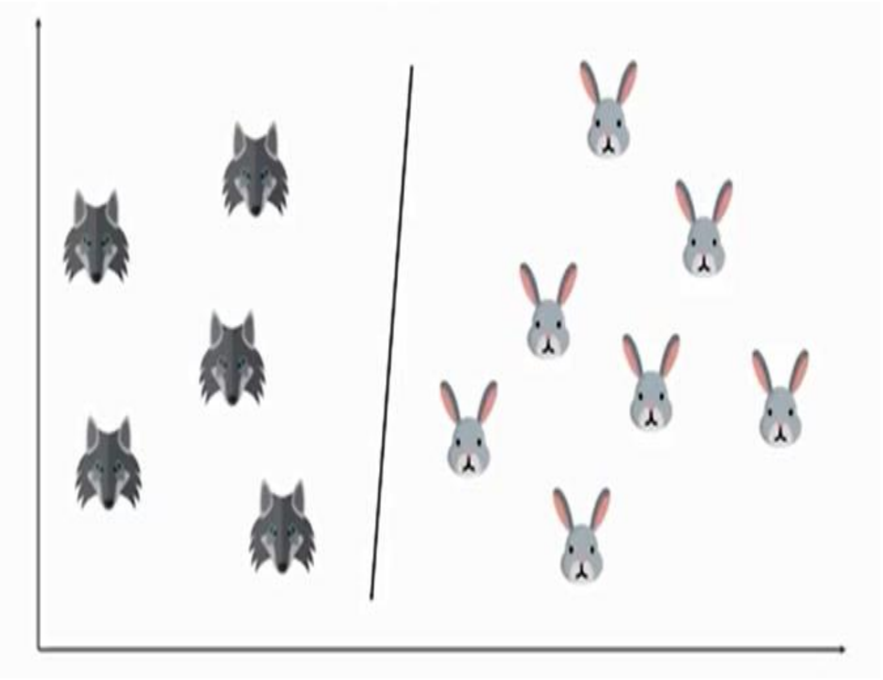
Why Support Vector Machine



How does Support Vector Machine Work

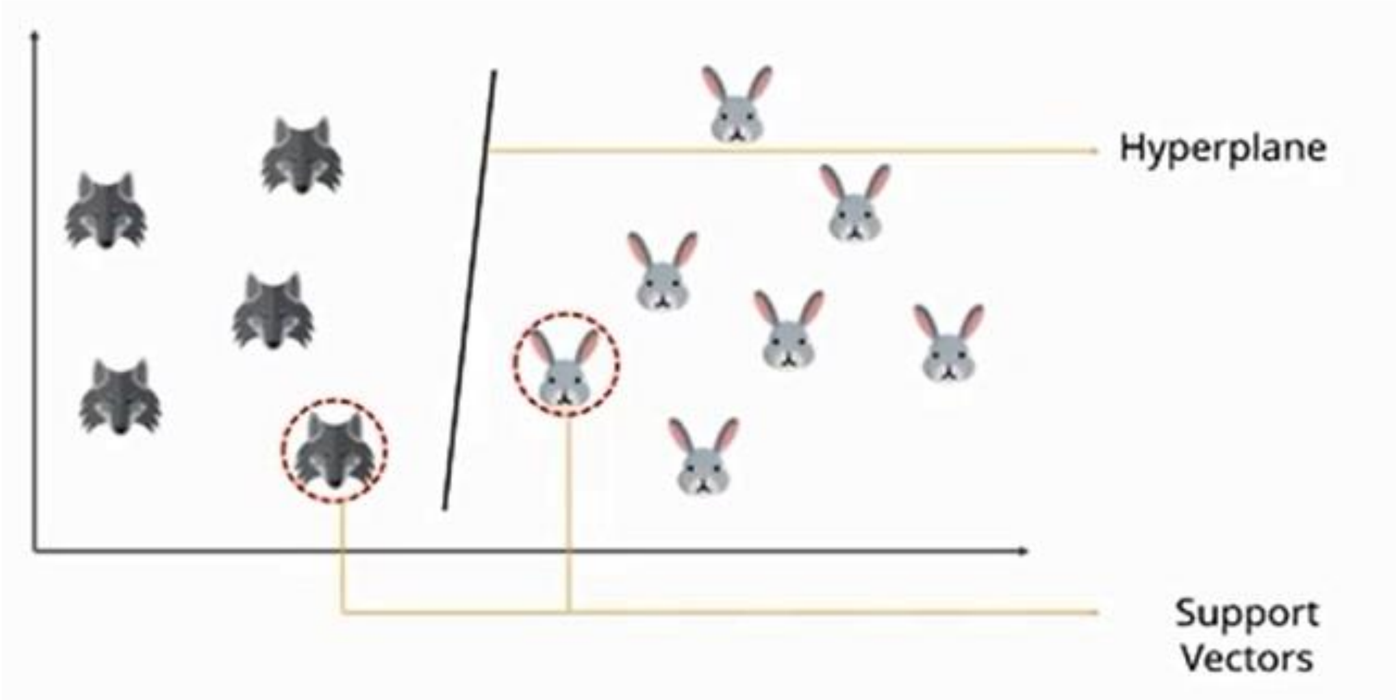
Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.





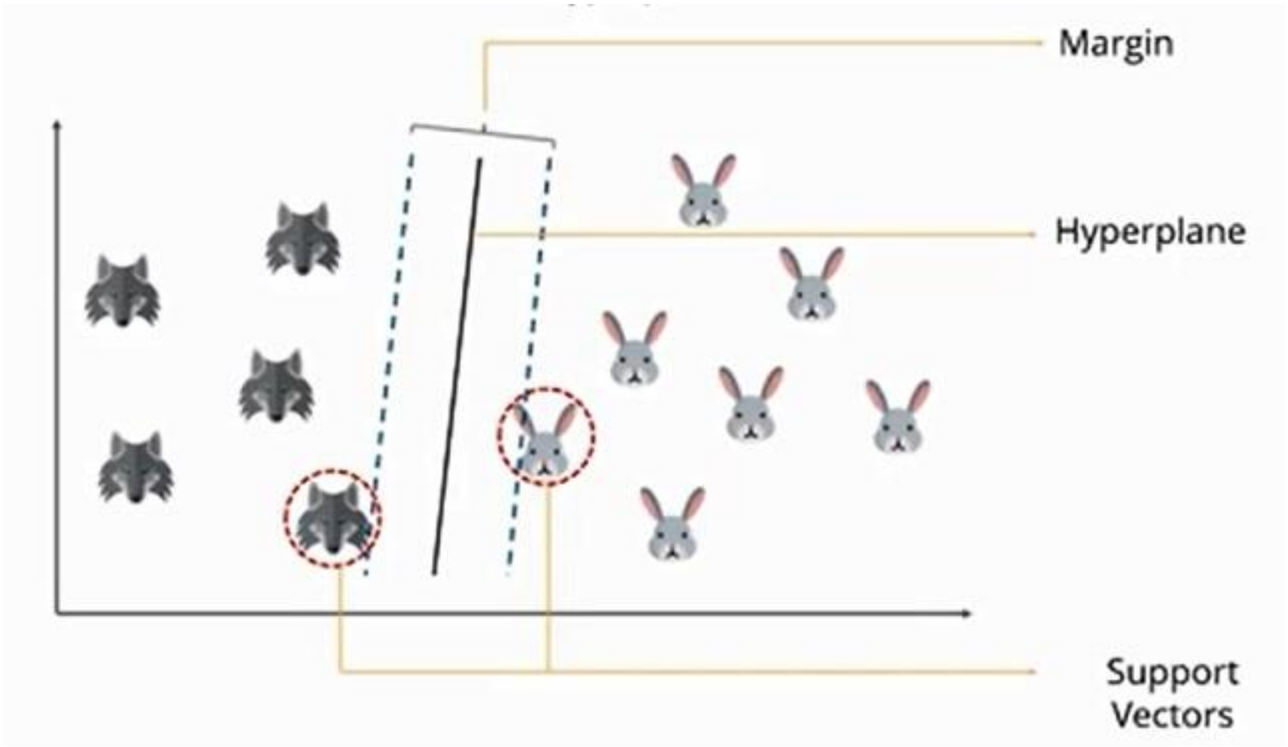
How does Support Vector Machine Work

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



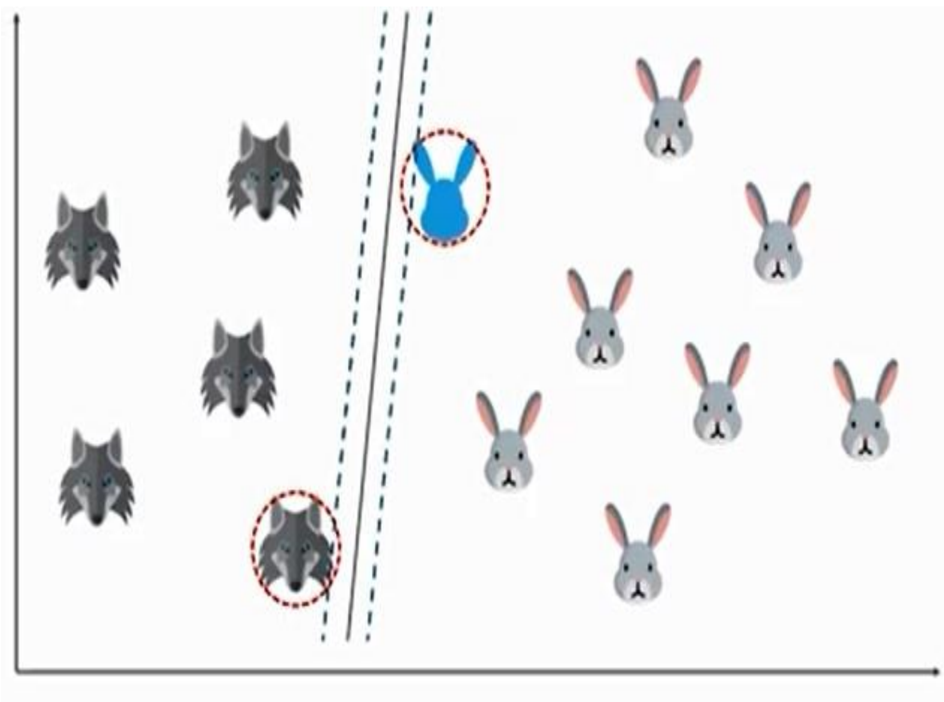
How does Support Vector Machine Work

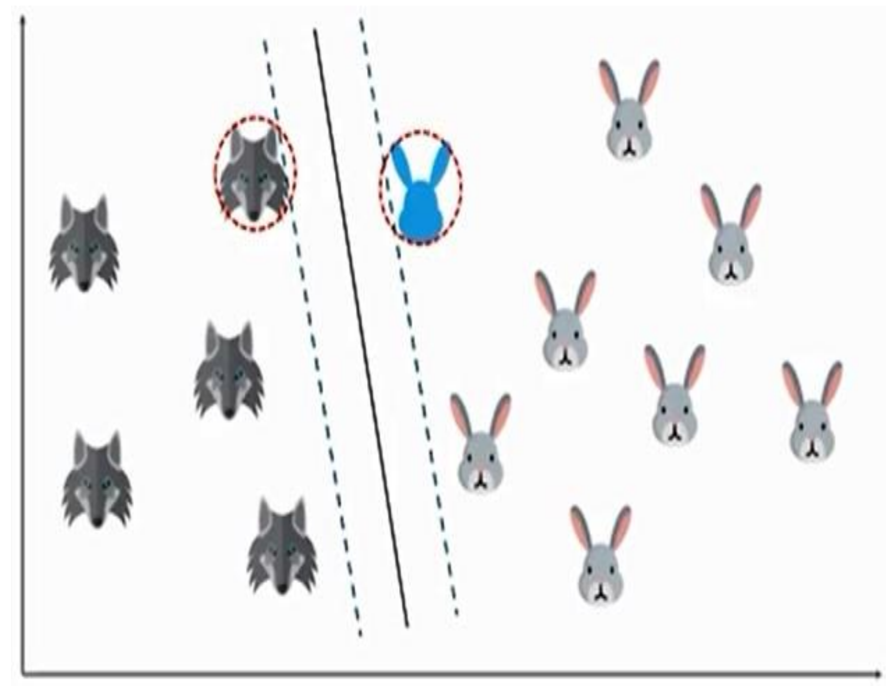
Support Vector Machine (SVM) is a Supervised Classification Method that separates data using Hyperplanes.



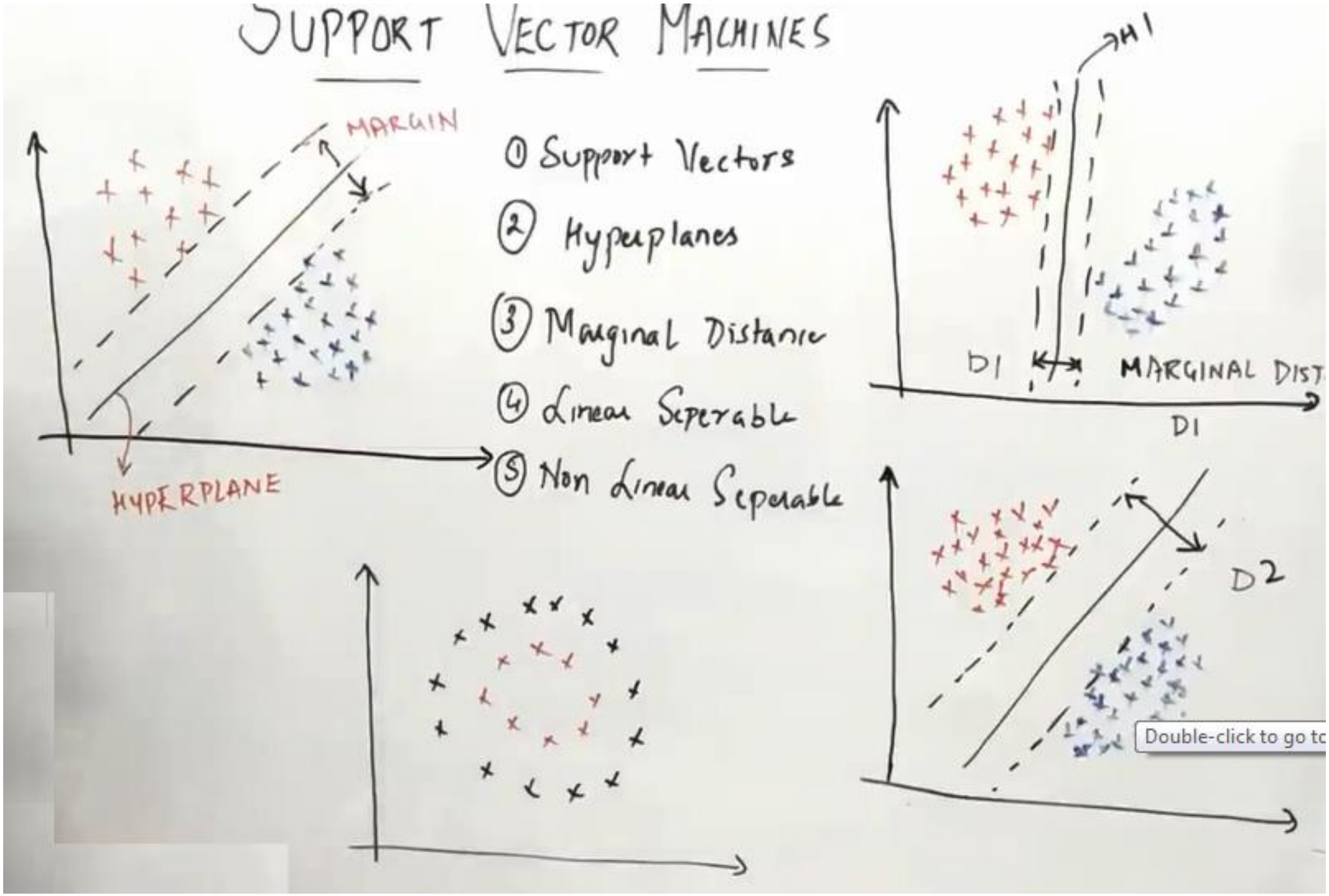
How does Support Vector Machine Work

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





Support Vector Machine-Use Cases



Support Vector Machine-Example

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

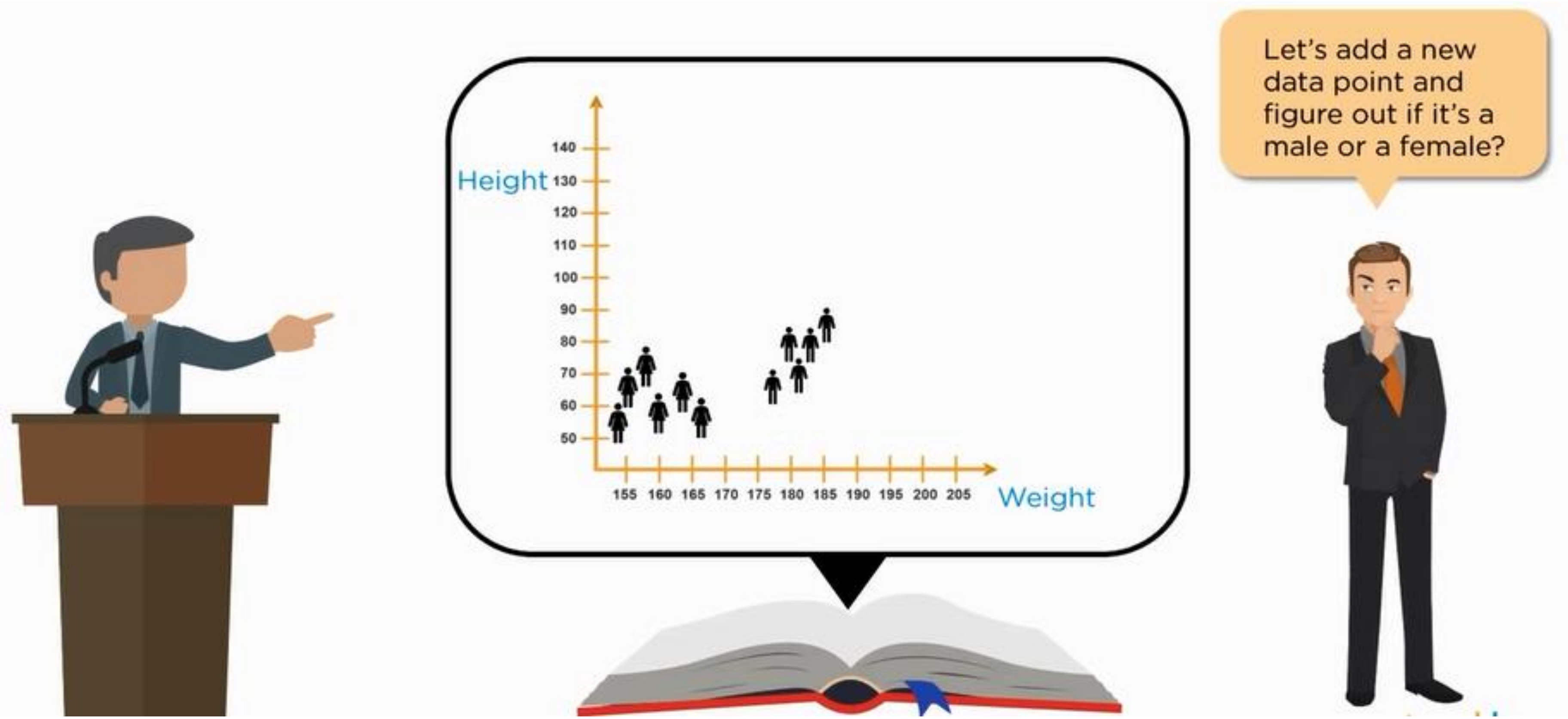
Female

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

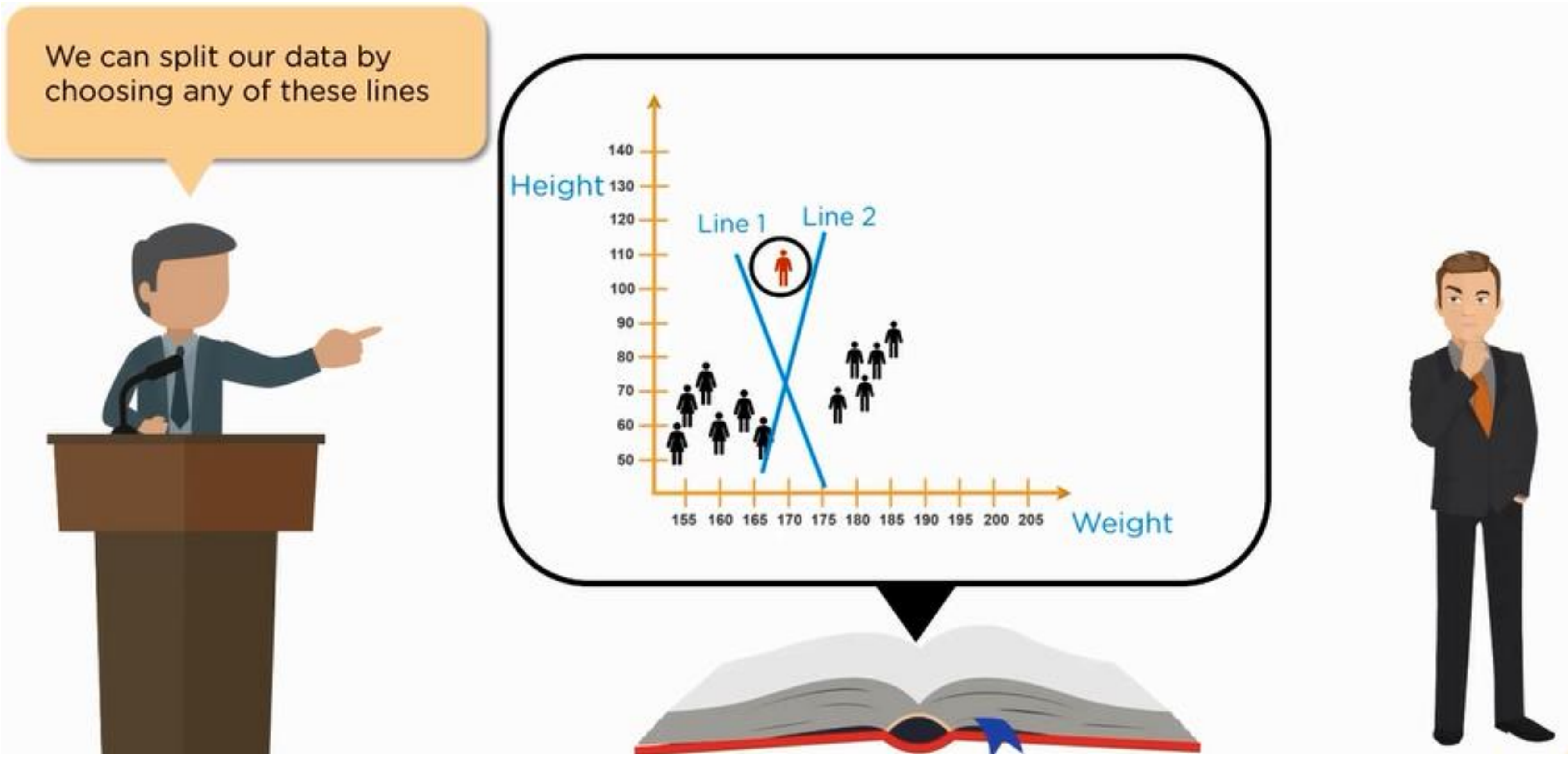
male

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

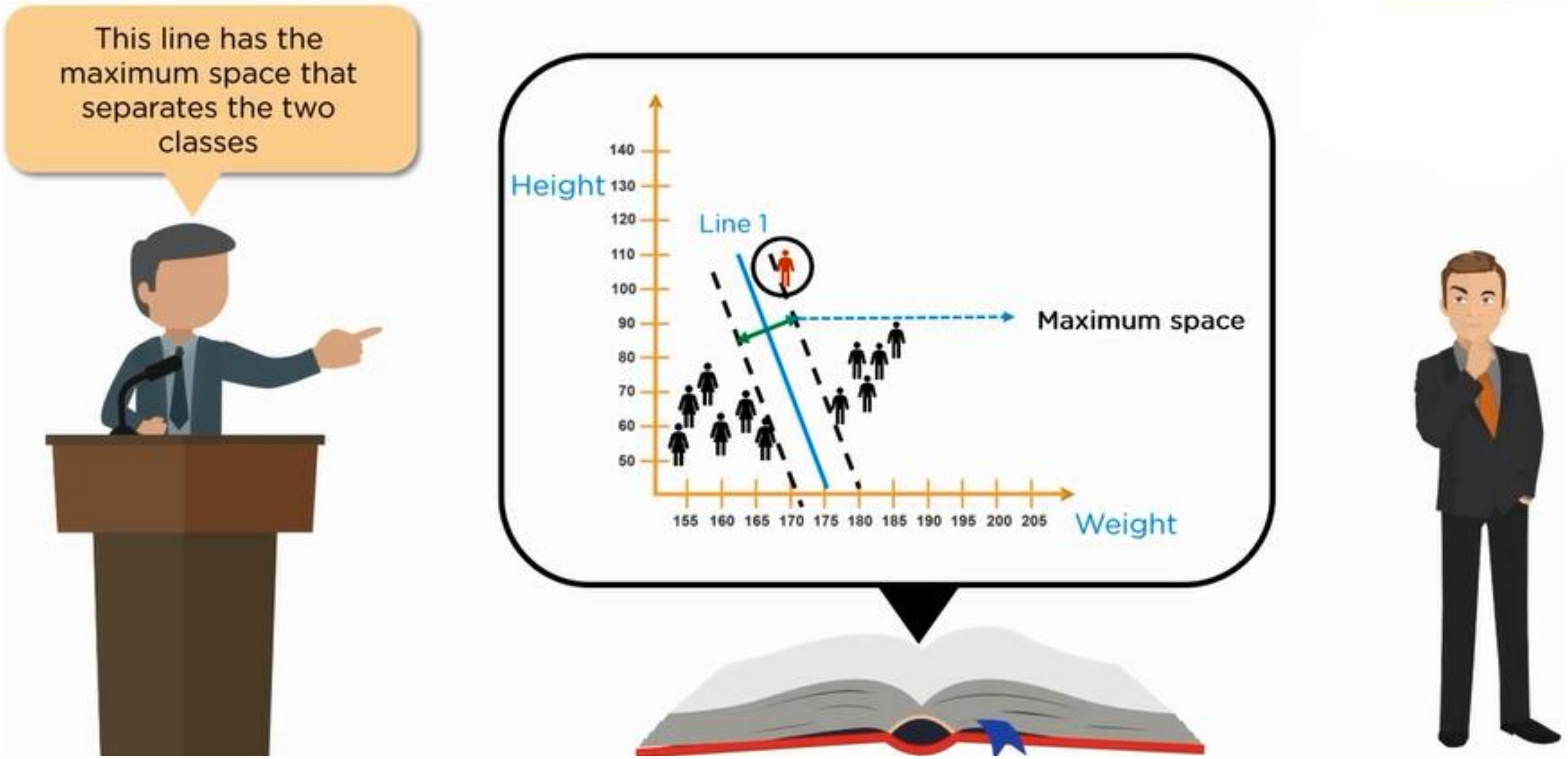
Support Vector Machine-Example



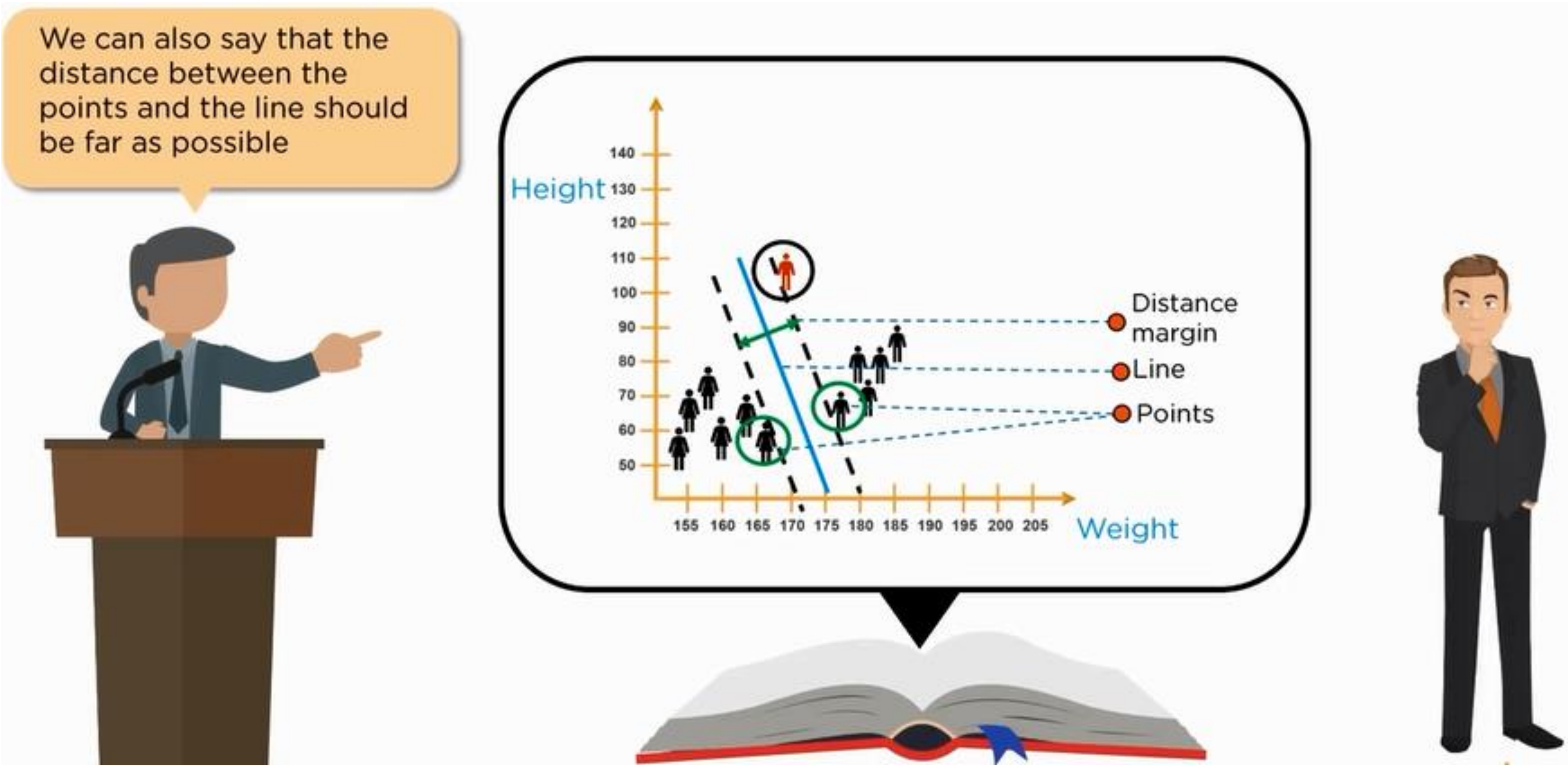
Support Vector Machine-Example



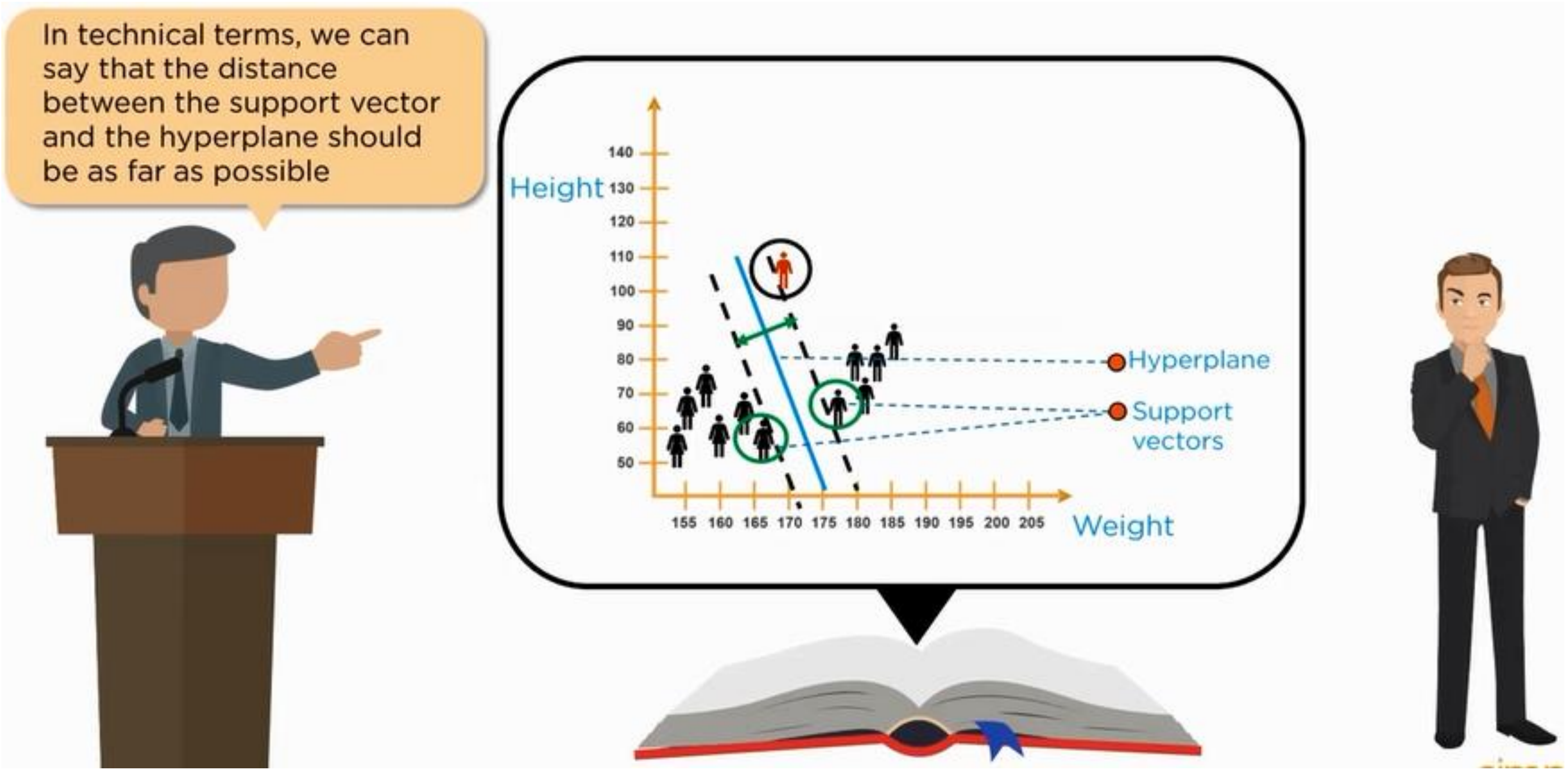
Support Vector Machine-Example



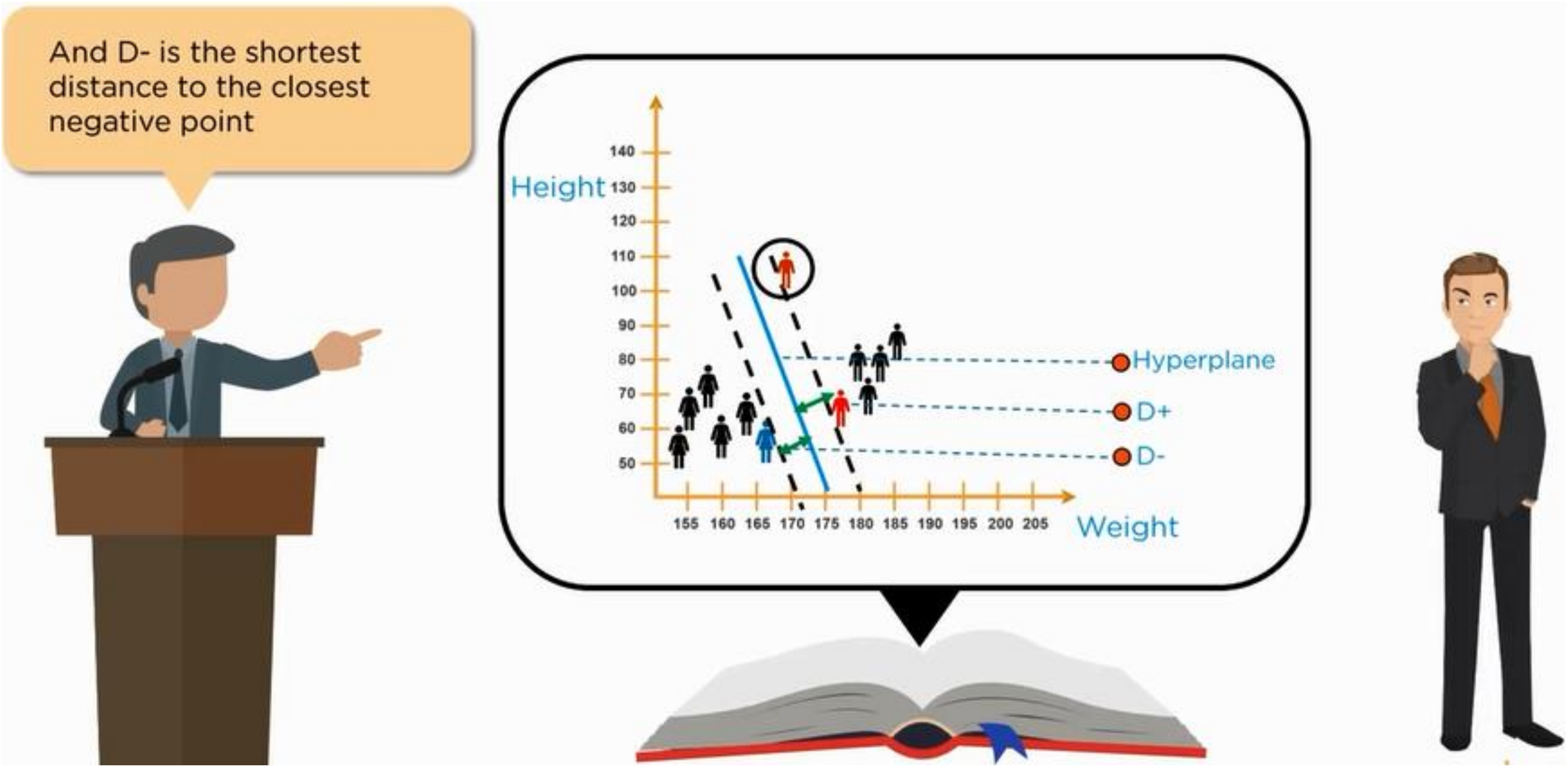
Support Vector Machine-Example



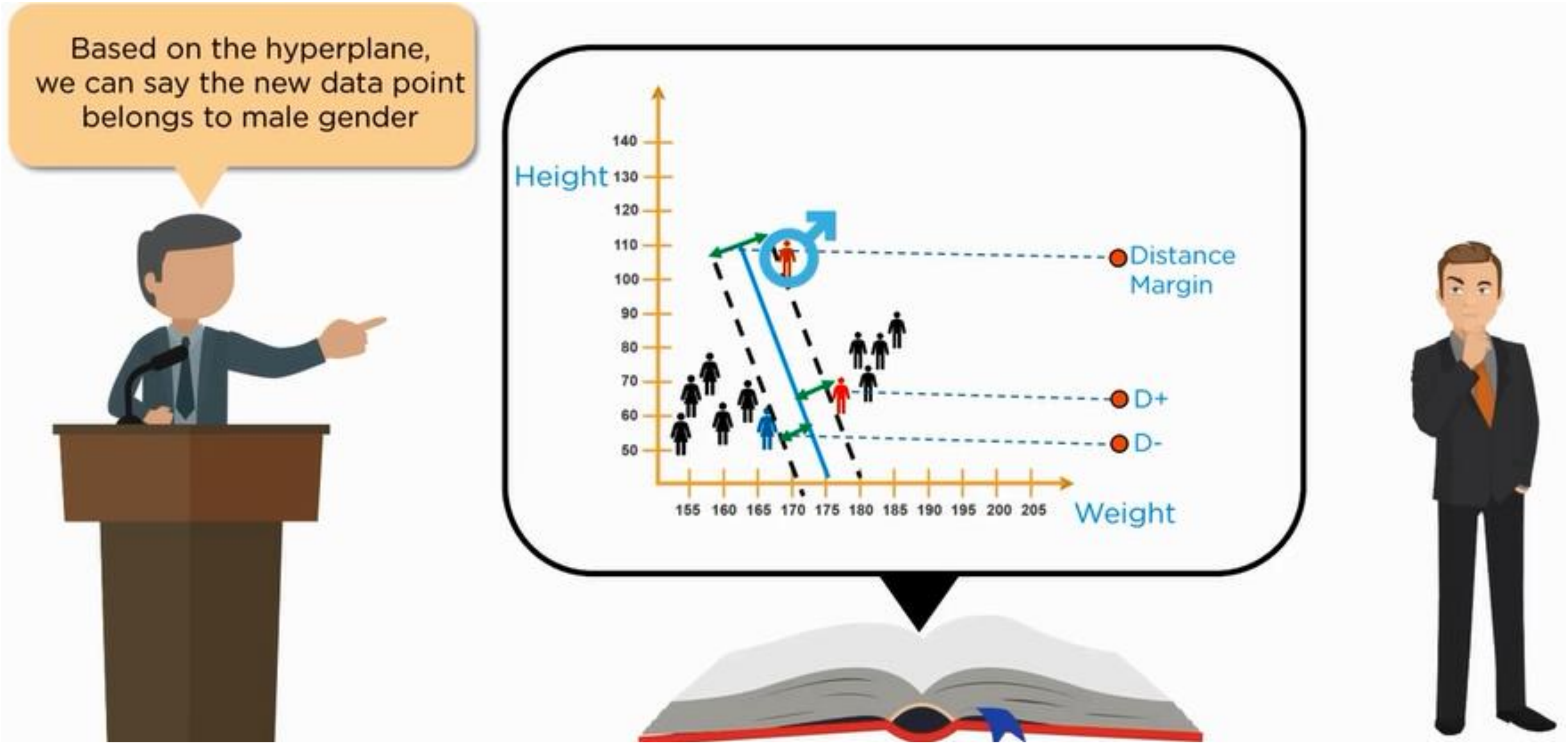
Support Vector Machine-Example



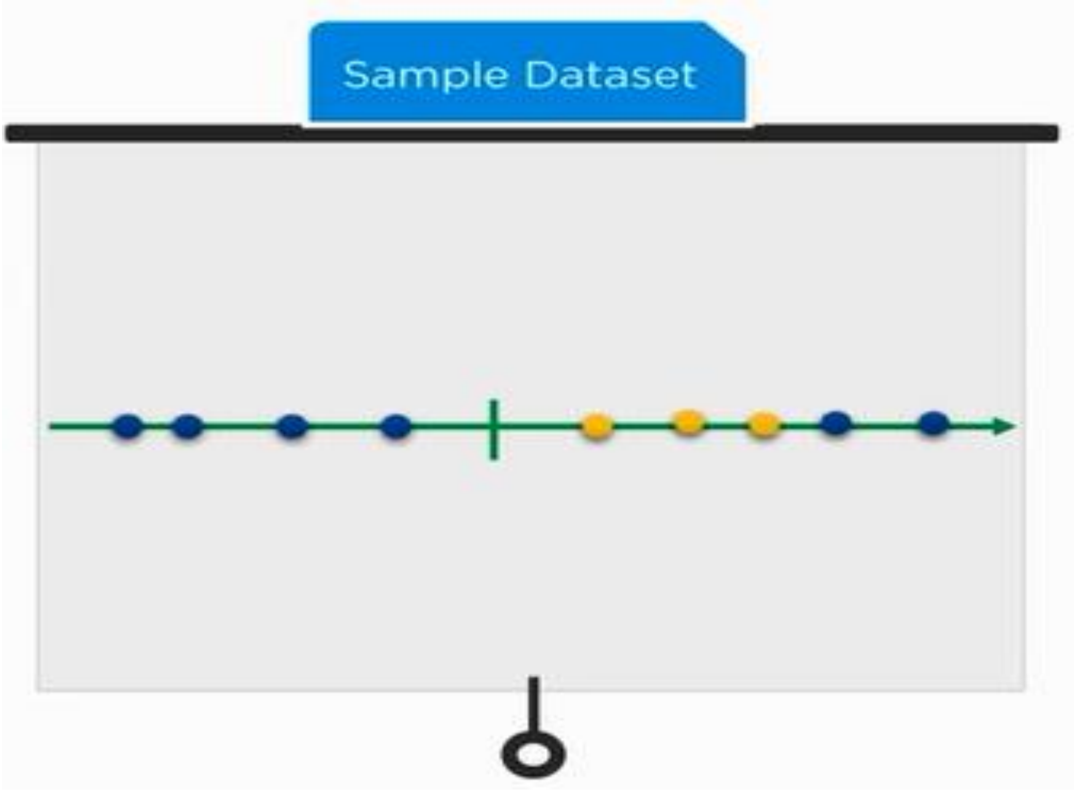
Support Vector Machine-Example



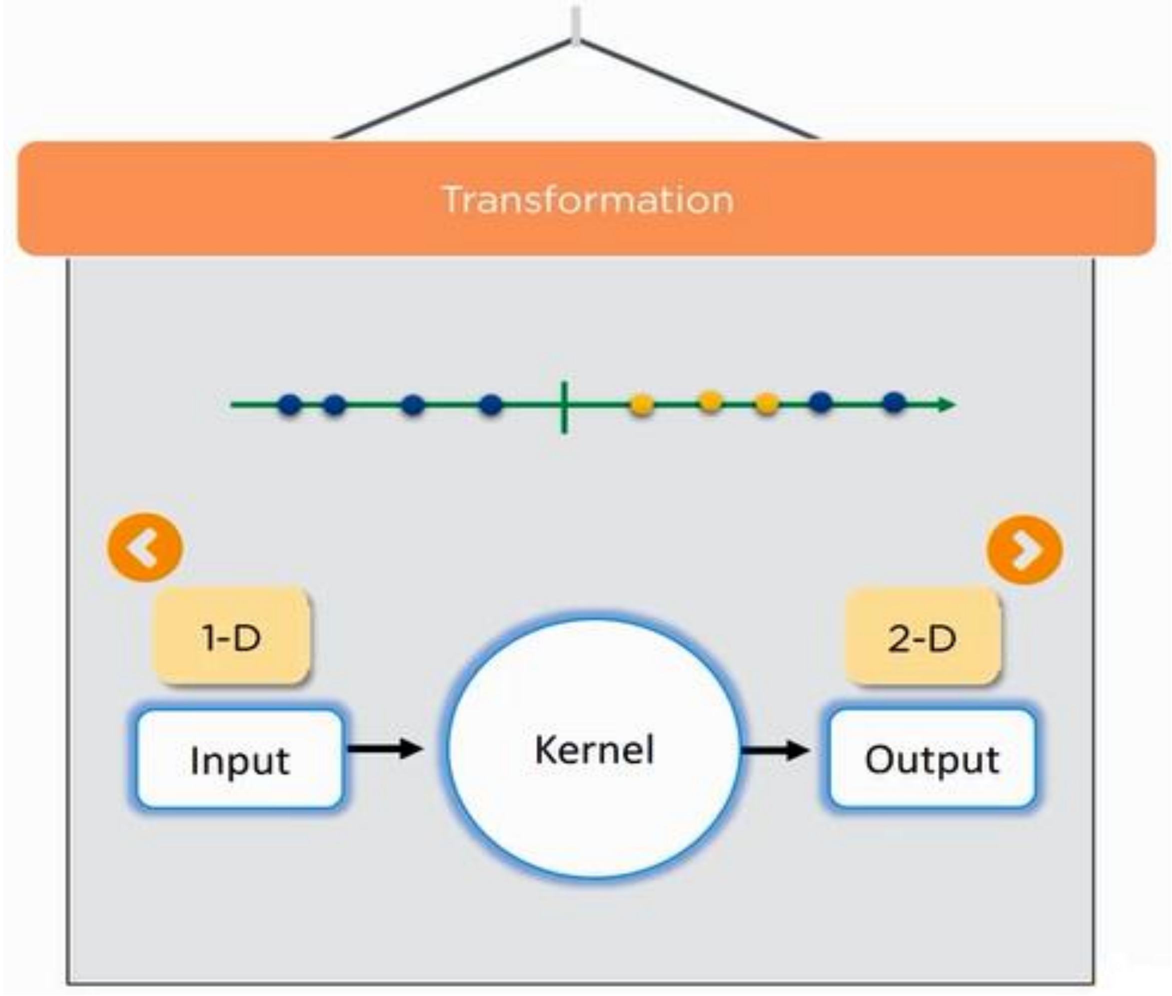
Support Vector Machine-Example



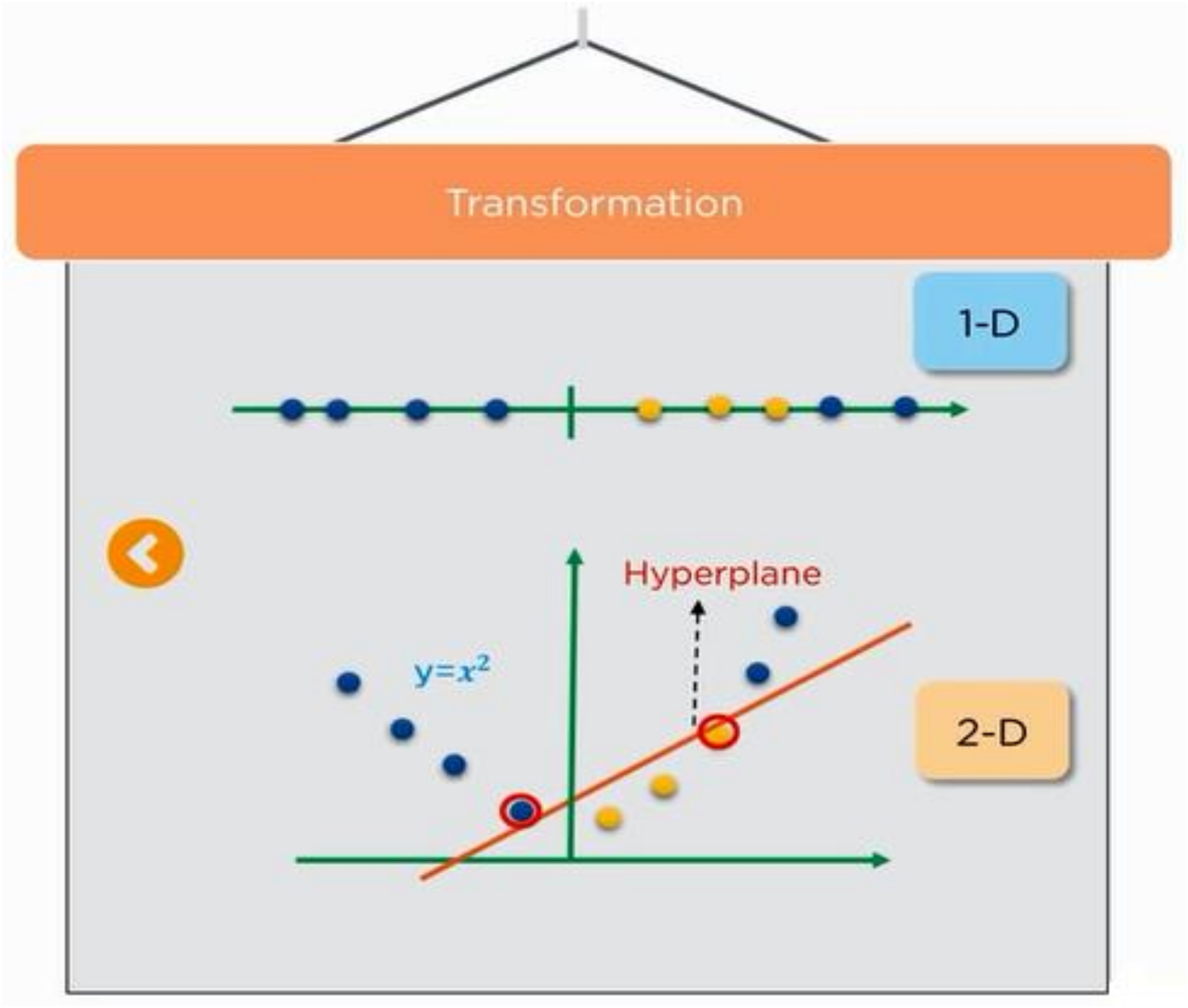
Support Vector Machine-Example



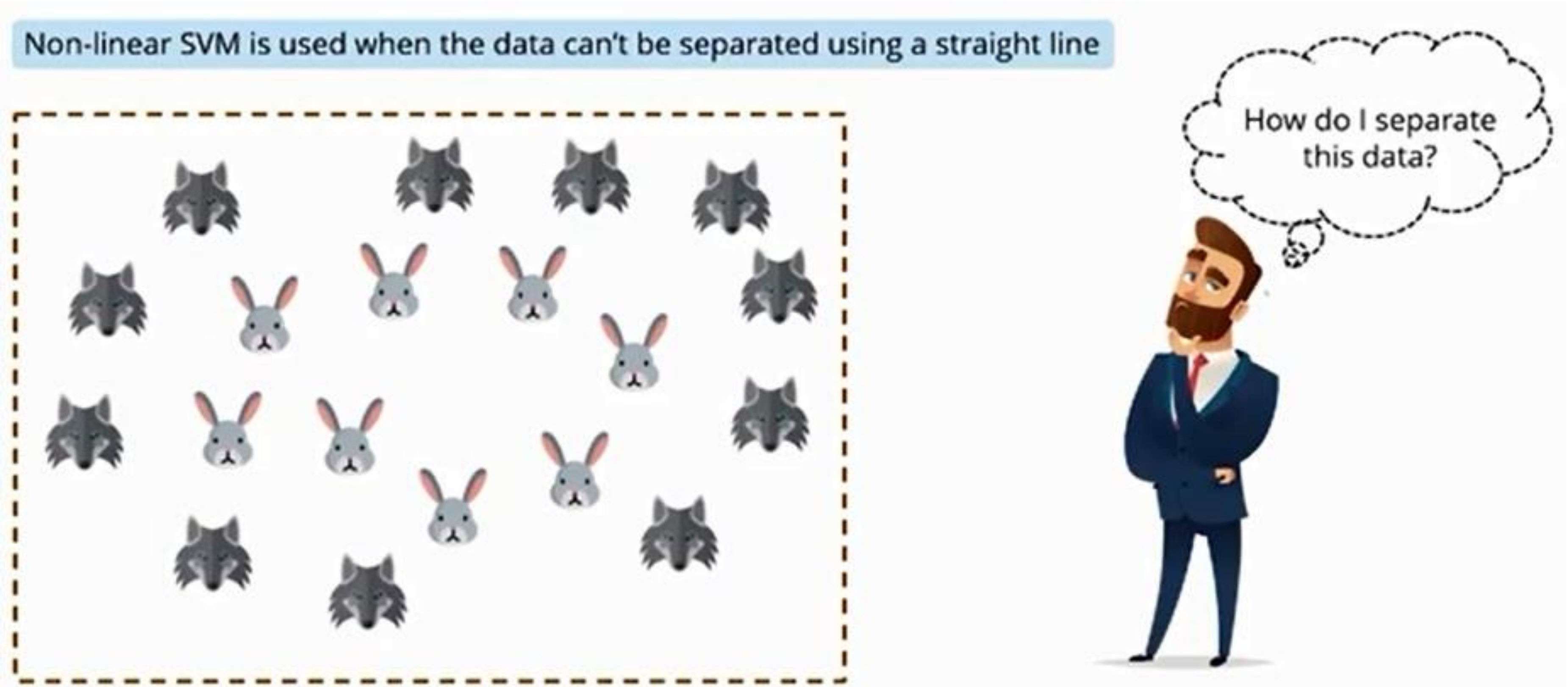
Support Vector Machine-Example



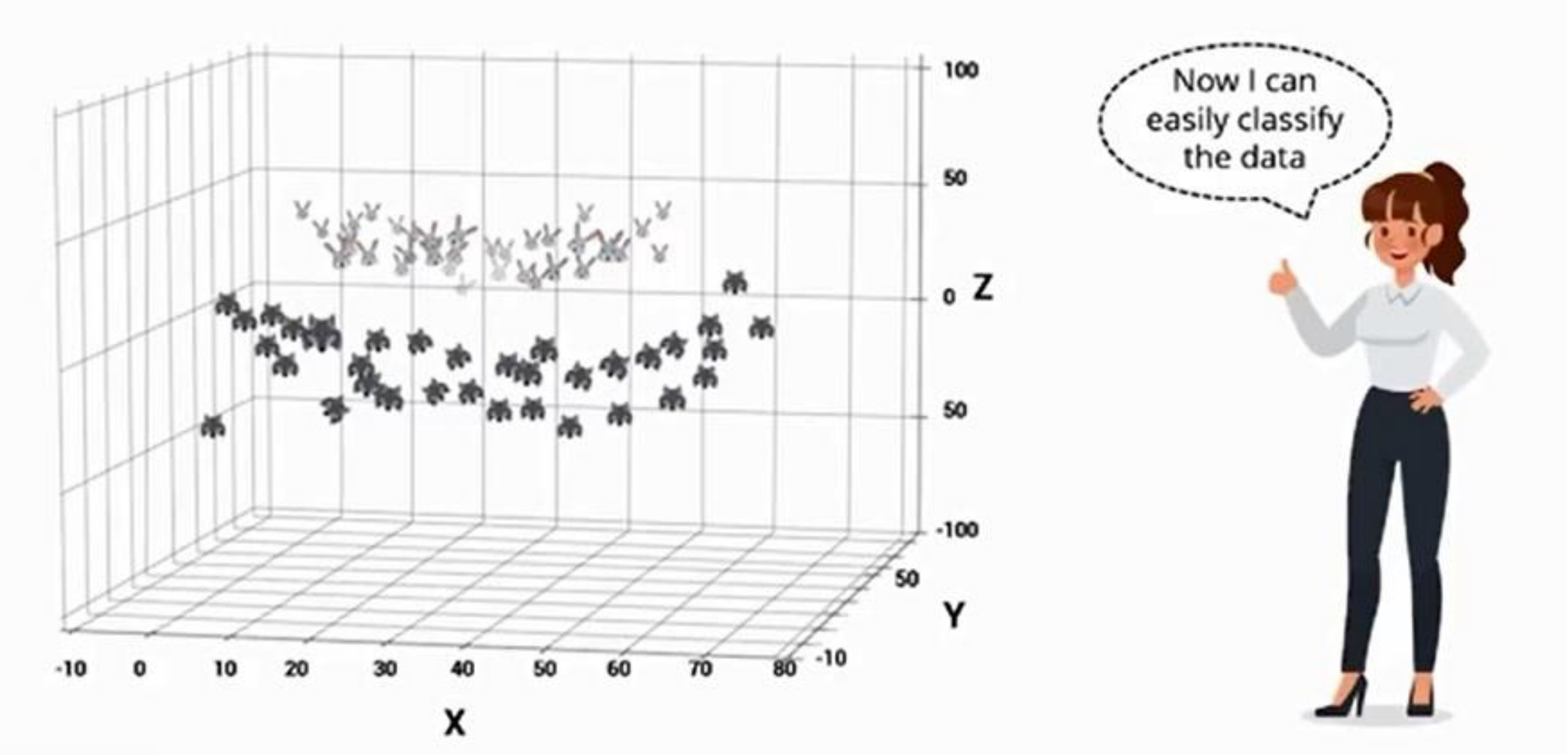
Support Vector Machine-Example



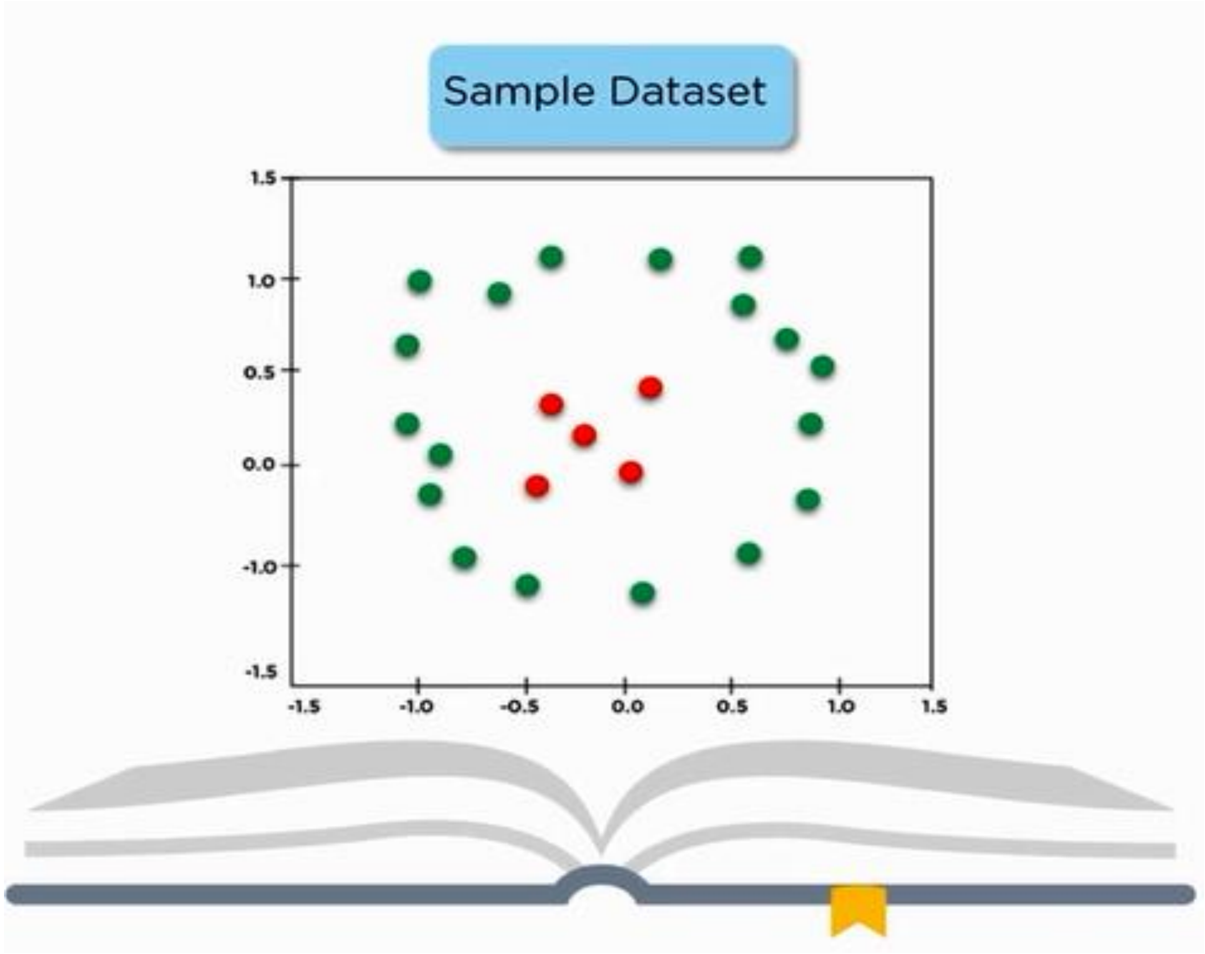
Non Linear Support Vector Machine



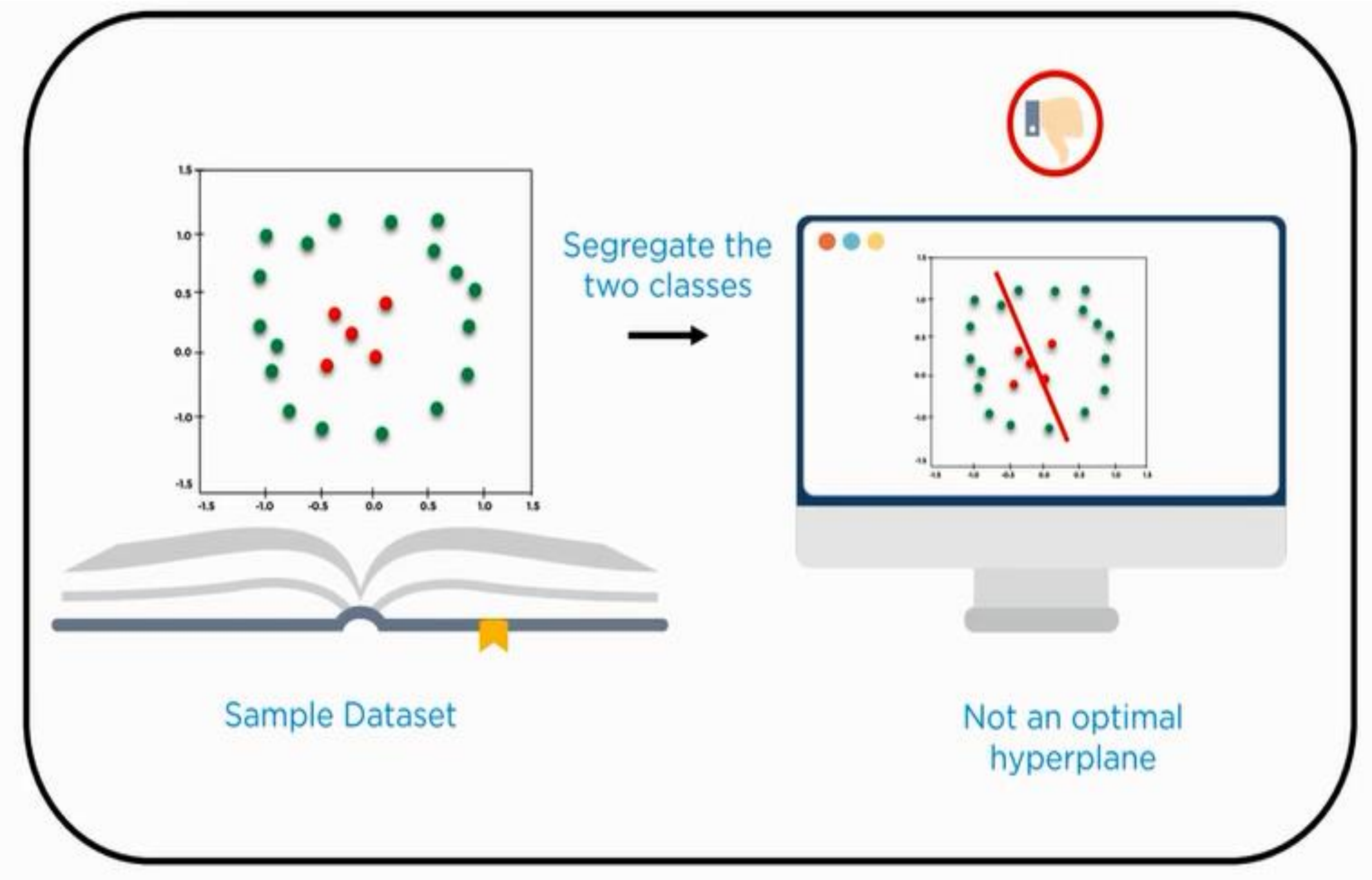
Non Linear Support Vector Machine



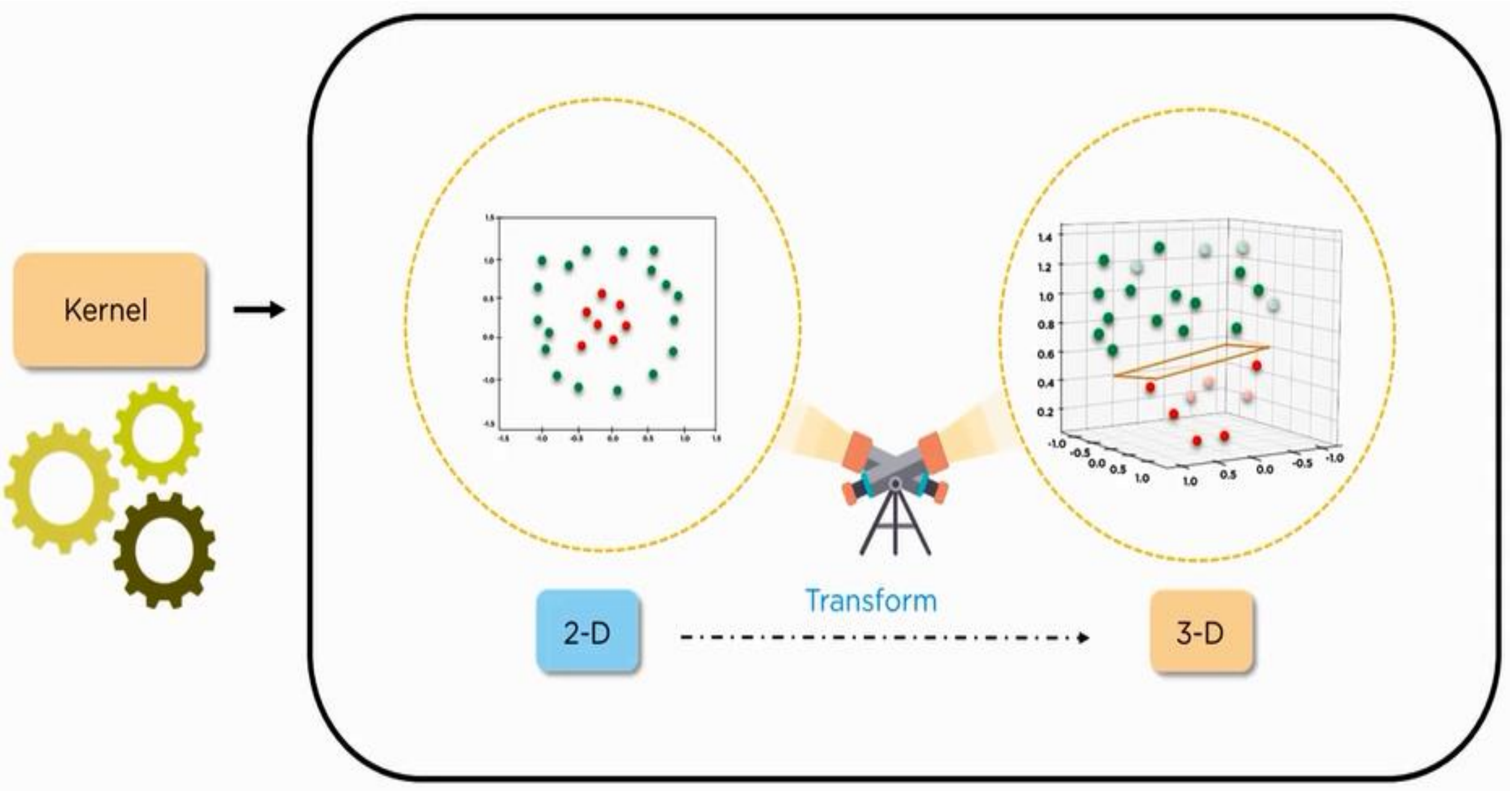
Support Vector Machine-Example



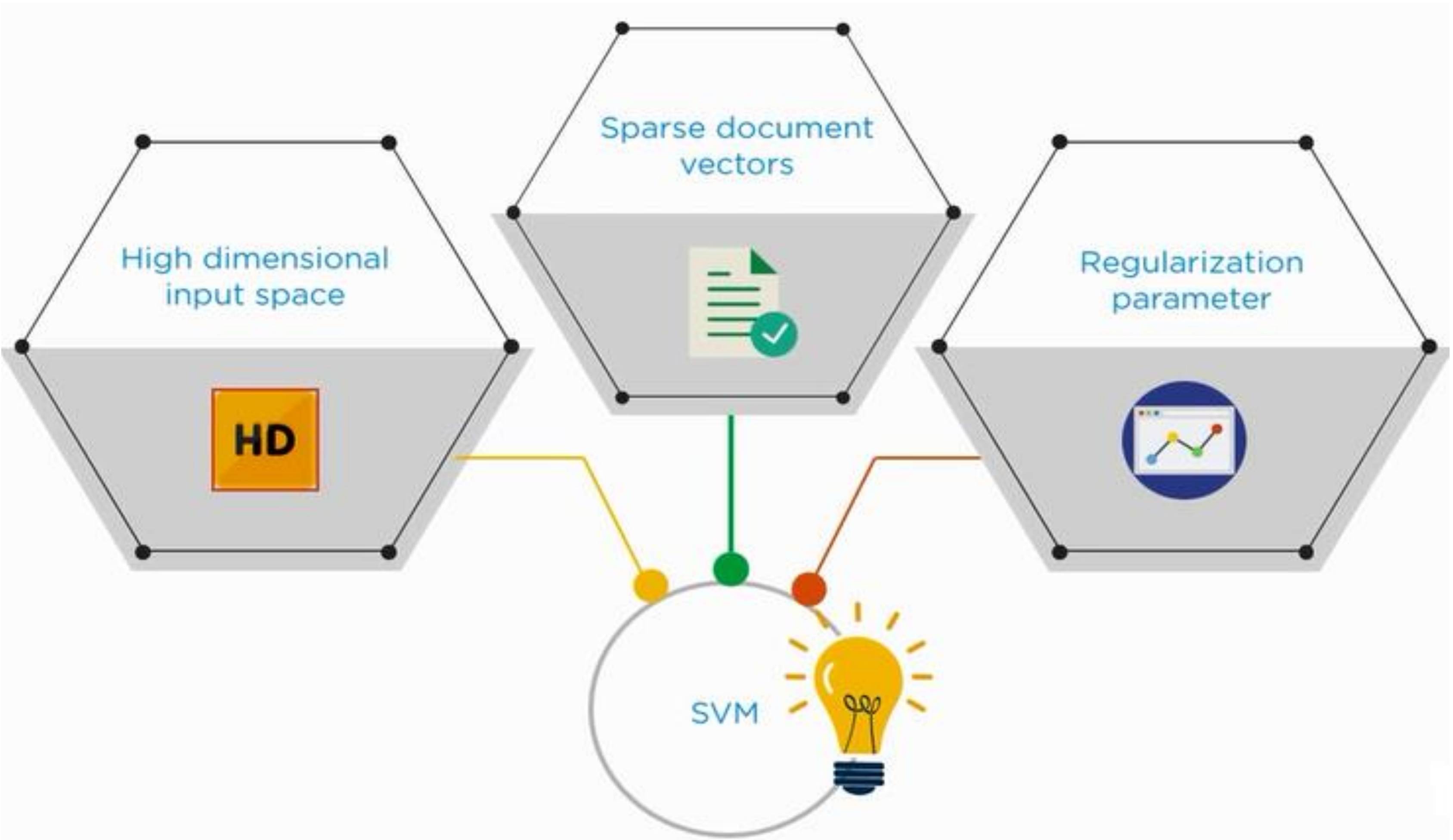
Support Vector Machine-Example



Support Vector Machine-Example

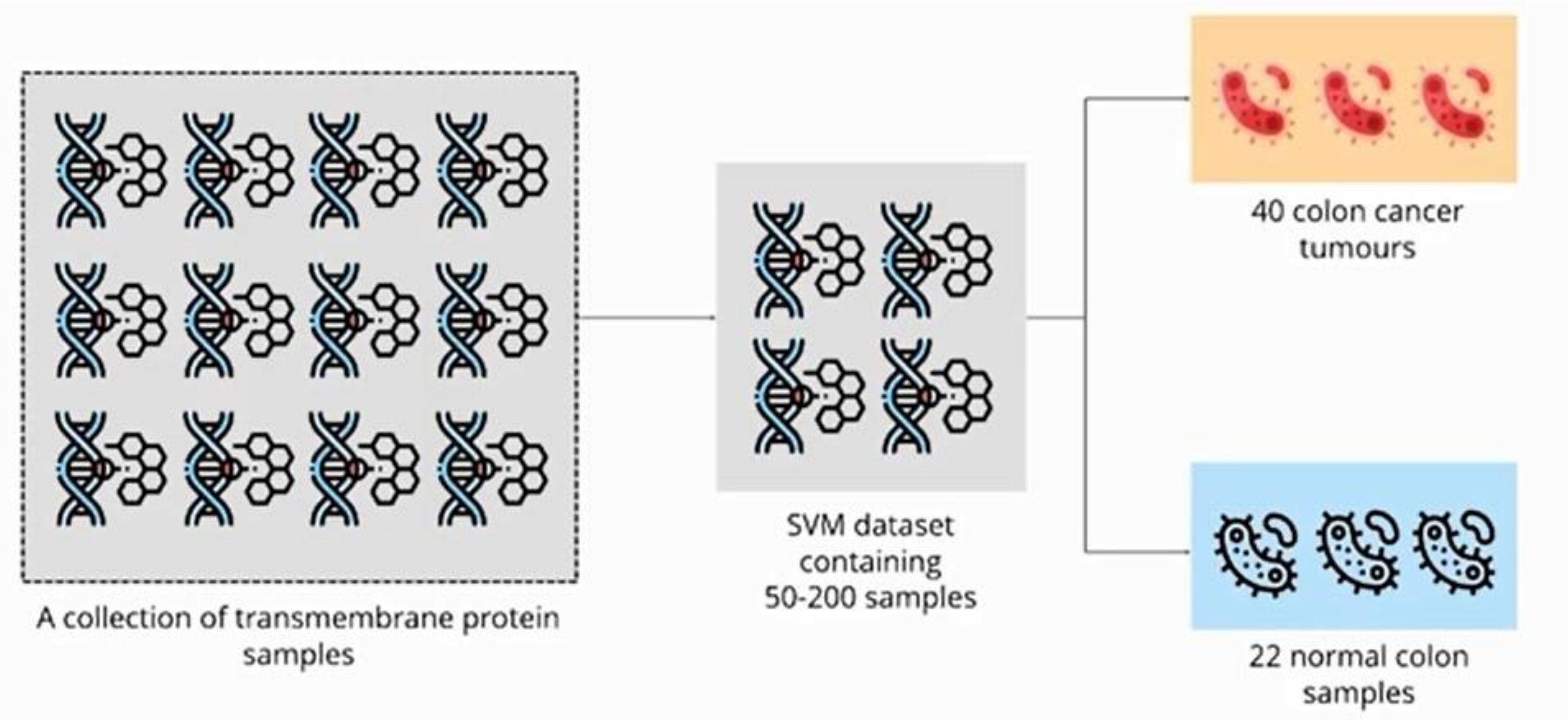


Support Vector Machine-Example



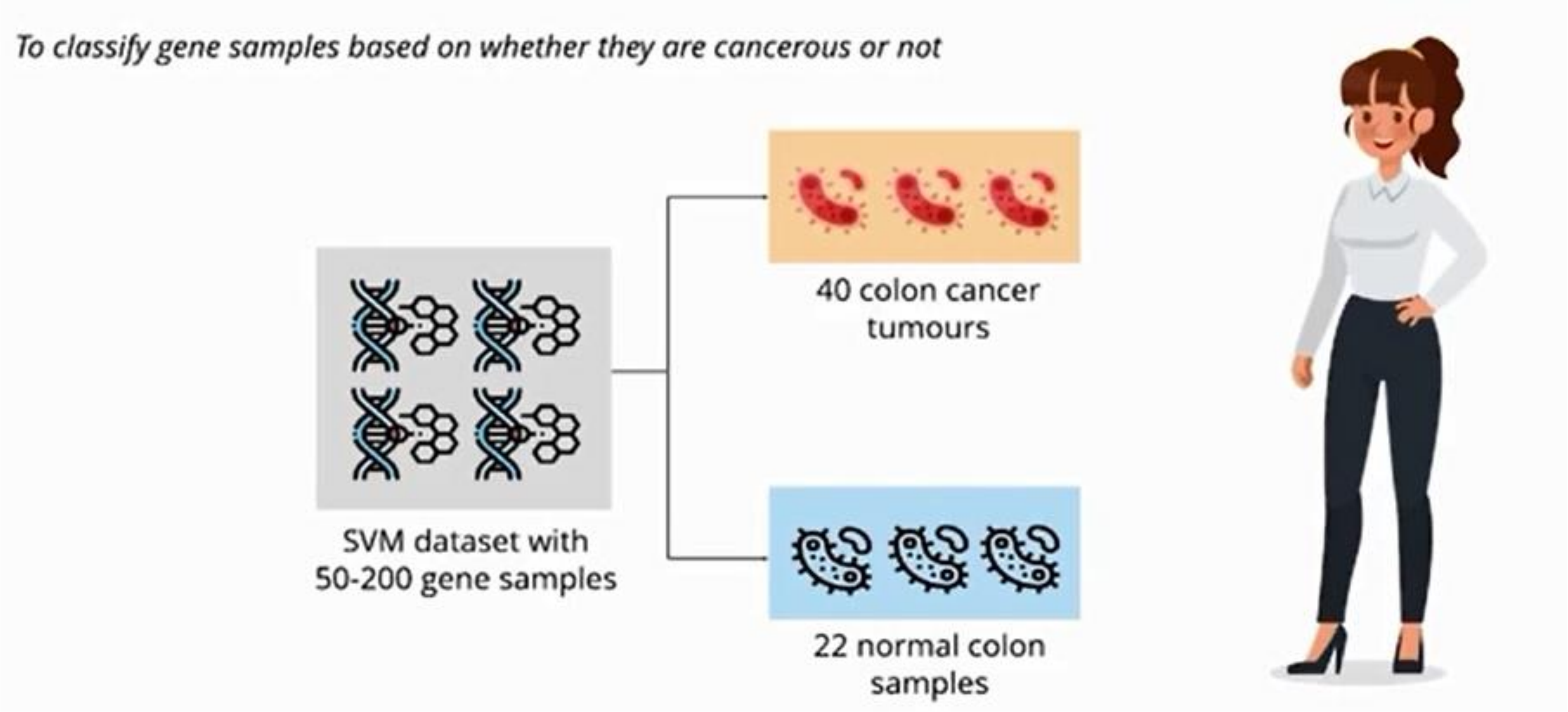
Support Vector Machine-Use Cases

Colon Cancer Classification



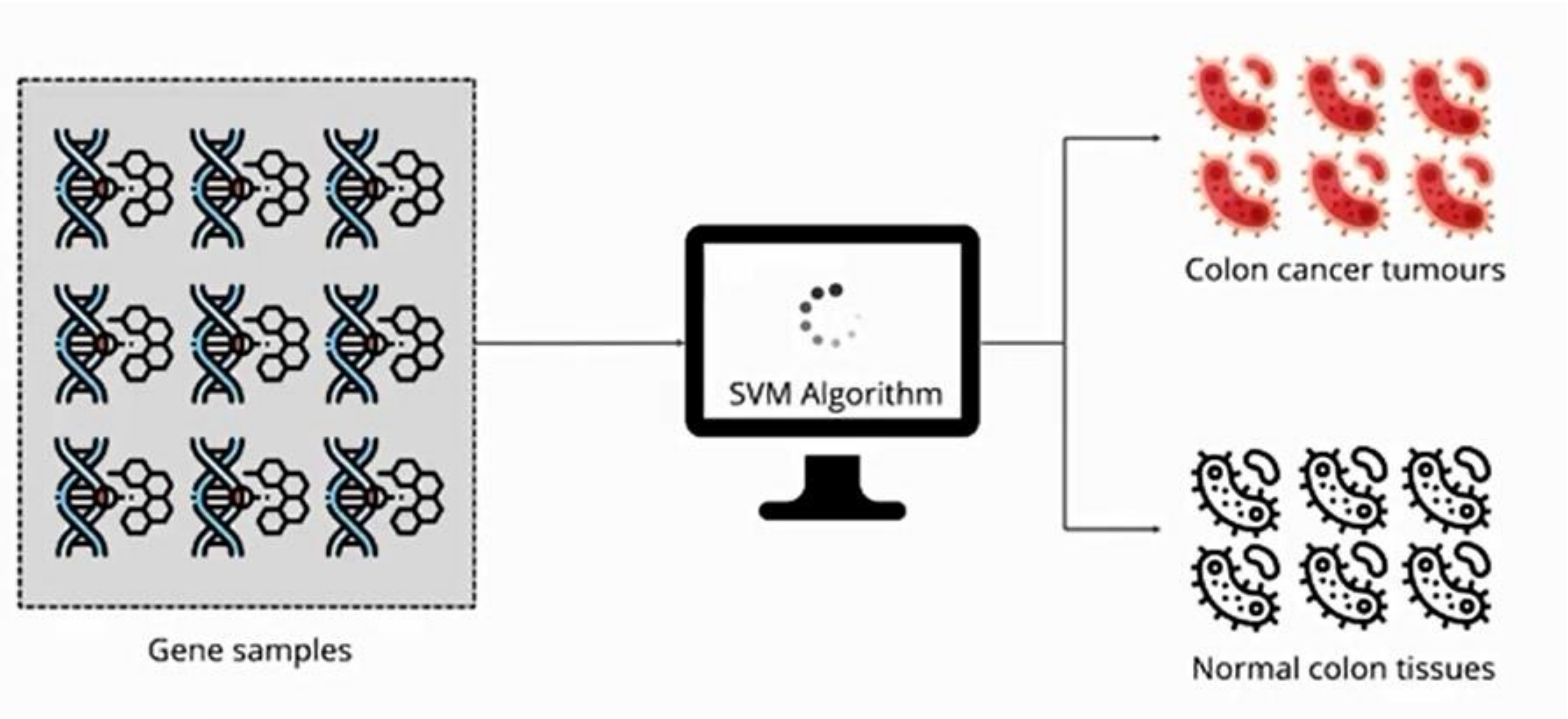
Support Vector Machine-Use Cases

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



Support Vector Machine-Use Cases

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



Support Vector Machine-Applications


Diolch
Kiitos
Sheun
umesc
Dekuju/Dekujeme
Kasih
Todah
Shnorhakalutiun
Mamnoon
Te
ekkür
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ači
Gamsahapnida
Ngiyabonga
Cam
Gra
Xie
Dank
Dhanyavadaalu
Takk
or
Dhanyavaad
al
Dakujem
Waad
Kop
Salamat
Merci
Dankie
Go
Grazie
Faleminderit
Daw
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Or
Dhonnobaad
Shukriya
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun
Hain
Dhan
daa