What's in it for you?

- Why Machine Learning?
- What is Machine Learning?
- Types Of Machine Learning
- Machine Learning Algorithms
  - Linear Regression
  - Decision Trees
  - Support Vector Machine
- Use Case: Classify whether a recipe is of a cupcake or a muffin using SVM

Why Machine Learning?

Because Machine can drive your car for you!!

Because Machine can now detect 50 eye diseases

Because Machine can unlock your phone with your face!!


g
August 22 at 8:47 AM
For Halloween! Comment "YES" if you want it!
Order here: https://www.teefinding.com/limited-edition Share With Your Friends! Worldwide shipping!
57
1 Comment 5 Shares
April 6 at 11:08 AM
TWICE TT LIKE,SHARE,COMMENT "YES" IF YOU LIKE KPOP
44
4 Comments 14 Shares


For Halloween! Comment "YES" if you want it!
Order here: https://www.teefinding.com/limited-edition Share With Your Friends! Worldwide shipping!
Nobody likes spam posts on Facebook that annoy them into interacting with likes, shares, comments, and other actions


This tactic, known as "Engagement Bait," takes advantage of Facebook's Newsfeed algorithm by boosting engagement in order to get greater reach

To eliminate engagement bait, the company reviewed and categorized hundreds of thousands of posts to train a machine learning model that detects different types of engagement bait

Facebook scroll GIF will be replaced

This is a tag bait!

Block this post

Data fed to the machine

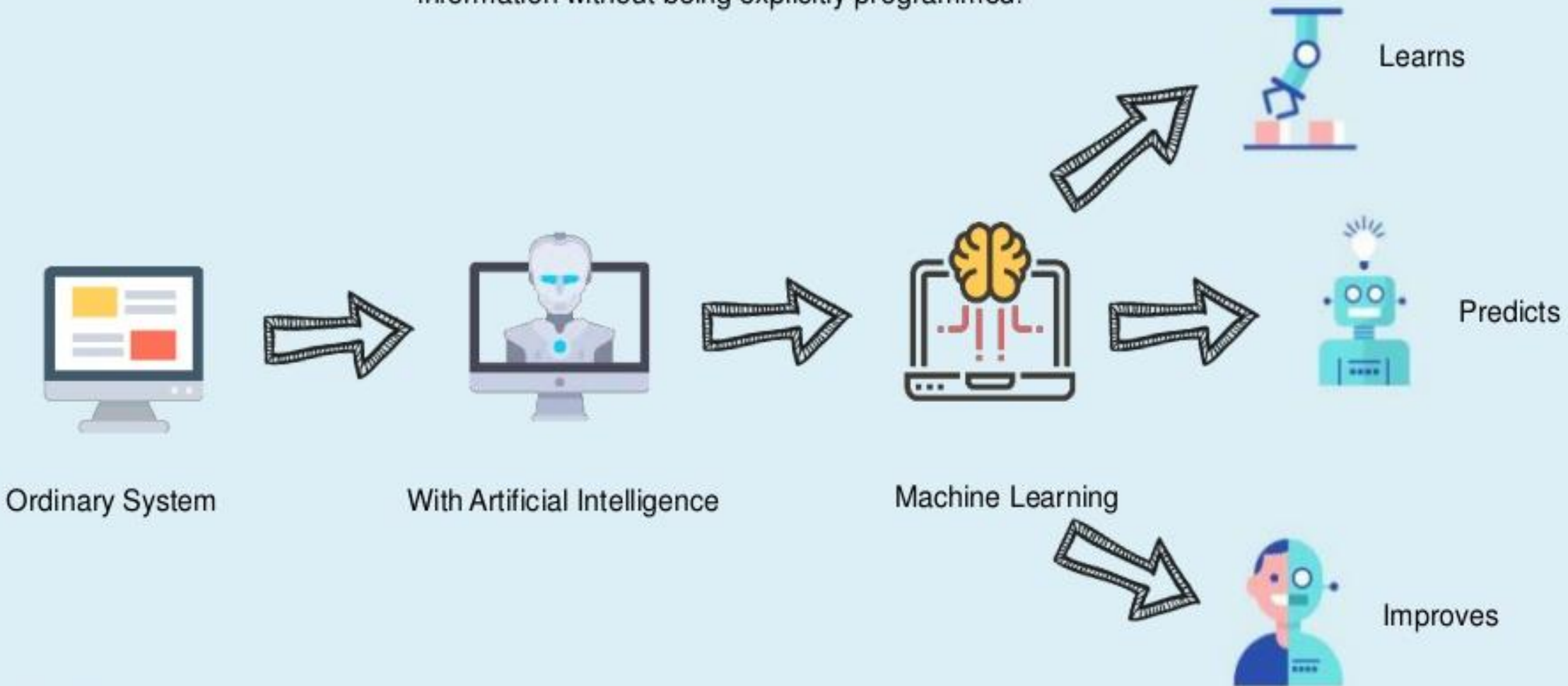
Scans the keywords and phrases like “This” and checks the click through rate

New Post

Google's DeepMind project "AlphaGO", a computer program that plays the board game 'GO' has defeated the world's number one Go player Ke Jie

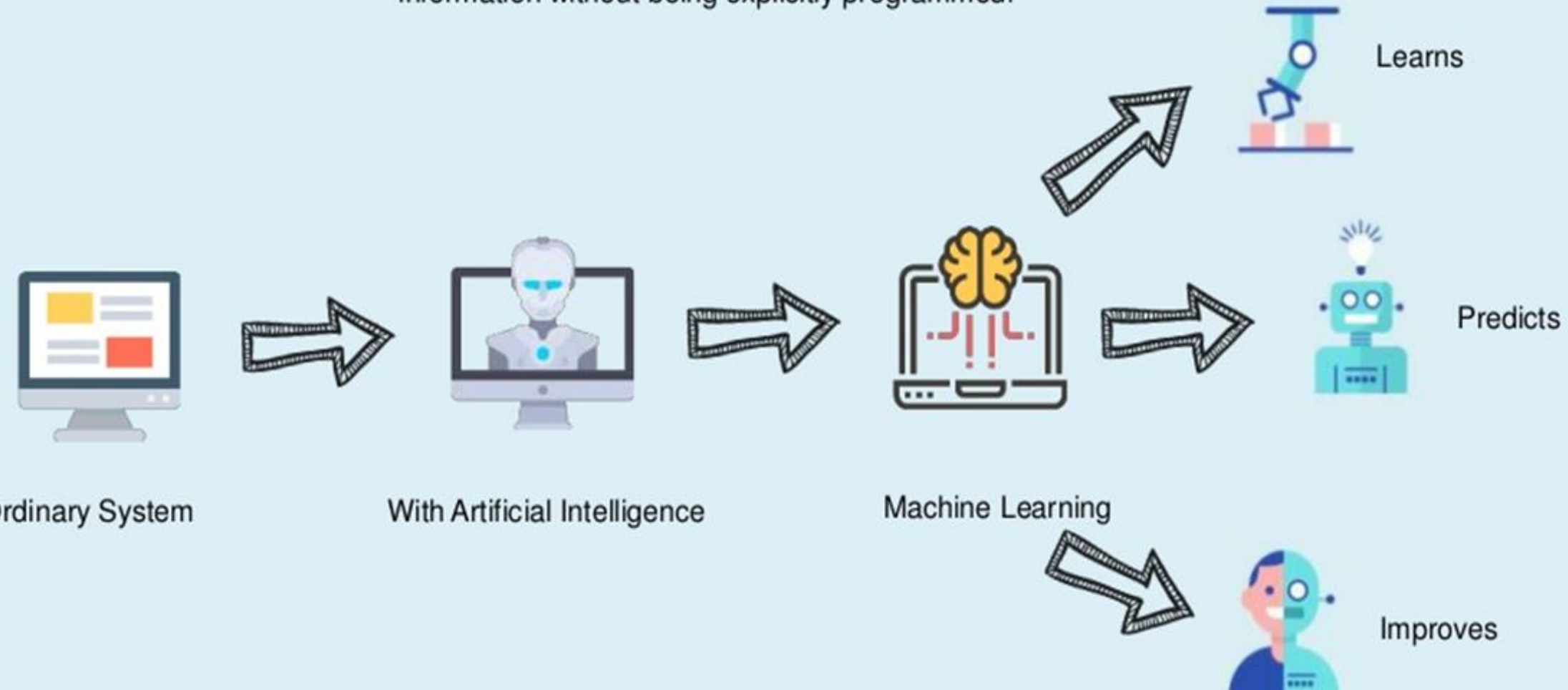
What is Machine Learning?

Machine learning is the science of making computers learn and act like humans by feeding data and information without being explicitly programmed!



What is Machine Learning?

Machine learning is the science of making computers learn and act like humans by feeding data and information without being explicitly programmed!



What is Machine Learning?

Do you want to predict a category? That's classification!

For instance, whether the stock price will increase or decrease

Do you want to predict a quantity? That's regression!

For instance, predicting the age of a person based on the height, weight, health and other factors

Do you want to detect an anomaly? That's anomaly detection!

For instance, you want to detect money withdrawal anomalies

Do you want to discover structure in unexplored data? That's clustering

For instance: Finding groups of customers with similar behavior given a large database of customer data containing their demographics and past buying records


? QUIZ ?
TIME

Can you tell what's happening in the following cases?

A. Grouping documents into different categories based on the topic and content of each document

Can you tell what's happening in the following cases?

A. Grouping documents into different categories based on the topic and content of each document

B. Identifying hand-written digits in images correctly

Can you tell what's happening in the following cases?

A. Grouping documents into different categories based on the topic and content of each document

B. Identifying hand-written digits in images correctly

C. Behavior of a website indicating that the site is not working as designed

Can you tell what's happening in the following cases?

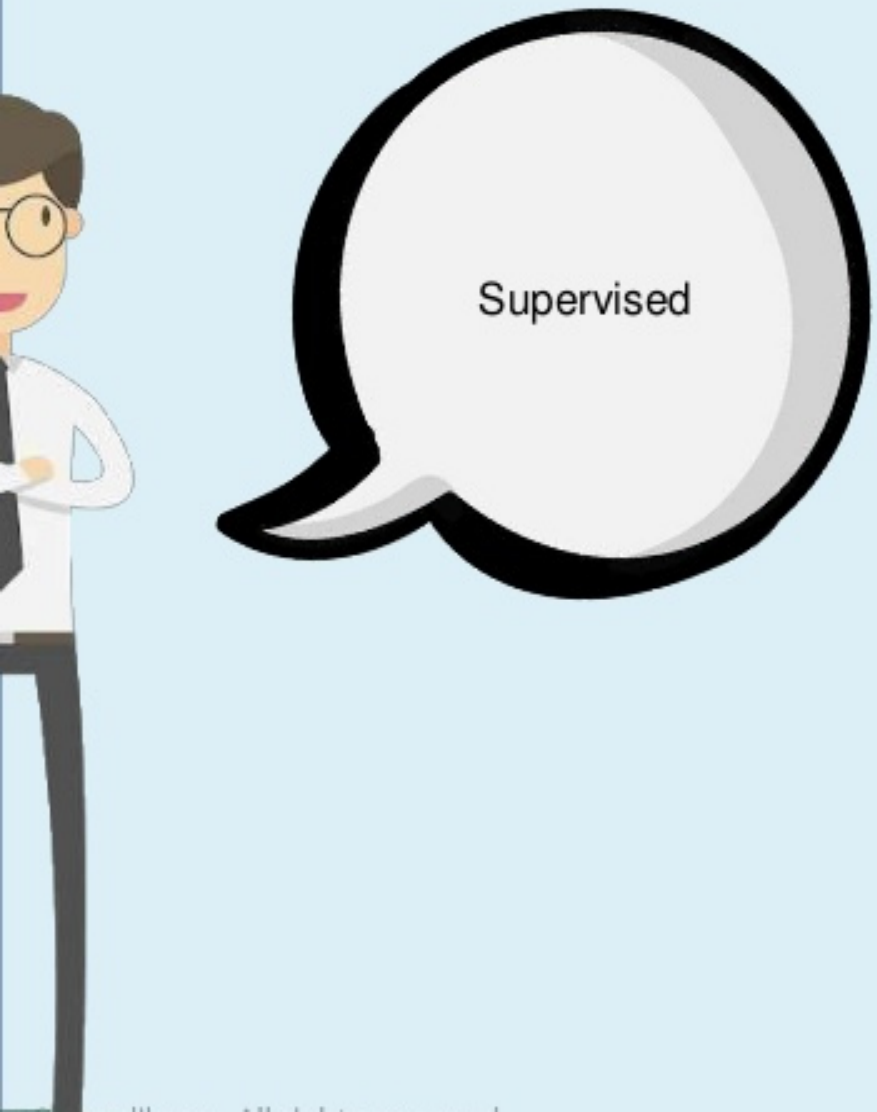
A. Grouping documents into different categories based on the topic and content of each document

B. Identifying hand-written digits in images correctly

C. Behavior of a website indicating that the site is not working as designed

D. Predicting salary of an individual based his/her years of experience

Types of Machine Learning

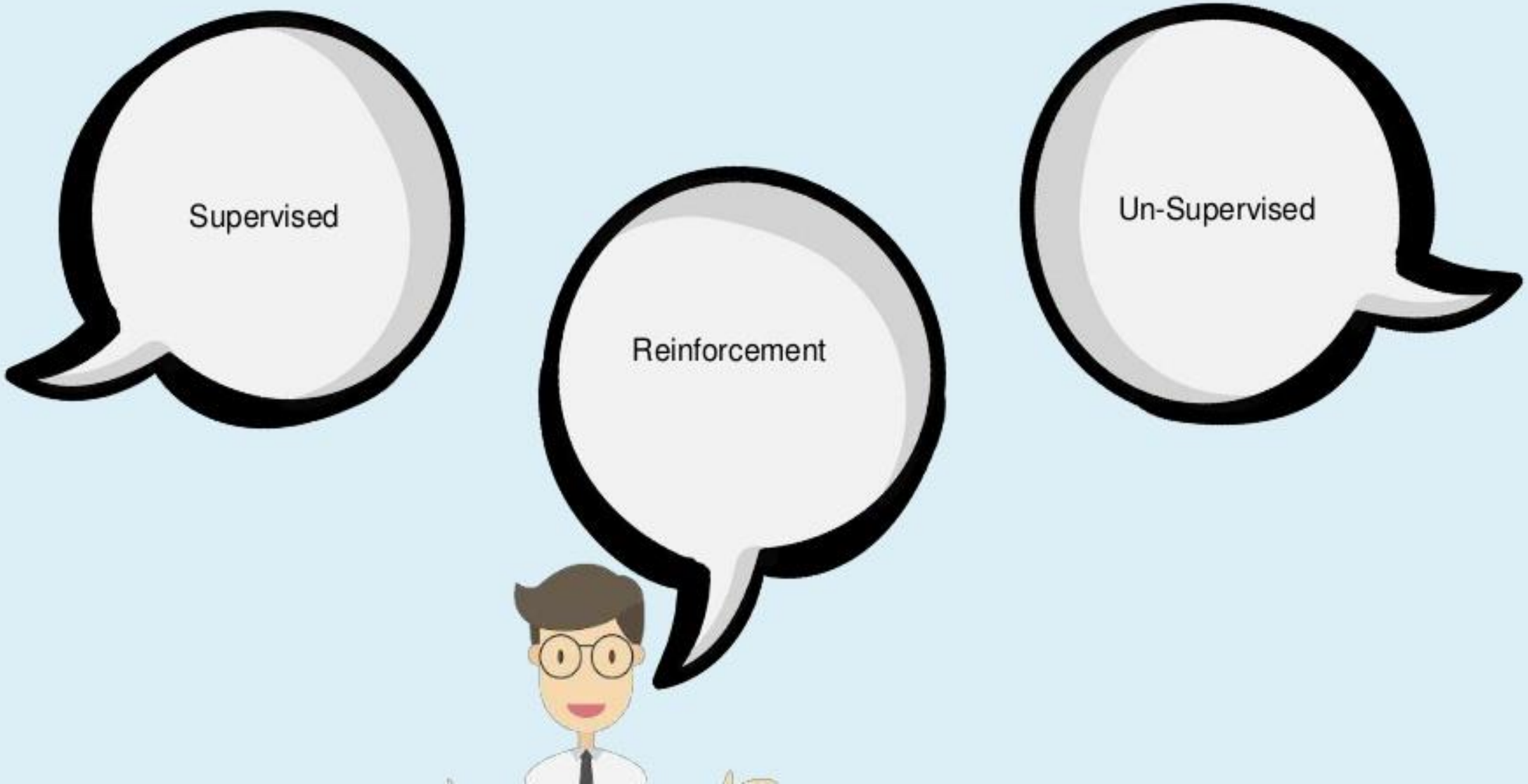


Types of Machine Learning

Supervised

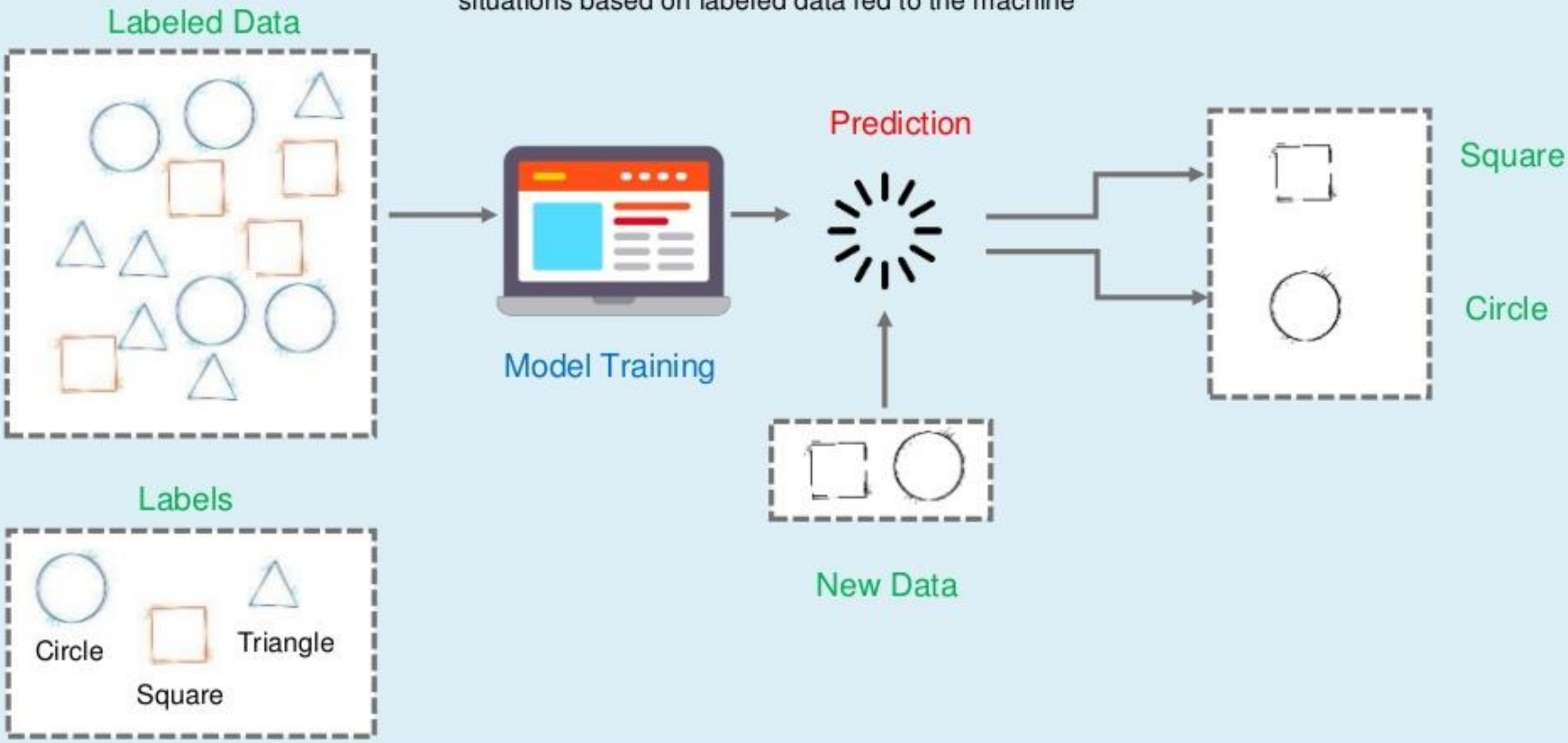
Un-Supervised

Types of Machine Learning



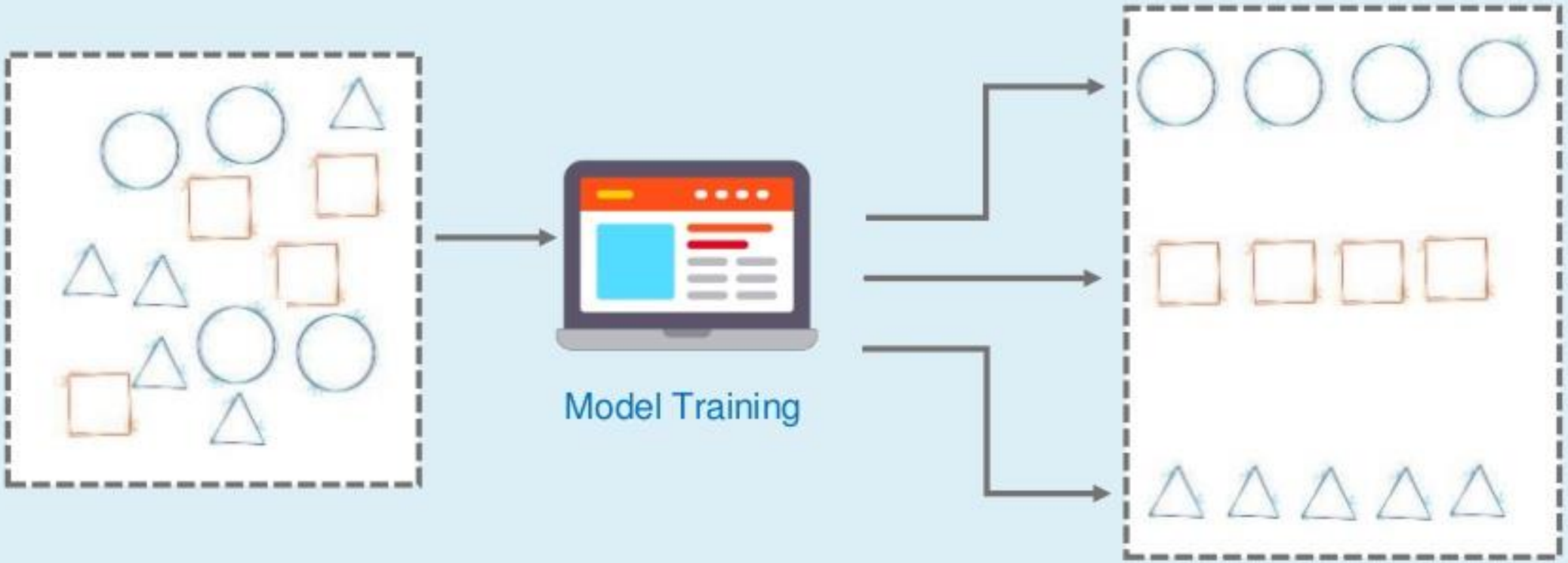
Supervised Learning

Supervised learning is a method used to enable machines to classify/ predict objects, problems or situations based on labeled data fed to the machine



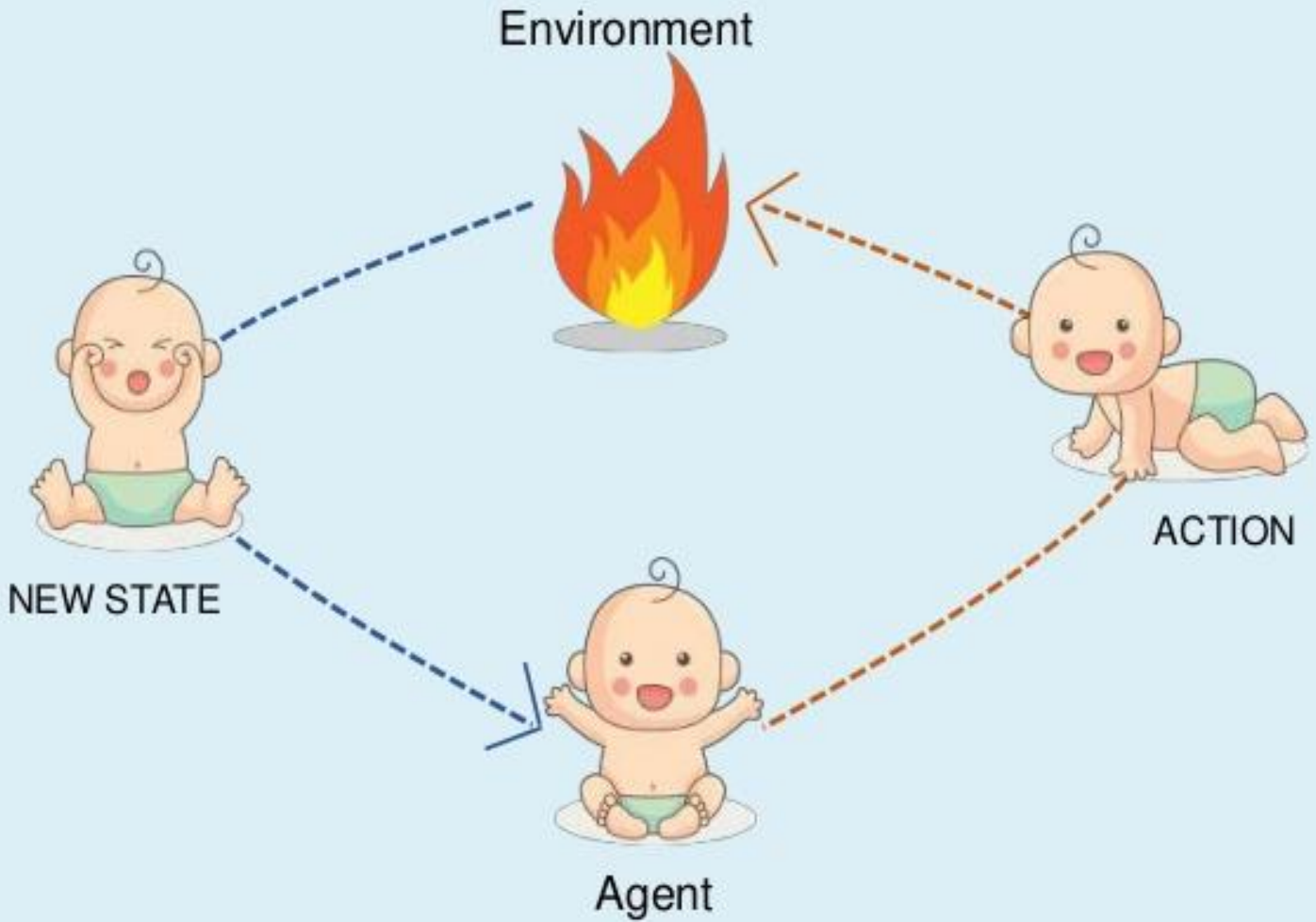
Unsupervised Learning

In Unsupervised learning, Machine Learning model finds the hidden pattern in an unlabeled data



Reinforcement Learning

Reinforcement learning is an important type of Machine Learning where an agent learns how to behave in an environment by performing actions and seeing the results



Supervised VS Unsupervised

Machine Learning Algorithms

There are many interesting Machine Learning algorithms, let's have a look at a few of them

Linear Regression

Decision Trees

Support Vector Machine

Linear Regression

Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning!

Linear regression is a **linear model**,

e.g. a model that assumes a linear relationship between

the input variables (x)

and a single output variable (y)

$$y = mx + c$$

Linear Regression

Imagine, we are predicting distance travelled (y) from speed (x). Our linear regression model representation for this problem would be:

$$y = m * x + c$$

Or

$$distance = m * speed + c$$

c = coefficient
m = y-intercept

Time is constant

Speed = **10m/s**

Distance = 36 km

Time is constant

Speed = **10m/s**

Distance = 36 km

Speed = **20m/s**

Distance = 52 km

Time is constant

Speed = **10m/s**

Distance = 36 km

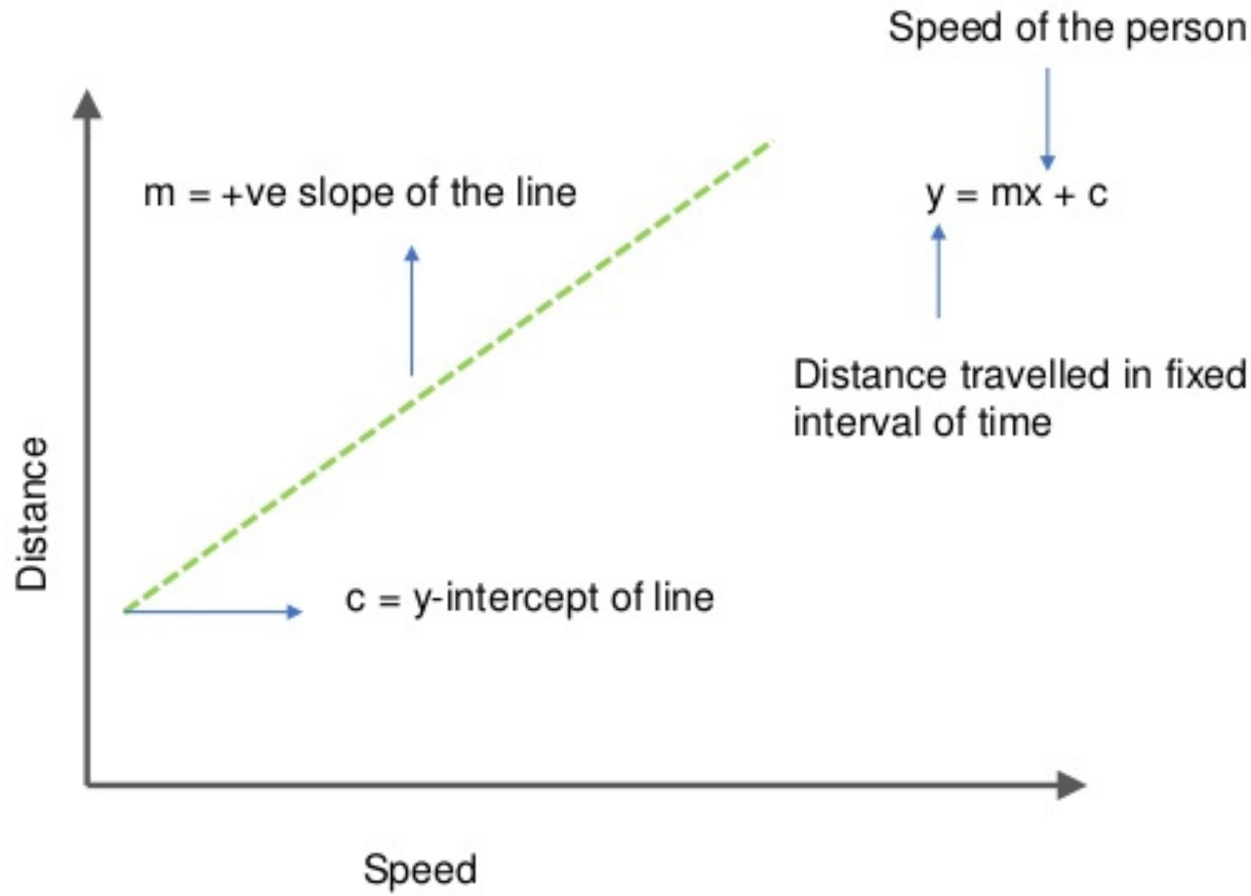
Speed = **20m/s**

Distance = 52 km

Speed = **30m/s**

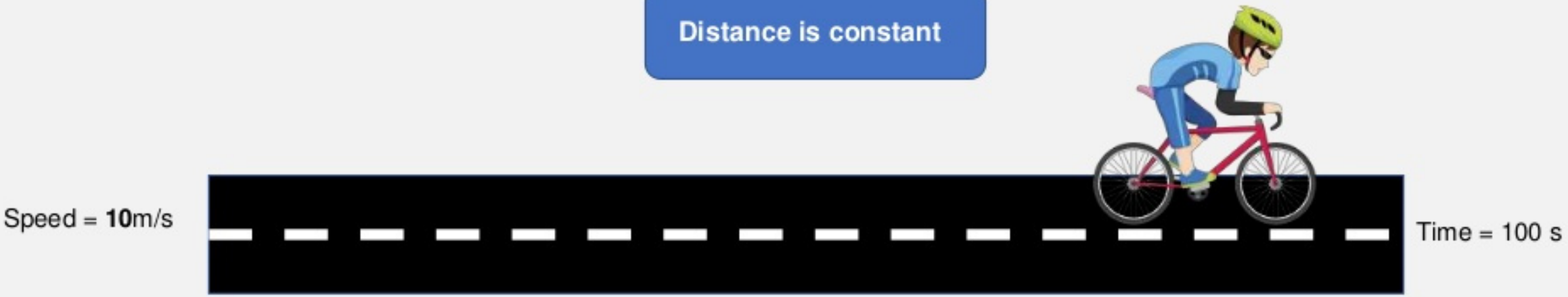
Distance = ?

Linear Regression



As the speed increases, distance also increases, hence the variables have a positive relationship

Distance is constant



Distance is constant

Speed = **10**m/s

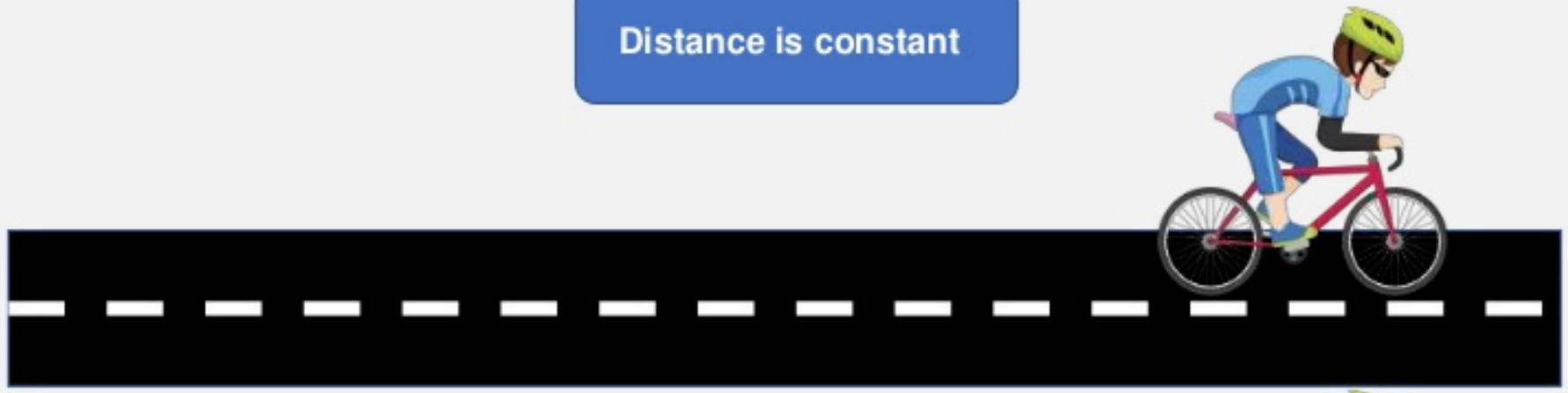
Time = 100 s

Speed = **20m/s**

Time = 50 s

Distance is constant

Speed = **10**m/s



Time = 100 s

Speed = **20m/s**

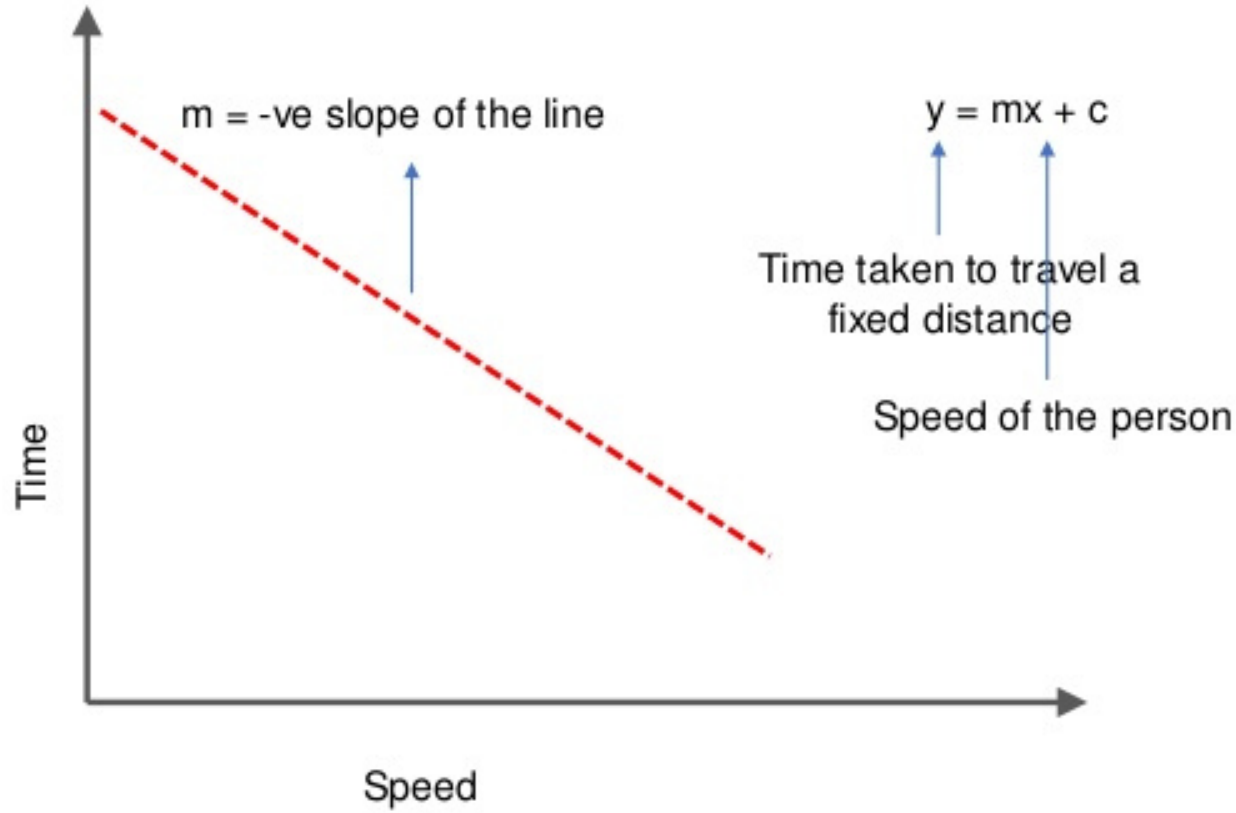
Time = 50 s

Speed = **30m/s**

Time = ?

Linear Regression

If distance is assumed to be constant, let's see the relationship between speed and time



As the speed increases, time decreases, hence the variables have a negative relationship

Linear Regression

Let's see the mathematical implementation of Linear Regression!

Suppose we have a dataset that looks like:

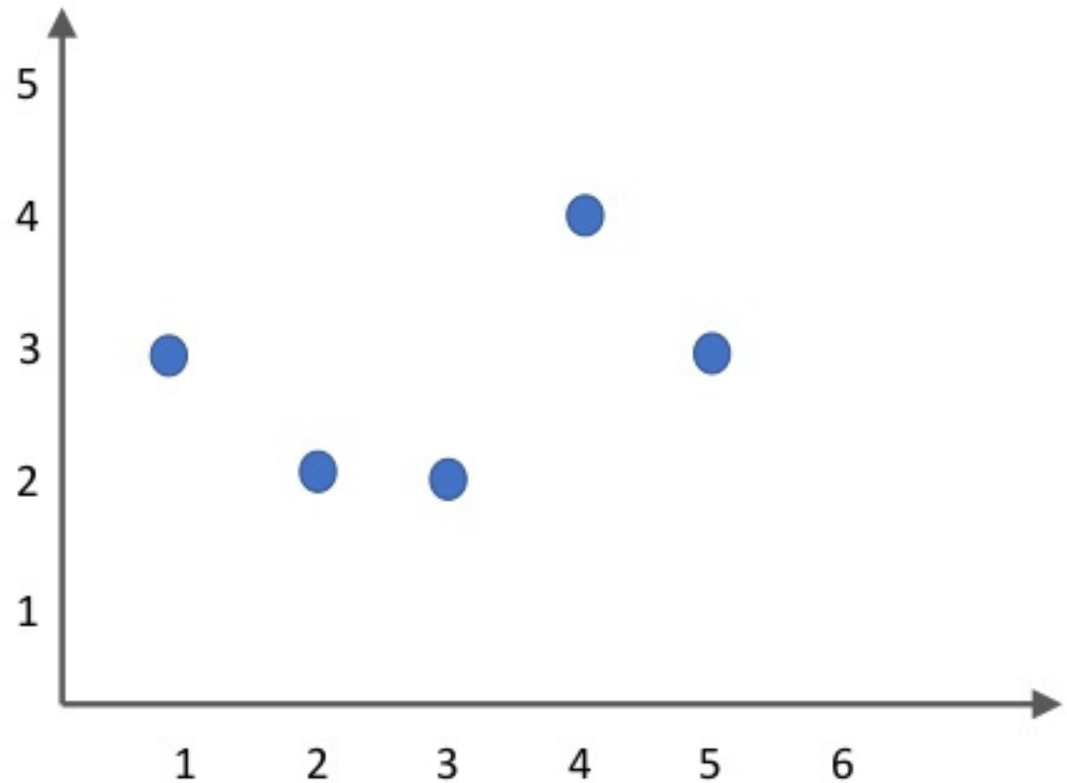
x	y
1	3
2	2
3	2
4	4
5	3

Linear Regression

x	y
1	3
2	2
3	2
4	4
5	3

Mean(xi) = 3

Let's plot these points!!

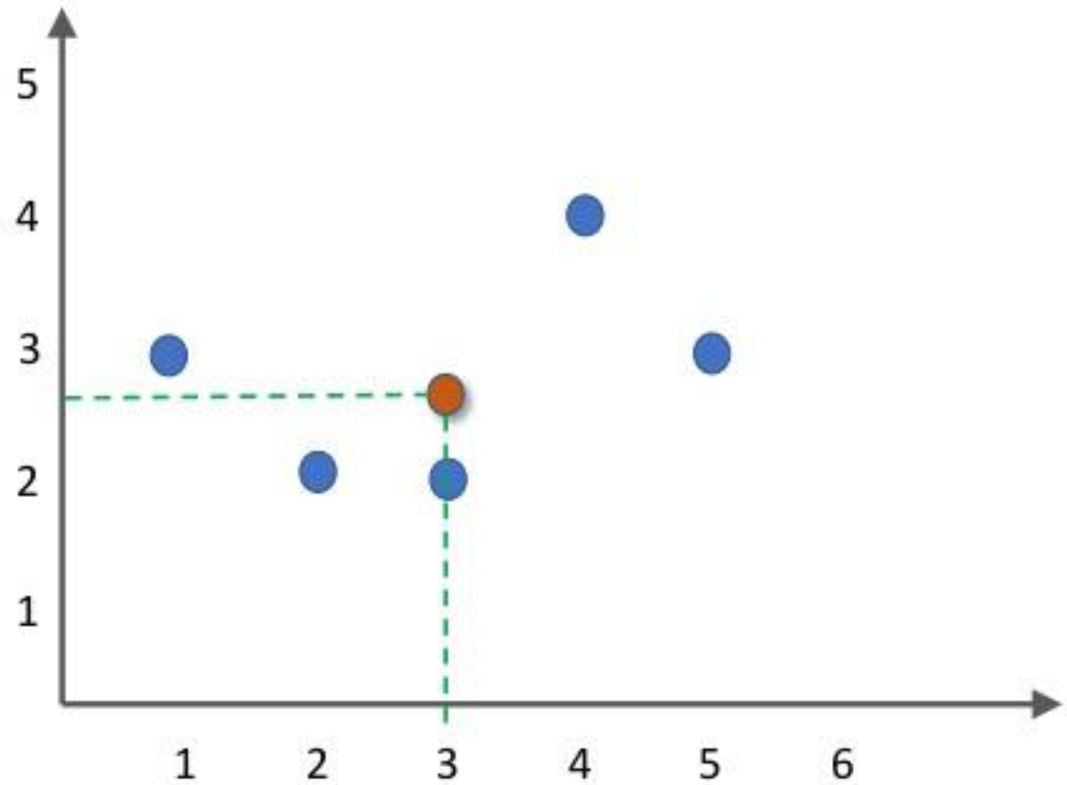


Linear Regression

x	y
1	3
2	2
3	2
4	4
5	3

Mean(xi) = 3 Mean(yi) = 2.8

Let's plot these points!!



Linear Regression

Now, lets find regression equation to find the best fit line!

$$y = mx + c$$

To find this equation for our data, we need to find our slope (m) and coefficient (c)

Linear Regression

$$y = mx + c$$

$$m = \frac{\sum(x- xi) (y - yi)}{\sum(x- xi)^2}$$

x	y	x- xi	y – yi	(x- xi) ^{2}	(x- xi) (y – yi)
1	3	-2	0.2	4	-0.4
2	2	-1	-0.8	1	0.8
3	2	0	-0.8	0	0
4	4	1	1.2	1	1.2
5	3	2	0.2	4	0.4

Linear Regression

$$y = mx + c$$

$$m = \frac{\sum(x- xi) (y - yi)}{\sum(x- xi)^2}$$

x	y	x- xi	y – yi	(x- xi) ^{2}	(x- xi) (y – yi)
1	3	-2	0.2	4	-0.4
2	2	-1	-0.8	1	0.8
3	2	0	-0.8	0	0
4	4	1	1.2	1	1.2
5	3	2	0.2	4	0.4
				Total = 10	Total = 2

Linear Regression

$$y = mx + c$$

$$m = \frac{\sum(x- xi) (y - yi)}{\sum(x- xi)^2} = 2/10 = 0.2$$

x	y	x- xi	y – yi	(x- xi) ^{2}	(x- xi) (y – yi)
1	3	-2	0.2	4	-0.4
2	2	-1	-0.8	1	0.8
3	2	0	-0.8	0	0
4	4	1	1.2	1	1.2
5	3	2	0.2	4	0.4
				Total = 10	Total = 2

Linear Regression

So, we can calculate the value of c

$$y = mx + c$$

$$m = \frac{\sum(x- xi) (y - yi)}{\sum(x- xi)^2} = 2/10 = 0.2$$

y = 0.2 x + c

Mean values = (3, 2.8)

2.8 = 0.2 * 3 + c

2.8 = 0.6 + c

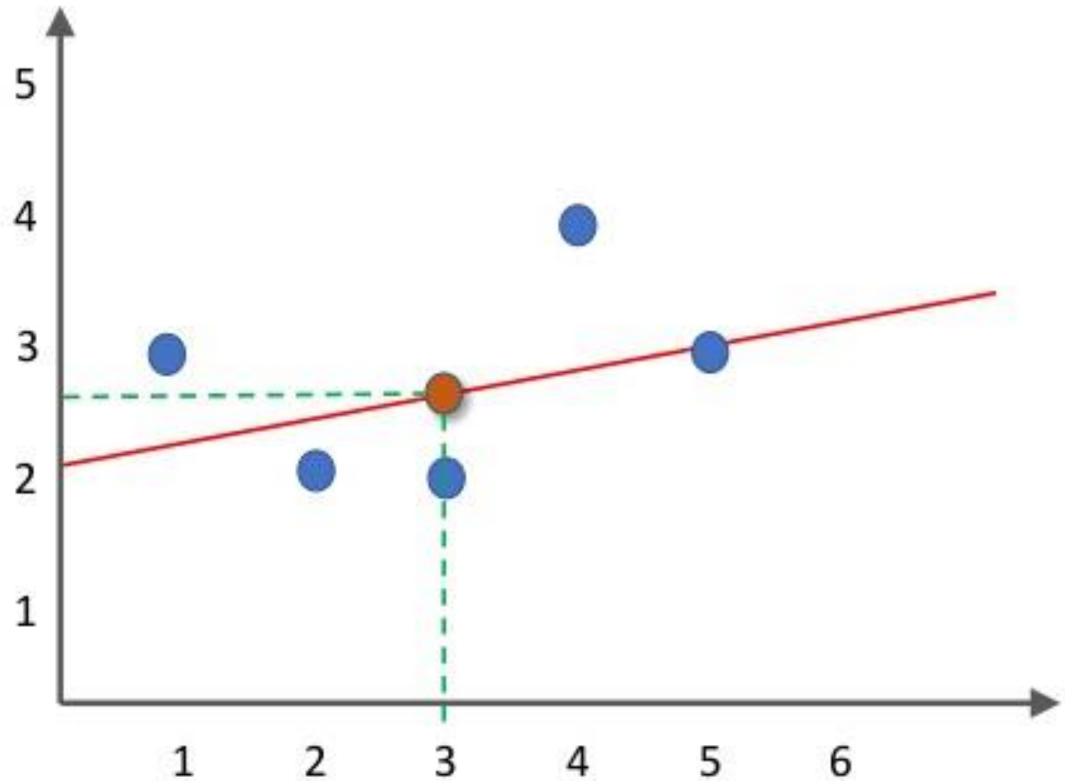
c = 2.8 - 0.6

c = 2.2

Linear Regression

Hence this is our regression line!

y = (0.2 *x) + 2.2



Linear Regression

Now, let's predict the values of y using x = {1,2,3,4,5} and plot the points!

y = (0.2 *x) + 2.2

$y_p = (0.2 * 1) + 2.2 = 2.4$
$y_p = (0.2 * 2) + 2.2 = 2.6$
$y_p = (0.2 * 3) + 2.2 = 2.8$
$y_p = (0.2 * 4) + 2.2 = 3.0$
$y_p = (0.2 * 5) + 2.2 = 3.2$

y_p = Predicted values of y

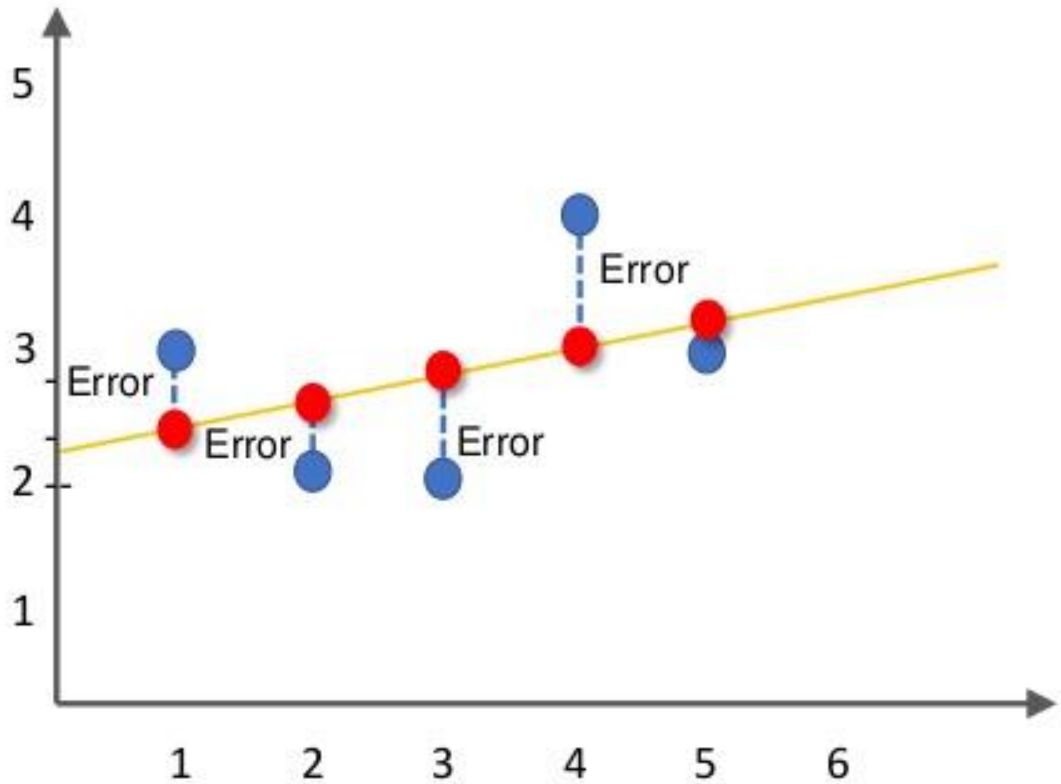
Linear Regression

Plot the predicted values along with the actual values to see the difference

x	y	y_p
1	3	2.4
2	2	2.6
3	2	2.8
4	4	3
5	3	3.2

Linear Regression

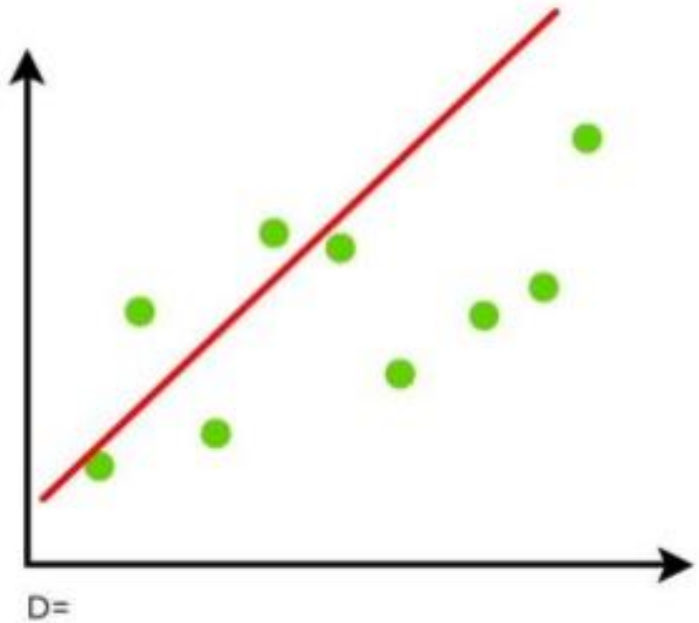
So, our goal is to reduce this error!



Linear Regression

Minimizing the Distance: There are lots of ways to minimize the distance between the line and the data points

like Sum of Squared errors, Sum of Absolute errors, Root Mean Square error etc.

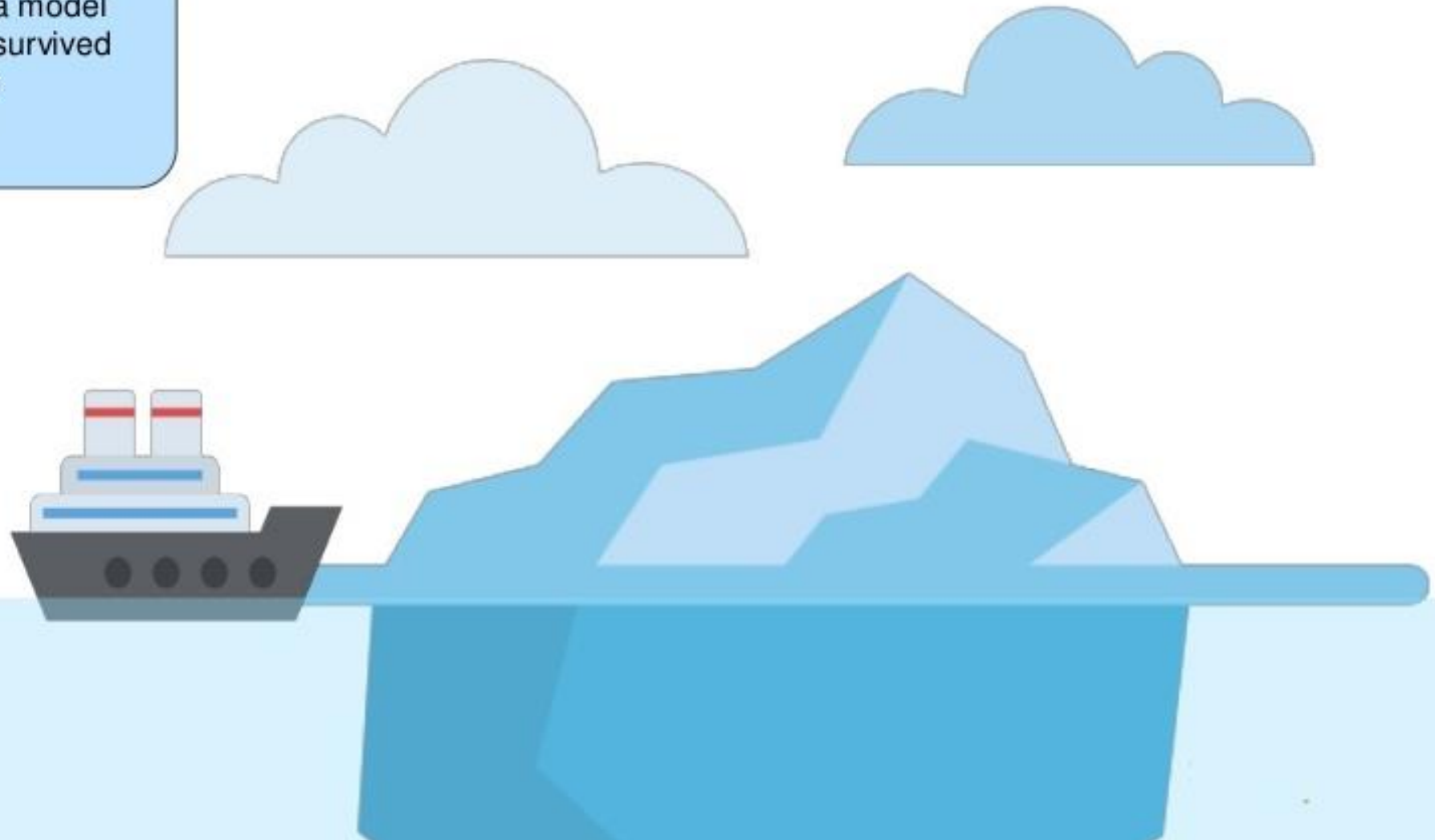


We keep moving this line through the data points to make sure the best fit line has the least square distance between the data points and the regression line

What is Logistic Regression?

Surviving the Titanic

Suppose, you have to build a model to predict how many people survived the Titanic shipwreck



Surviving the Titanic

Suppose, you have to build a model to predict how many people survived the Titanic shipwreck

Surviving the Titanic

Teaching the model with the passenger dataset

Dropping the non-essential components of the dataset

Determining the survival of passengers and evaluating the model

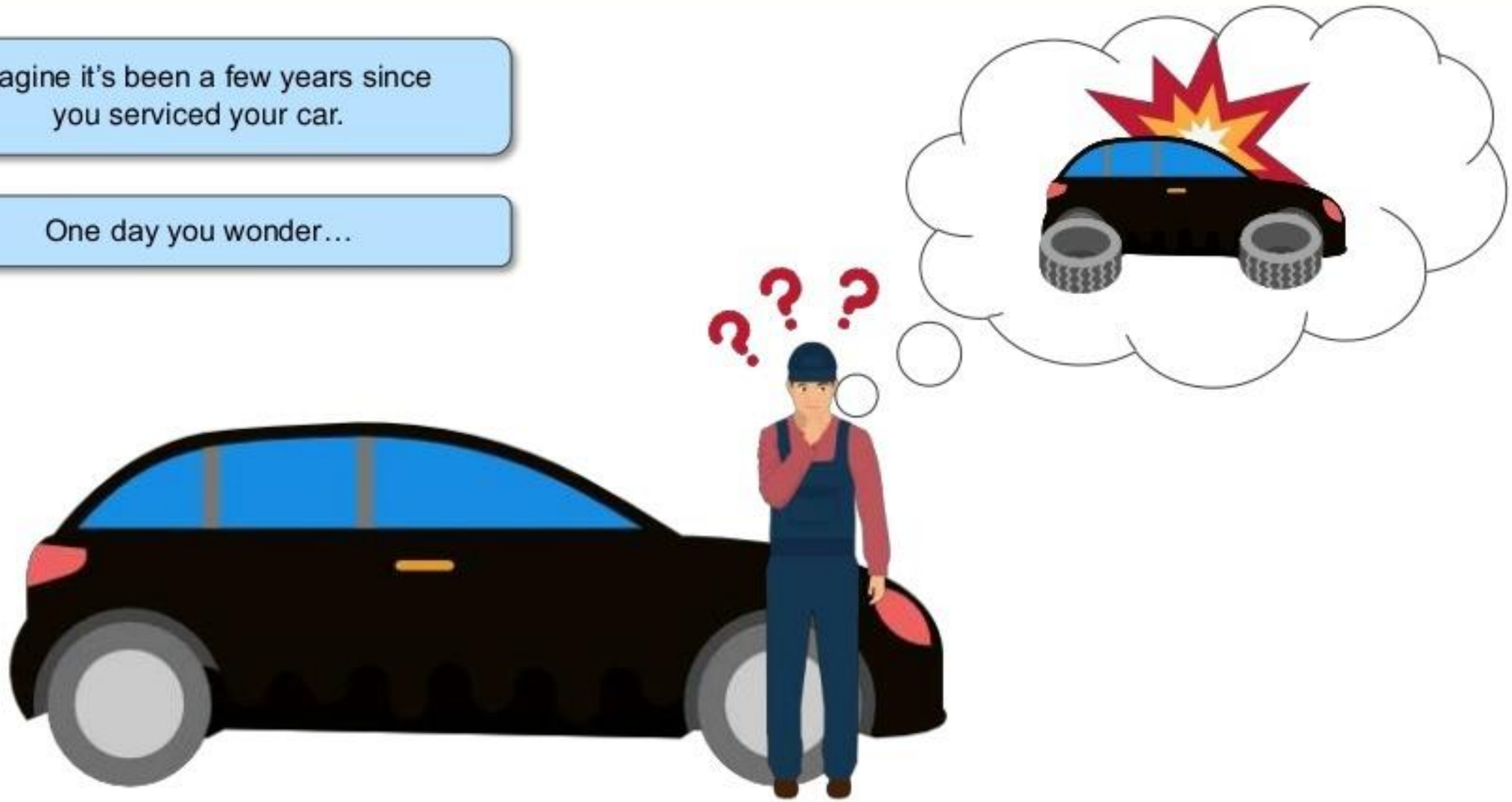
Agenda

- What is Supervised Learning?
- What is Classification? What are some of its solutions?
- What is Logistic Regression?
- Comparing Linear and Logistic Regression
- Logistic Regression applications
- Use Case – Predicting the number in an image

What is Logistic Regression?

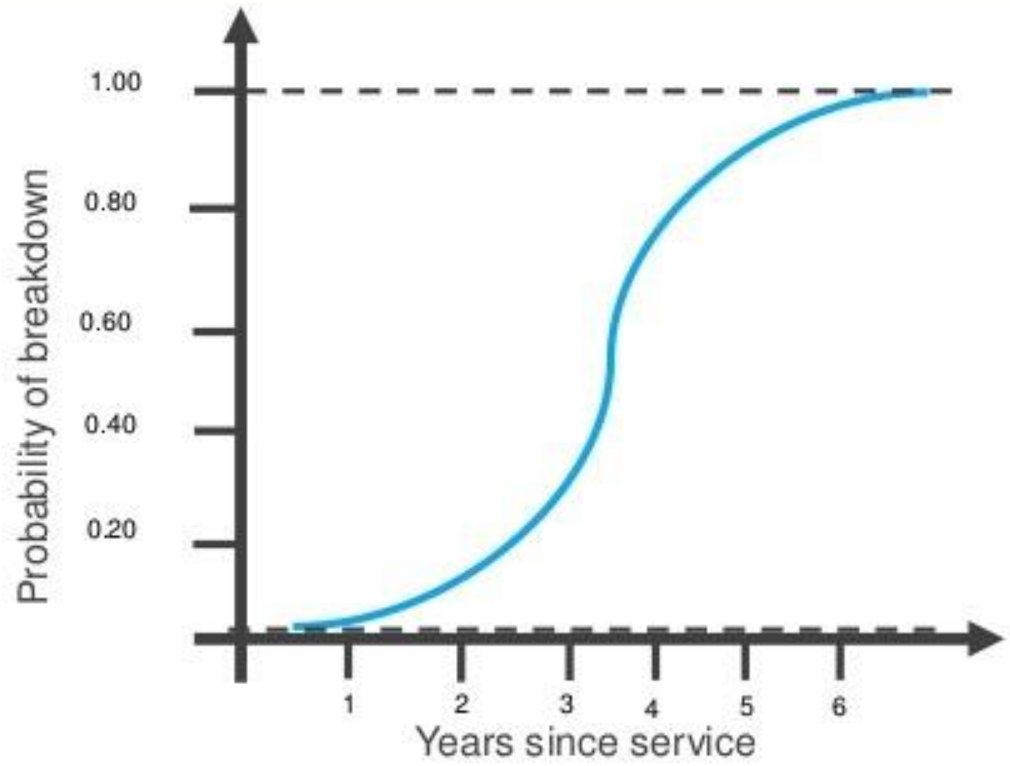
Imagine it's been a few years since you serviced your car.

One day you wonder…



What is Logistic Regression?

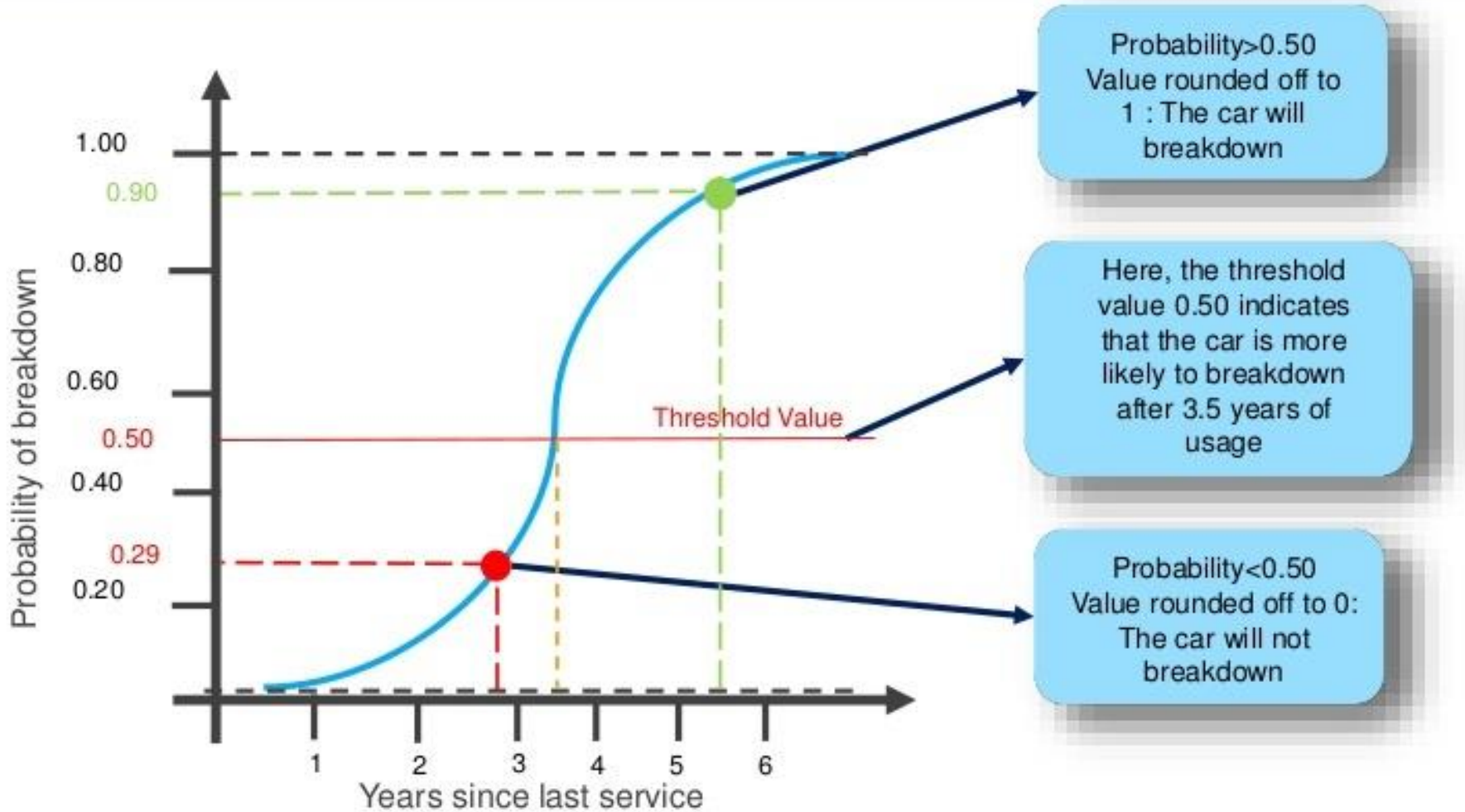
You provide years since last service



Regression model created based on other users' experience

It is a classification algorithm, used to predict binary outcomes for a given set of independent variables. The dependent variable's outcome is discrete.

What is Logistic Regression?



Model makes predictions

Logistic Regression

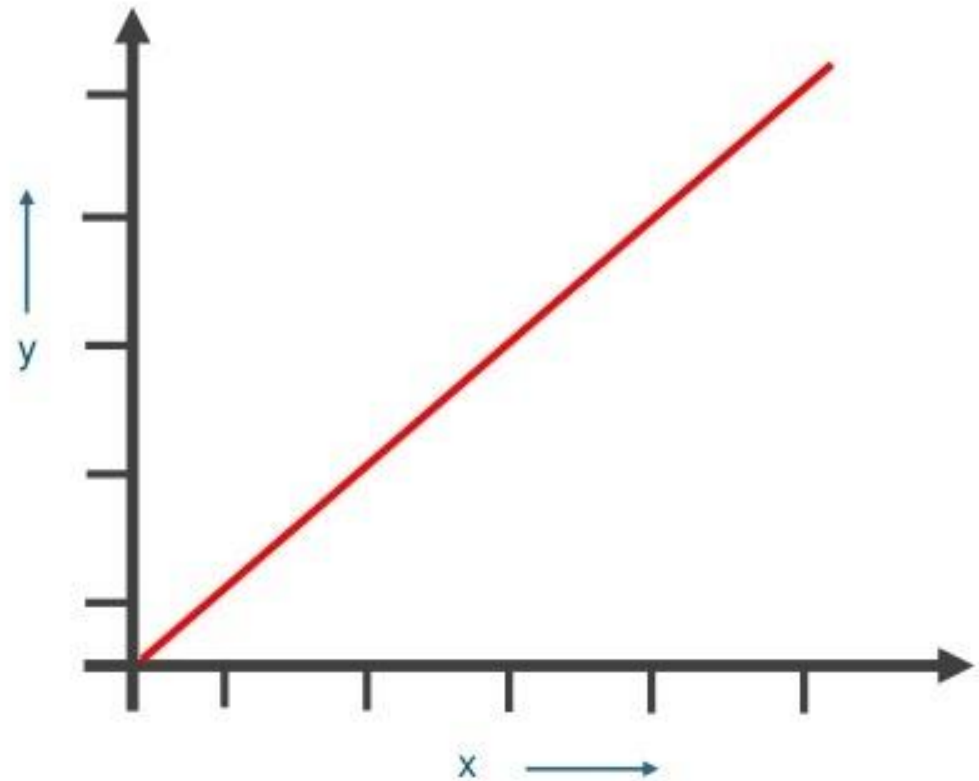
- It is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable.
- In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).
- In other words, the logistic regression model predicts P(Y=1) as a function of X.

Logistic Regression Assumptions

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little or no multicollinearity.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

Linear and Logistic Regression

Here's the graph of how linear regression would be, for a given scenario



Linear and Logistic Regression

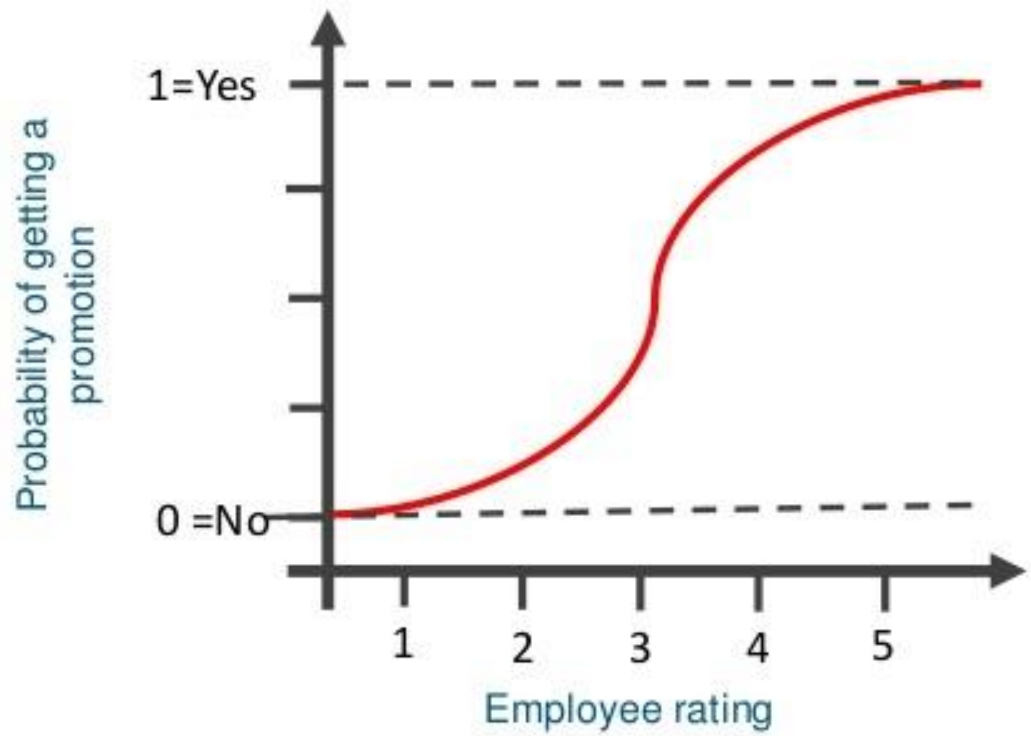
What if you wanted to know whether the employee would get a promotion or not based on their rating

Linear and Logistic Regression

This graph would not be able to make such a prediction. So we clip the line at 0 and 1.

Linear and Logistic Regression

So, how did this…

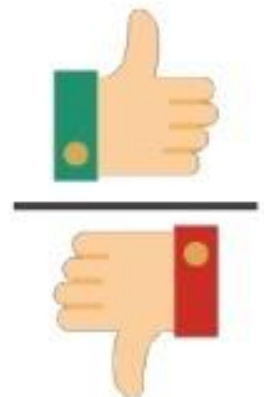


…become this?

The Math behind Logistic Regression

To understand Logistic Regression, let's talk about the odds of success

Odds (θ) =

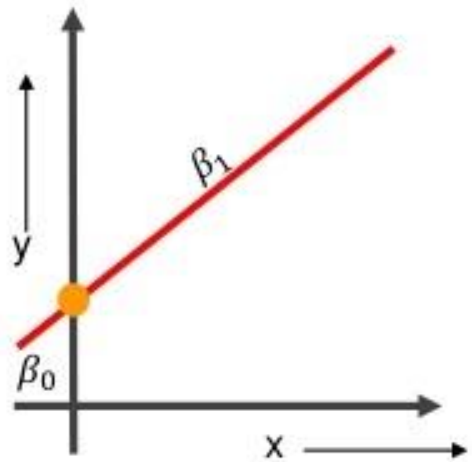


$$\frac{\text{Probability of an event happening}}{\text{Probability of an event not happening}} \quad or \quad \theta = \frac{p}{1-p}$$

The values of odds range from 0 to ∞
The values of probability change from 0 to 1

The Math behind Logistic Regression

Take the equation of the straight line



Here, β_0 is the y-intercept
β_1 is the slope of the line
x is the value of the x co-ordinate
y is the value of the prediction

The equation would be: $y = \beta_0 + \beta_1 x$

The Math behind Logistic Regression

Now, we predict the odds of success

$$\log\left(\frac{p(x)}{1-P(x)}\right) = \beta_0 + \beta_1 x$$

Exponentiating both sides:

$$e^{ln}\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

$$\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

Let Y = $e^{\beta_0+\beta_1 x}$

Then $\frac{p(x)}{1-p(x)} = Y$

$$p(x) = Y(1-p(x))$$
$$p(x) = Y - Y(p(x))$$
$$p(x) + Y(p(x)) = Y$$
$$p(x)(1+Y) = Y$$
$$p(x) = \frac{Y}{1+Y}$$
$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

The equation of a sigmoid function:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

The Math behind Logistic Regression

A sigmoid curve is obtained!

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems

Logistic Regression

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems

Logistic Regression

- Used to solve Classification Problems

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature

Logistic Regression

- Used to solve Classification Problems

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature

Logistic Regression

- Used to solve Classification Problems
- The response variable is categorical in nature

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature
- It helps estimate the dependent variable when there is a change in the independent variable.

Logistic Regression

- Used to solve Classification Problems
- The response variable is categorical in nature

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature
- It helps estimate the dependent variable when there is a change in the independent variable.

Logistic Regression

- Used to solve Classification Problems
- The response variable is categorical in nature
- It helps calculate the possibility of a particular event taking place.

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature
- It helps estimate the dependent variable when there is a change in the independent variable.

- Is a straight line.

Logistic Regression

- Used to solve Classification Problems
- The response variable is categorical in nature
- It helps calculate the possibility of a particular event taking place.

How is Linear and Logistic Regression different?

Linear Regression

- Used to solve Regression Problems
- The response variables are continuous in nature
- It helps estimate the dependent variable when there is a change in the independent variable.

- Is a straight line.

Logistic Regression

- Used to solve Classification Problems
- The response variable is categorical in nature
- It helps calculate the possibility of a particular event taking place.

- An S-curve. (S = Sigmoid)

Logistic Regression Applications

Weather Prediction

Helps determine the kind of weather that can be expected

Logistic Regression Applications

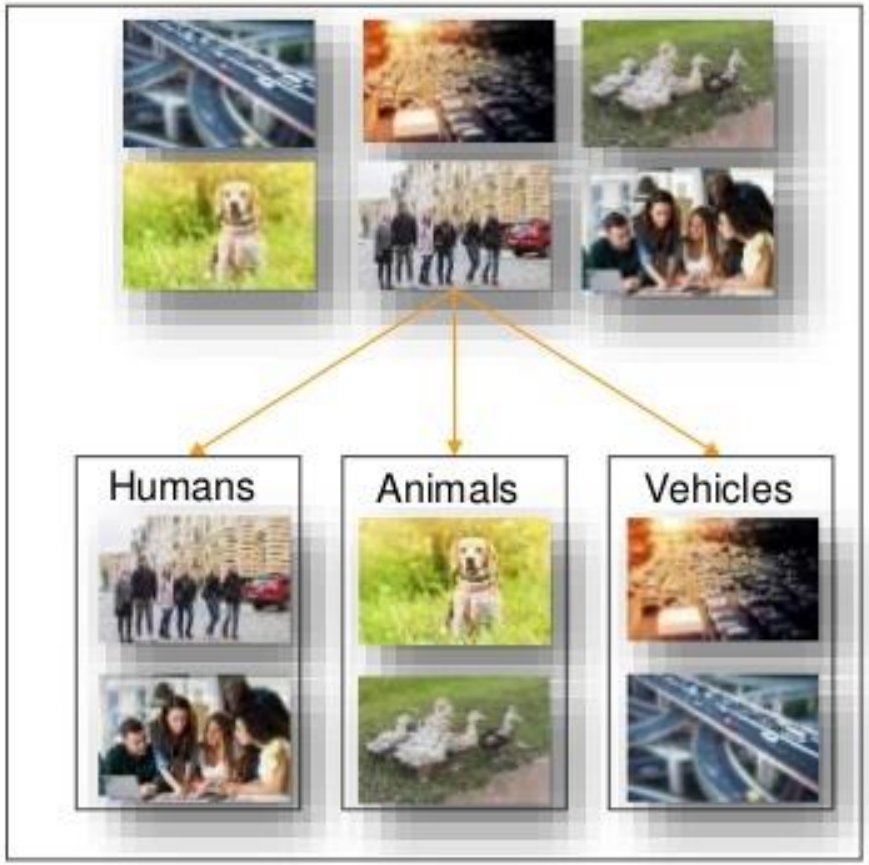
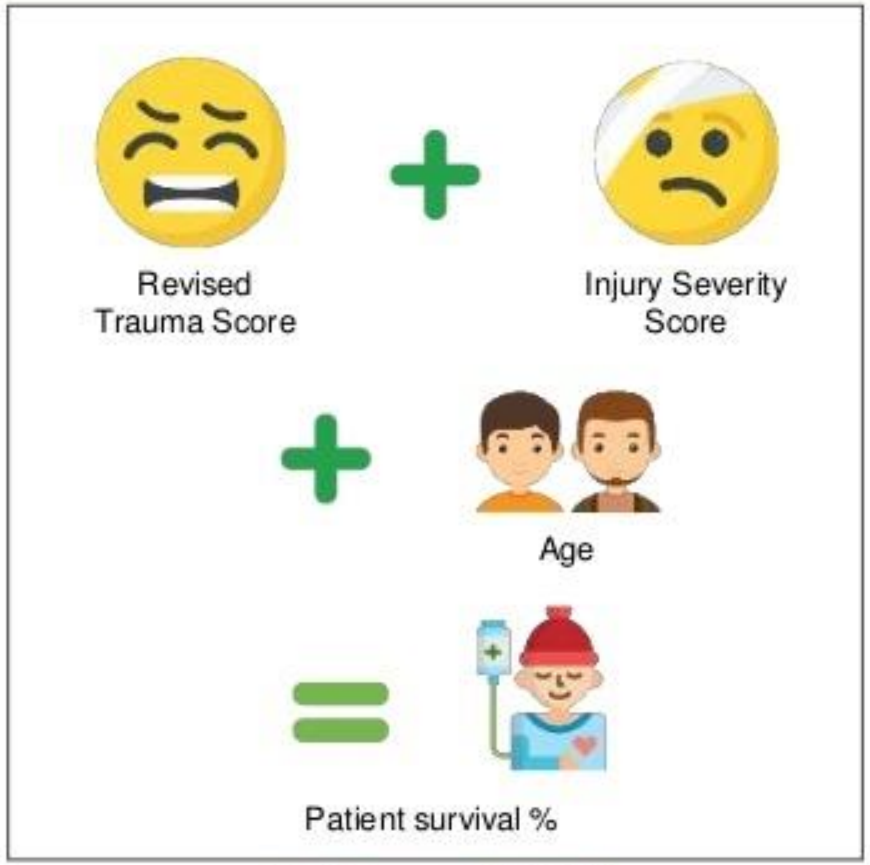


Image Categorization

Identifies the different components that are present in the image, and helps categorize them

Logistic Regression Applications



Healthcare (TRISS)

Determines the possibility of patient survival, taking age, ISS and RTS into consideration

Decision Trees

Decision Tree is a tree shaped algorithm used to determine a course of action

Each branch of the tree represents a possible decision, occurrence or reaction

Decision Trees

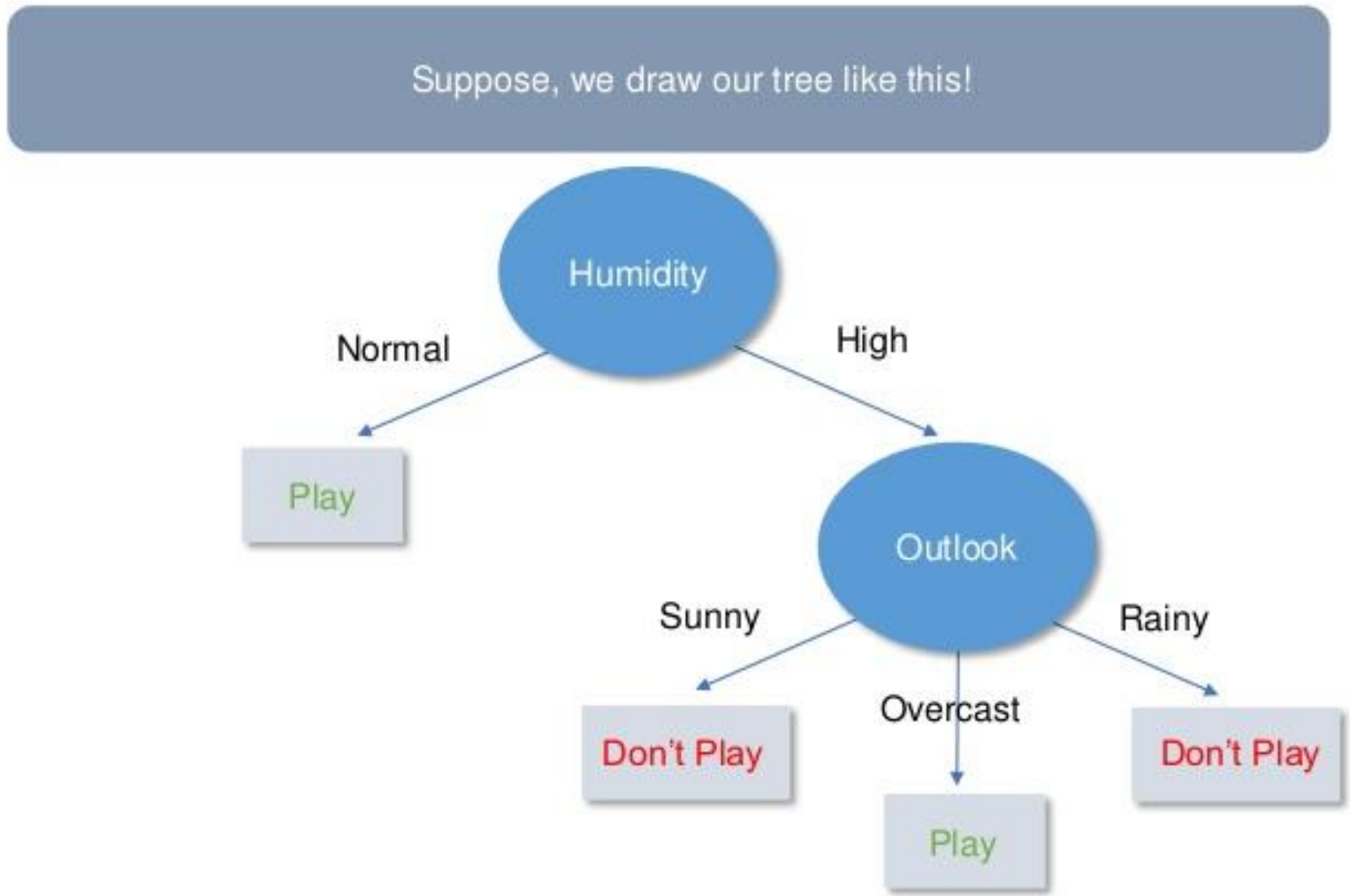
We have a data which tells us if it is a good day to play golf!

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Decision Trees

Let's determine if you should play golf when the day is sunny and windy?

Decision Trees



Decision Trees

But, is this the right decision tree?
For that, we should calculate Entropy and Information Gain!

Entropy

Entropy is the measure of randomness or *'impurity'* in the dataset

Entropy should be low!

Information Gain

It is the measure of decrease in entropy after the dataset is split

Also known as Entropy Reduction

Information Gain should be high!

Decision Trees

Let's look at entropy!

$$Entropy = I(p,n) = -\frac{p}{p+n} \times log_2(\frac{p}{p+n}) - \frac{n}{p+n} \times log_2(\frac{n}{p+n})$$

Better quality image
will be replaced

Decision Trees

Let's look at entropy!

a) Entropy of target class of the dataset (whole entropy):

Entropy (Play golf)

= E(5,9)

= I(5/14, 9/14)

= I(0.36, 0.64)

= $-(0.36 \log_2 0.36) - (0.64 \log_2 0.64)$

= 0.94

Play Golf	
Yes	No
9	5

Total = 14

Decision Trees

Let's look at entropy!

Entropy (Play golf, Outlook)

$$= P(\text{sunny}) * E(3,2) + P(\text{Overcast}) * E(4,0) + P(\text{rainy}) * E(2,3)$$

$$= 5/14 * I(3,2) + 4/14 * I(4,0) + 5/14 * I(2,3)$$

$$= 0.693$$

Similarly, we can calculate the entropy of other predictors like Temperature, Humidity, Windy!

		Play Golf		
Predictors		Yes	No	Total
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

Decision Trees

Now, let's look at Information Gain!

Gain(Outlook) = Entropy(PlayGolf) − Entropy(PlayGolf,Outlook)

= 0.940−0.693

=0.247

The information gain of the other three attributes can be calculated in the same way:

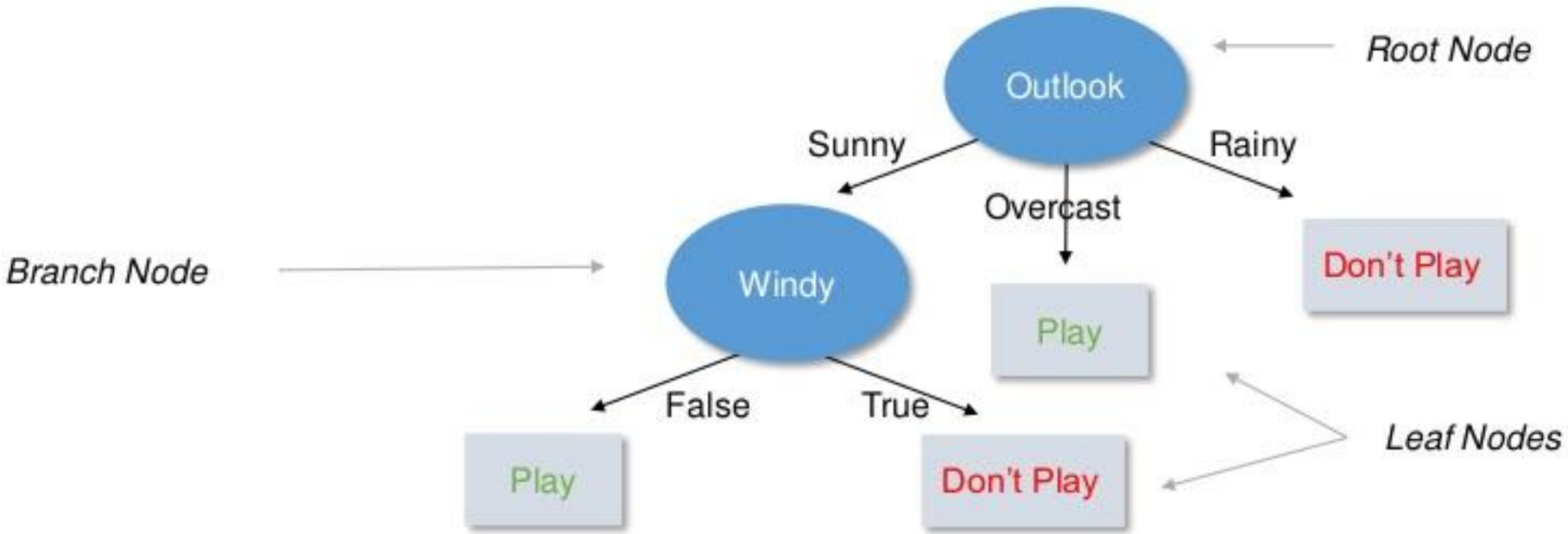
Gain(Temp) = Entropy(PlayGolf)−Entropy(PlayGolf,Temp) = 0.029

Gain(Humidity) = Entropy(PlayGolf)−Entropy(PlayGolf,Humidity) = 0.152

Gain(Windy) = Entropy(PlayGolf)−Entropy(PlayGolf,Windy) = 0.048

Decision Trees

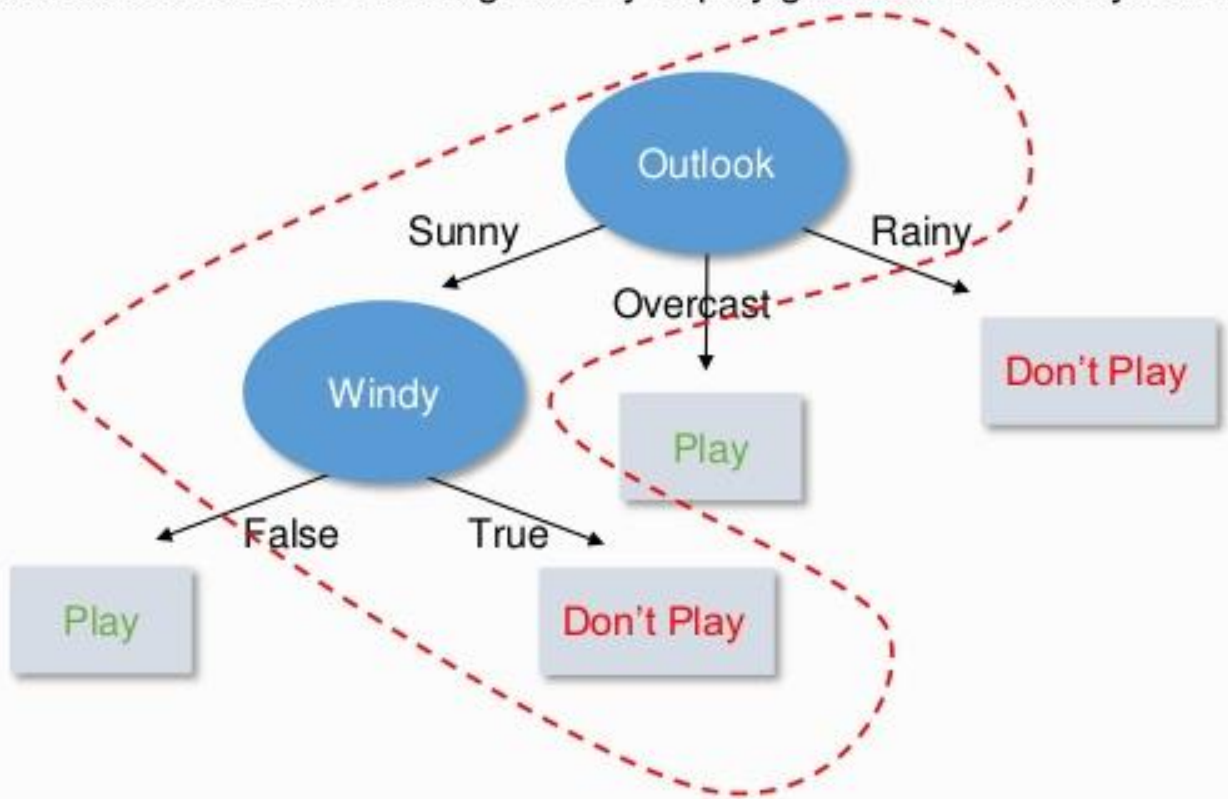
Now, let's build the decision tree!



We choose the attribute with largest information gain as the root node

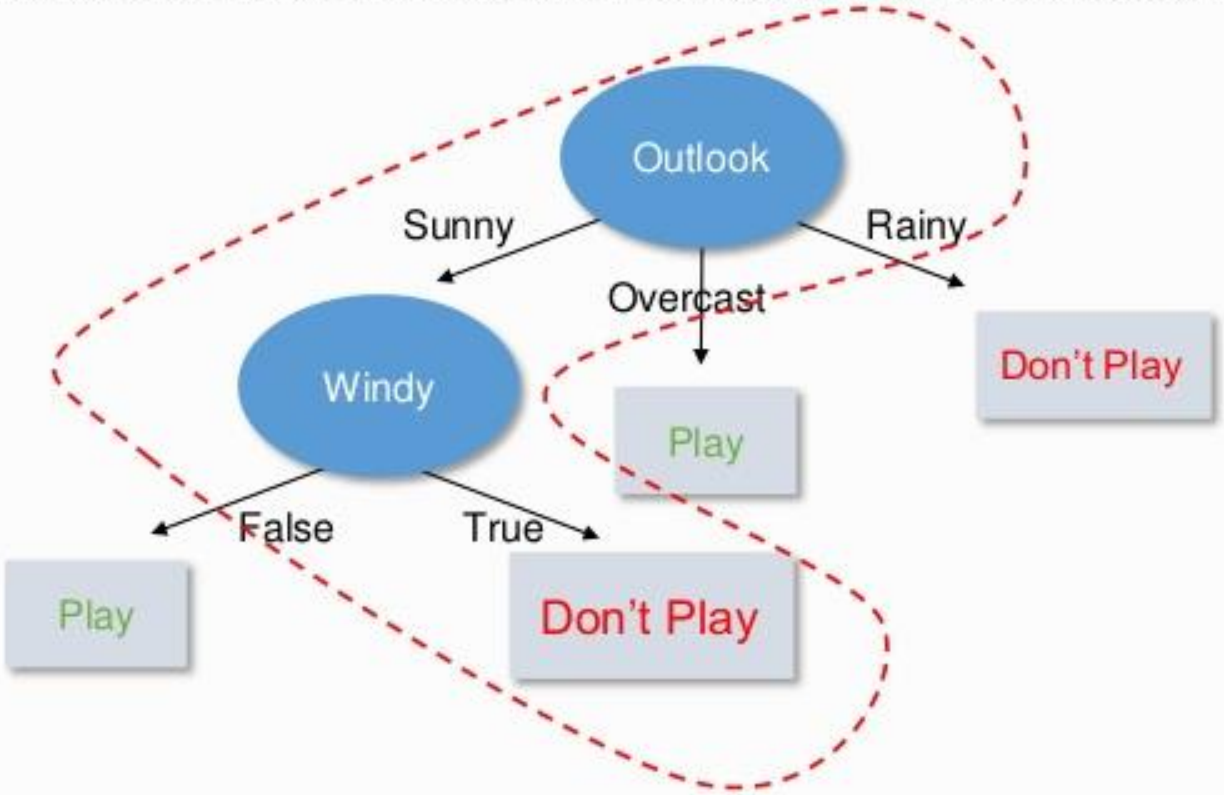
Decision Trees

So, we wanted to know if it's a good day to play golf when it's sunny and windy!



Decision Trees

So, we wanted to know if it's a good day to play golf when it's sunny and windy!



Decision Trees

Uh-Oh, it's not a good day to play golf!
You can watch a golf game at home! :D

Support Vector Machine

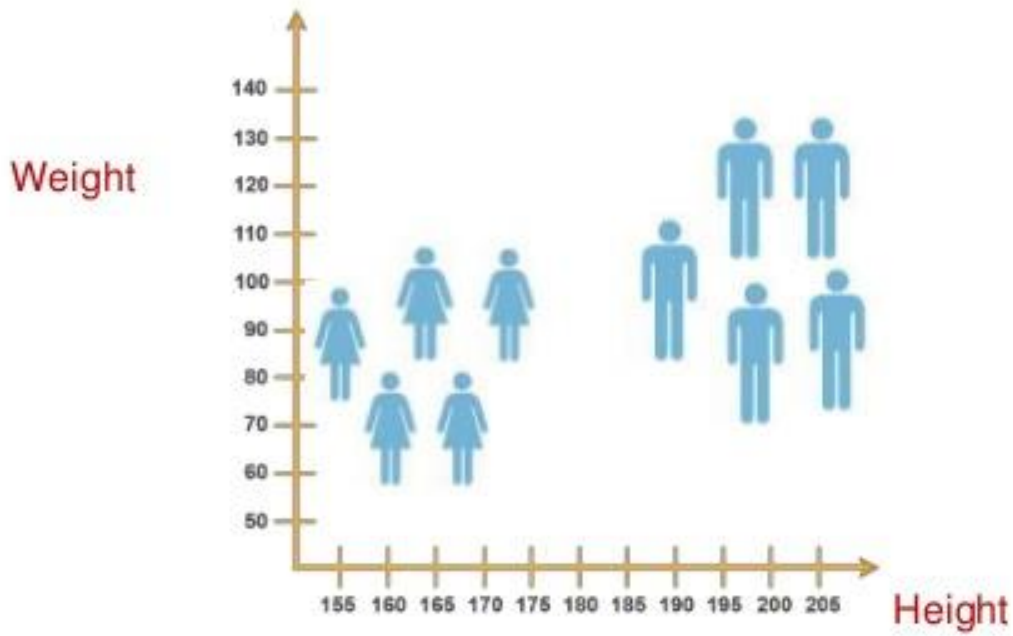
Support Vector Machine is a widely used classification algorithm!

The idea of Support Vector Machines is simple: The algorithm creates a separation line which divides the classes in the best possible manner

For example, dog or cat, disease or no disease

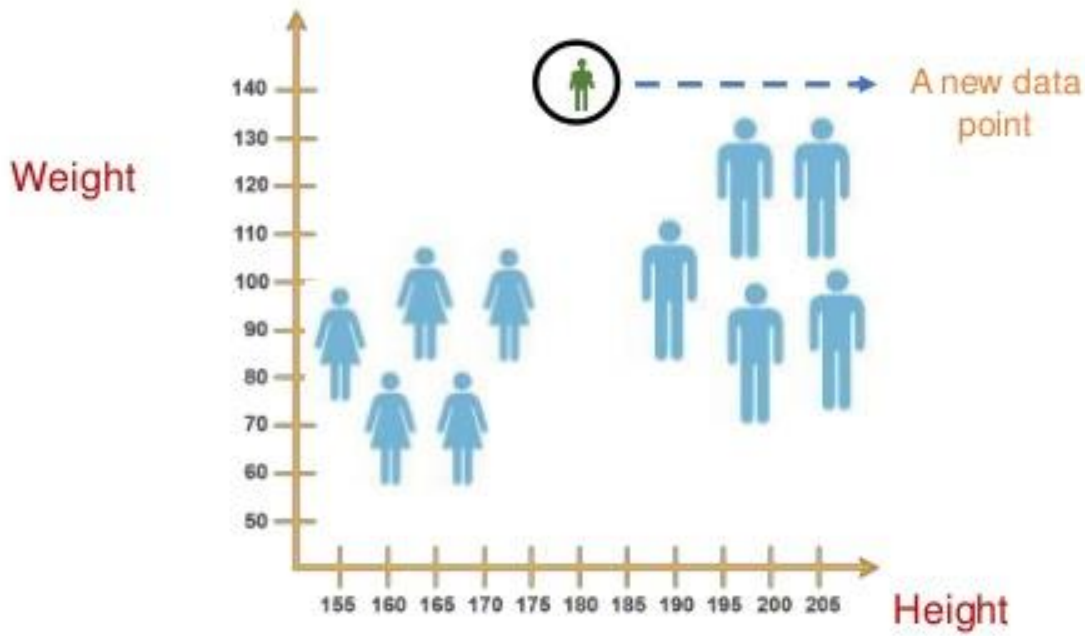
Support Vector Machine

Suppose, we have labeled sample data, which tells height and weight of males and females



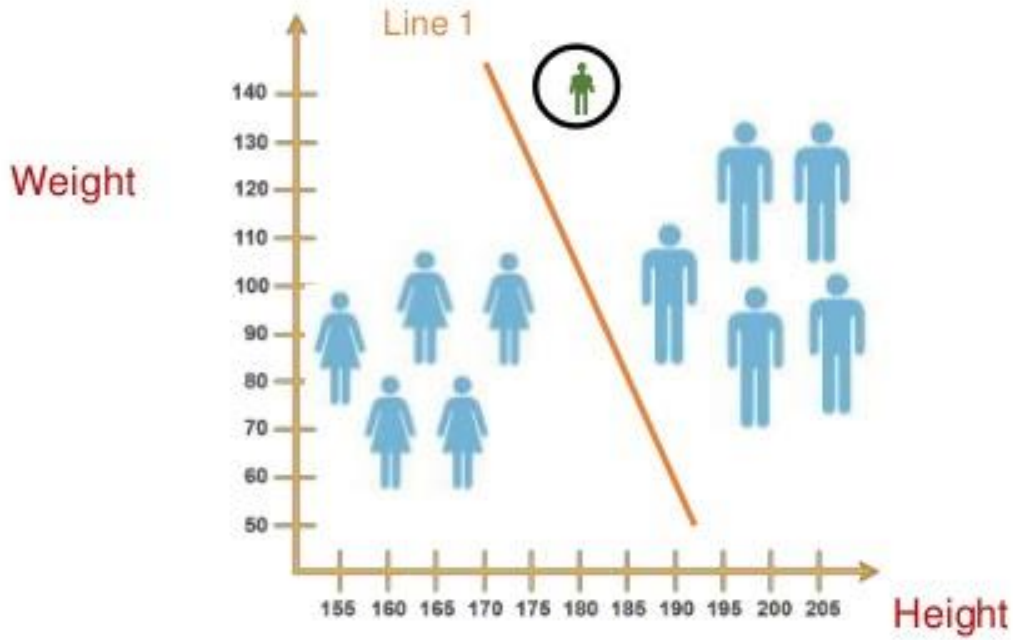
Support Vector Machine

How can a machine classify whether a new data point is a male or a female?



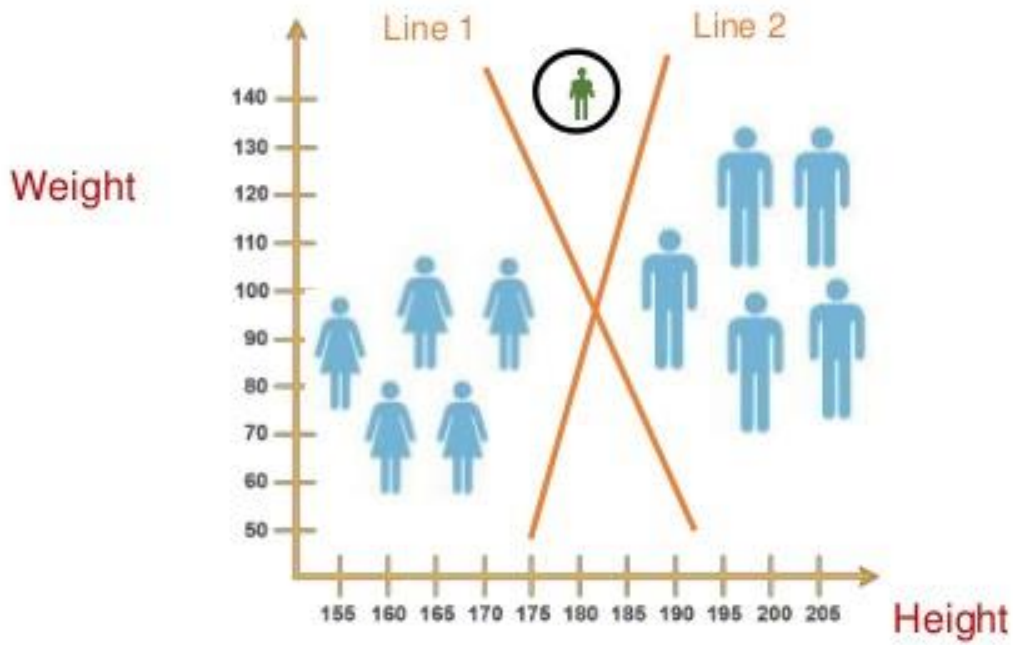
Support Vector Machine

We draw decision lines, but if we consider decision **line 1** then we will classify it as a male



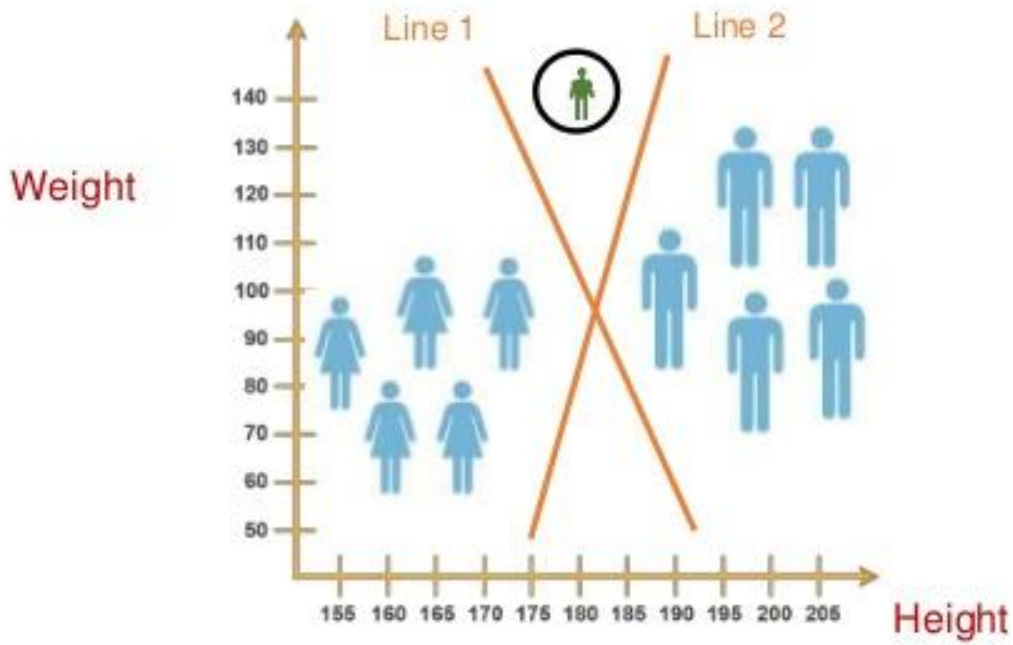
Support Vector Machine

And if we consider decision **line 2**, then it will be a female!



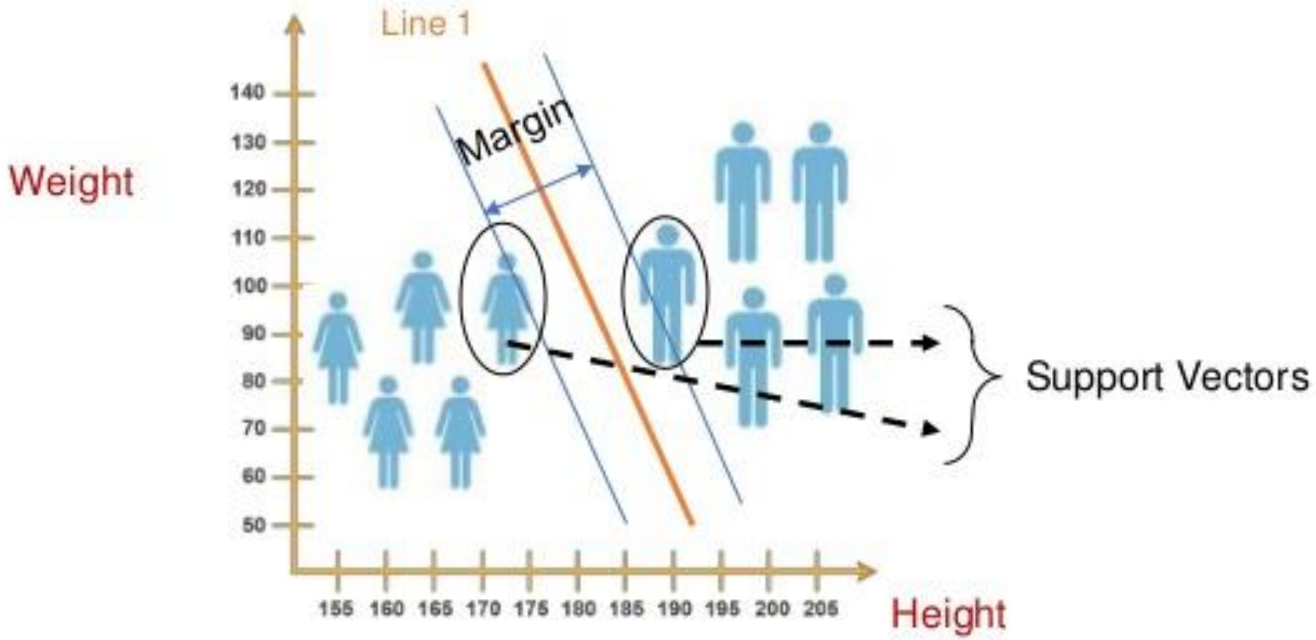
Support Vector Machine

We need to know which line divides the classes correctly, but how?



Support Vector Machine

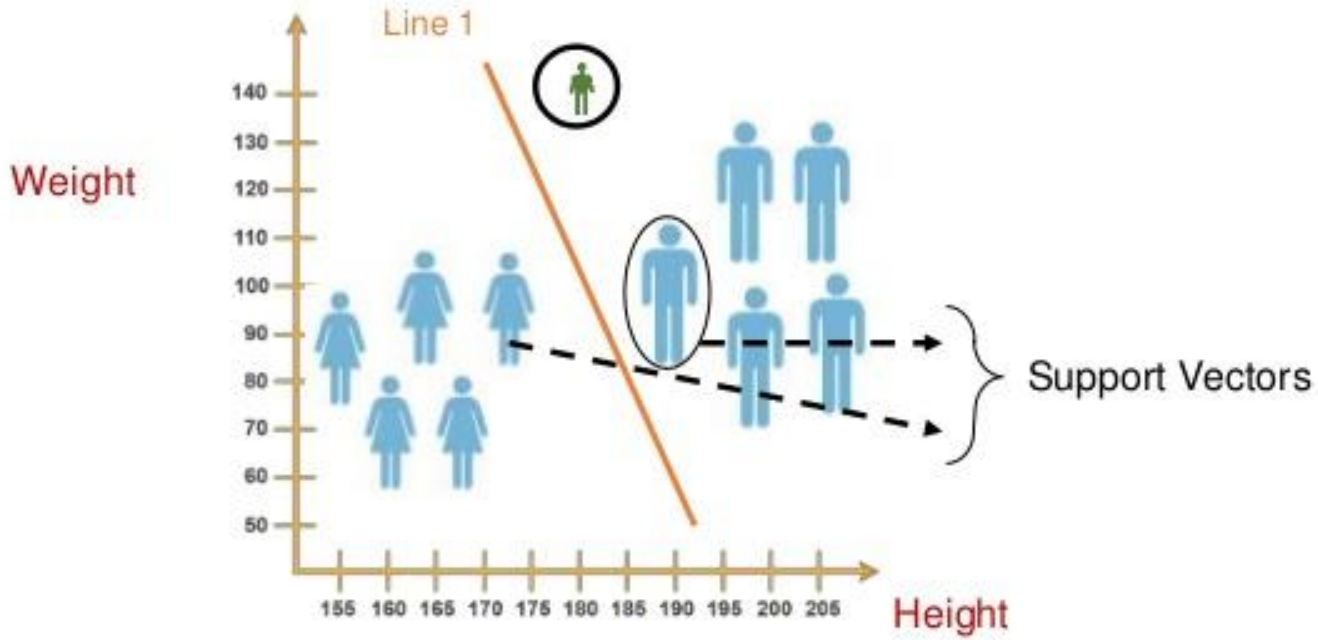
The goal is to choose a hyperplane with the greatest possible margin between the decision line and the nearest point within the training set



Distance Margin: The distance between the hyperplane and the nearest data point from either set

Support Vector Machine

When we draw the hyperplanes, we observe that Line 1 has the maximum distance margin so it will classify the new data point correctly



Result: New data point is male!

Support Vector Machine

Let's understand this with the help of an example!

Support Vector Machine

Problem Statement: Classifying muffin and cupcake recipes using support vector machines

VS

Support Vector Machine

Let's have a look at our dataset:

What's the difference between a muffin and a cupcake? Turns out muffins have more flour, while cupcakes have more butter and sugar

Type	Flour	Milk	Sugar	Butter	Egg	Baking Powder	Vanilla	Salt
Muffin	55	28	3	7	5	2	0	0
Muffin	47	24	12	6	9	1	0	0
Muffin	47	23	18	6	4	1	0	0
Muffin	45	11	17	17	8	1	0	0
Muffin	50	25	12	6	5	2	1	0
Muffin	55	27	3	7	5	2	1	0
Muffin	54	27	7	5	5	2	0	0
Muffin	47	26	10	10	4	1	0	0
Muffin	50	17	17	8	6	1	0	0
Muffin	50	17	17	11	4	1	0	0
Cupcake	39	0	26	19	14	1	1	0
Cupcake	42	21	16	10	8	3	0	0
Cupcake	34	17	20	20	5	2	1	0
Cupcake	39	13	17	19	10	1	1	0
Cupcake	38	15	23	15	8	0	1	0
Cupcake	42	18	25	9	5	1	0	0
Cupcake	36	14	21	14	11	2	1	0
Cupcake	38	15	31	8	6	1	1	0
Cupcake	36	16	24	12	9	1	1	0
Cupcake	34	17	23	11	13	0	1	0

Why Support Vector Machine?

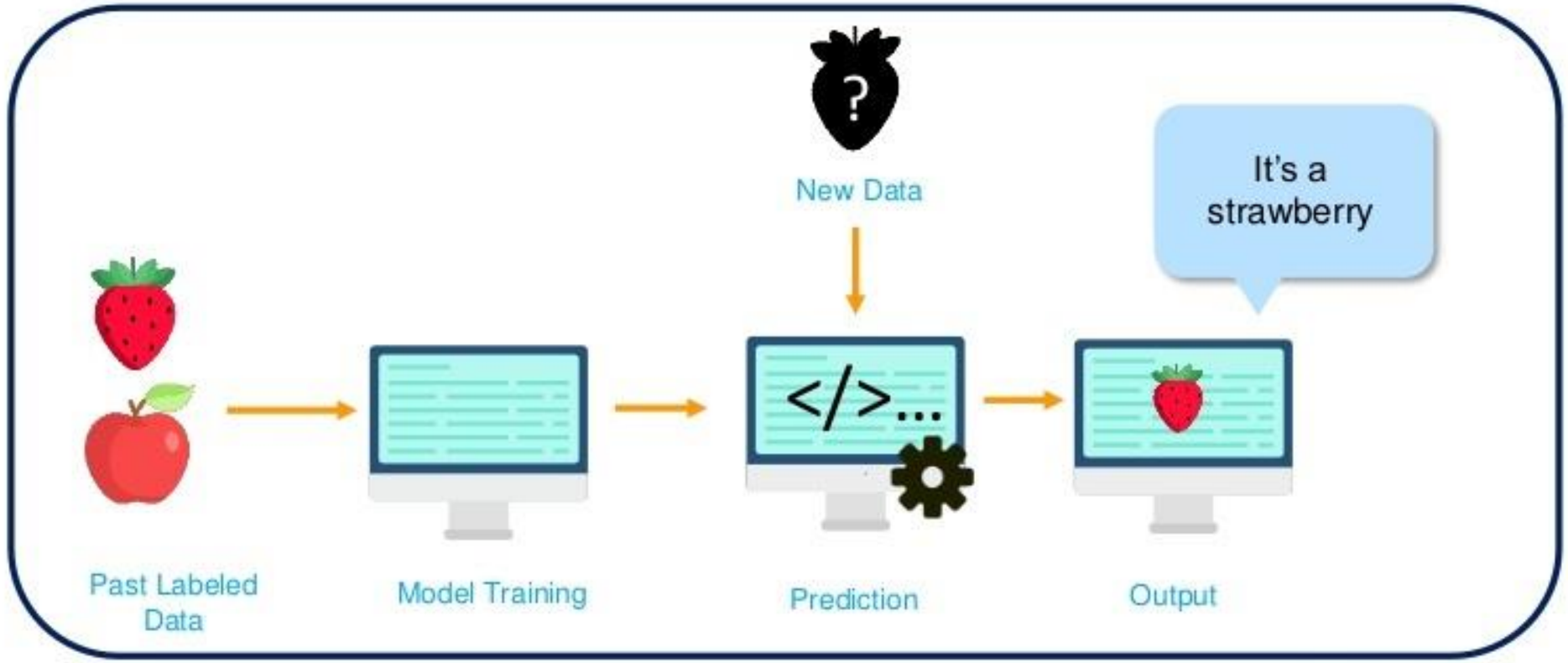
Last week, my son and I visited a fruit shop

Why Support Vector Machine?

Why Support Vector Machine?

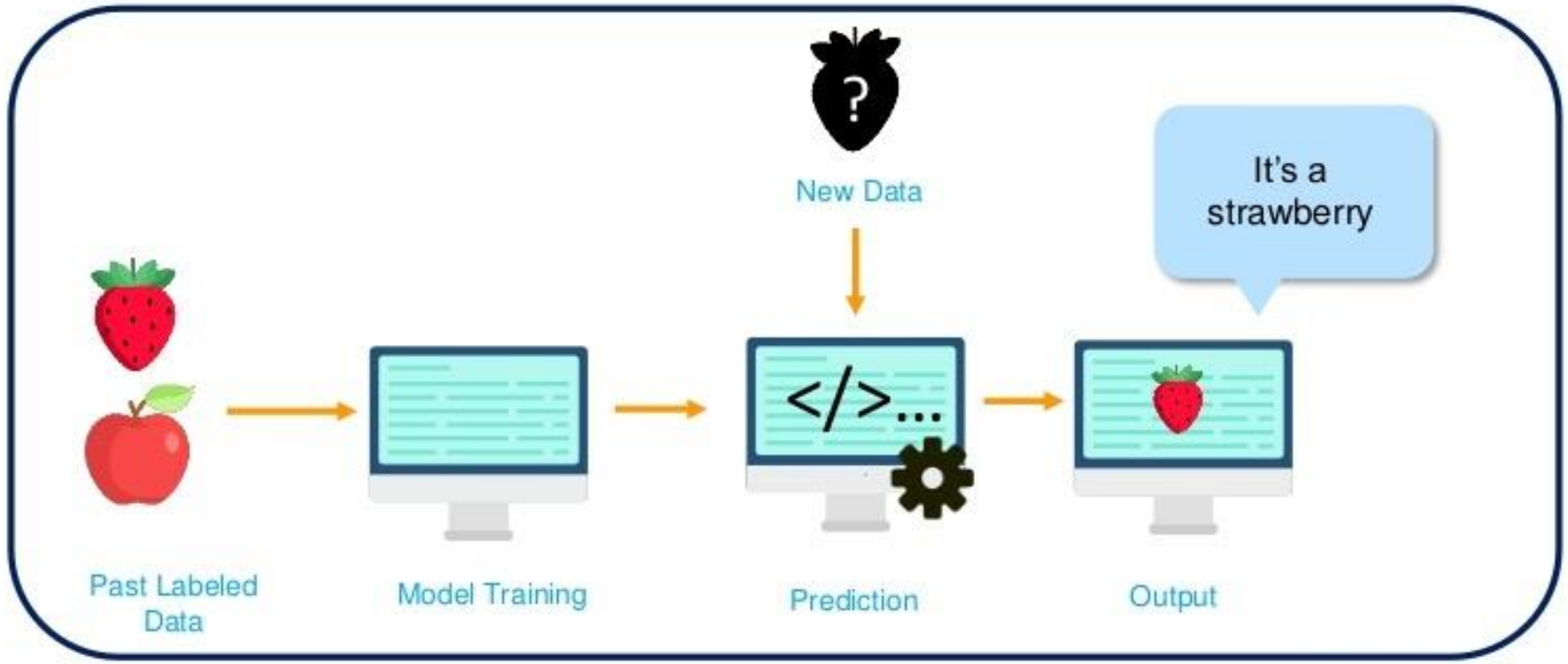
Why Support Vector Machine?

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



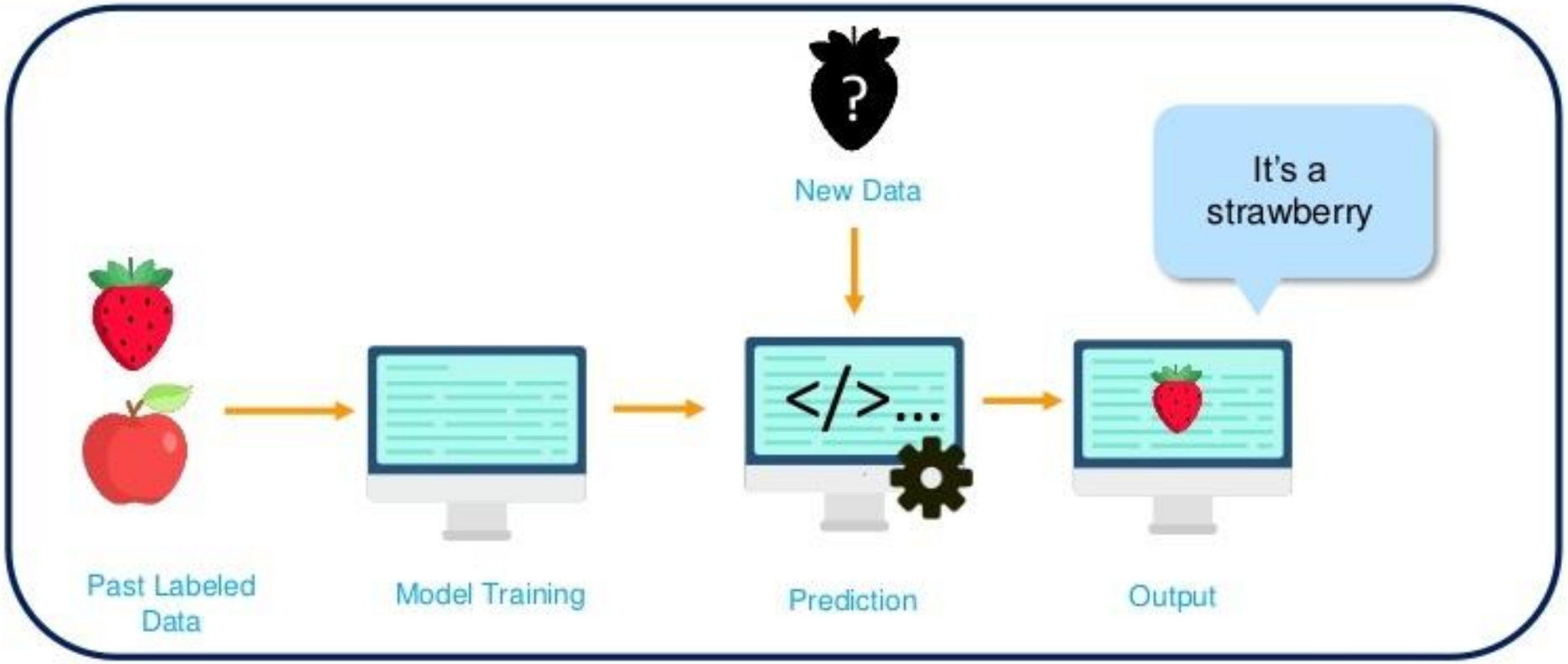
Why Support Vector Machine?

This is Support Vector Machine



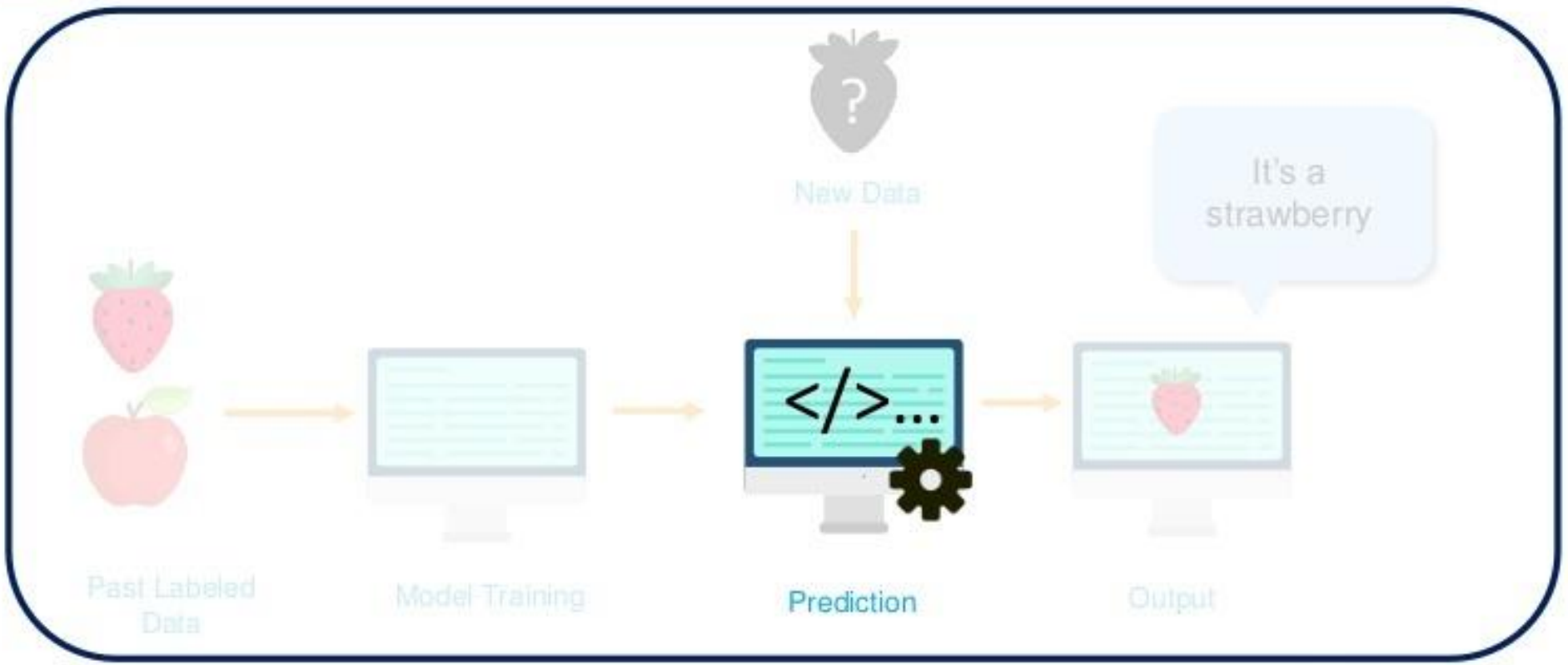
Why Support Vector Machine?

SVM is a supervised learning method that looks at data and sorts it into one of the two categories

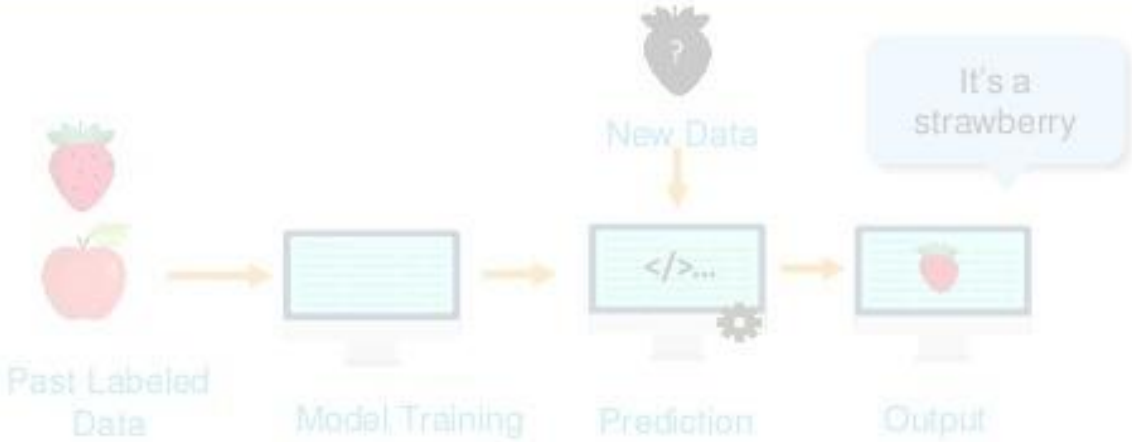


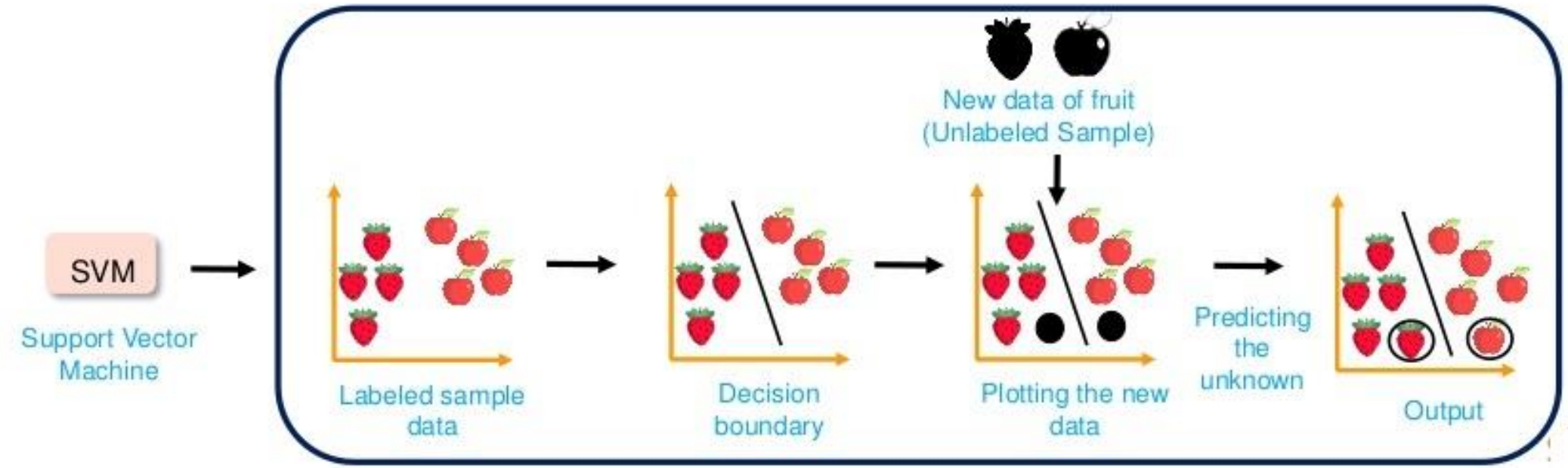
Why Support Vector Machine?

But how does prediction work?

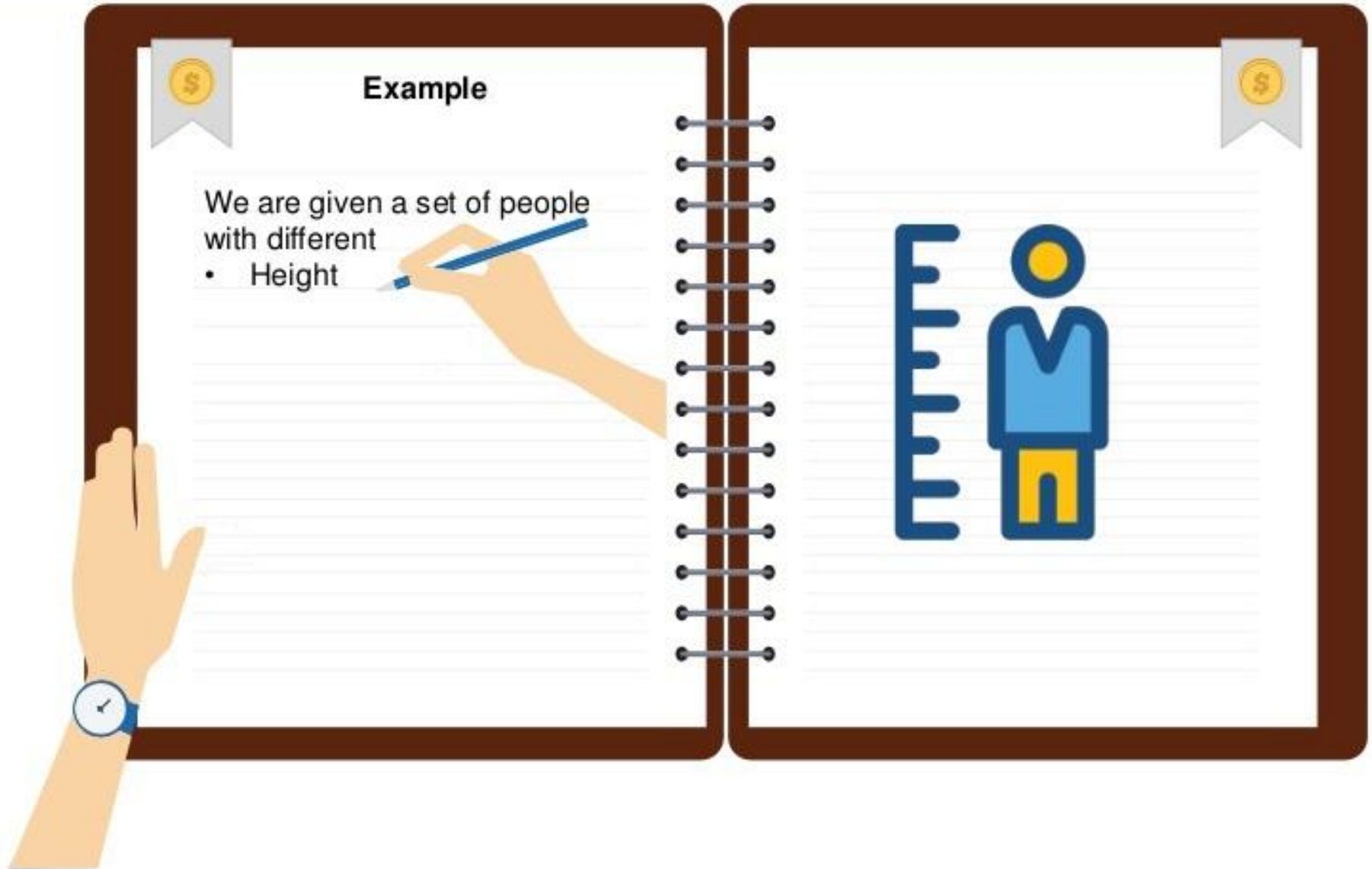


Why Support Vector Machine?

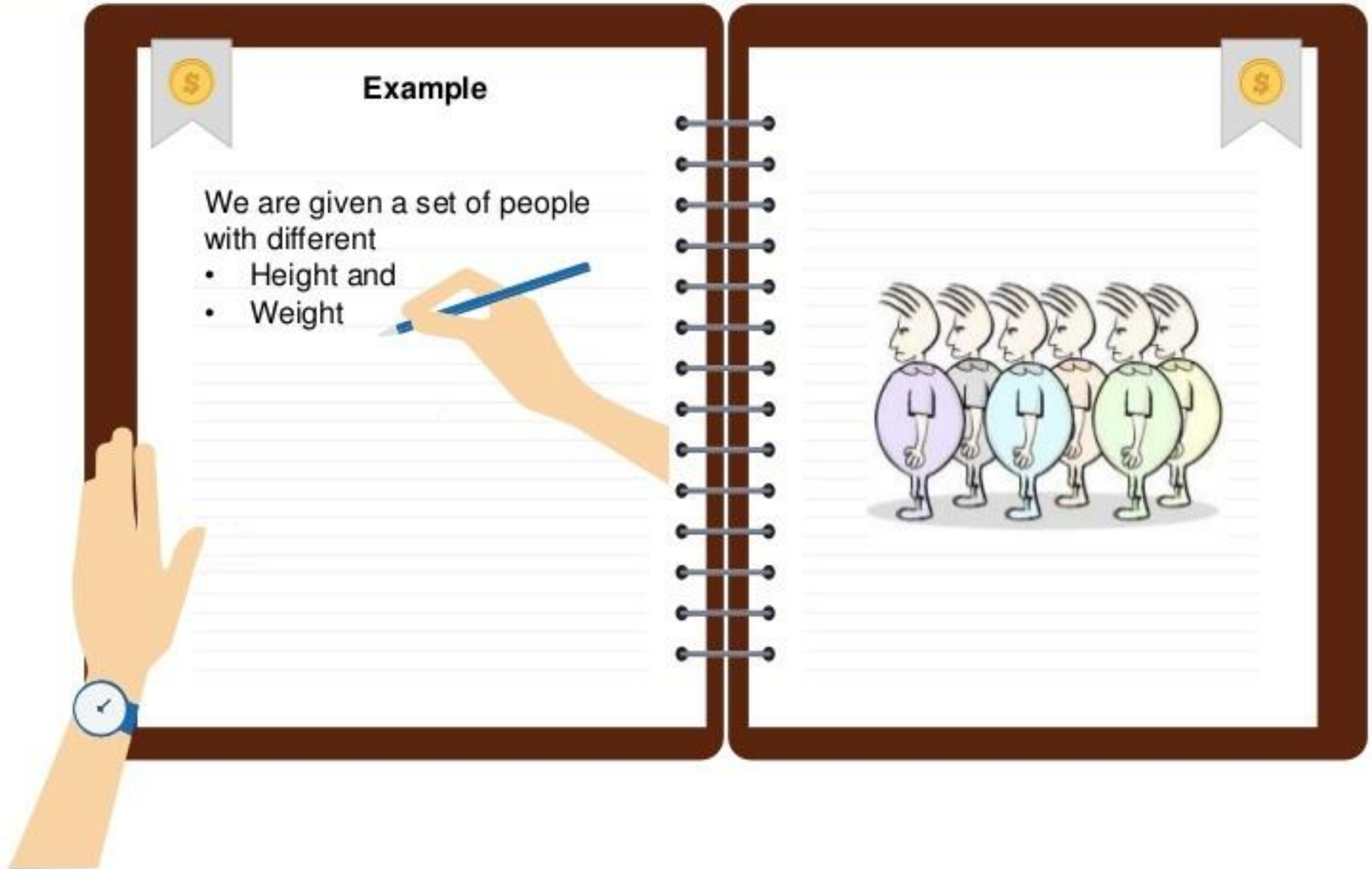




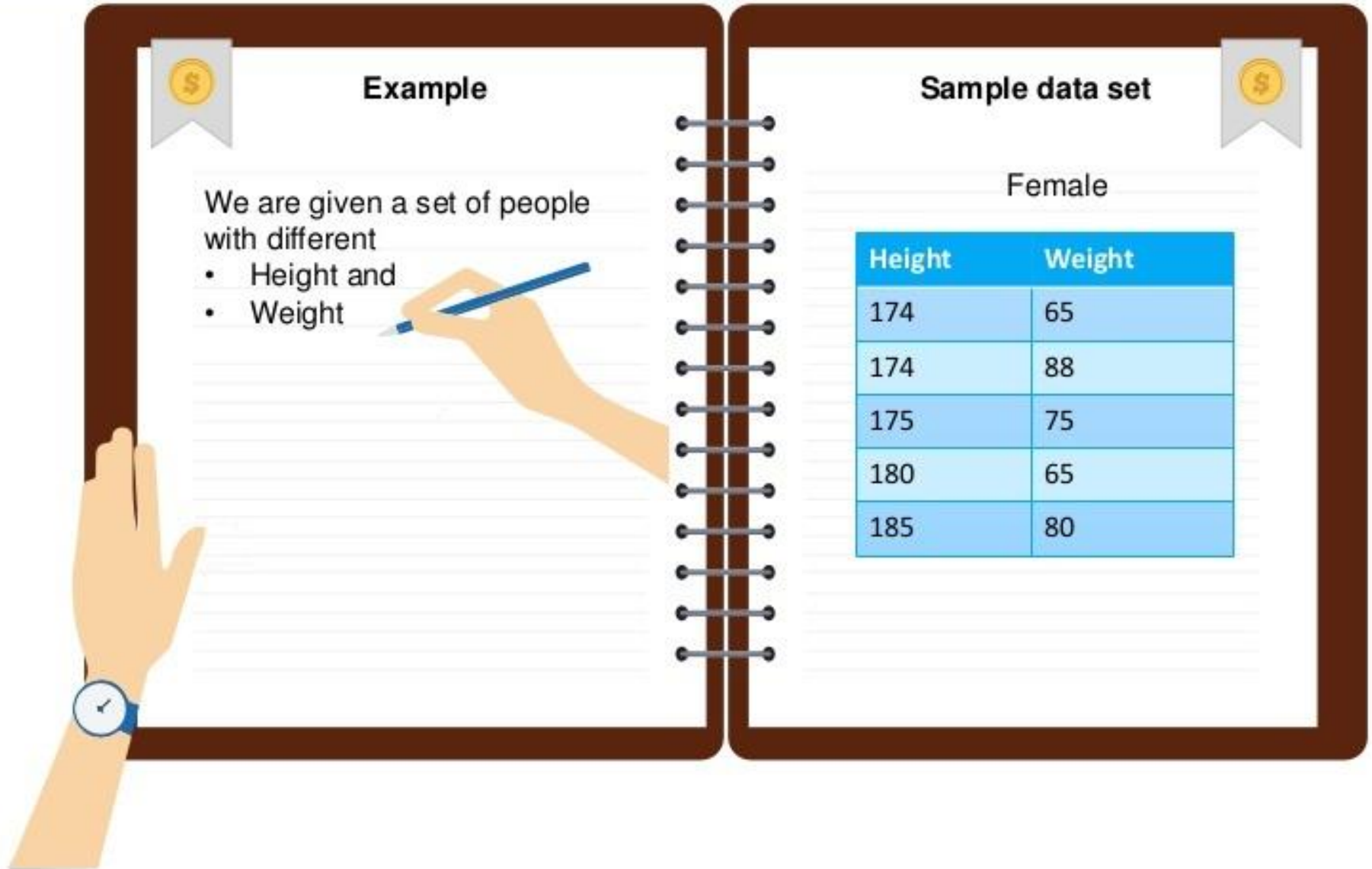
What is Support Vector Machine?



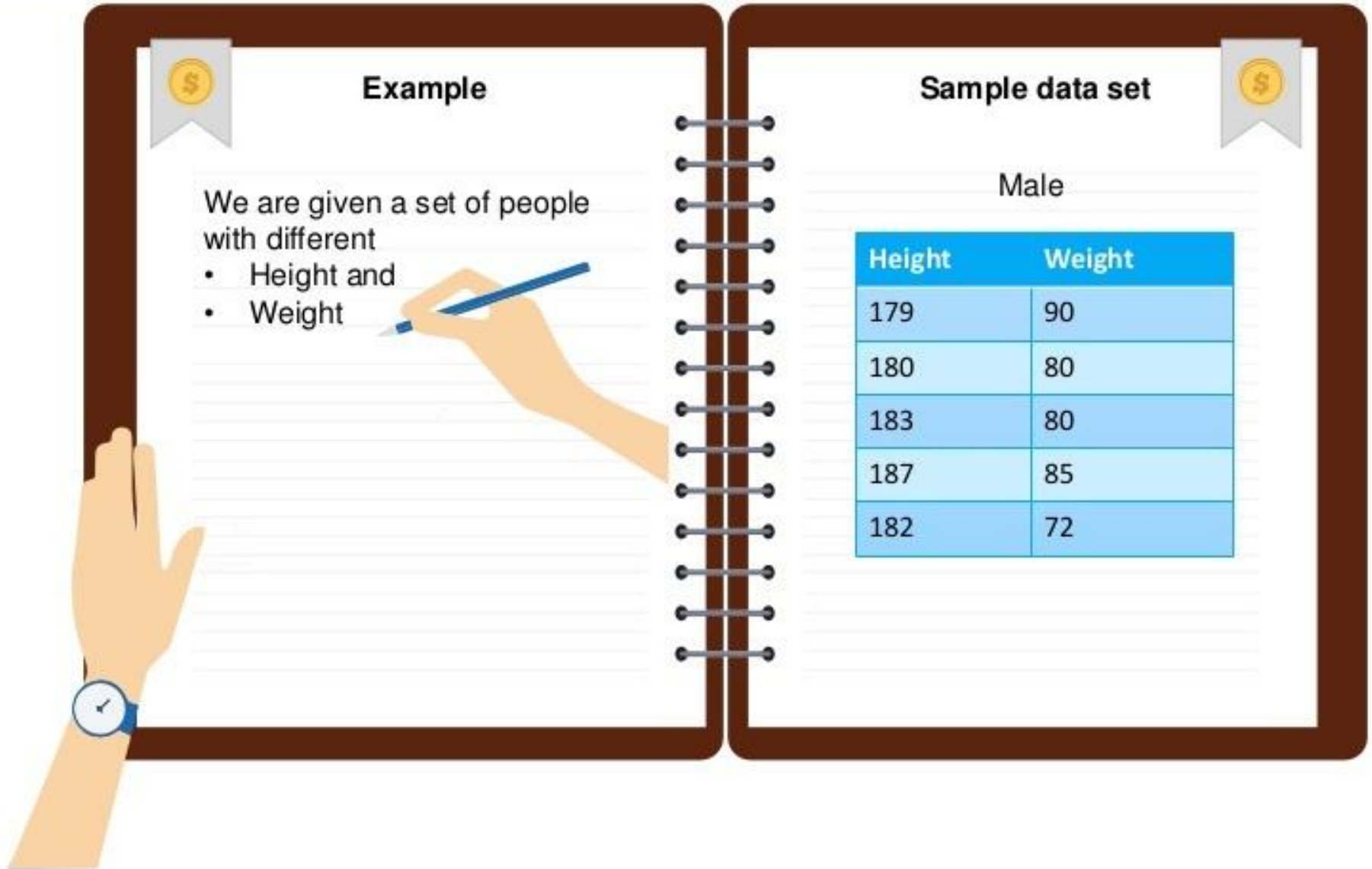
What is Support Vector Machine?



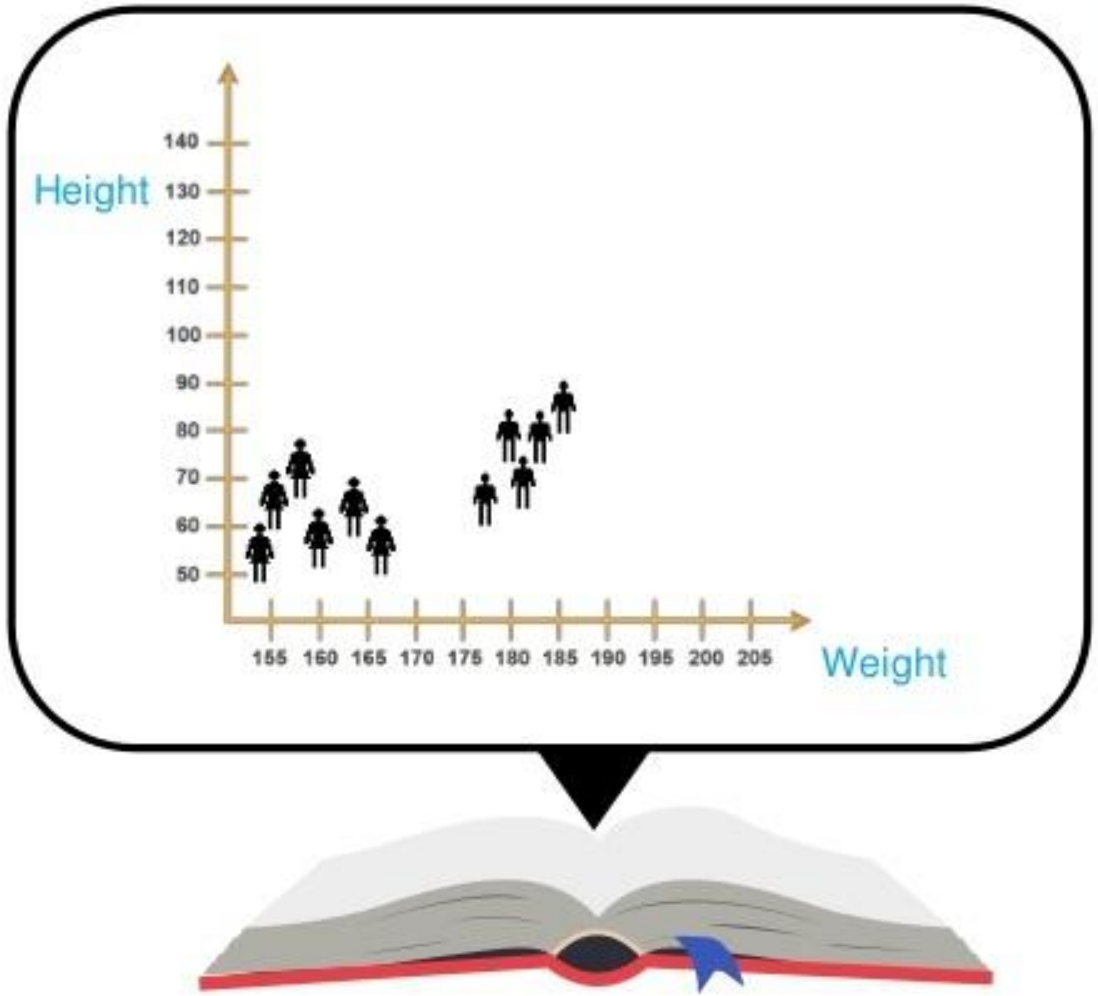
What is Support Vector Machine?



What is Support Vector Machine?

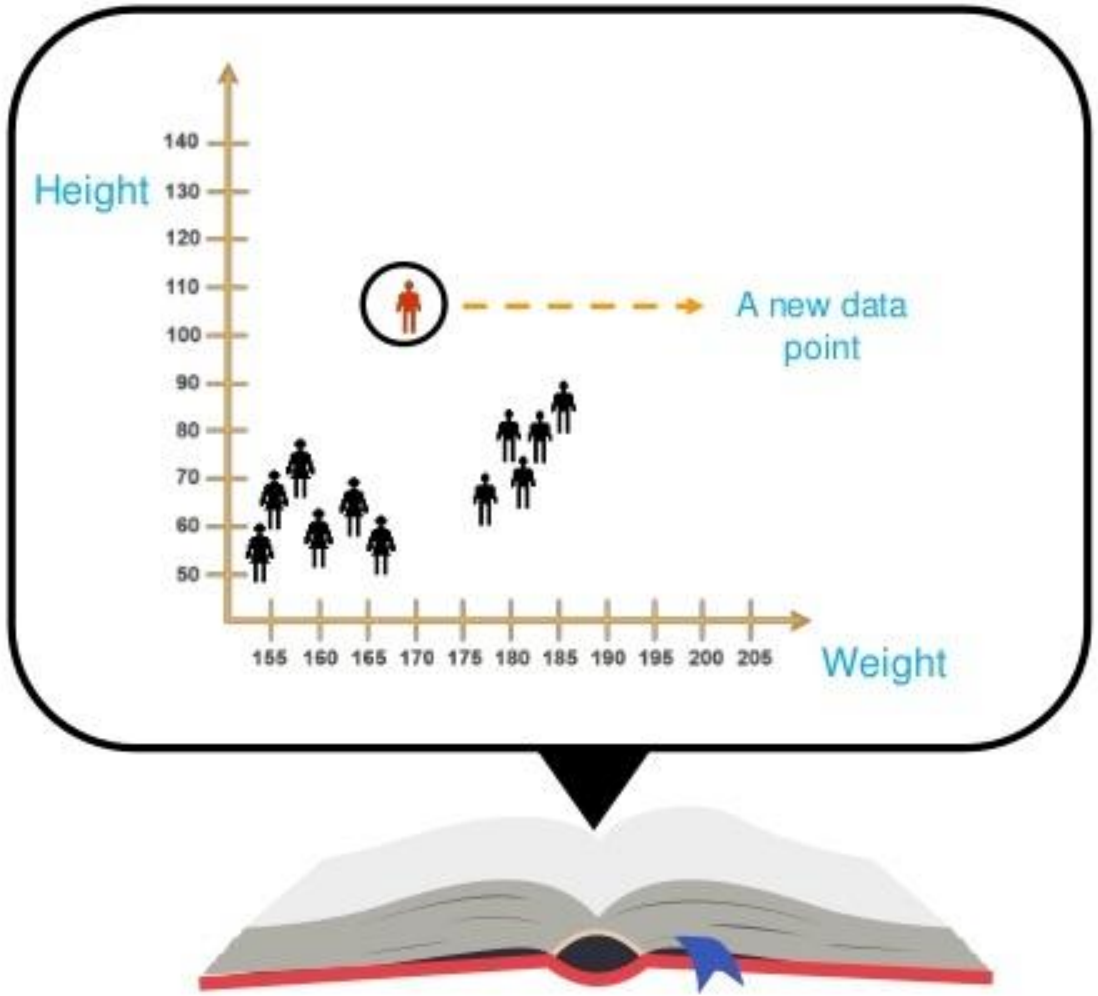


What is Support Vector Machine?



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

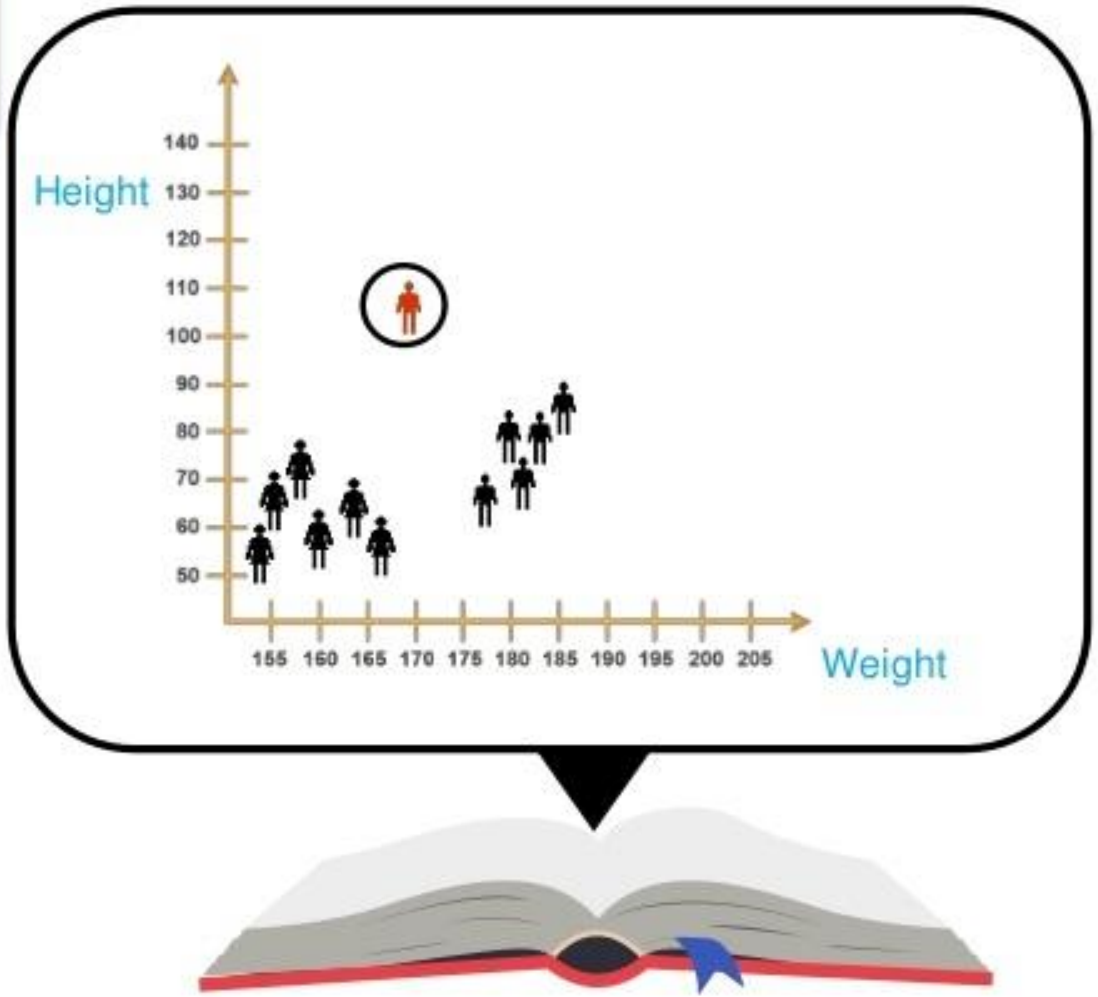
What is Support Vector Machine?



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

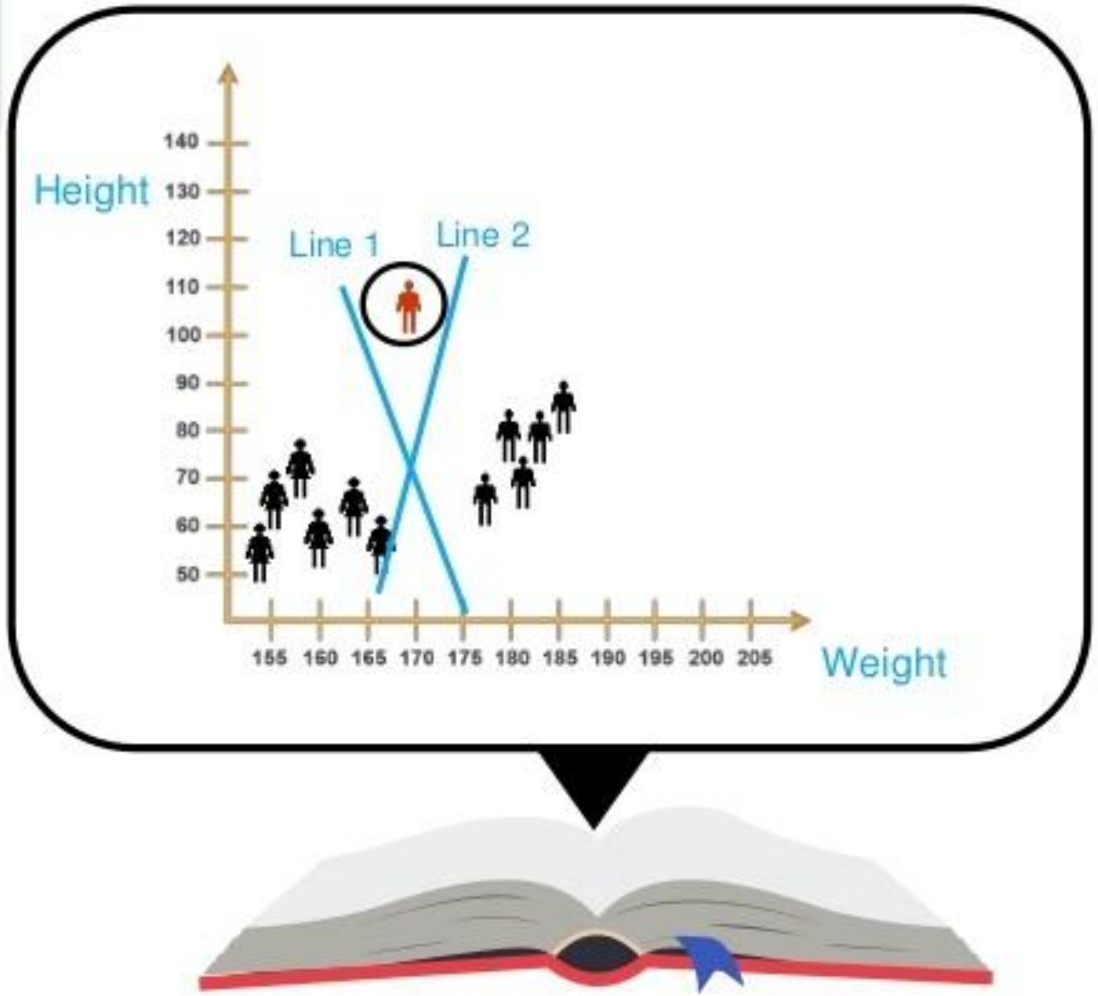
What is Support Vector Machine?

Sure.. For this task, we need to split our data first



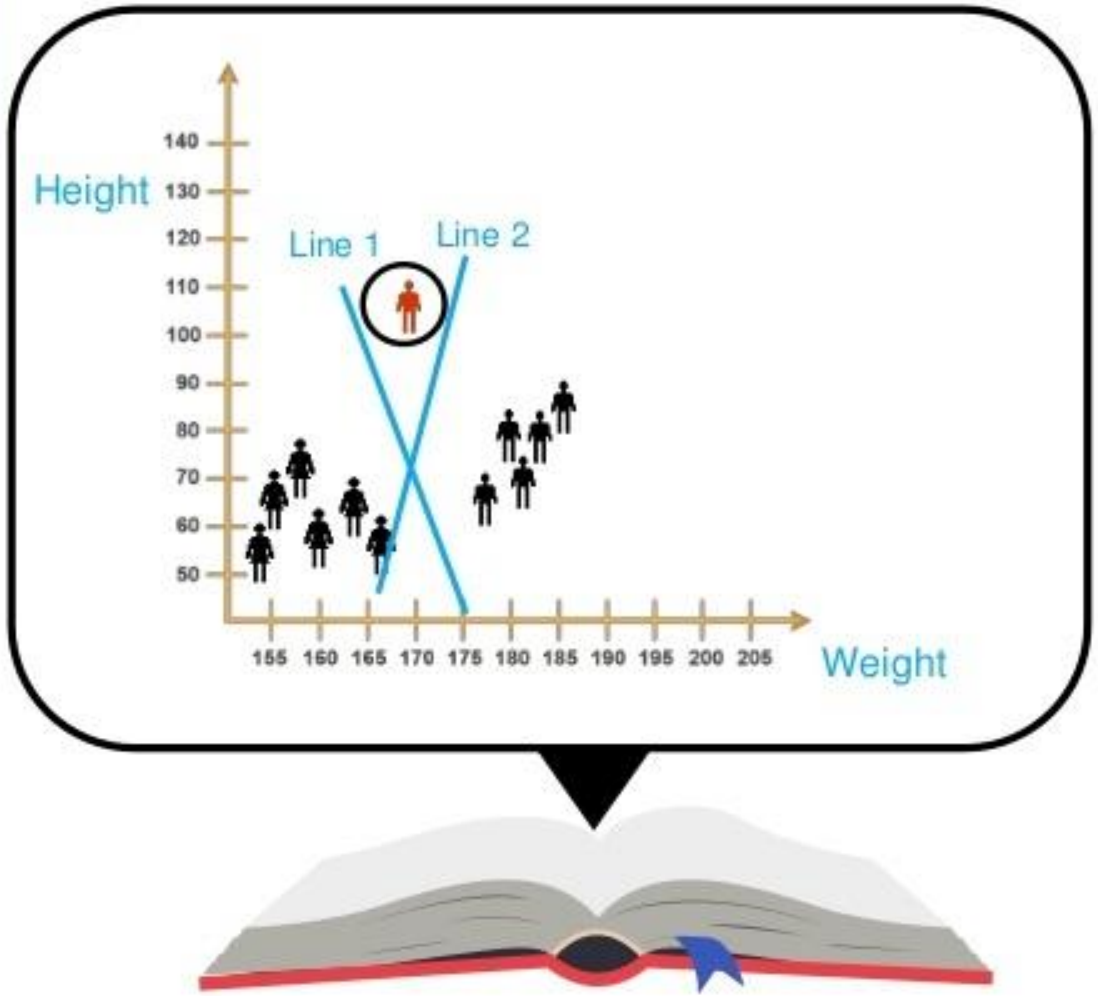
What is Support Vector Machine?

We can split our data by choosing any of these lines



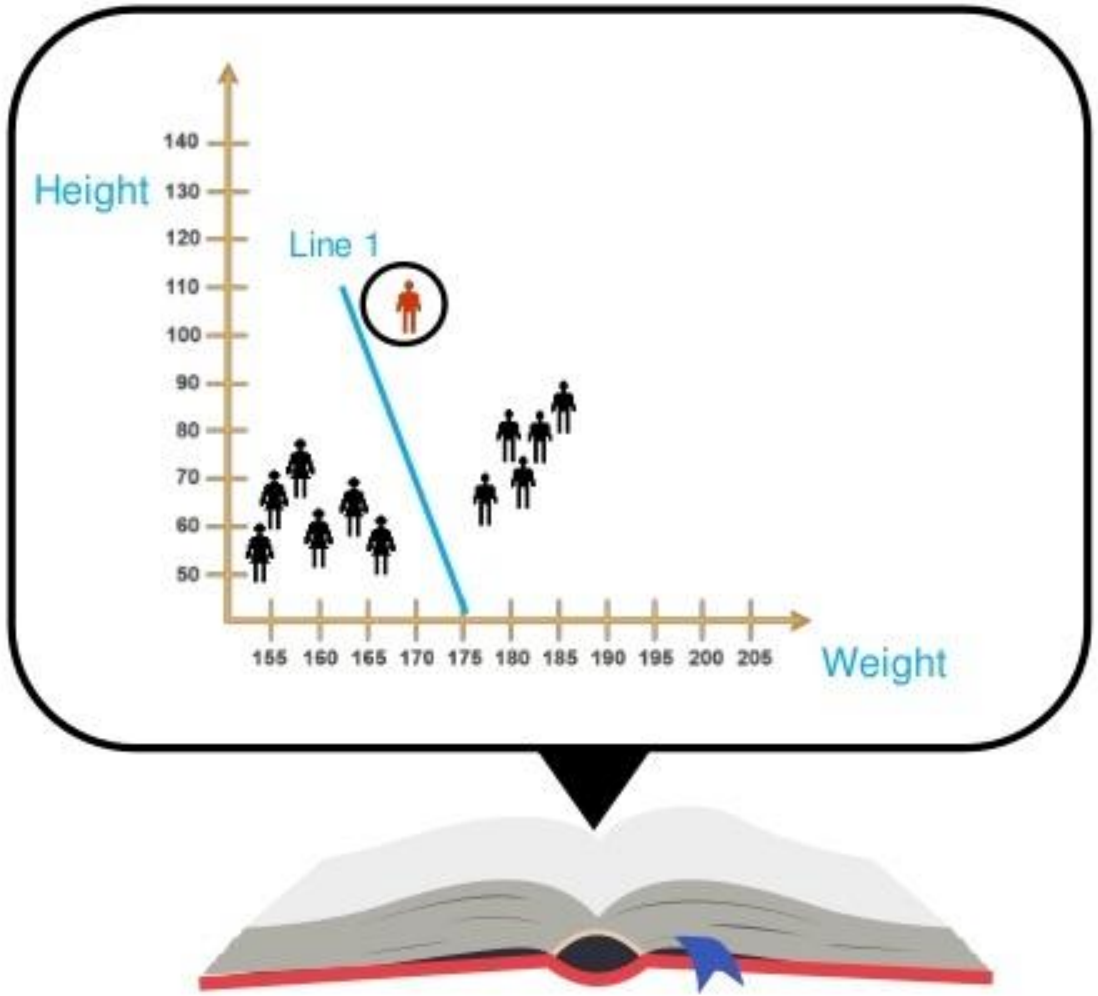
What is Support Vector Machine?

But to predict the gender of a new data point we should split the data in the best possible way



What is Support Vector Machine?

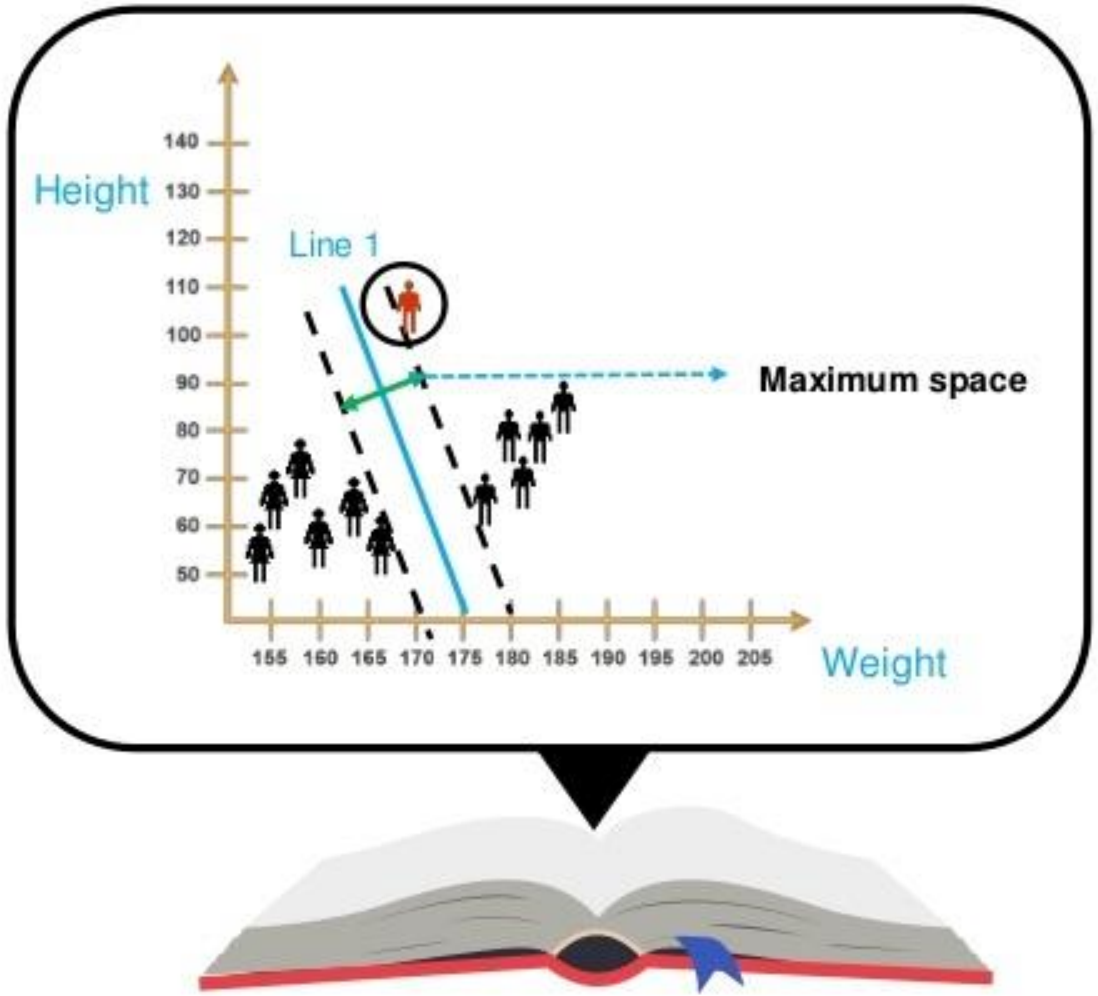
Then I would say, this line best splits the data



Why do you say it's the best split??

What is Support Vector Machine?

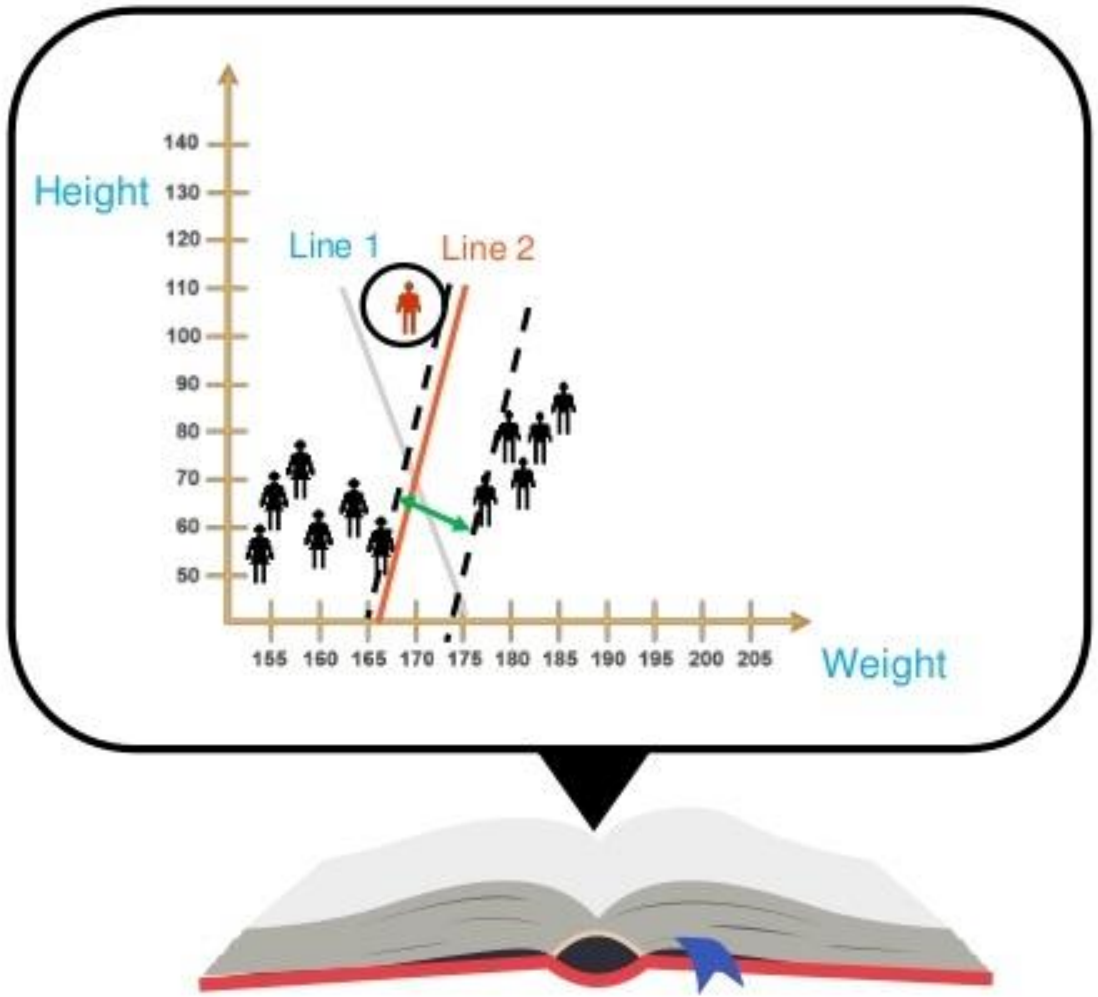
This line has the maximum space that separates the two classes



Why do you say it's the best split??

What is Support Vector Machine?

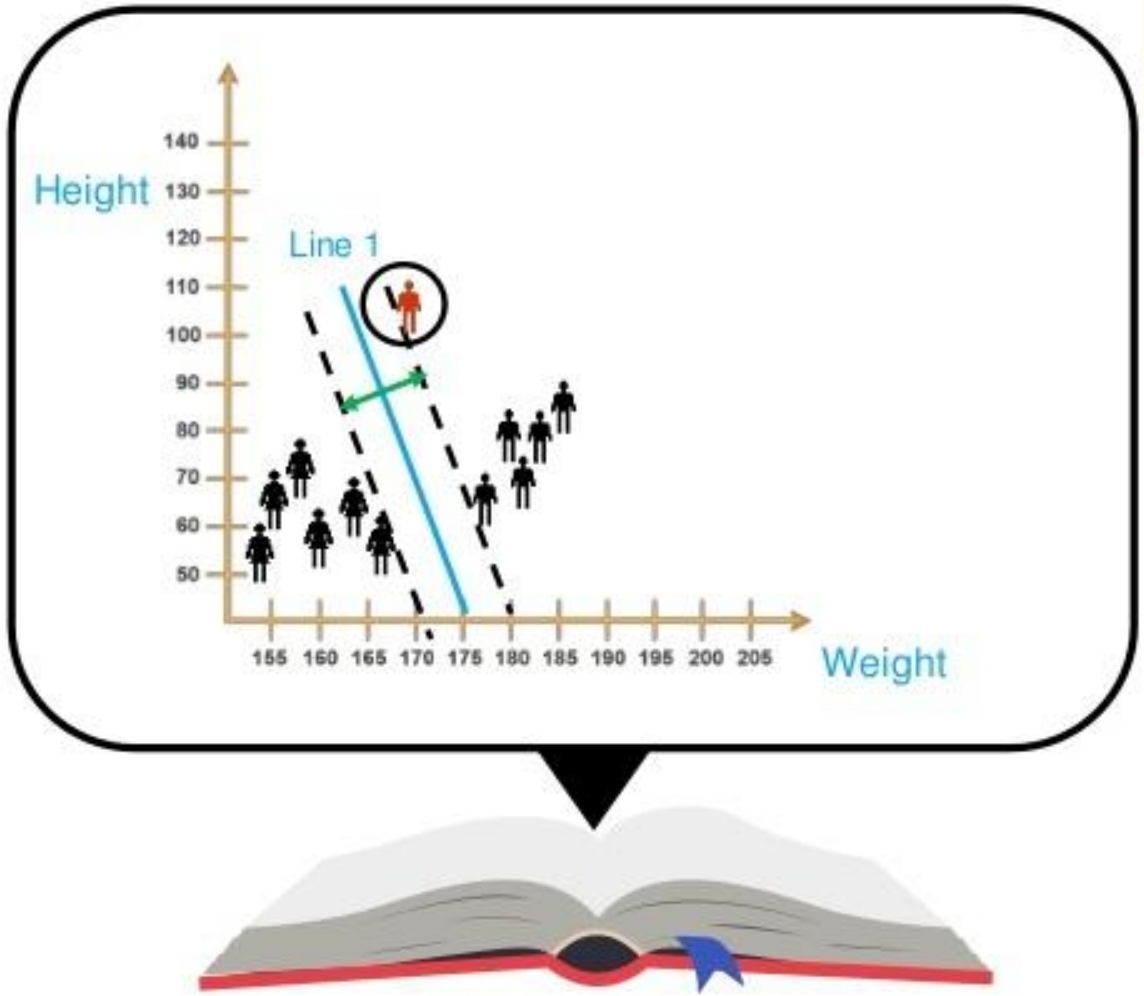
While the other line doesn't has the maximum space that separates the two classes



Why do you say it's the best split??

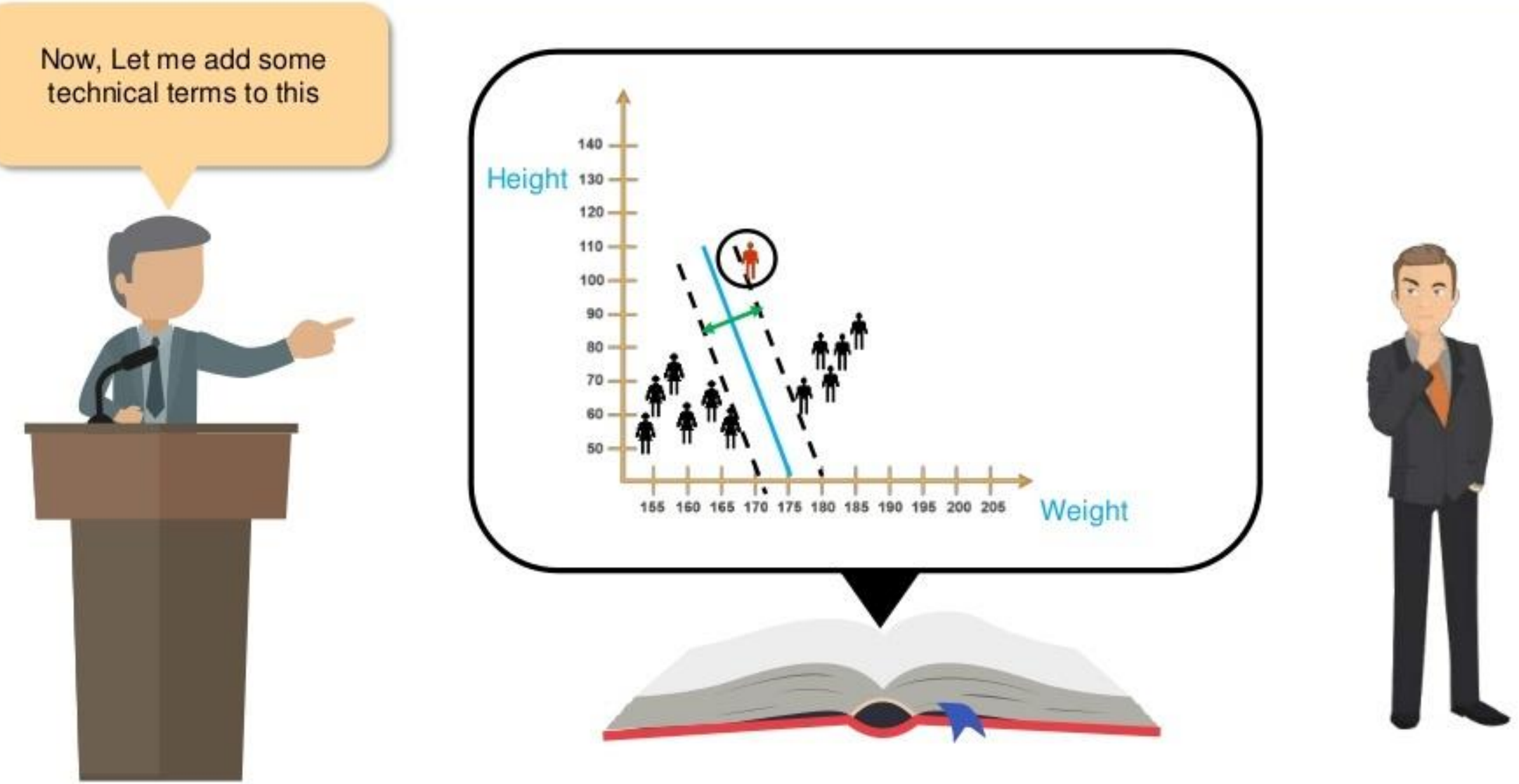
What is Support Vector Machine?

That is why this line best splits the data



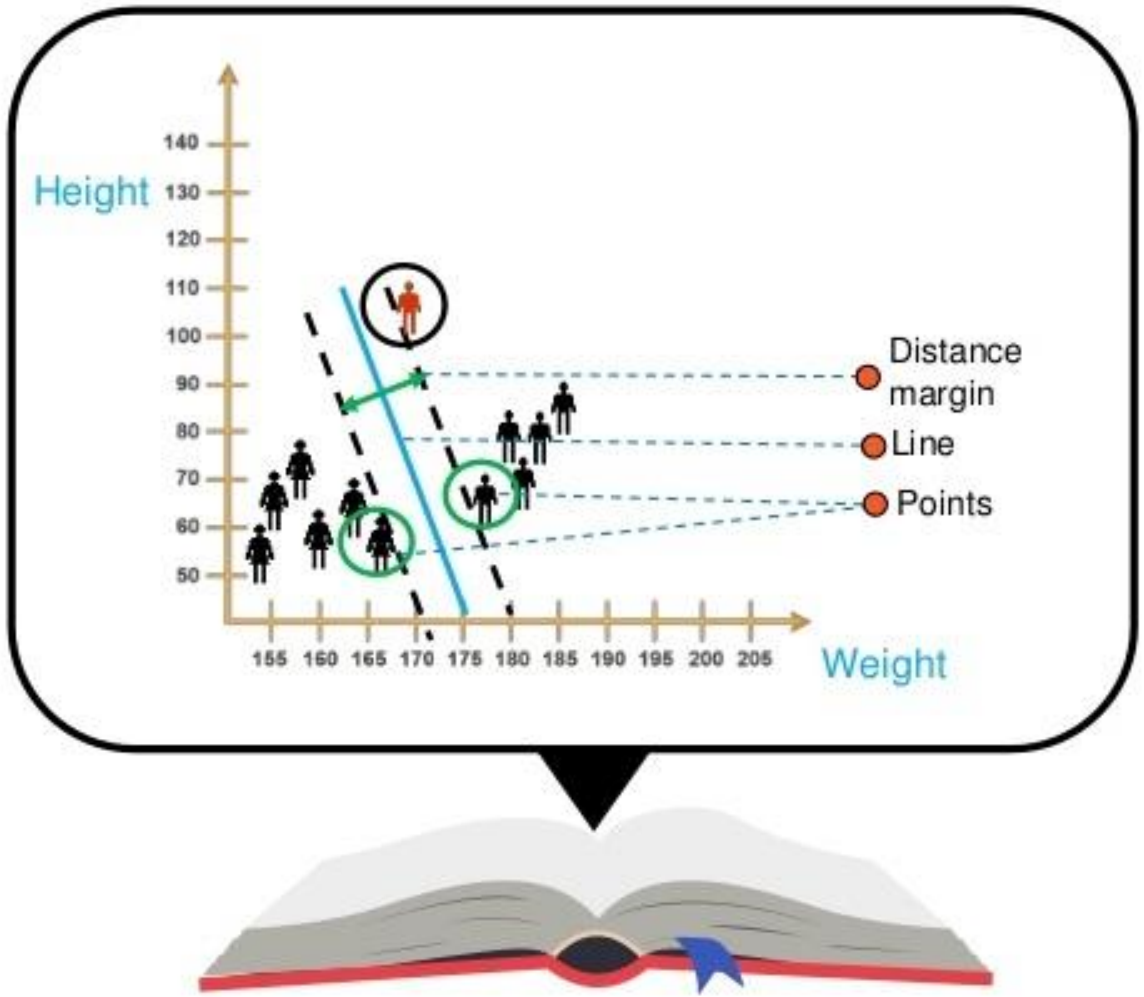
Well yes.. This is the best split!

What is Support Vector Machine?



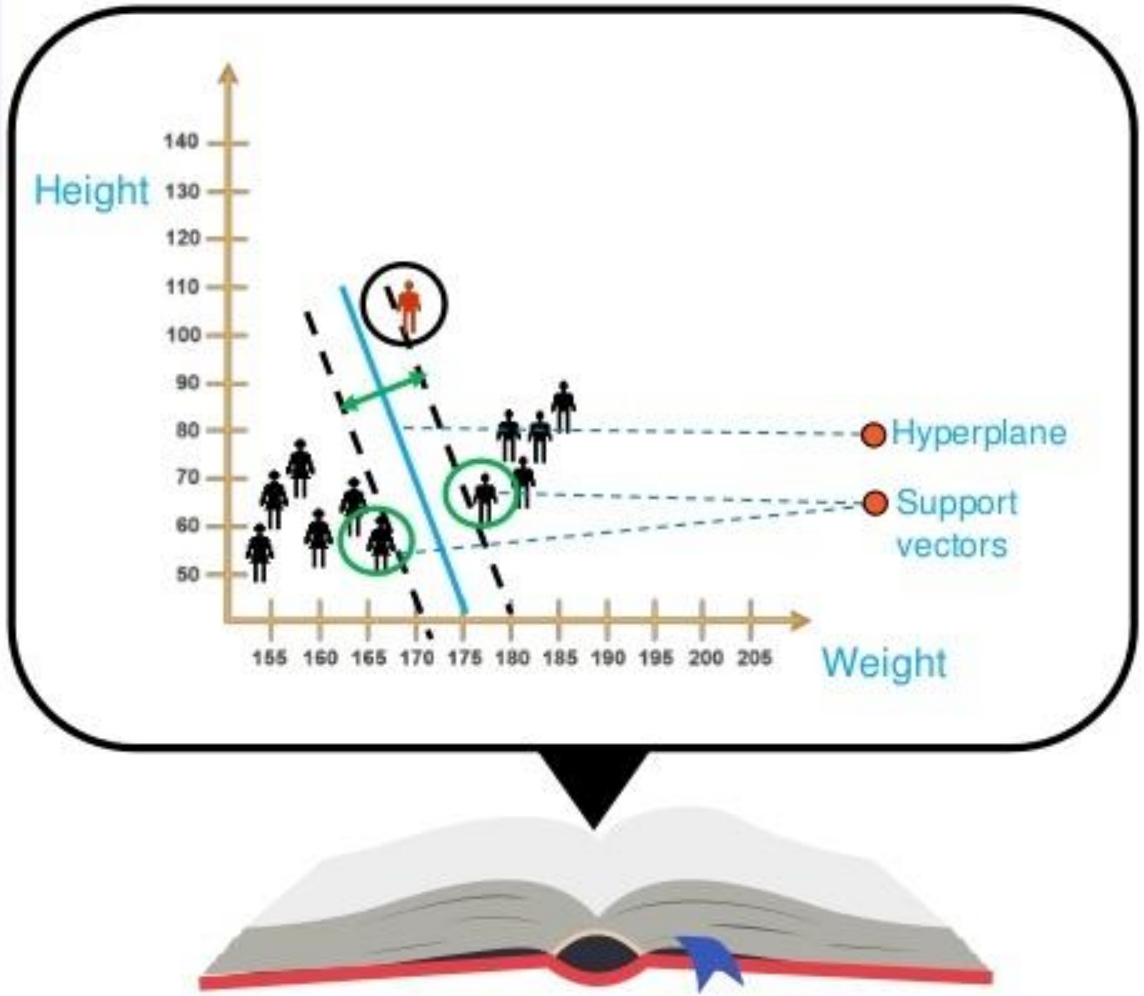
What is Support Vector Machine?

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



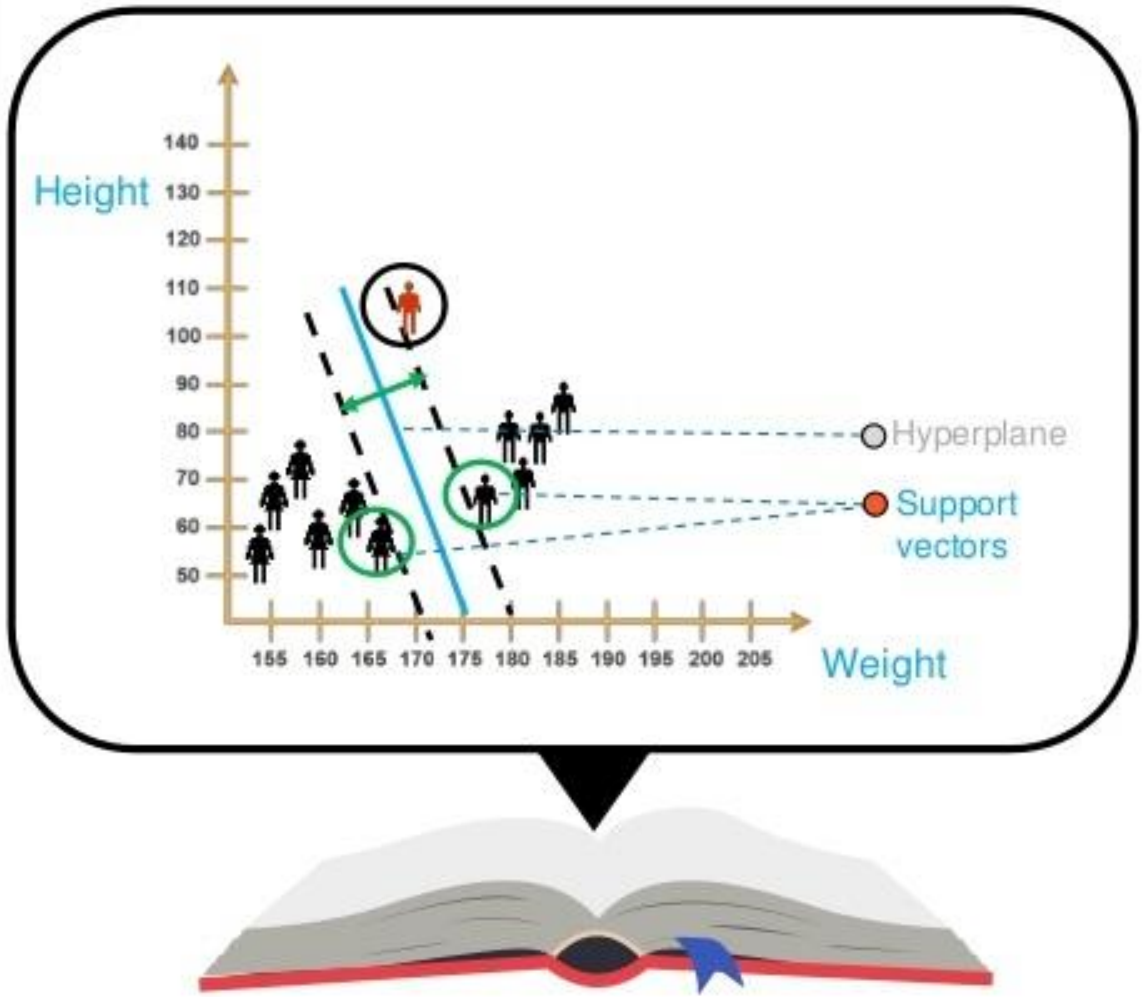
What is Support Vector Machine?

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



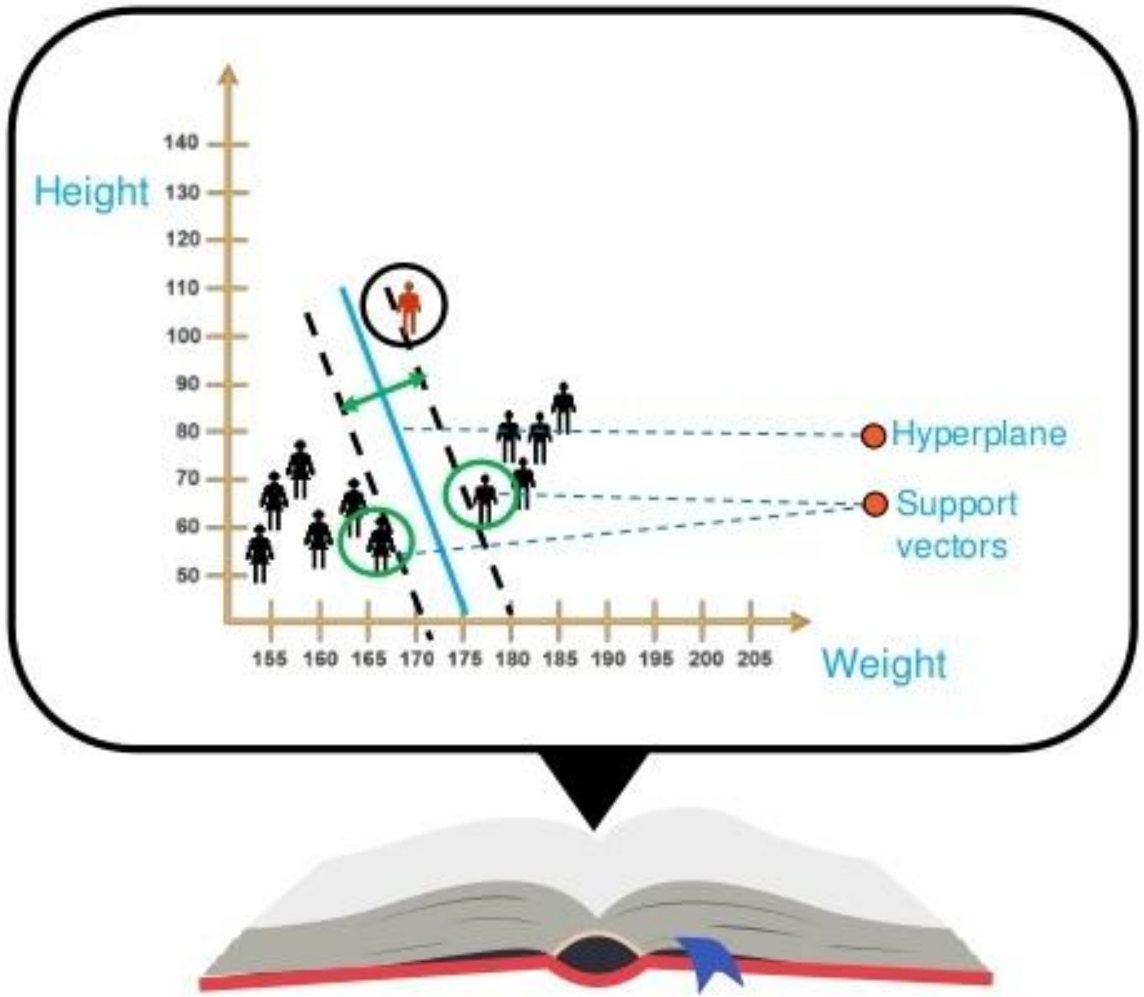
What is Support Vector Machine?

Where support vectors are the extreme points in the datasets



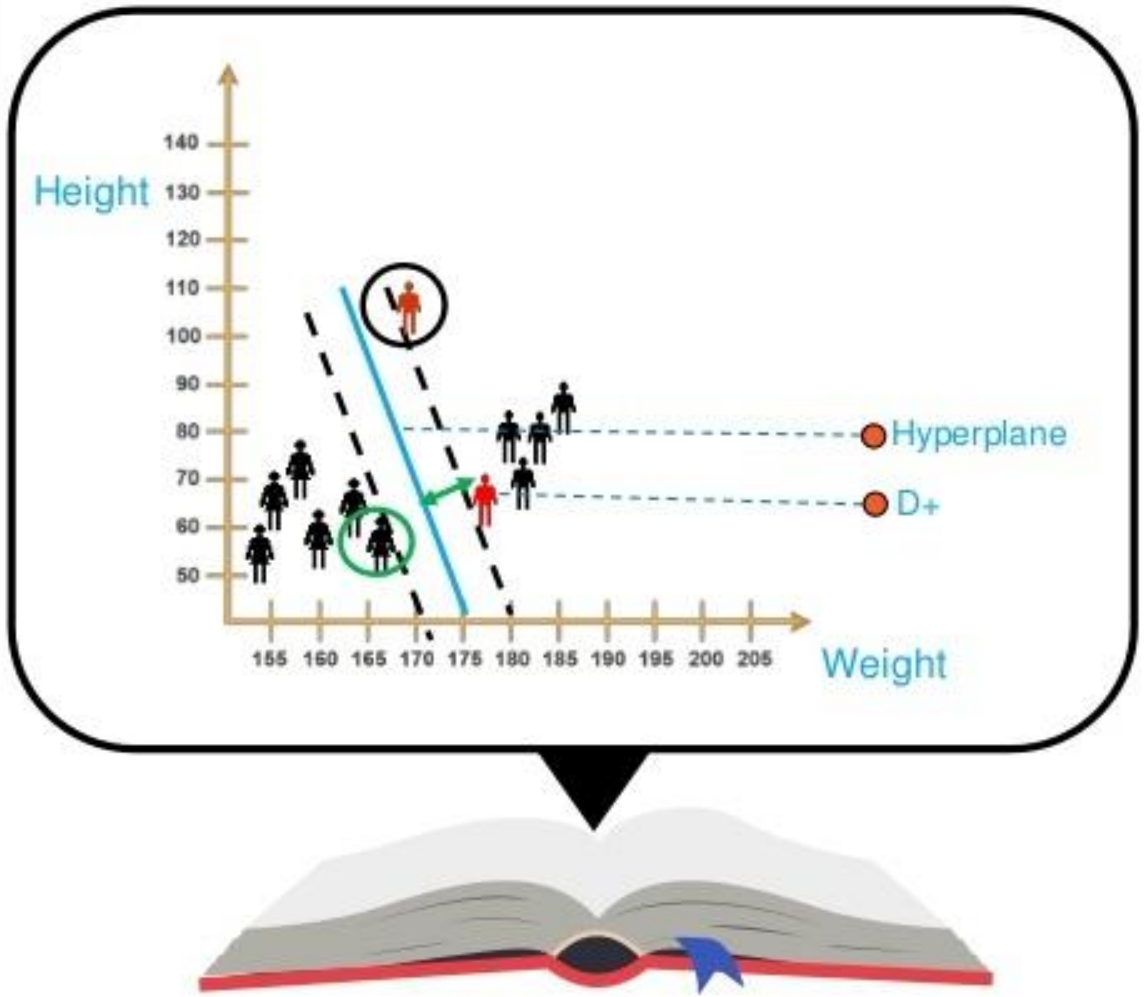
What is Support Vector Machine?

And a hyperplane has the maximum distance to the support vectors of any class



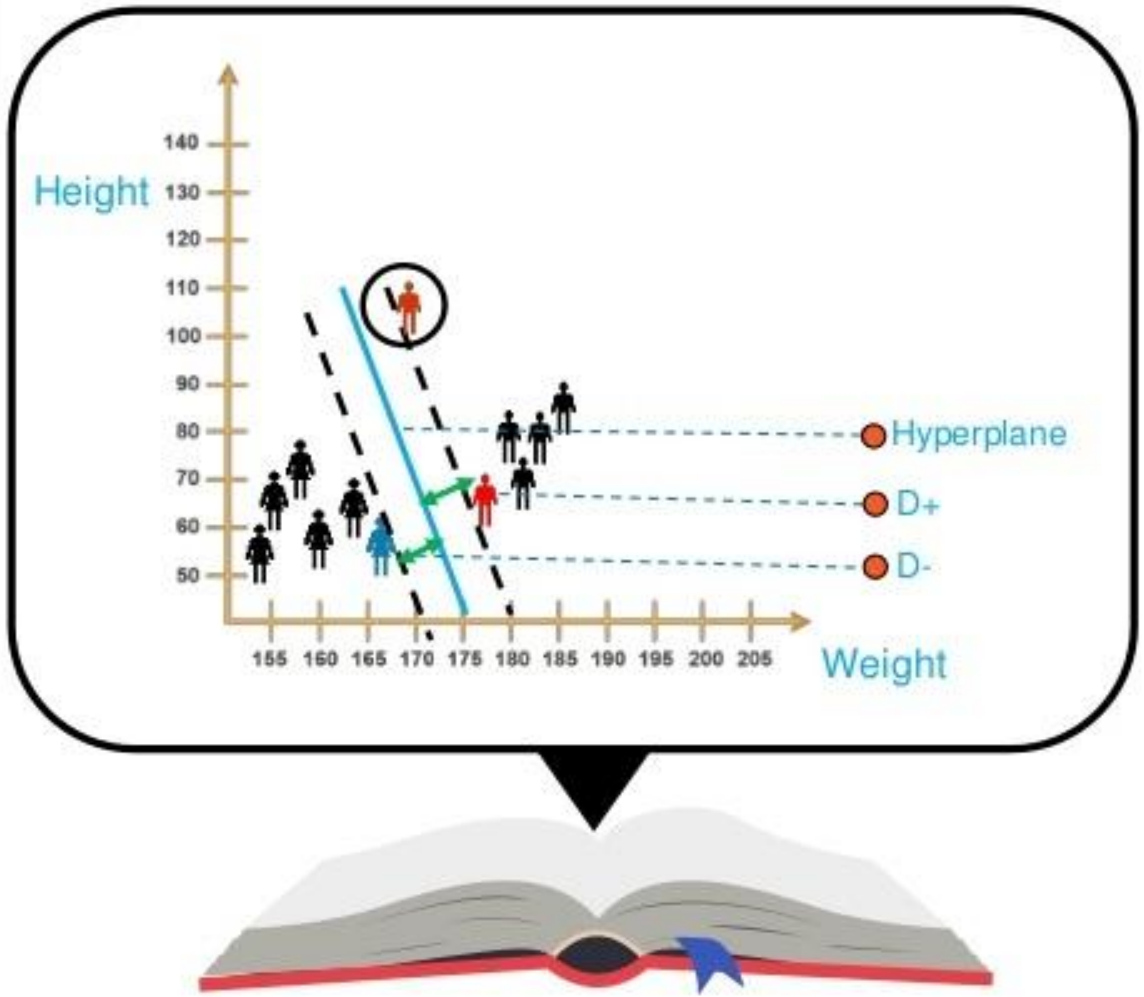
What is Support Vector Machine?

Here, D+ is the shortest distance to the closest positive point

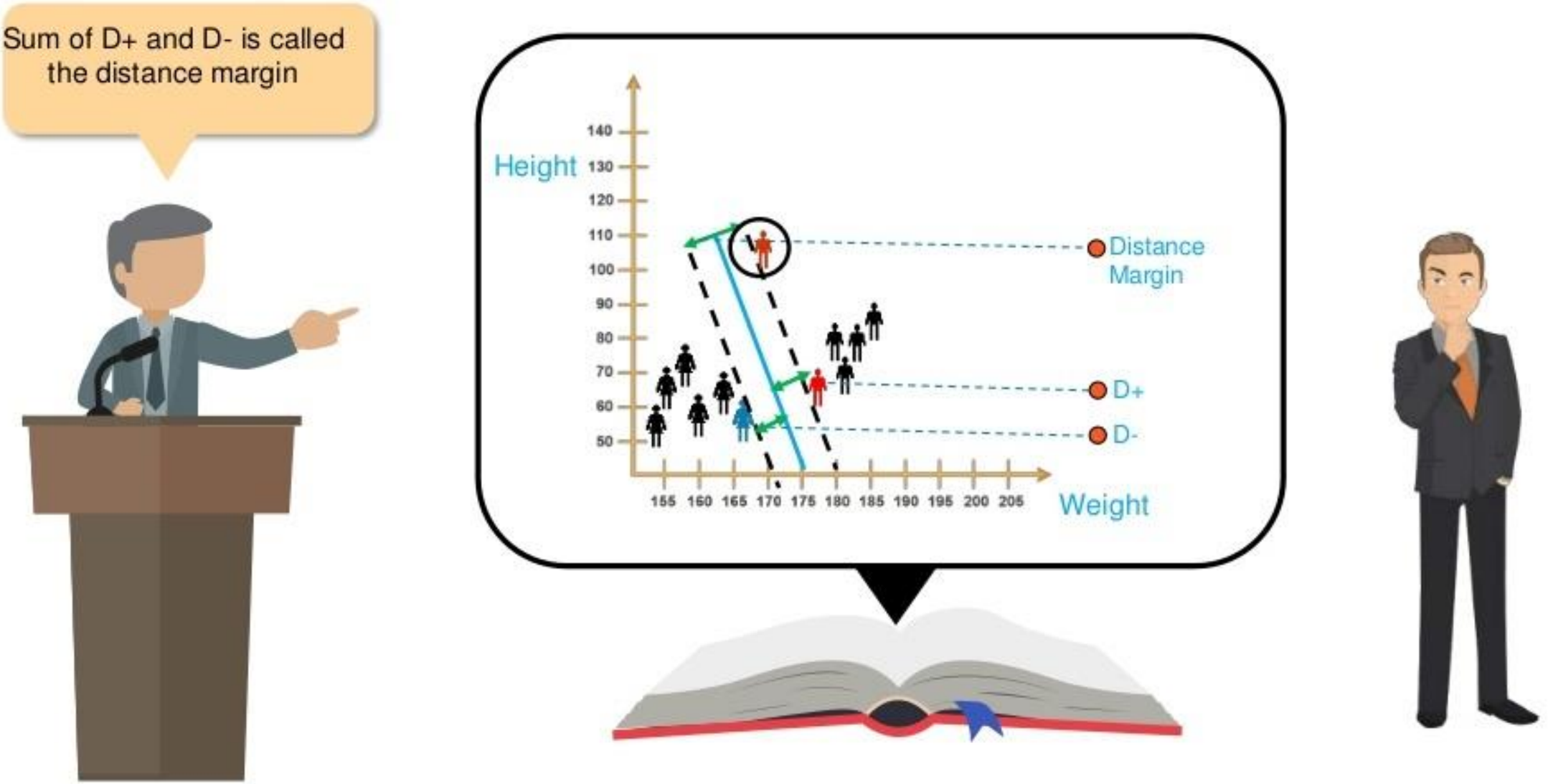


What is Support Vector Machine?

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

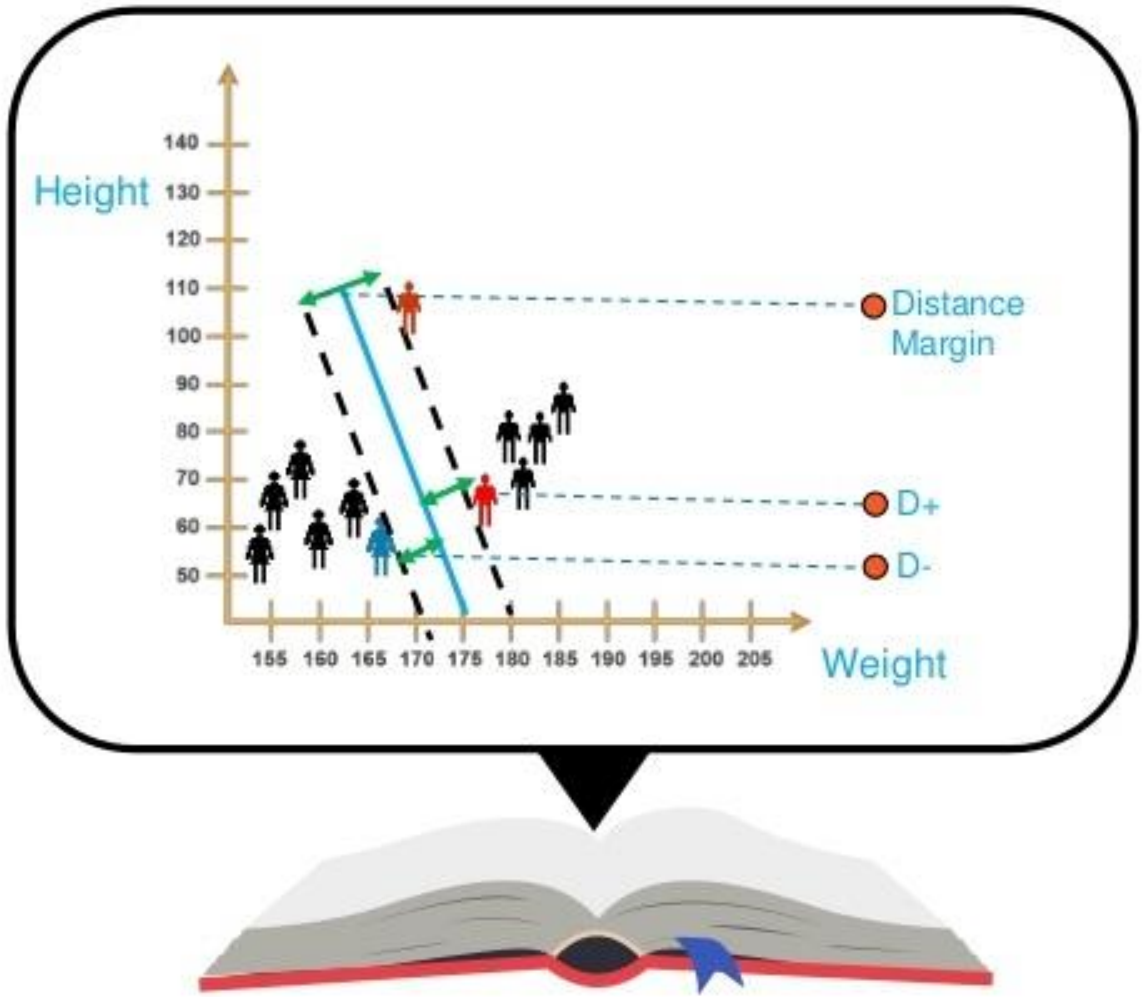


What is Support Vector Machine?



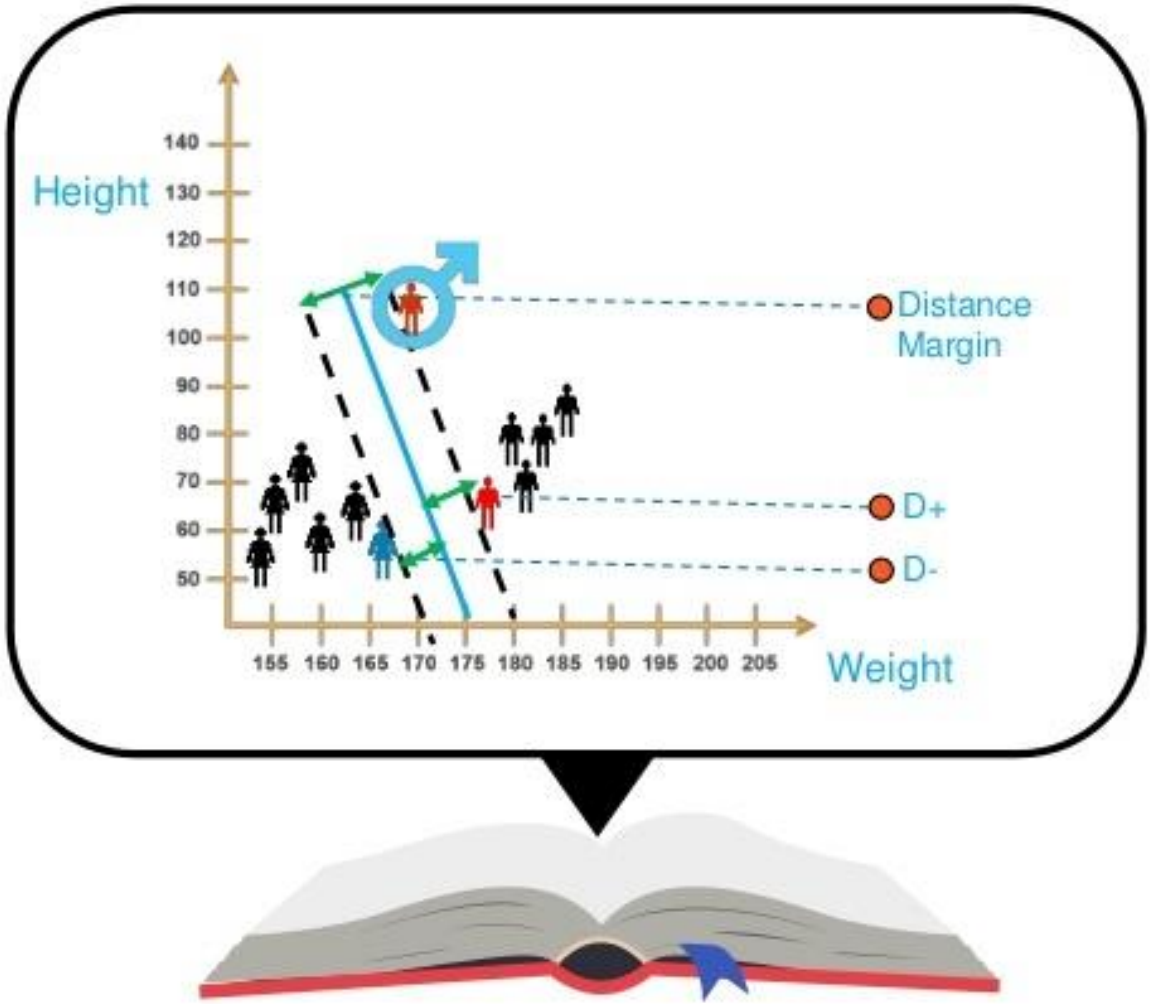
What is Support Vector Machine?

From the distance margin, we get an optimal hyperplane



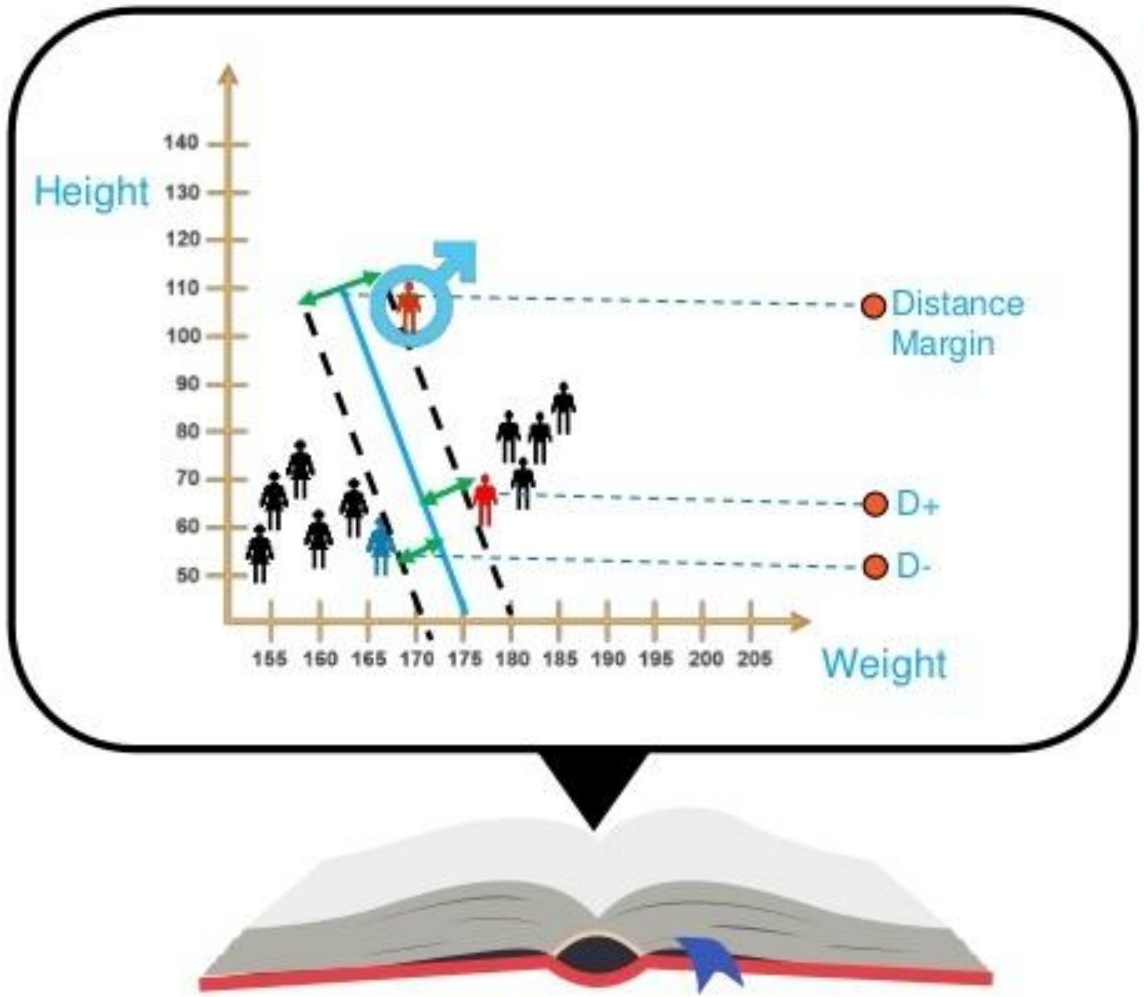
What is Support Vector Machine?

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



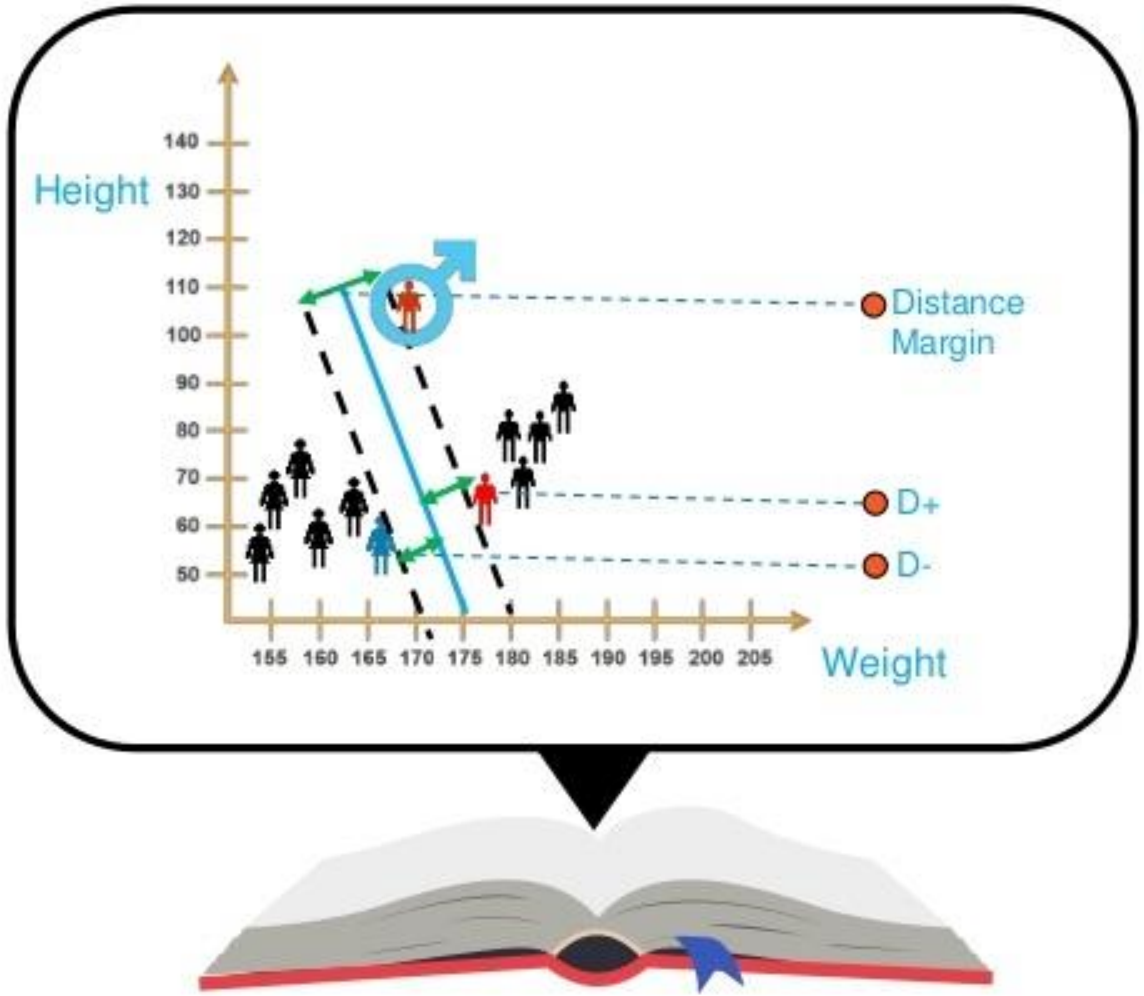
What is Support Vector Machine?

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



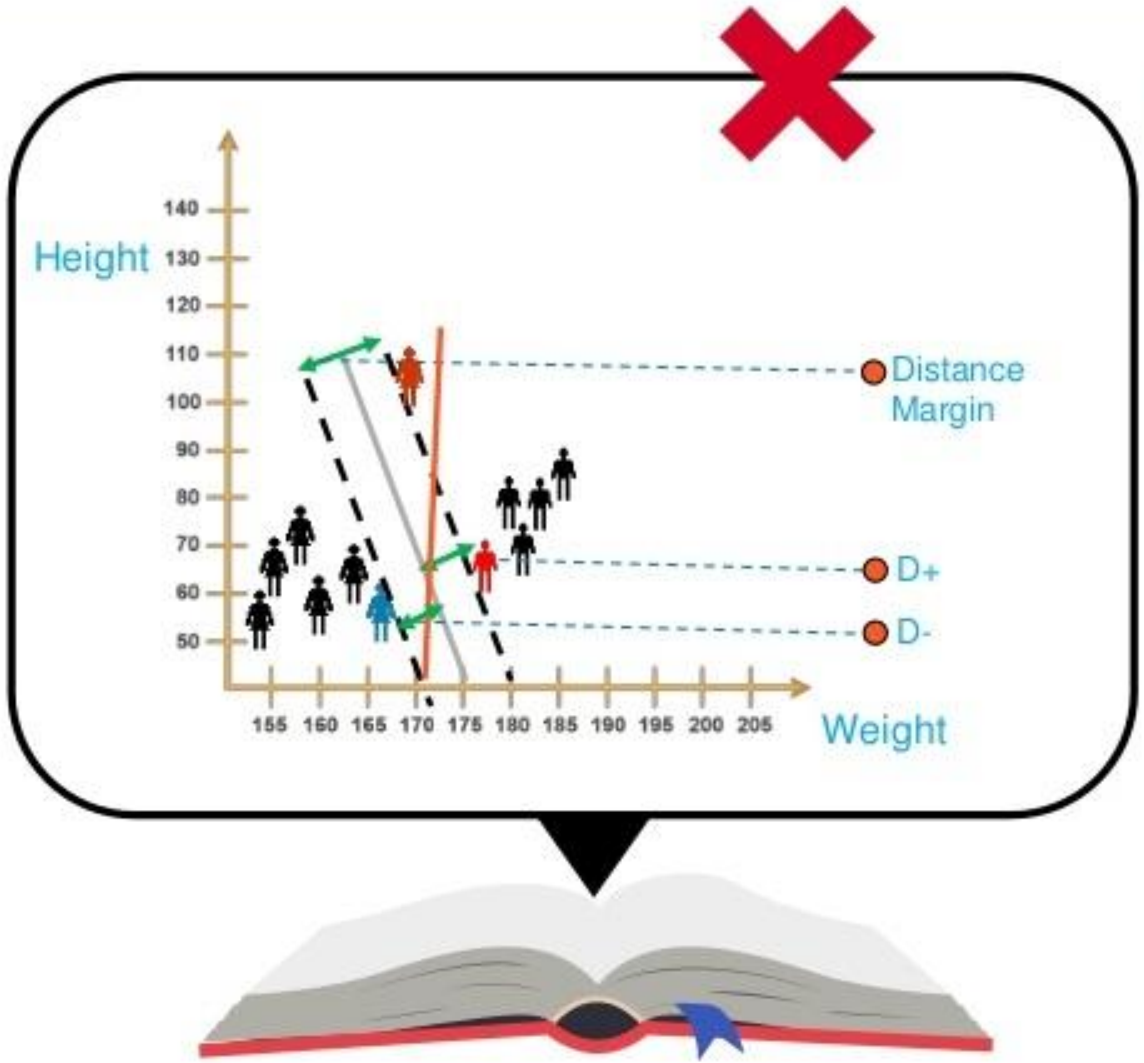
That was so clear!

What is Support Vector Machine?



What is Support Vector Machine?

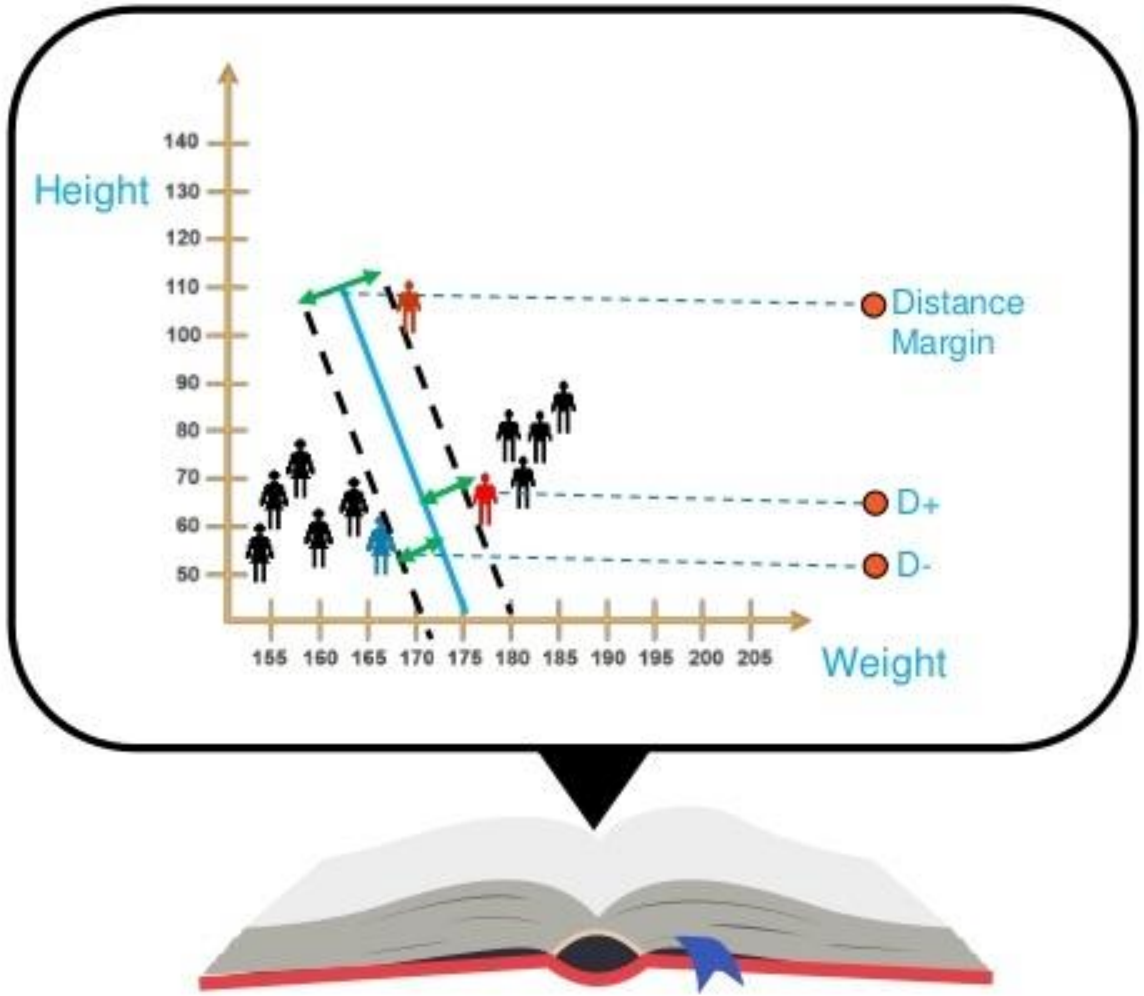
If we select a hyperplane having low margin then there is high chance of misclassification



But what happens if a hyperplane is not optimal?

What is Support Vector Machine?

What we discussed so far, is also called as **LSVM**

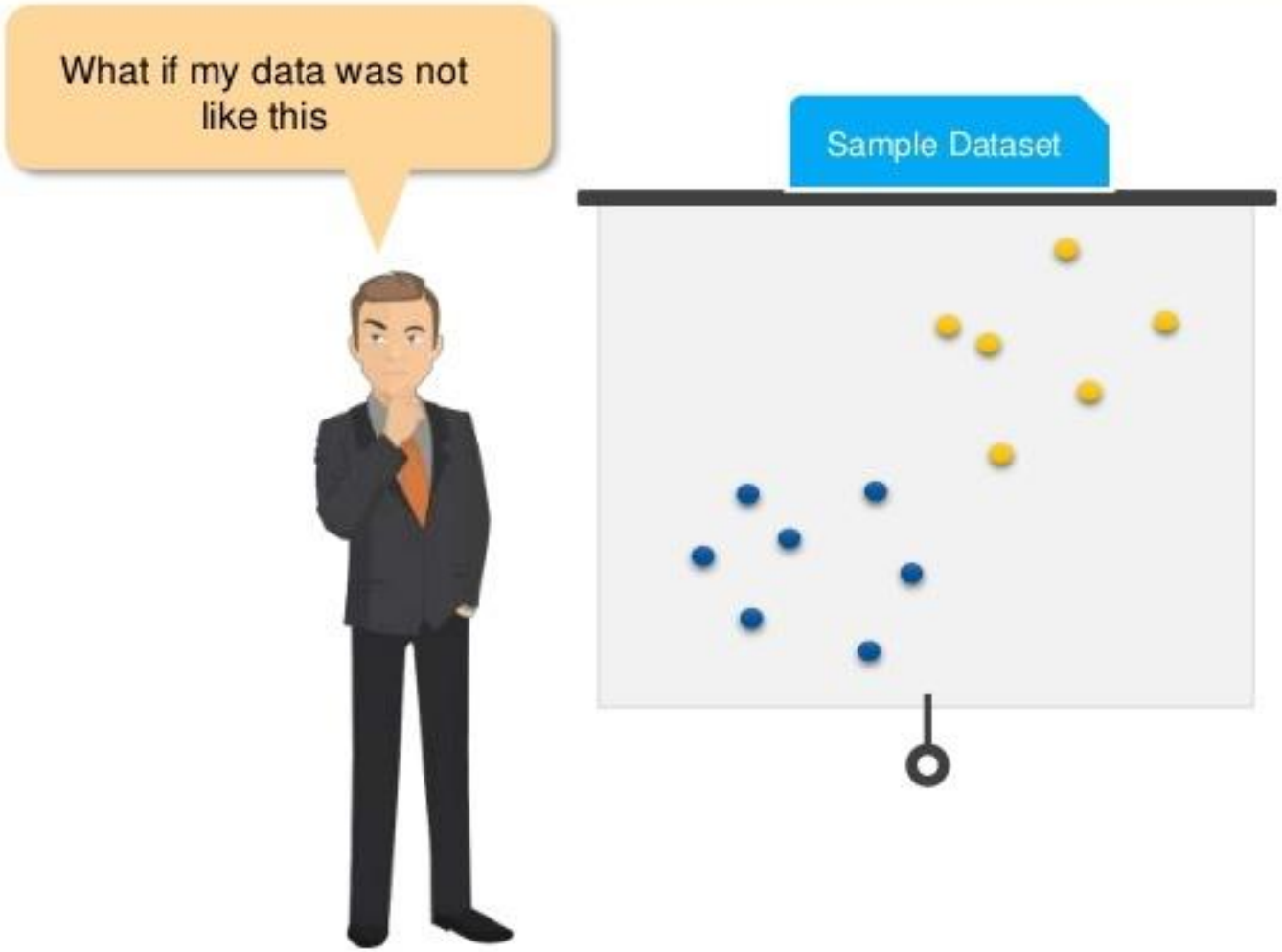


But what happens if a hyperplane is not optimal?

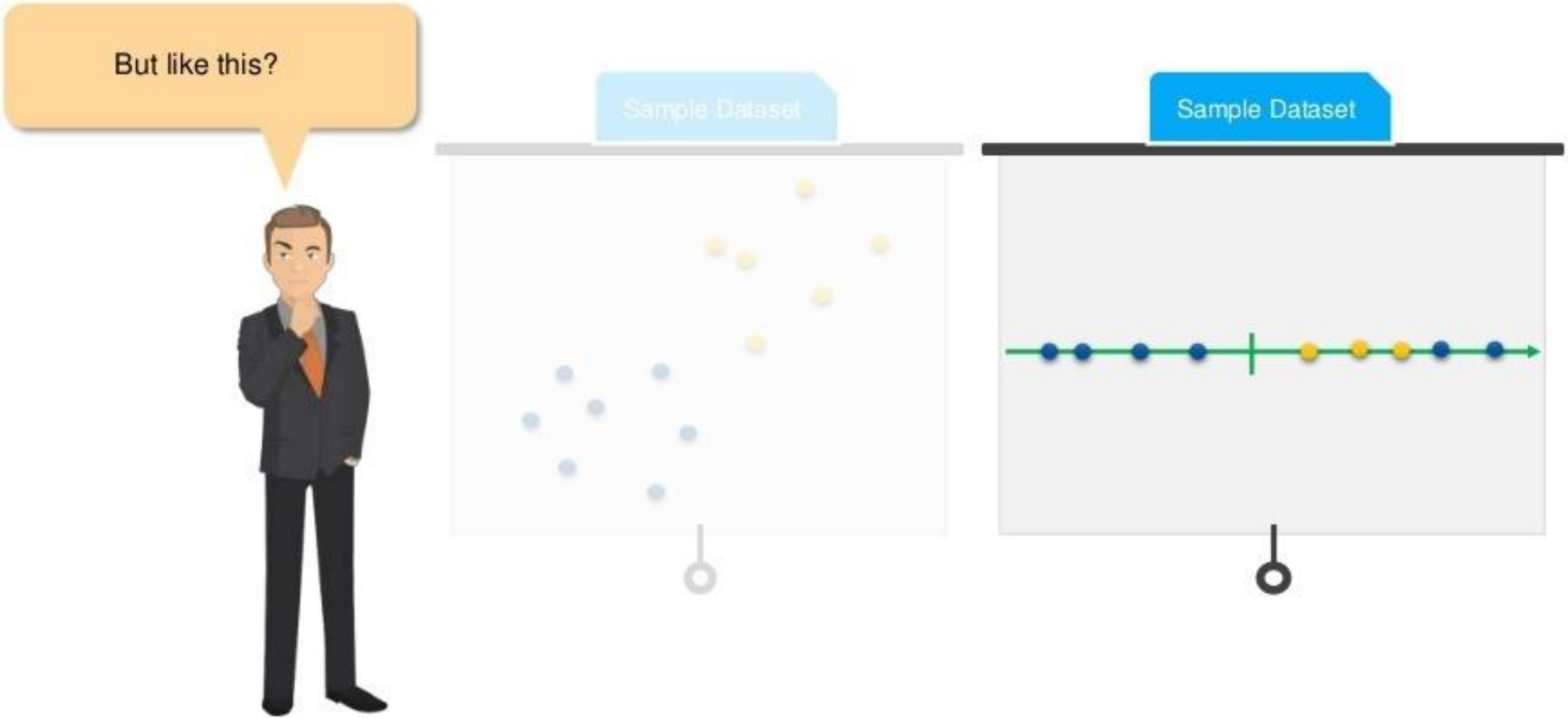
Understanding Support Vector Machine

Understanding Support Vector Machine

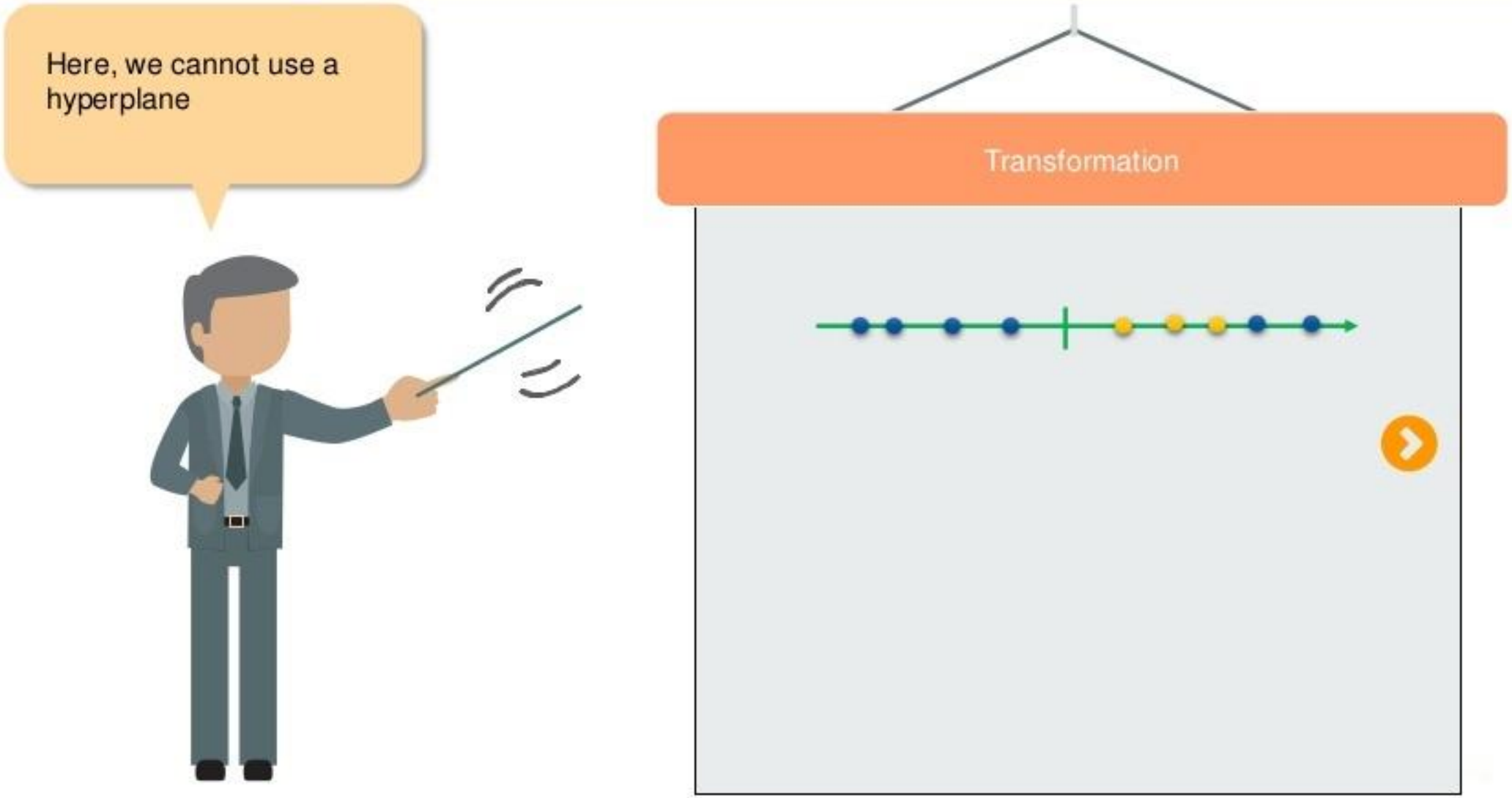
Understanding Support Vector Machine



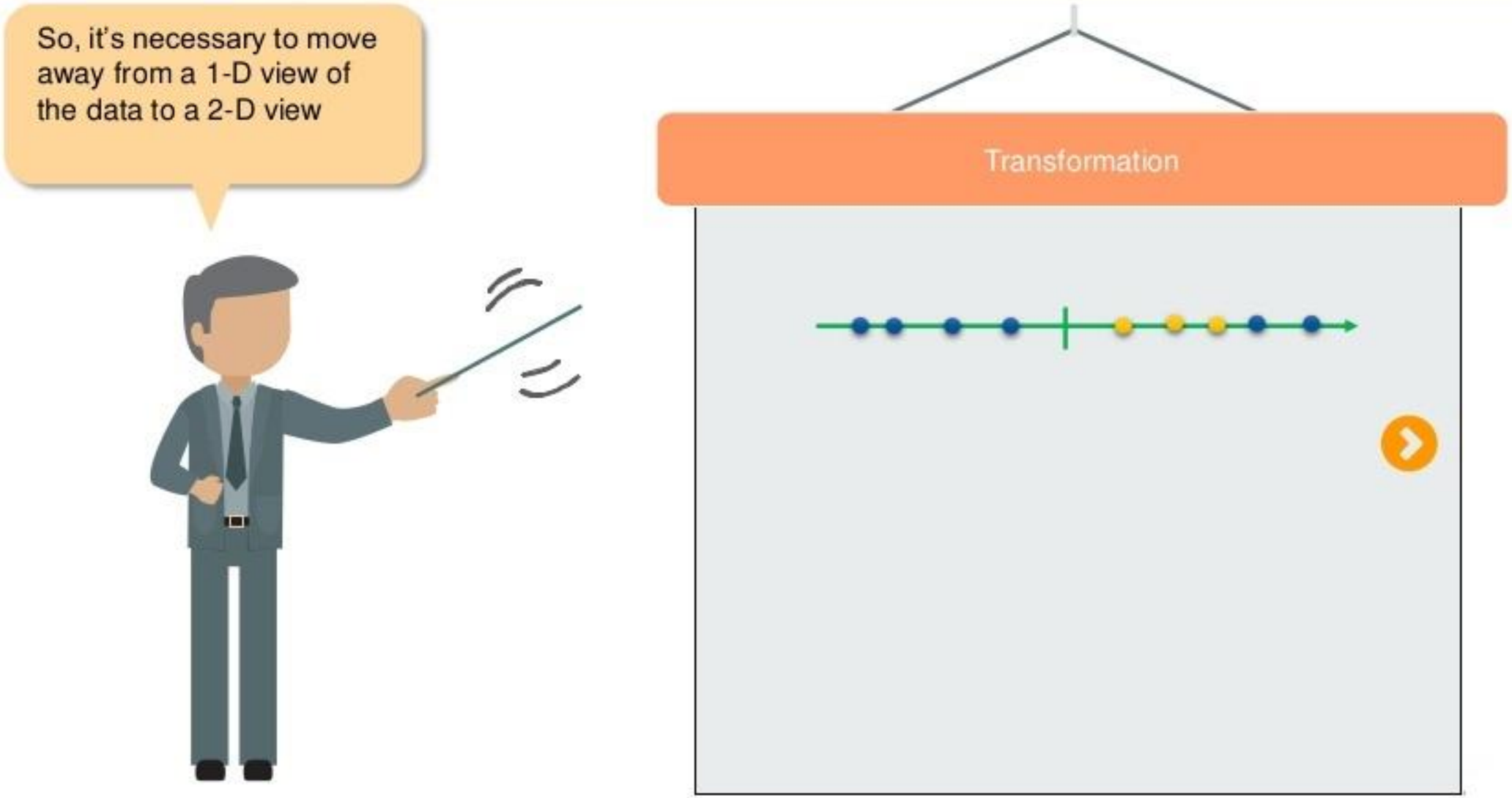
Understanding Support Vector Machine



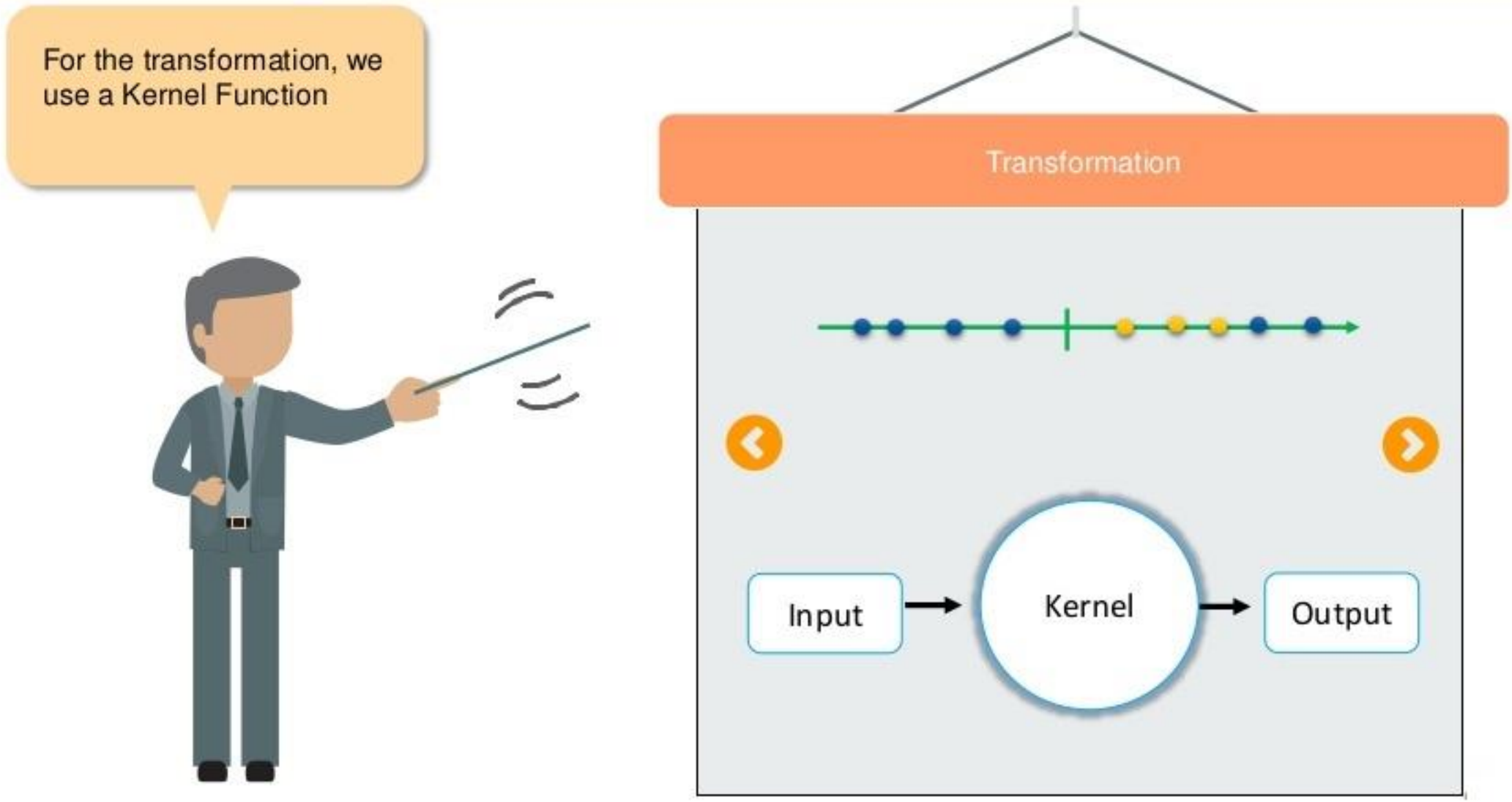
Understanding Support Vector Machine



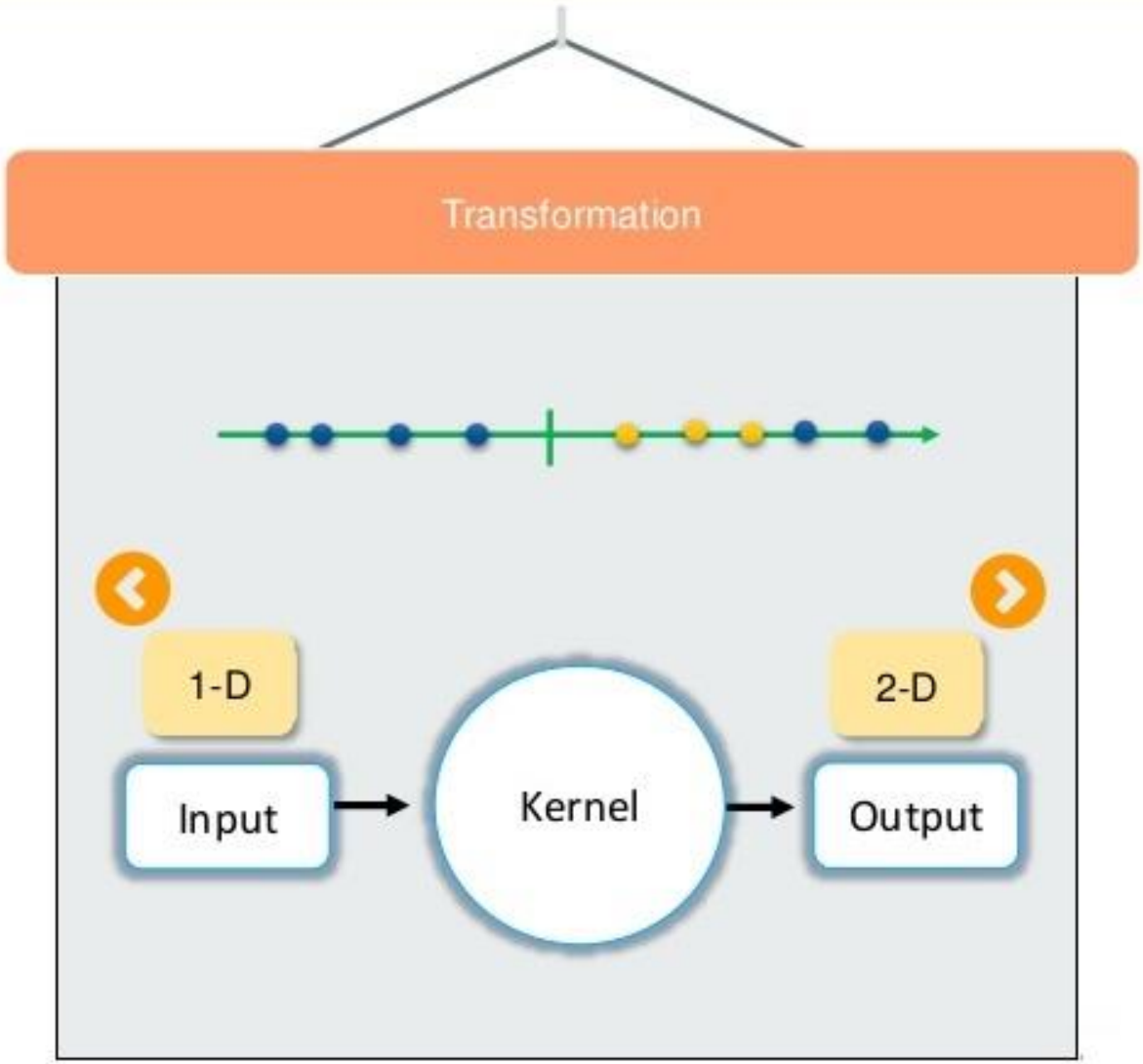
Understanding Support Vector Machine



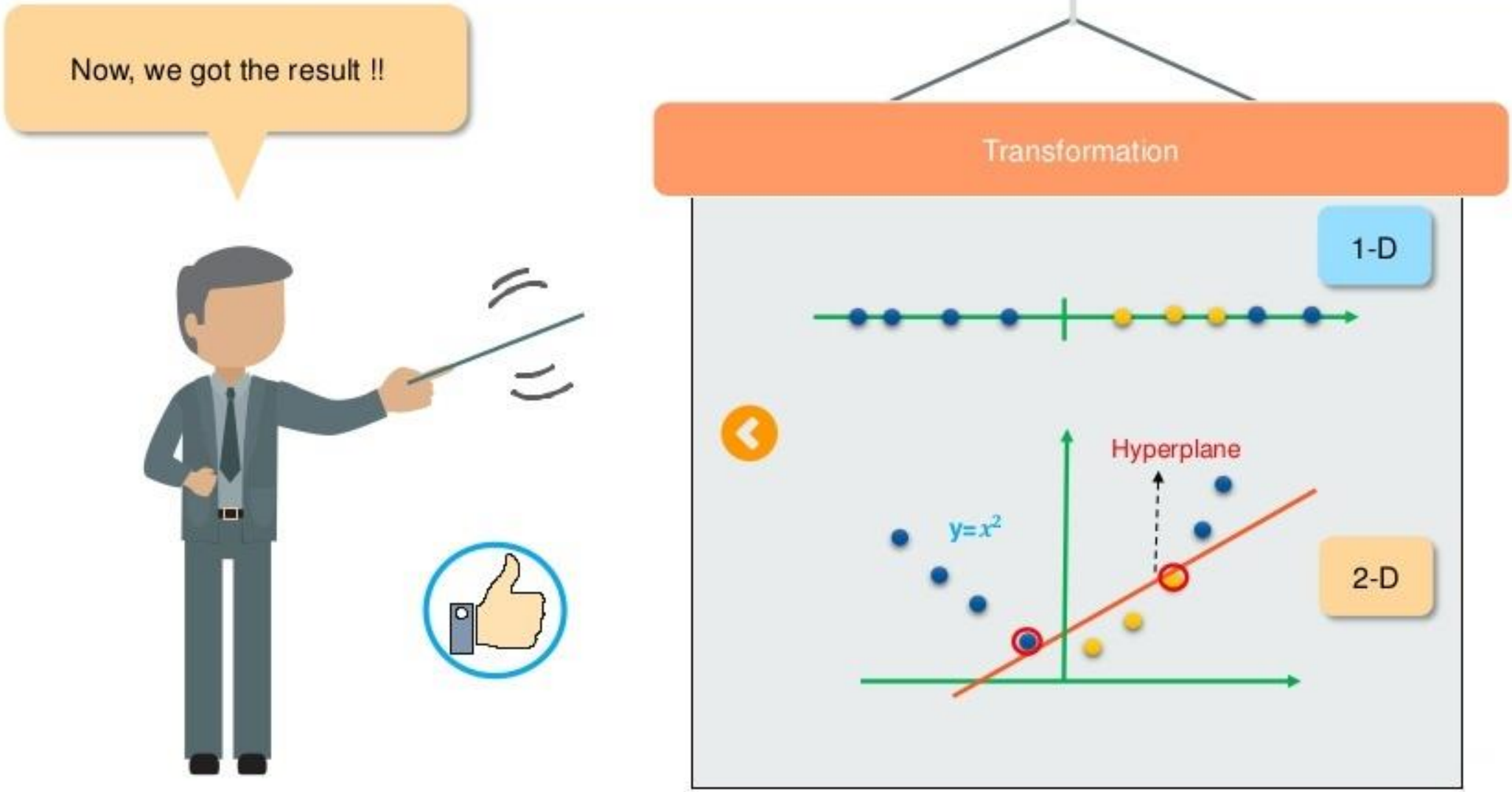
Understanding Support Vector Machine



Understanding Support Vector Machine



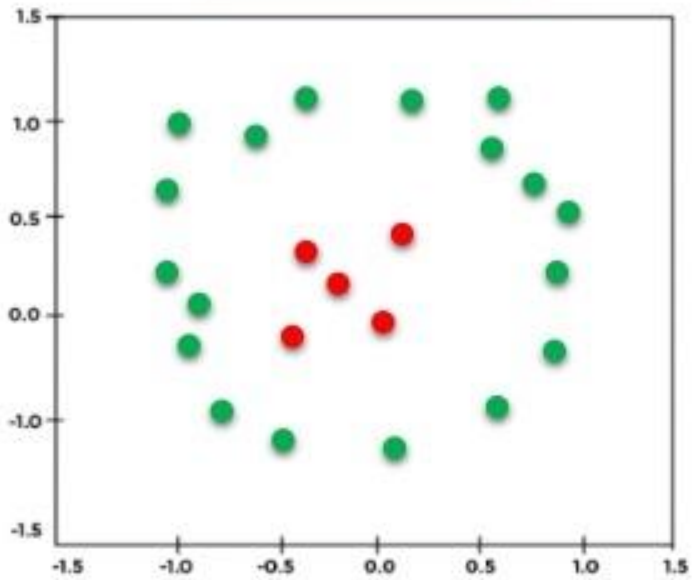
Understanding Support Vector Machine



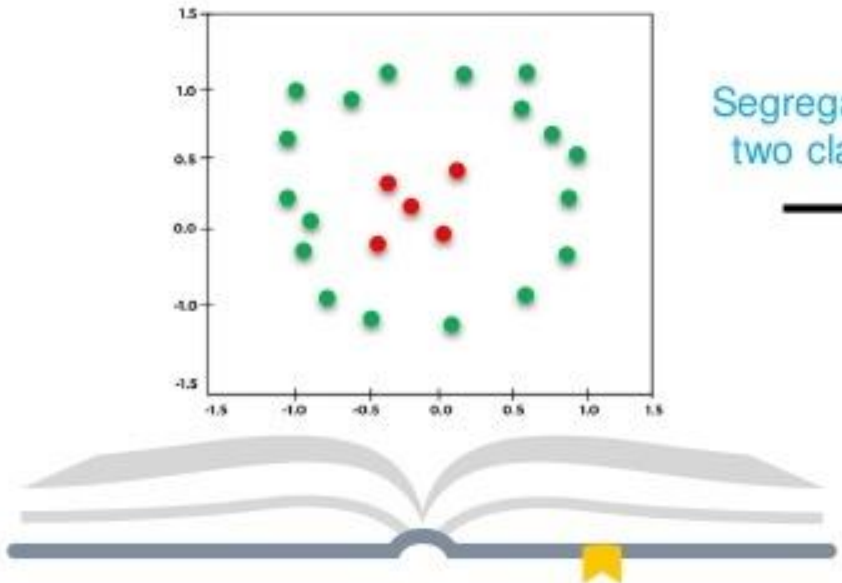
Understanding Support Vector Machine

How to perform SVM for this type of dataset?

Sample Dataset

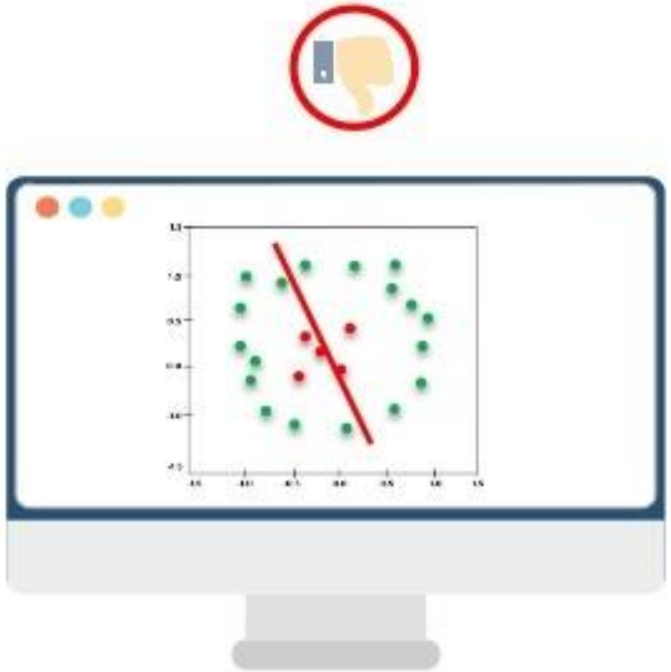


Understanding Support Vector Machine



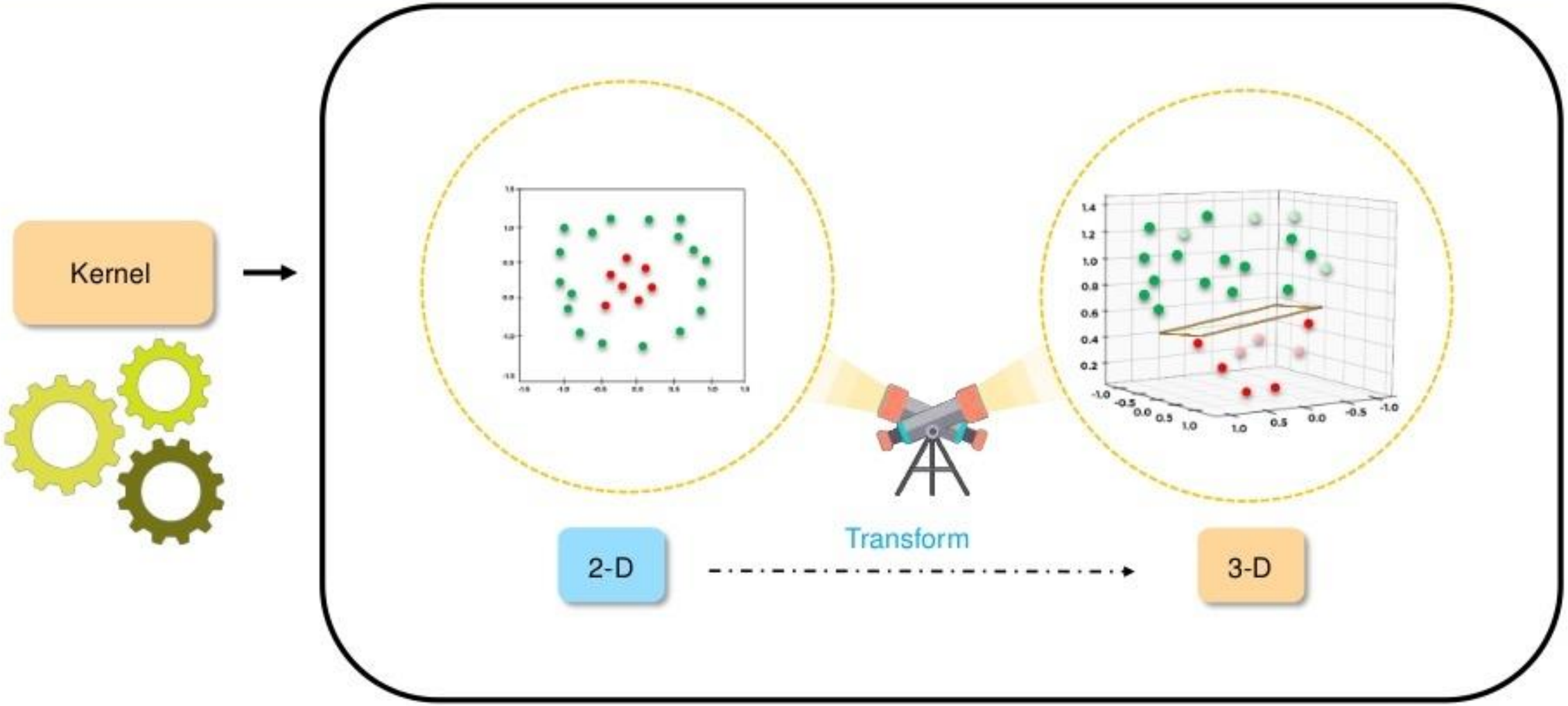
Sample Dataset

Segregate the two classes

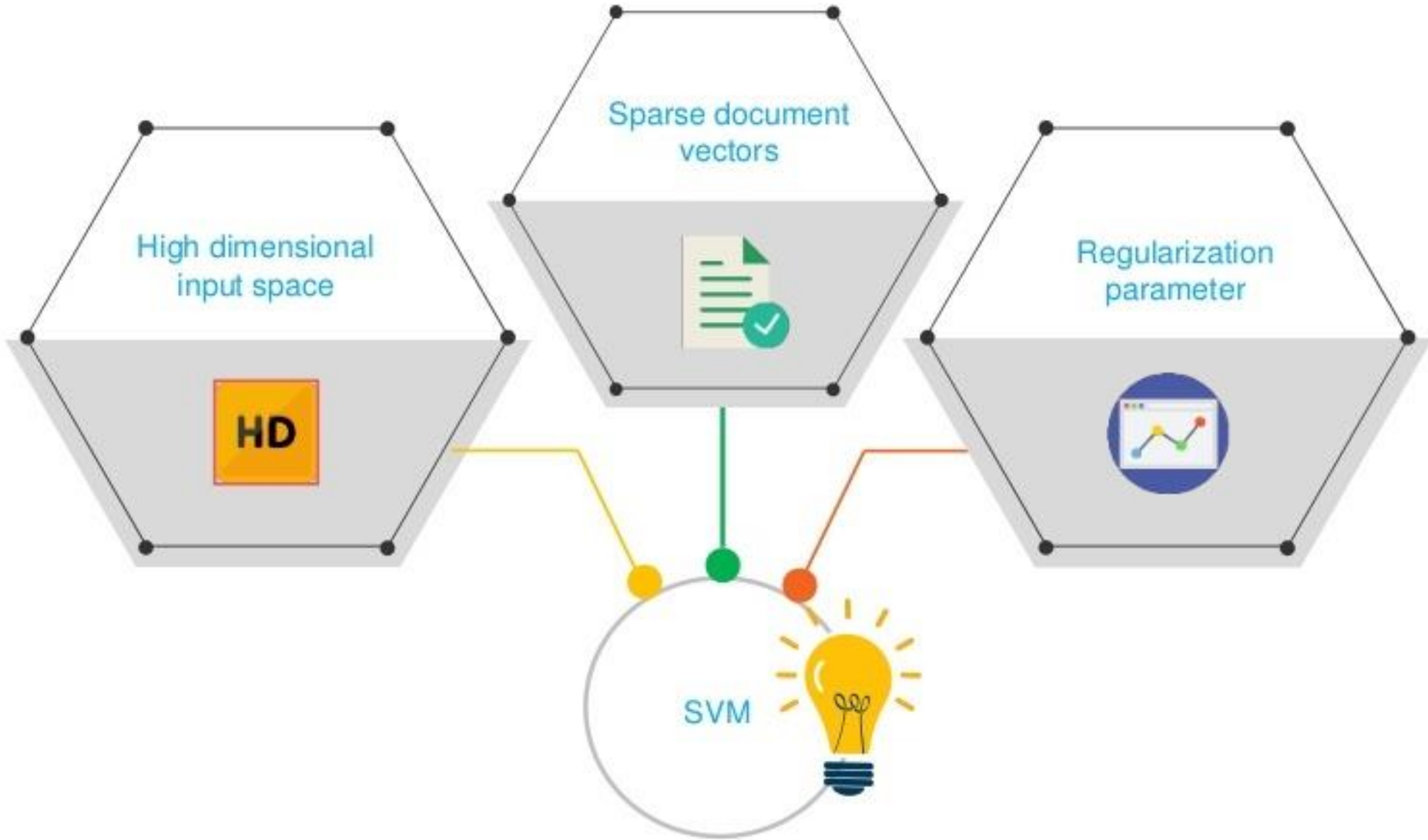


Not an optimal hyperplane

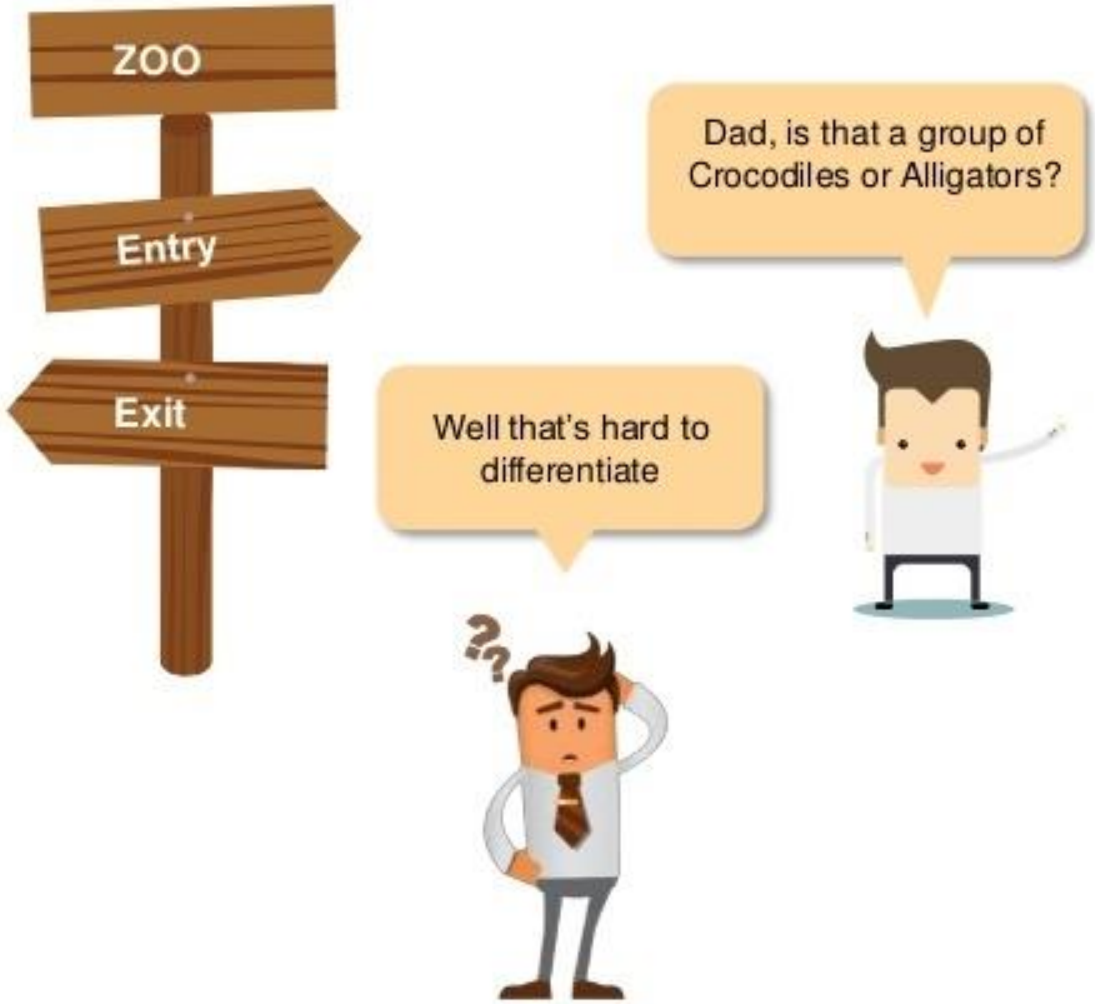
Understanding Support Vector Machine

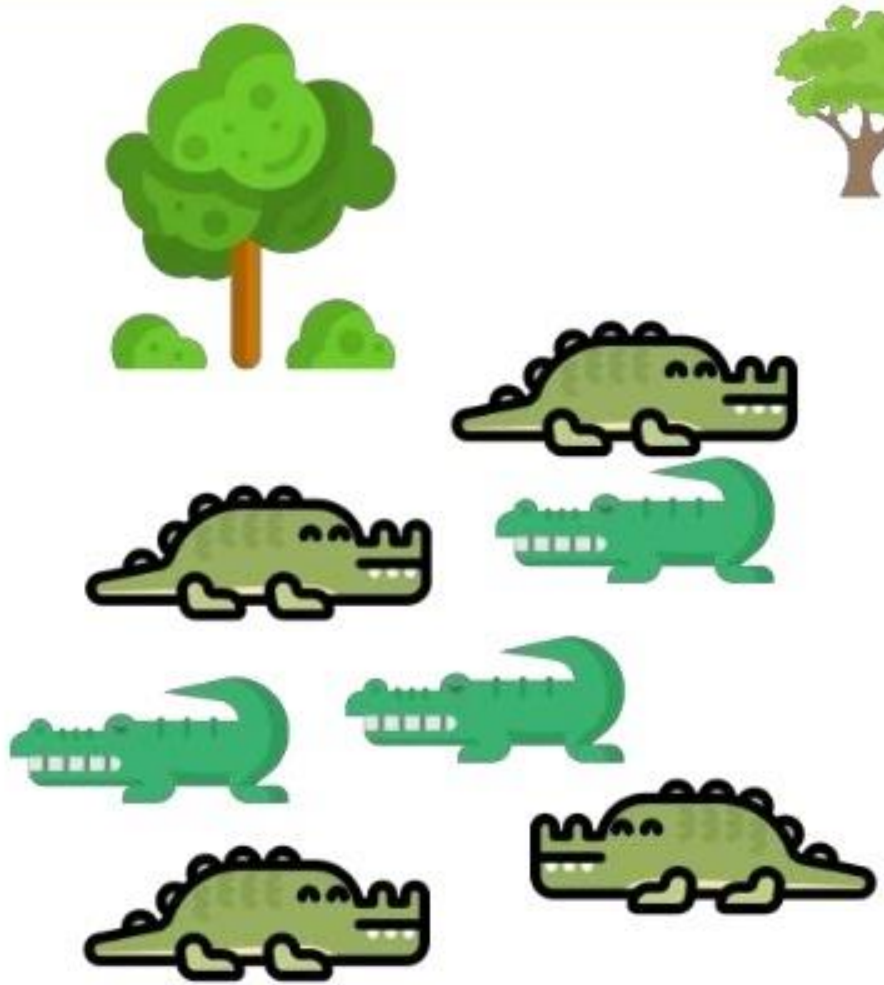


Advantages of Support Vector Machine



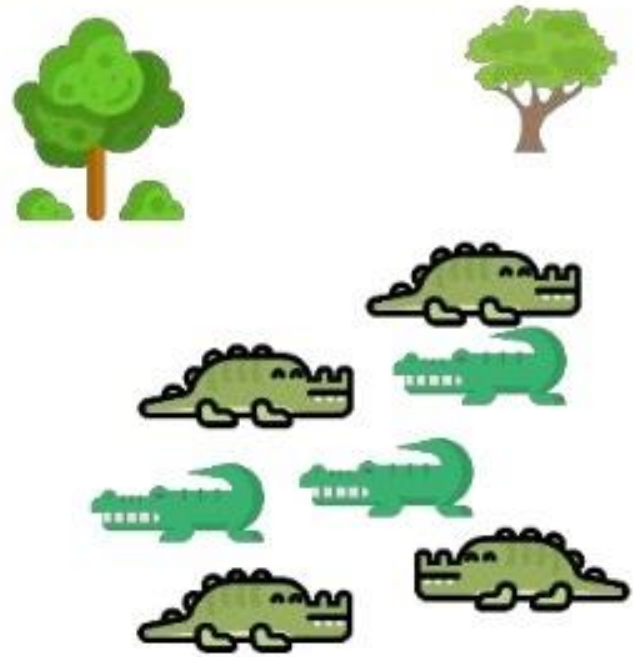
Use case – Problem Statement



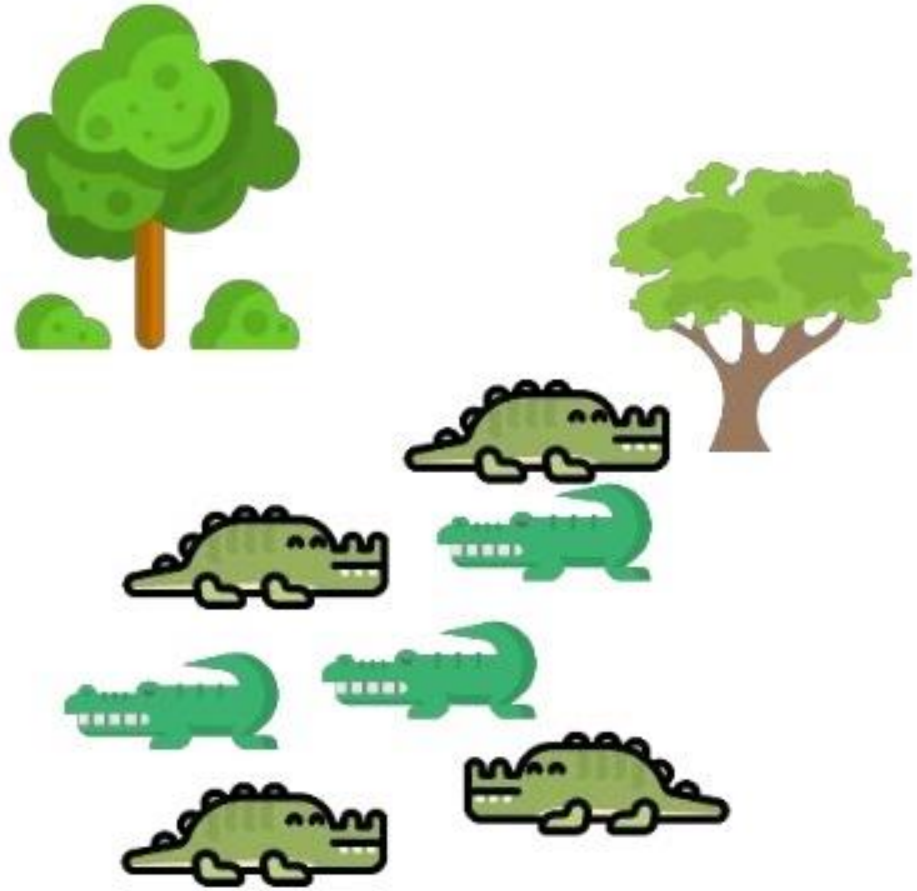


Use case – Problem Statement

Difference		
Size	• Crocodiles are larger in size	• Alligators are smaller in size
Snout Width	• Crocodiles have narrow snout	• Alligators have wider snout



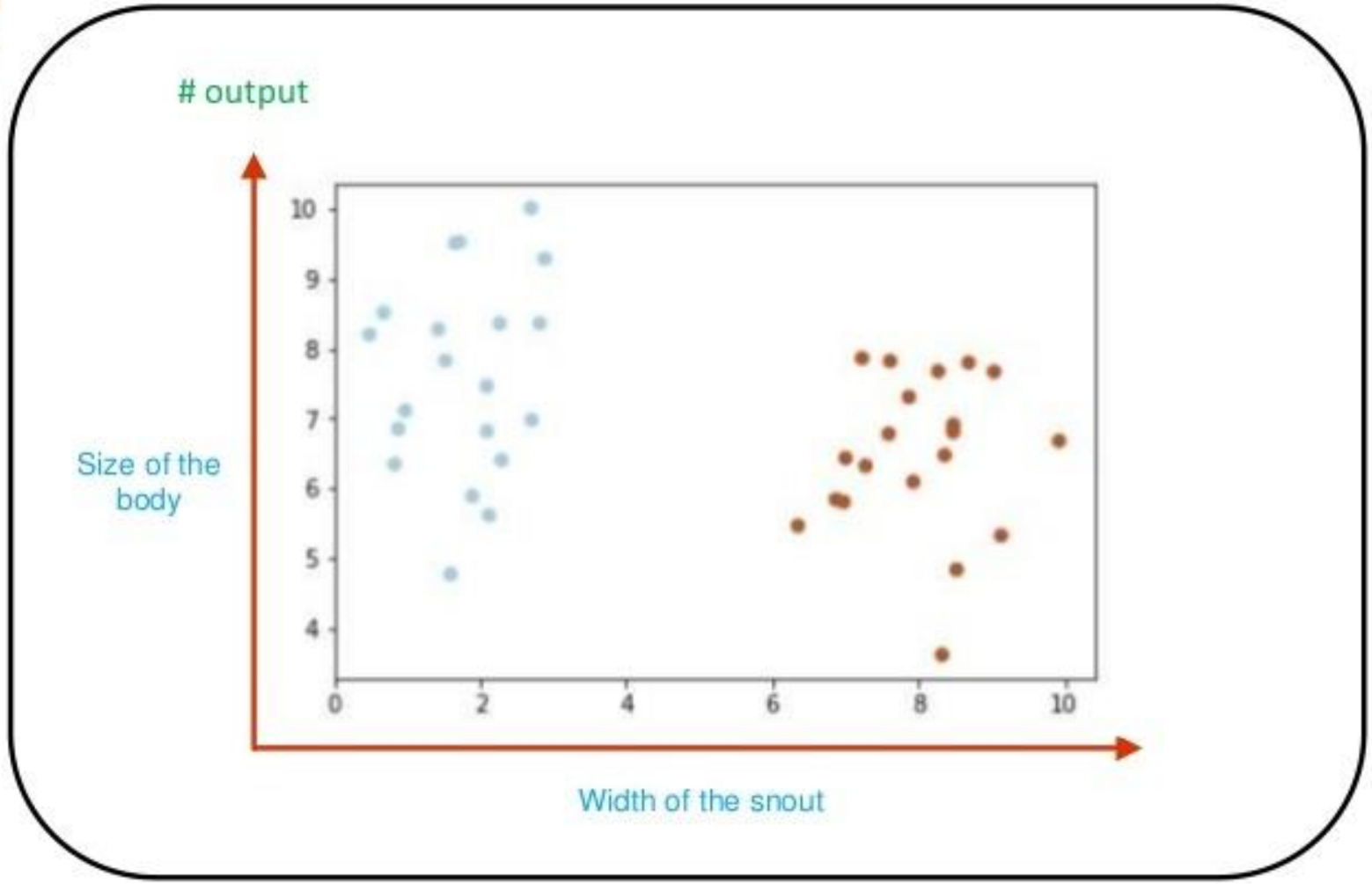
Use case – Problem Statement



Use case - Implementation

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.datasets.samples_generator import make_blobs
# we create 40 separable points
X, y = make_blobs(n_samples=40, centers=2, random_state=20)

# fit the model, don't regularize for illustration purposes
clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)
```

Use case - Implementation



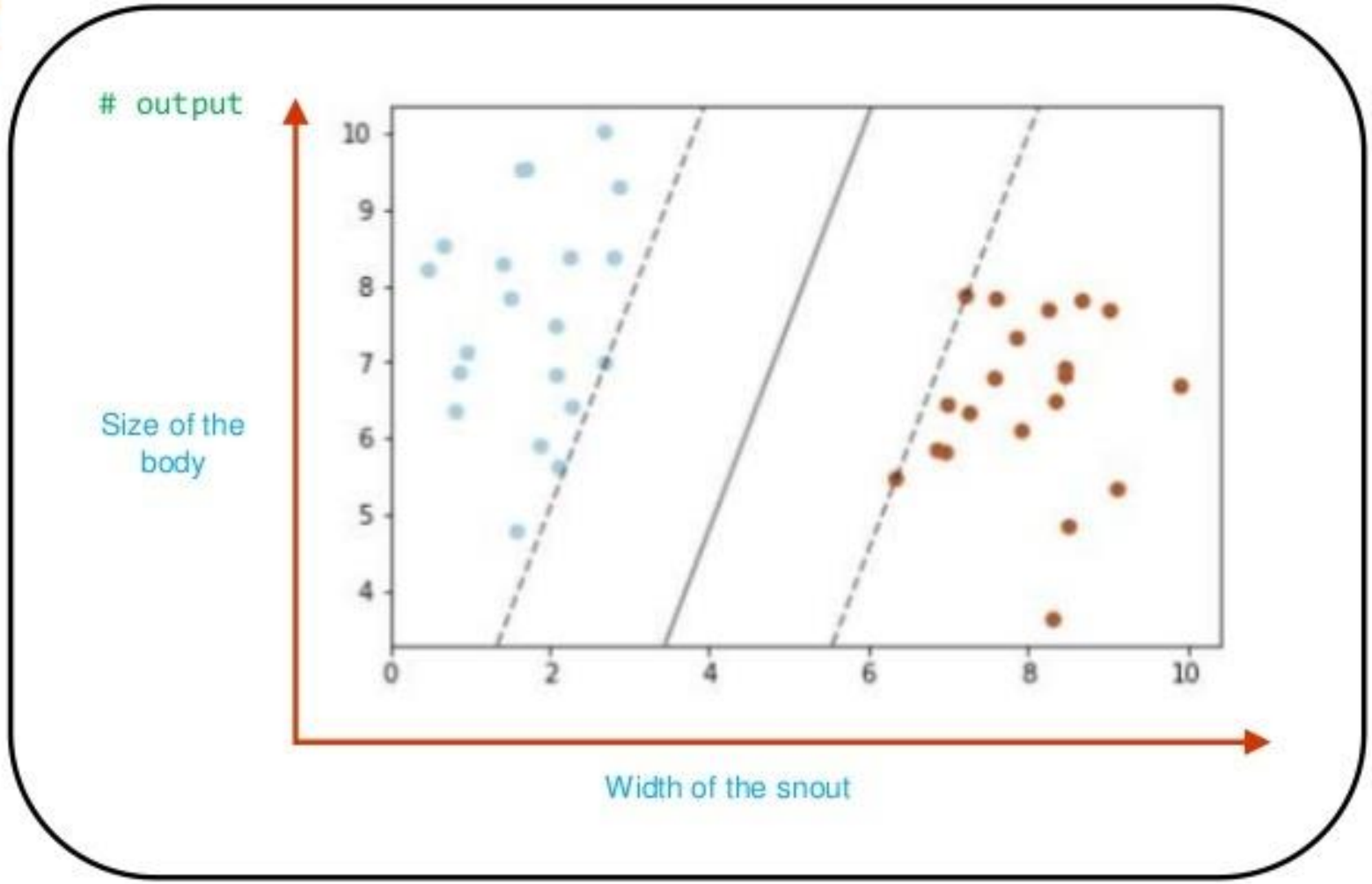
Use case - Implementation

```
# fit the model, don't regularize for illustration purposes
clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y, s=30,cmap=plt.cm.Paired)
plt.show

# plot the decision function
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# create grid to evaluate model
xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = clf.decision_function(xy).reshape(XX.shape)
```

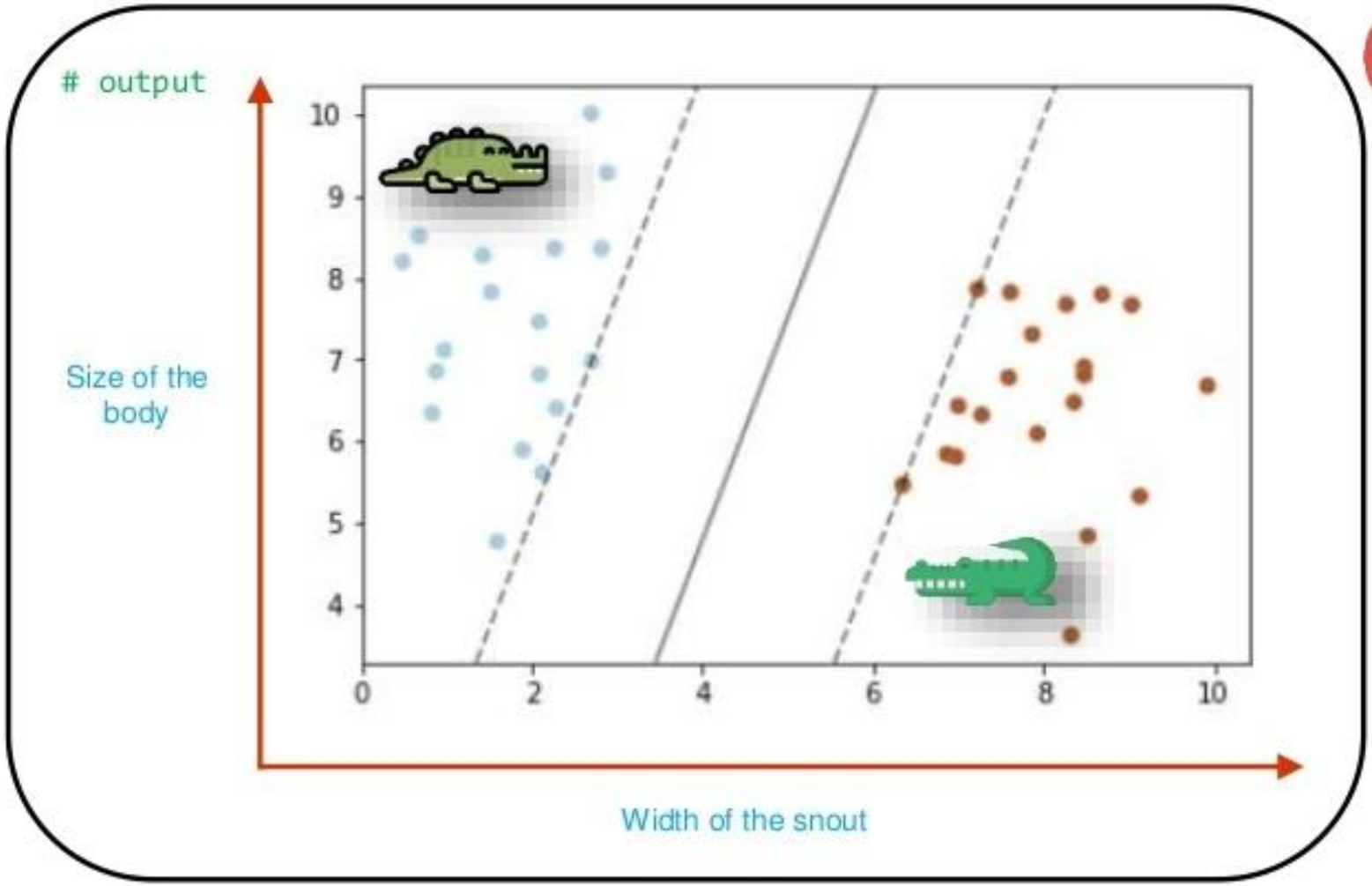
Use case - Implementation

```
# plot decision boundary and margins
ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])
# plot support vectors
ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
s=100,
           linewidth=1, facecolors='none')
plt.show()
```

Use case - Implementation



Use case - Implementation



Conclusion

Congratulations!

We have demonstrated Support vector machine by segregating the two classes

Where the blue data points represents crocodiles and the brown data points represents alligators

The hands on example will help you to encounter any Support Vector Machine project in future.

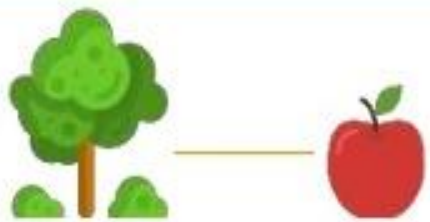
What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision

What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision

Decision Tree 1



Output 1

What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision

Decision Tree 1

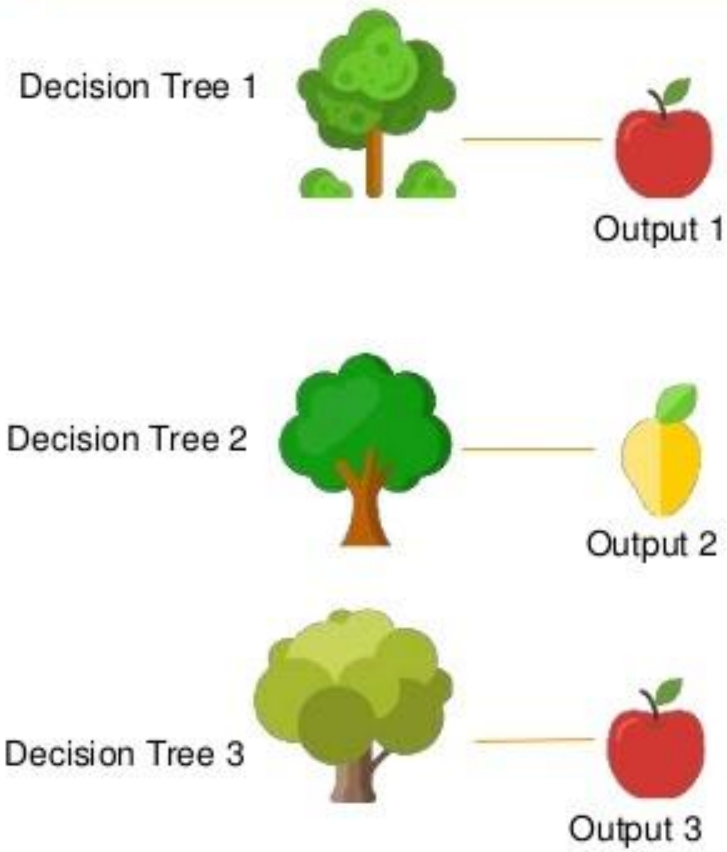
Output 1

Decision Tree 2

Output 2

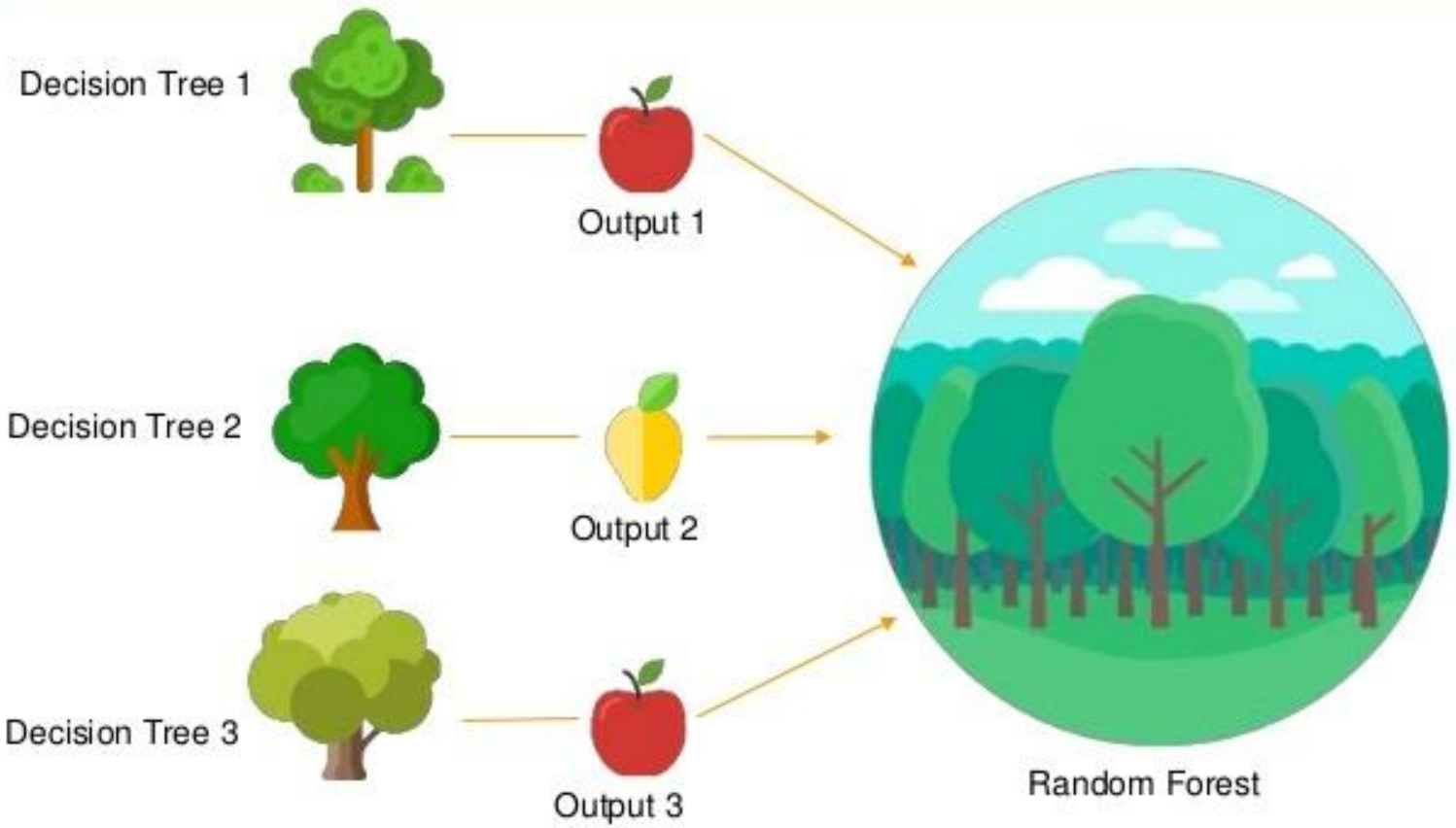
What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision



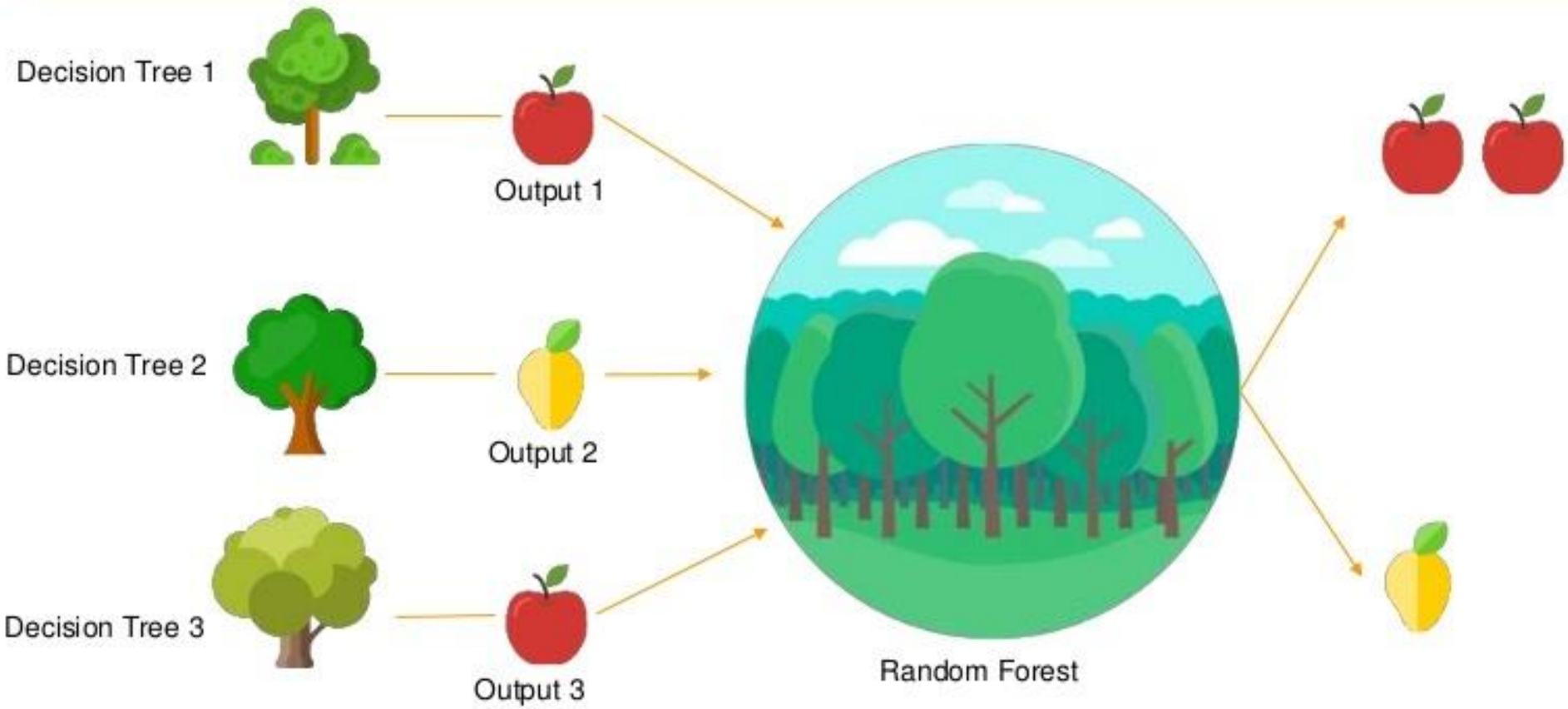
What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision



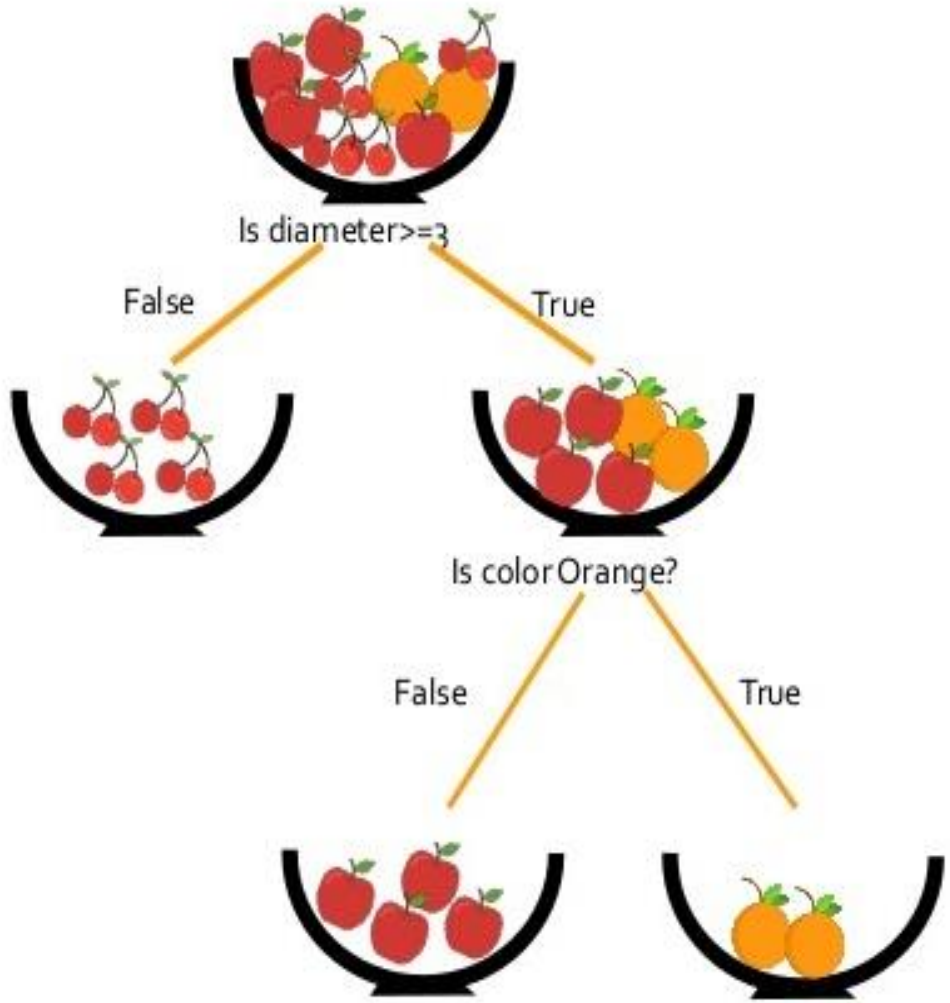
What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.
The Decision of the majority of the trees is chosen by the random forest as the final decision



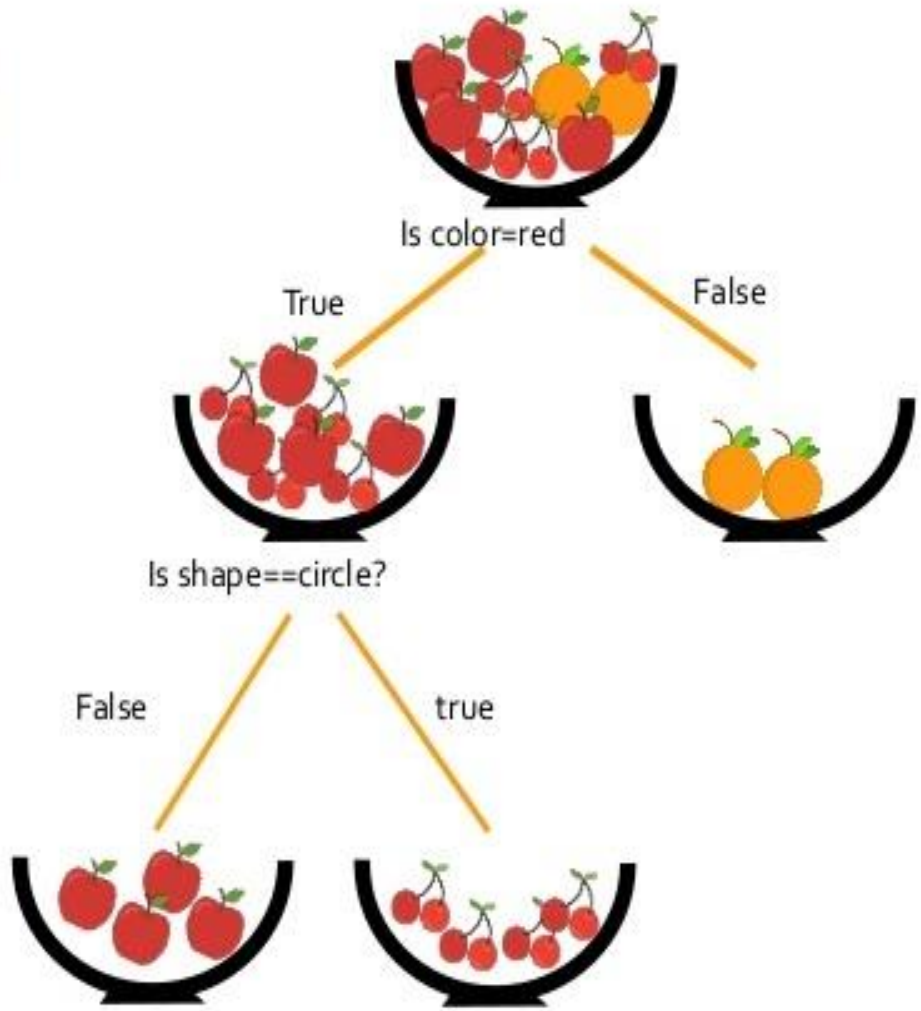
How does a Random Forest work?

Let this be Tree 1



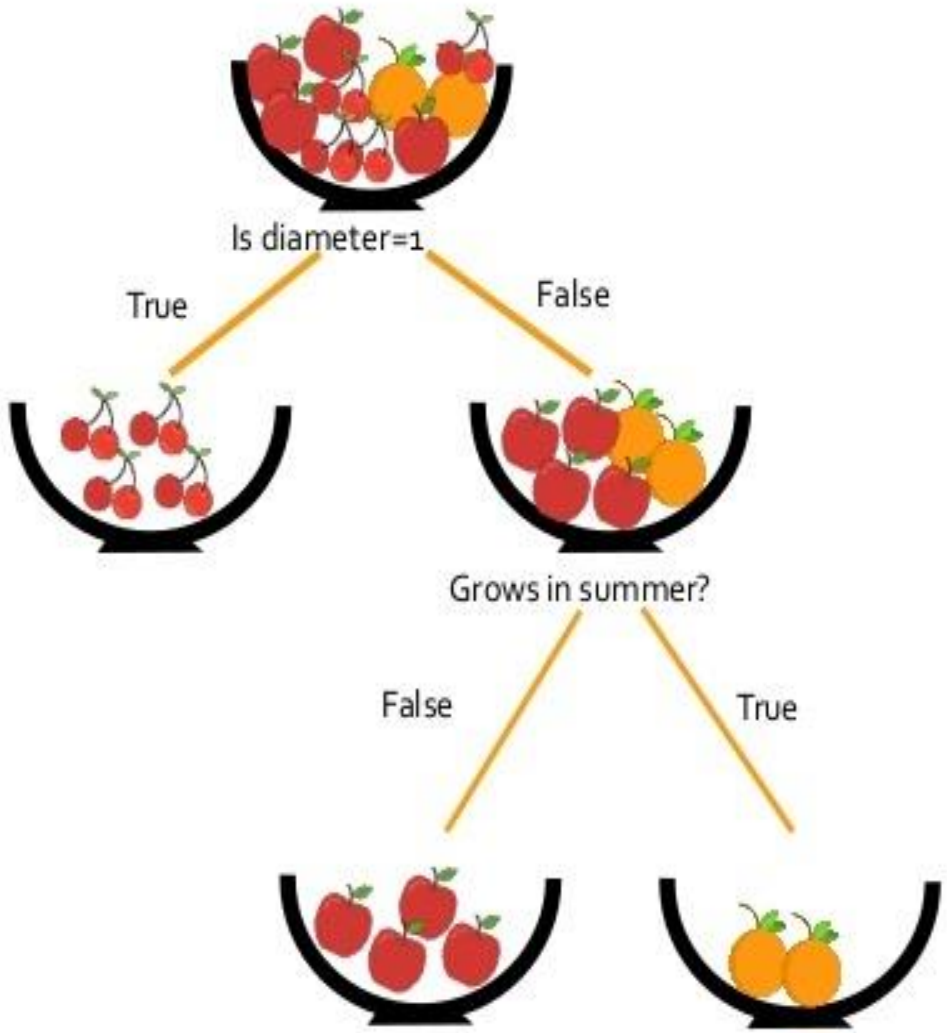
How does a Random Forest work?

Let this be Tree 2

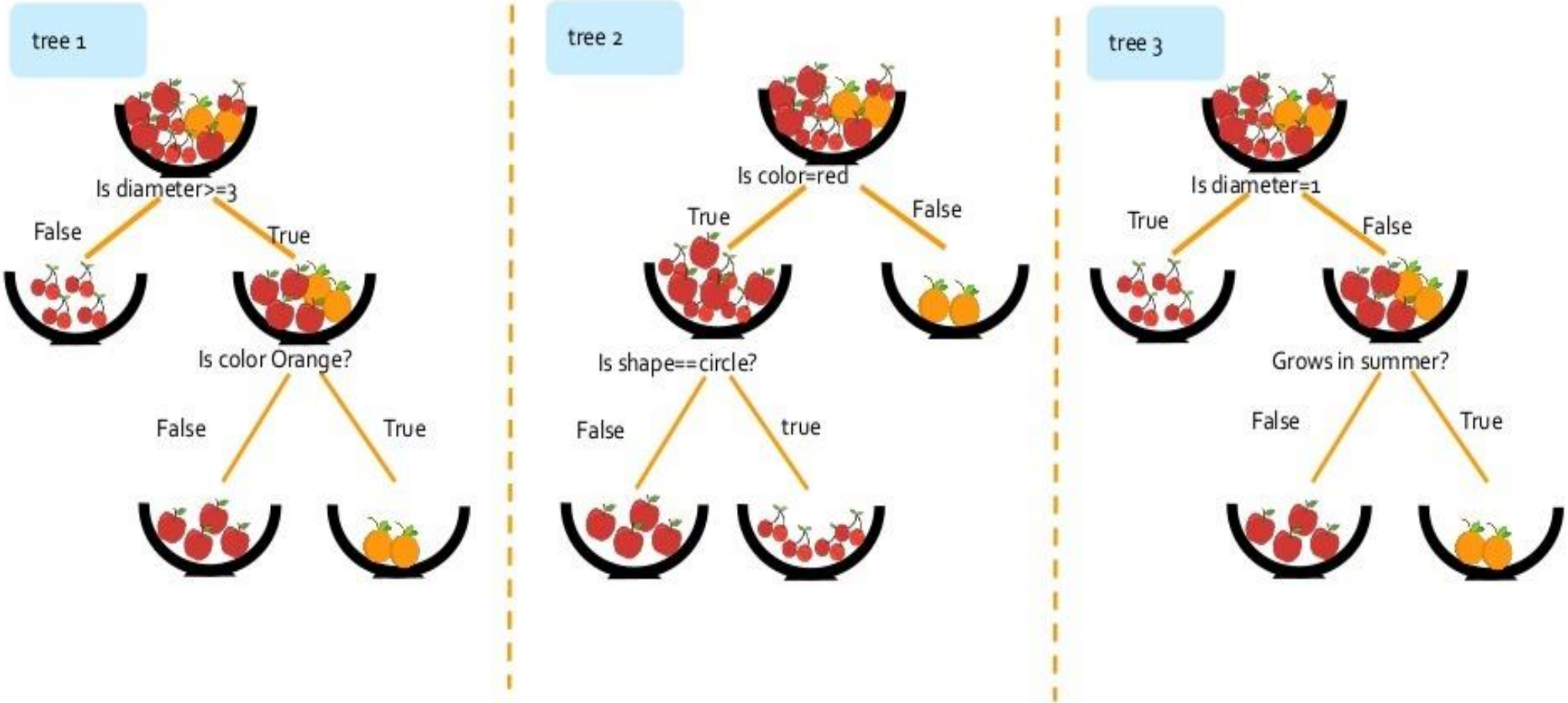


How does a Random Forest work?

Let this be Tree 3



How does a Random Forest work?



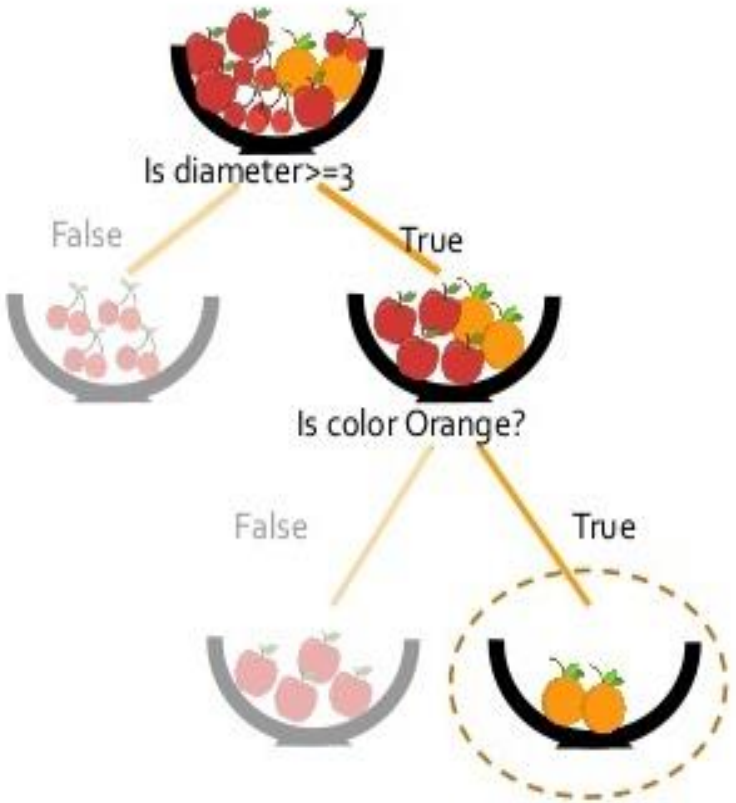
How does a Random Forest work?

Now Lets try to classify this fruit

How does a Random Forest work?

Tree 1 classifies it as an orange

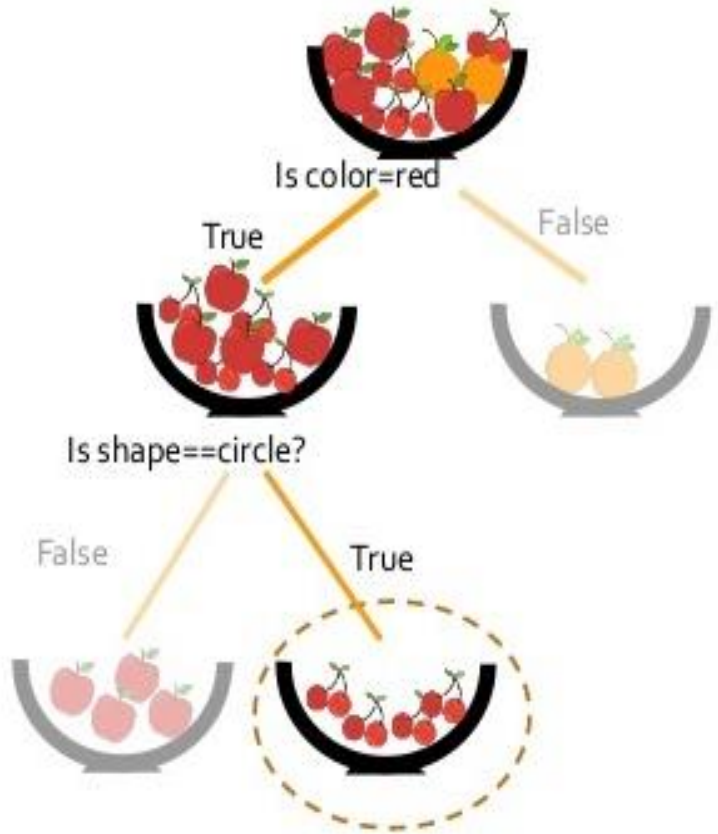
Diameter = 3
Colour = orange
Grows in summer = yes
SHAPE = CIRCLE



How does a Random Forest work?

Tree 2 classifies it as cherries

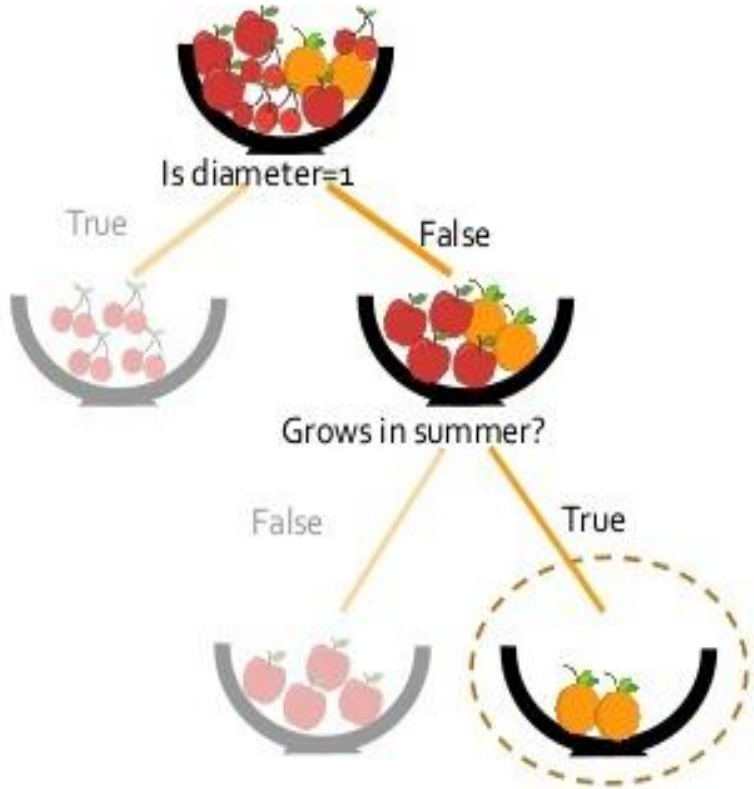
Diameter = 3
Colour = orange
Grows in summer = yes
SHAPE = CIRCLE



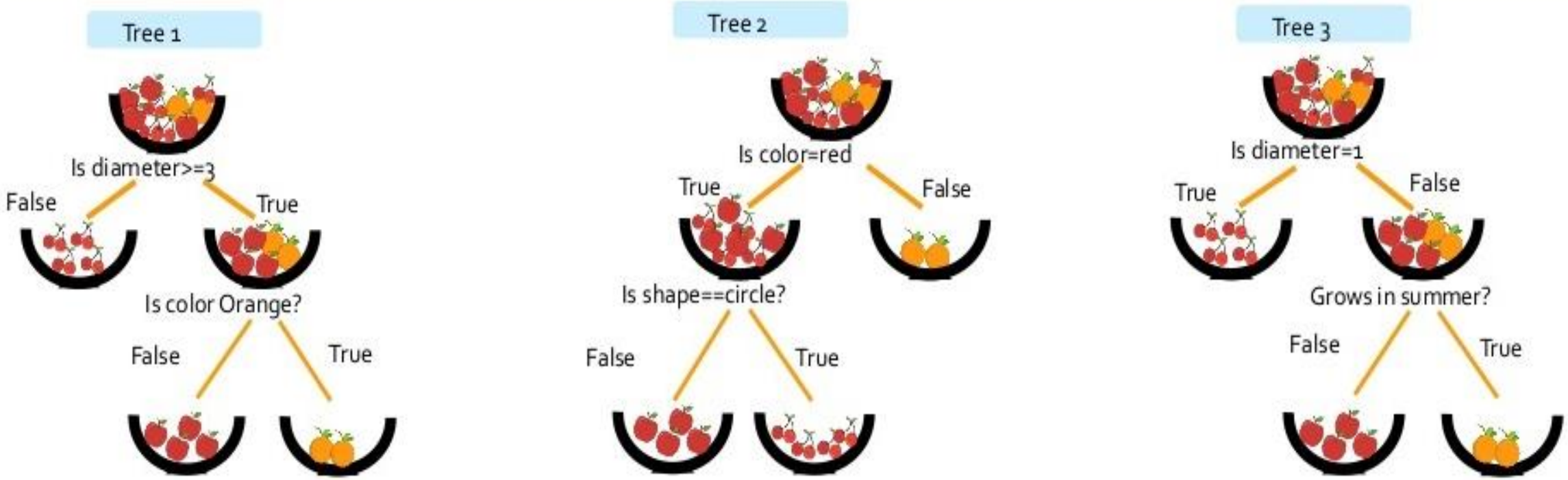
How does a Random Forest work?

Tree 3 classifies it as orange

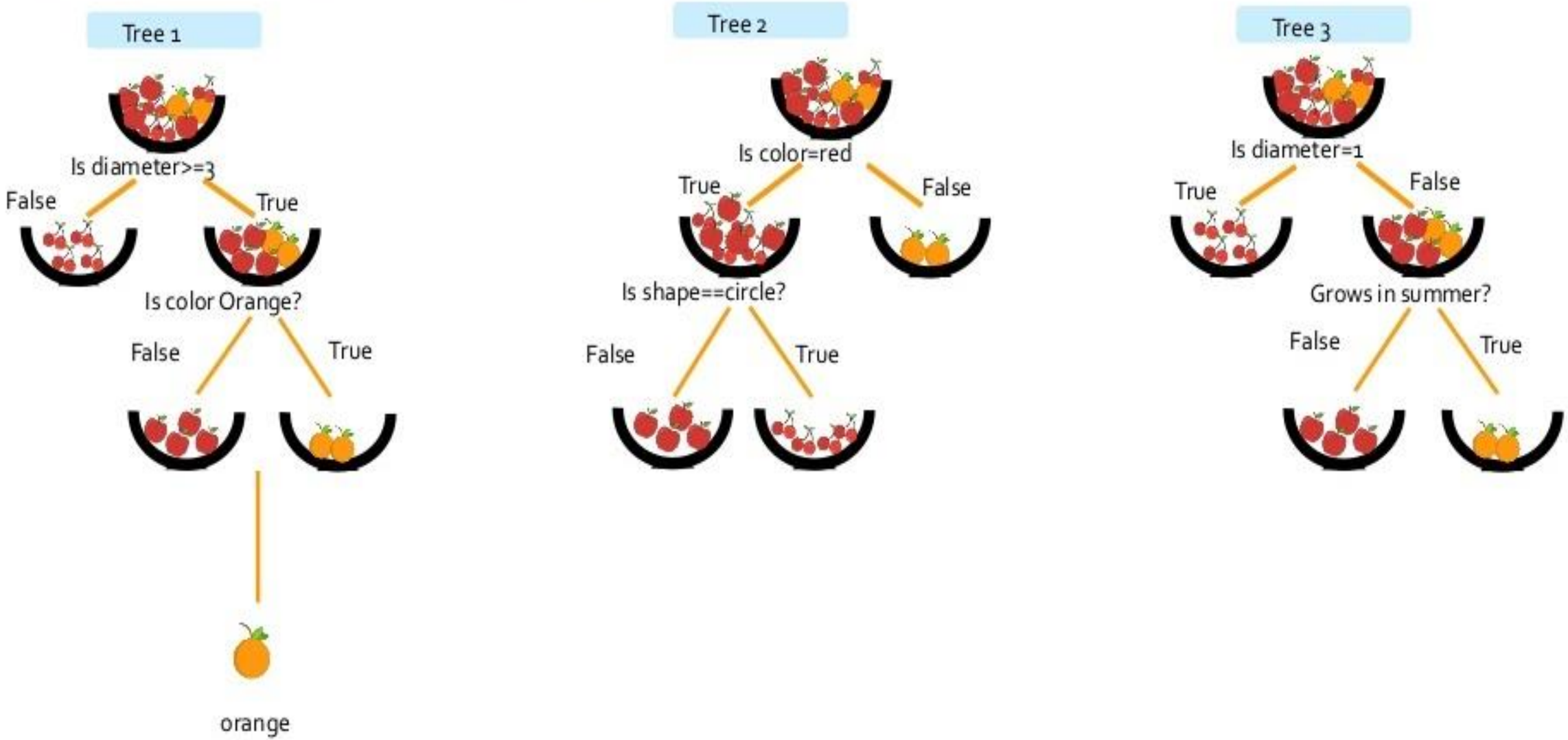
Diameter = 3
Colour = orange
Grows in summer = yes
SHAPE = CIRCLE



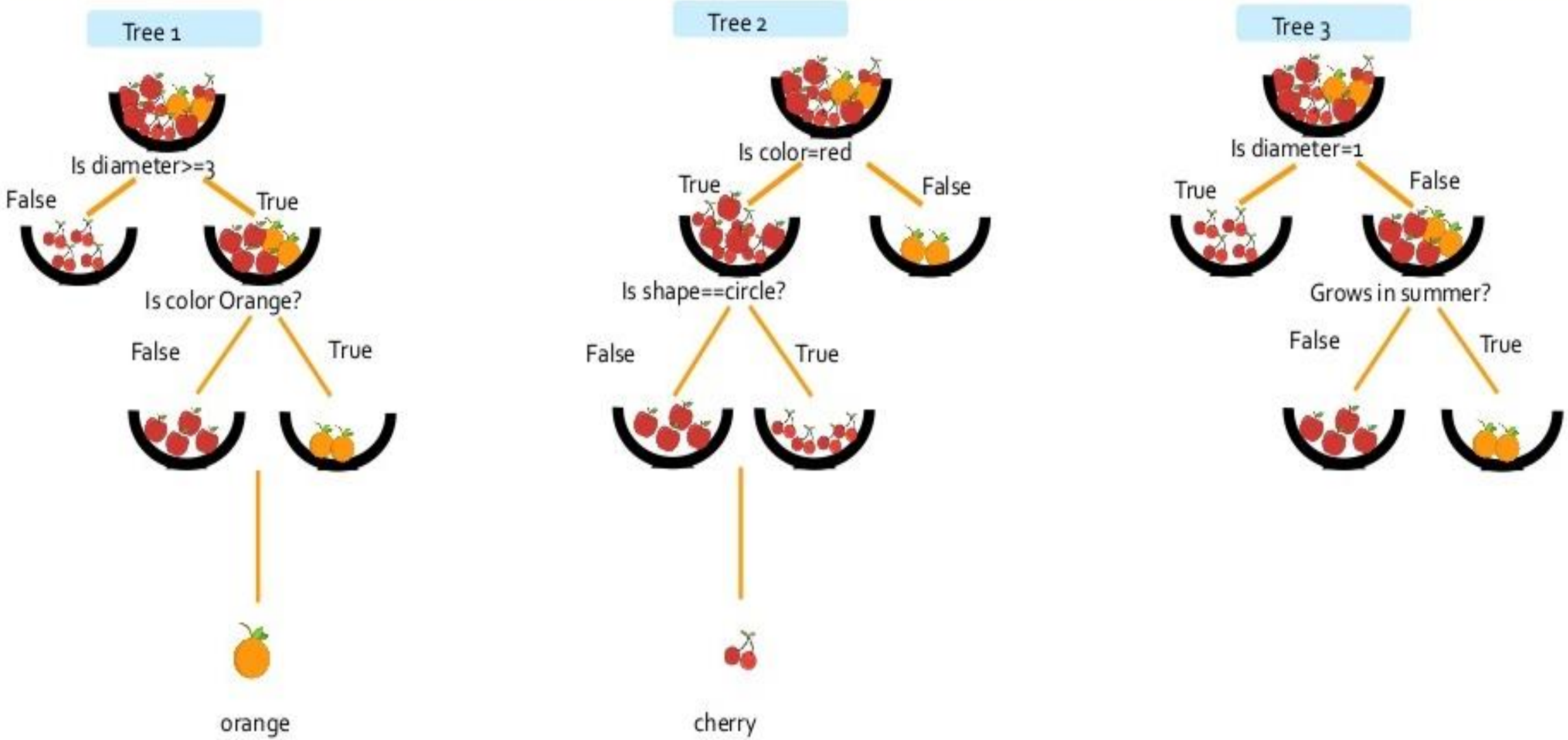
How does a Random Forest work?



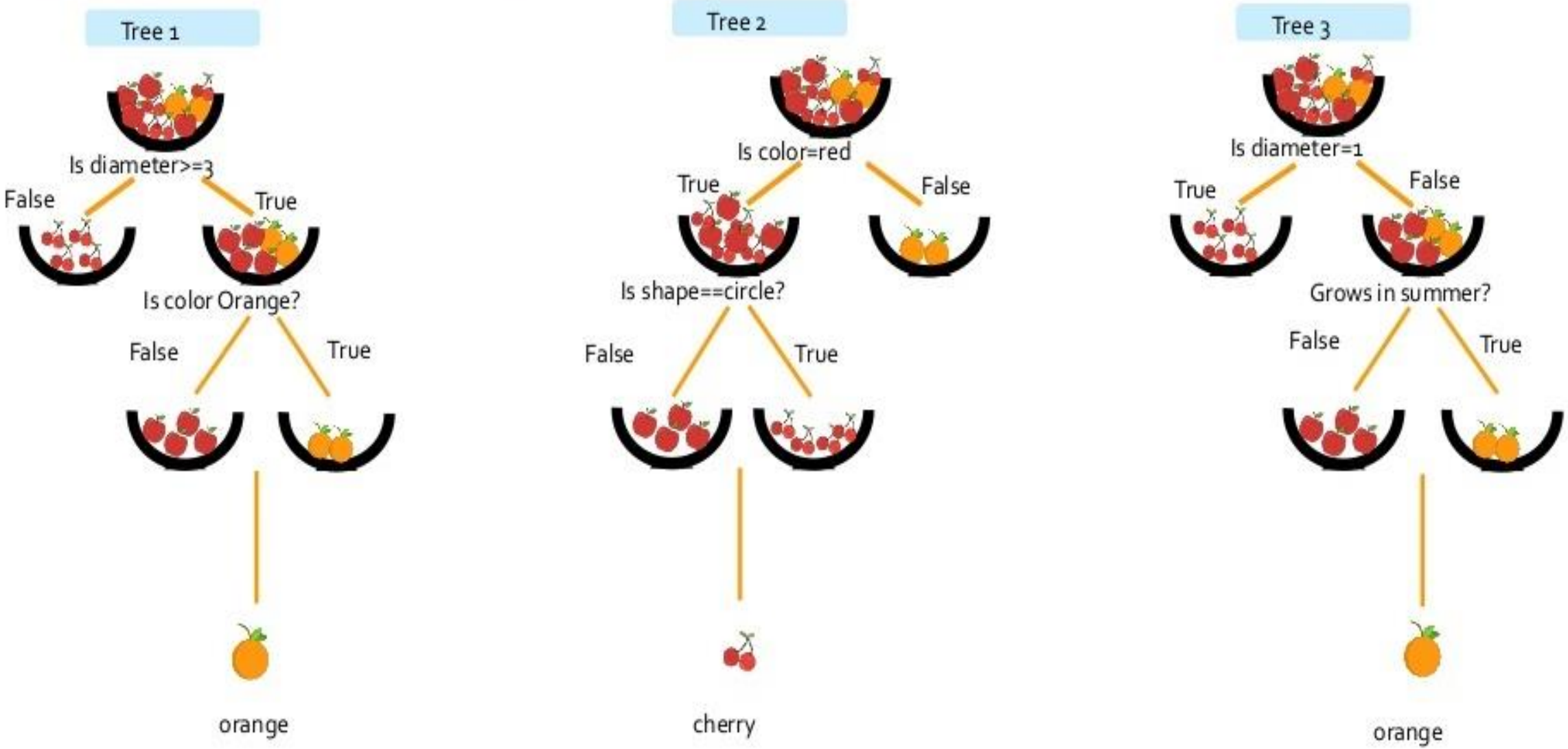
How does a Random Forest work?



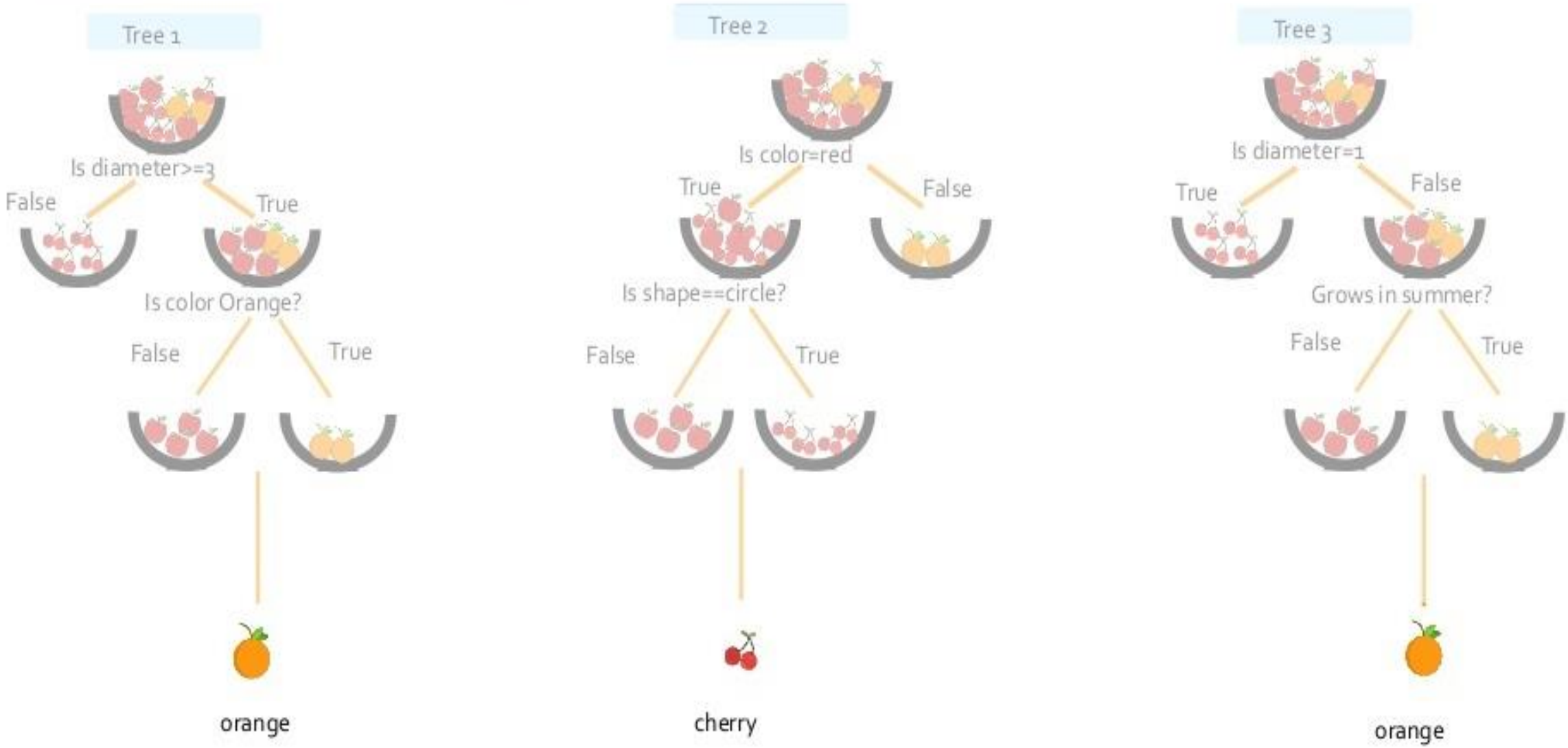
How does a Random Forest work?



How does a Random Forest work?



How does a Random Forest work?



How does a Random Forest work?

How does a Random Forest work?

So the fruit is classified as an orange

How does a Random Forest work?

So the fruit is classified as an orange

What is K-Means Clustering?

What is K-Means Clustering?

What is K-Means Clustering?

What is K-Means Clustering?

k-means performs division of objects into clusters which are “similar” between them and are “dissimilar” to the objects belonging to another cluster

What is K-Means Clustering?

Can you explain this with an example?

What is K-Means Clustering?

Can you explain this with an example?

Sure. For understanding K-Means in a better way, let's take an example of **Cricket**

What is K-Means Clustering?

Task: Identify bowlers and batsmen

What is K-Means Clustering?

Task: Identify bowlers and batsmen

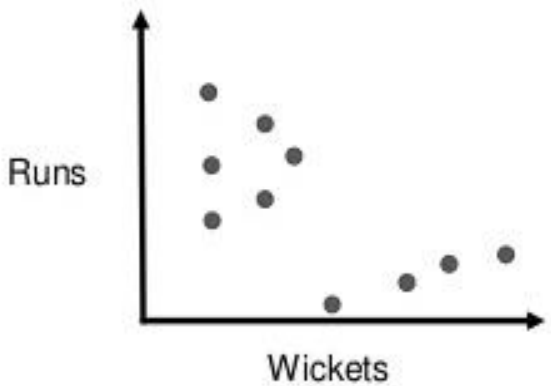
- The data contains runs and wickets gained in the last 10 matches
- So, the bowler will have more wickets and the batsmen will have higher runs

What is K-Means Clustering?

Assign data points

Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**

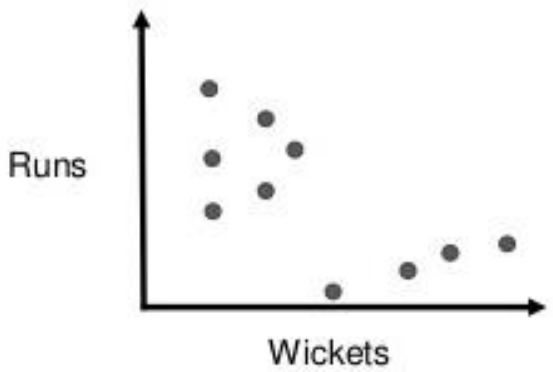


What is K-Means Clustering?

Assign data points

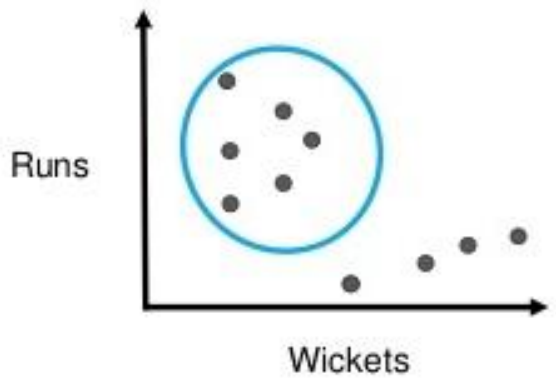
Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**



Cluster 1

We can see that this cluster has players with high runs and low wickets

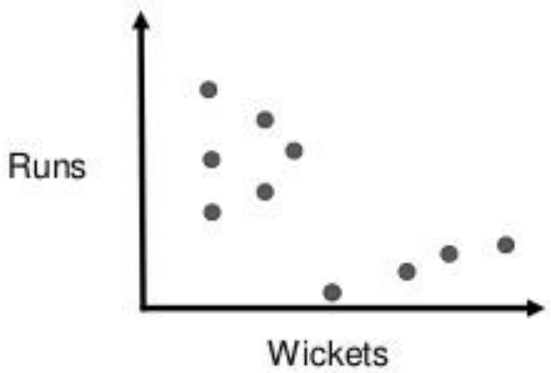


What is K-Means Clustering?

Assign data points

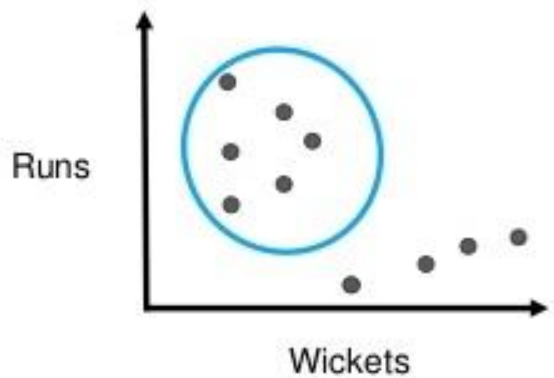
Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**



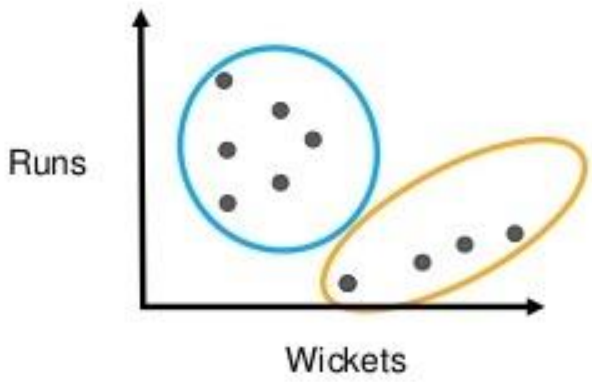
Cluster 1

We can see that this cluster has players with high runs and low wickets



Cluster 2

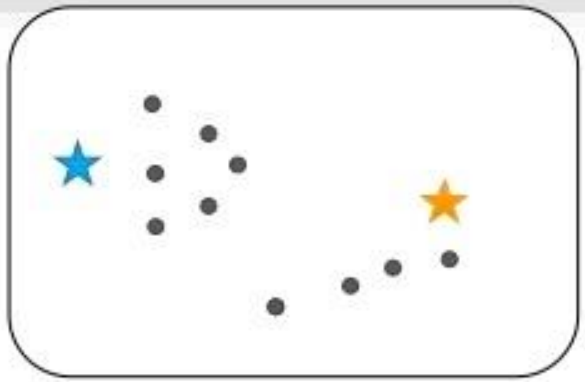
And here, we can see that this cluster has players with high wickets and low wickets



What is K-Means Clustering?

Consider the same data set of cricket

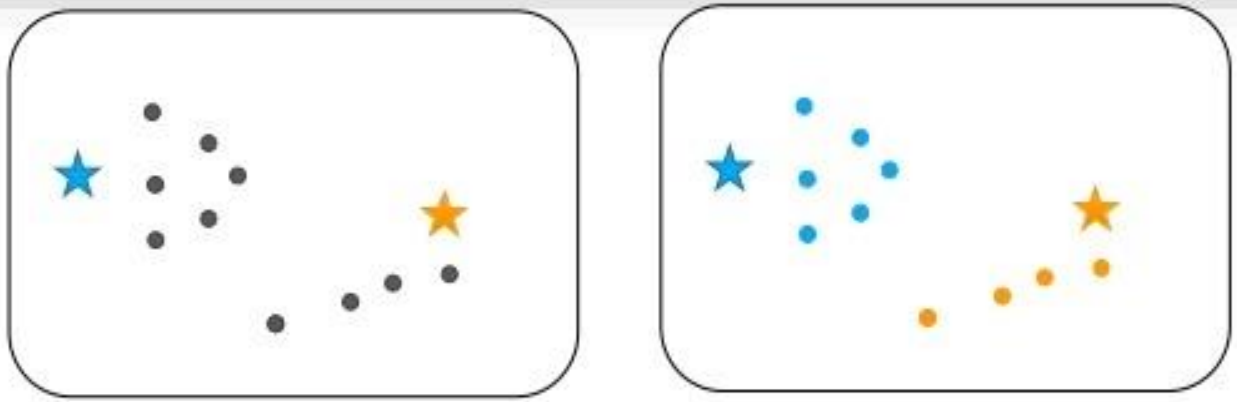
Solve the problem using K-Means



What is K-Means Clustering?

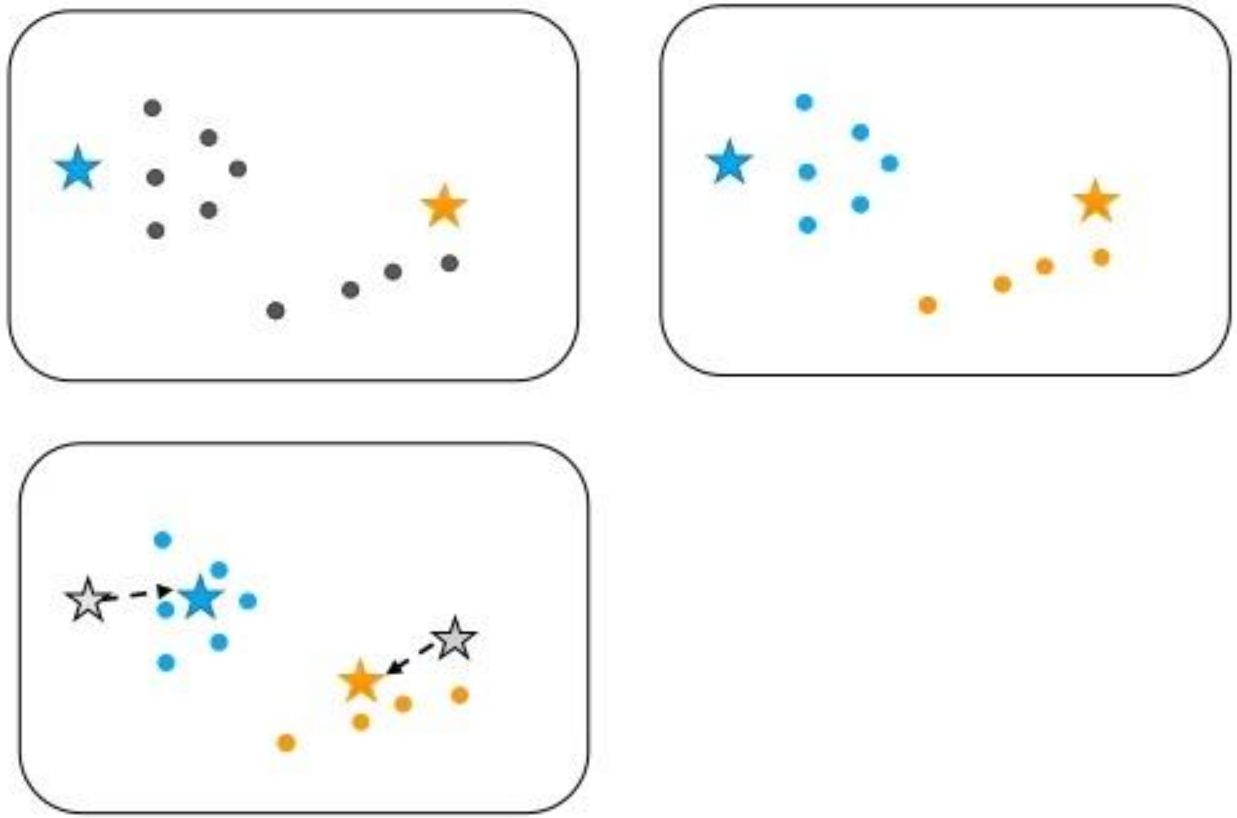
Initially, two centroids are assigned randomly

Euclidean distance to find out which centroid is closest to each data point and the data points are assigned to the corresponding centroids



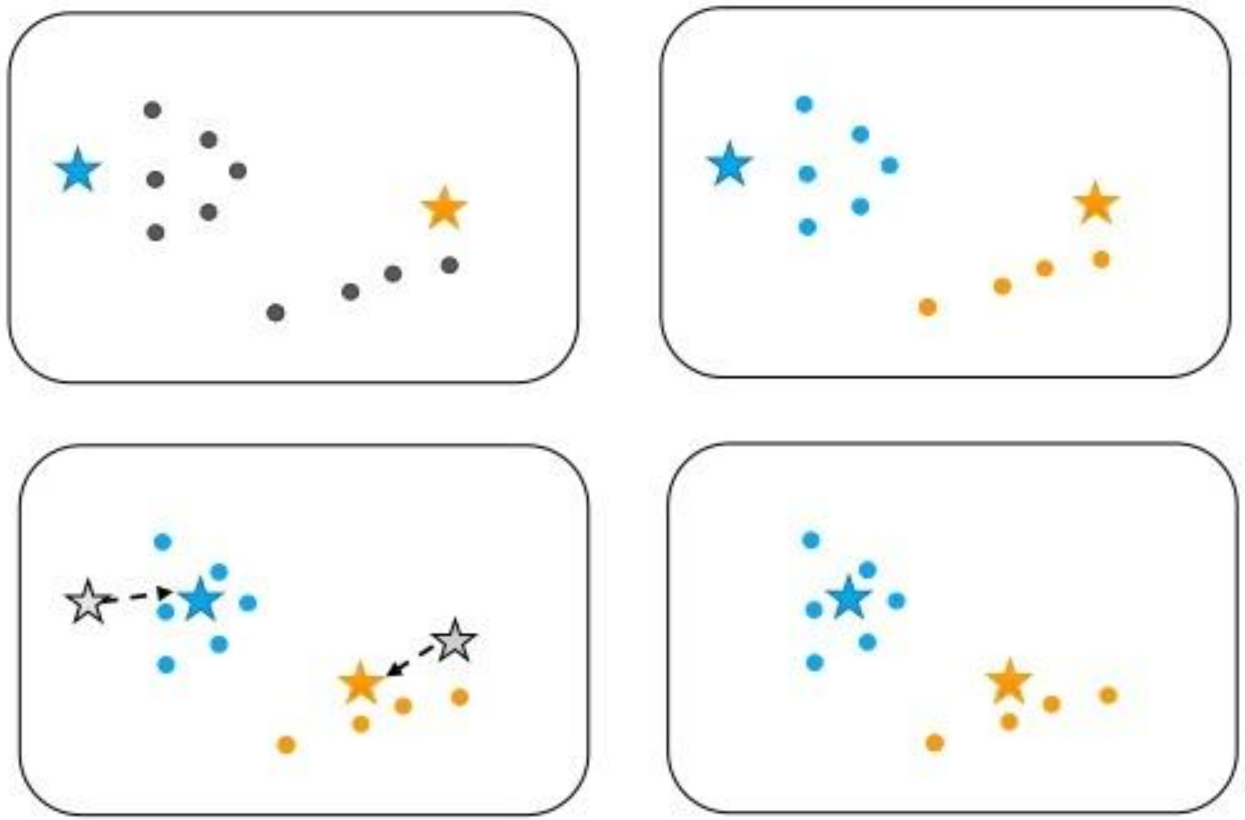
What is K-Means Clustering?

Reposition the two centroids for optimization.



What is K-Means Clustering?

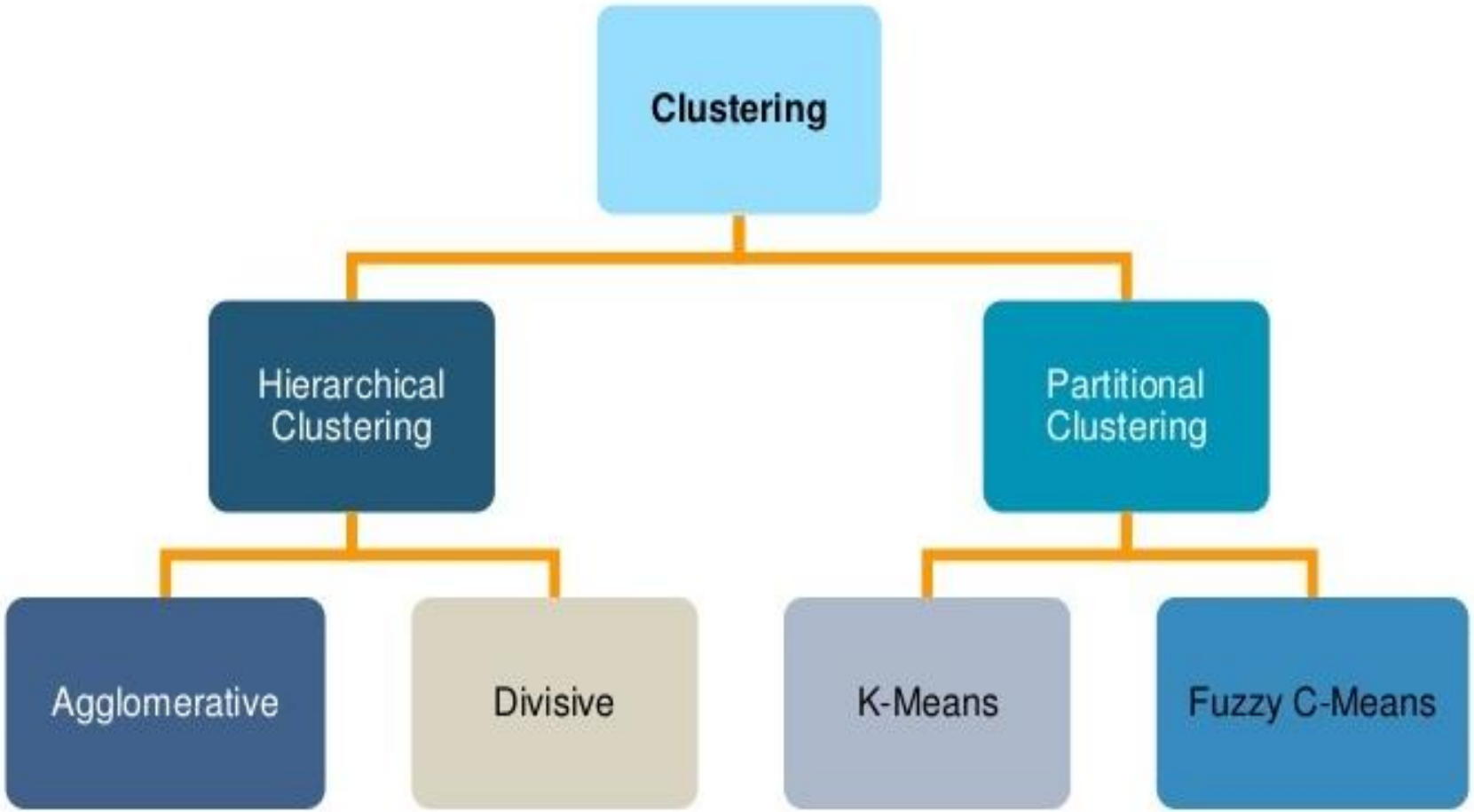
The process is iteratively repeated until our centroids become static



What's in it for you?

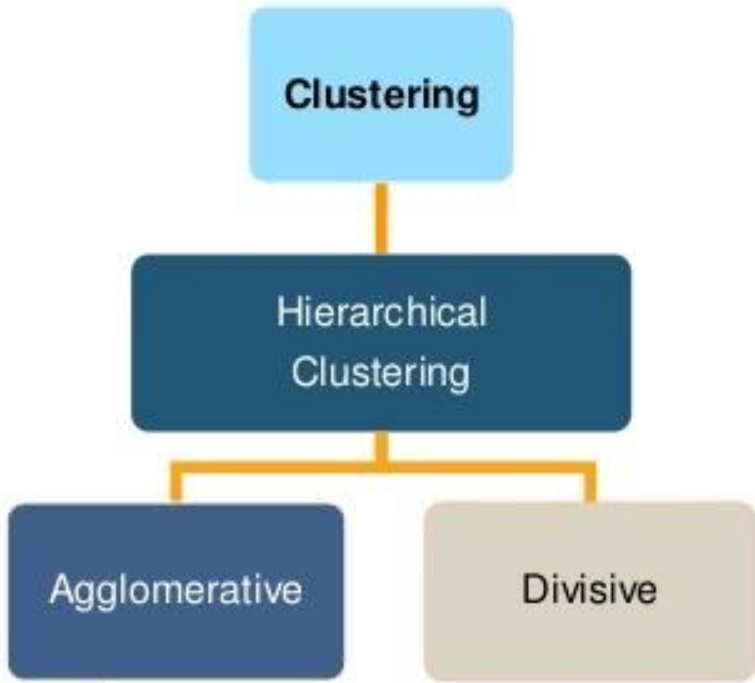
- Types of Clustering
- What is K-Means Clustering?
- Applications of K-Means clustering
- Common distance measure
- How does K-Means clustering work?
- K-Means Clustering Algorithm
- Demo: K-Means Clustering
- Use Case: Color Compression

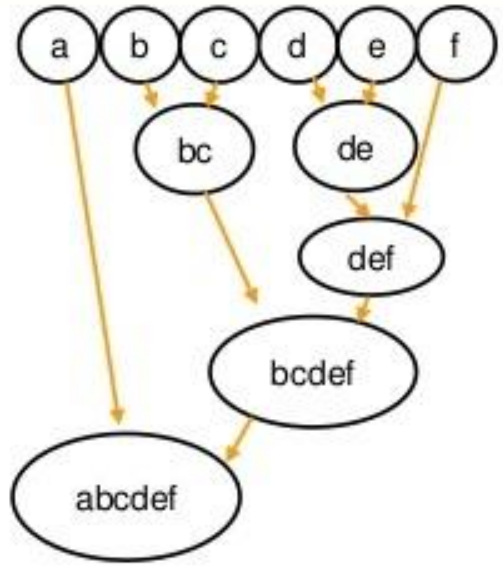
Types of Clustering



Types of Clustering

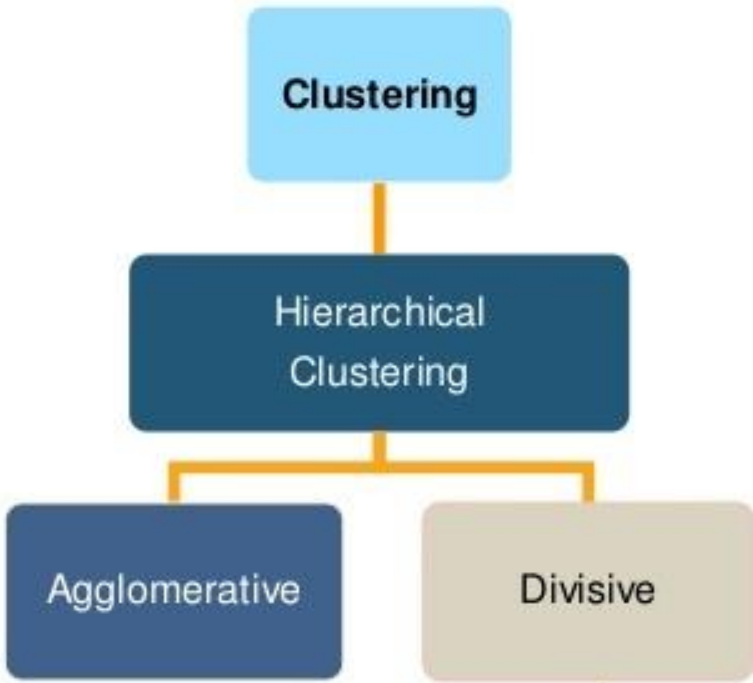
Types of Clustering

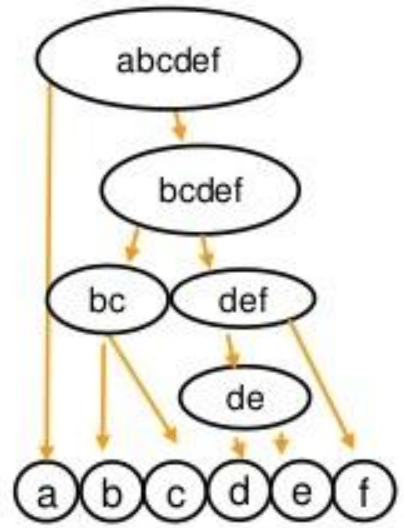




"Bottom up" approach: Begin with each element as a separate cluster and merge them into successively larger clusters

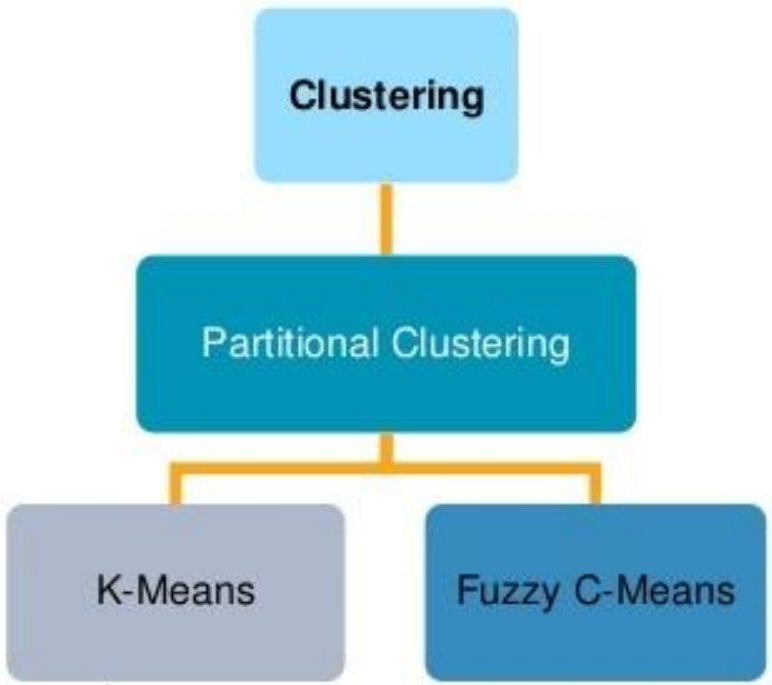
Types of Clustering

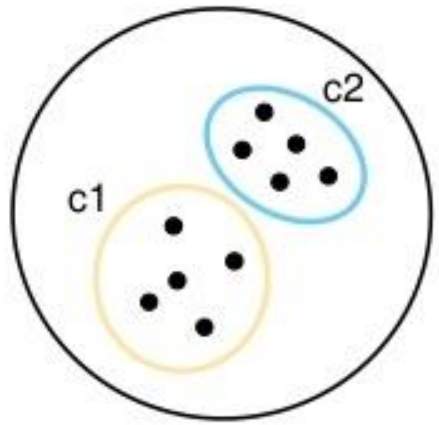




"Top down" approach begin with the whole set and proceed to divide it into successively smaller clusters.

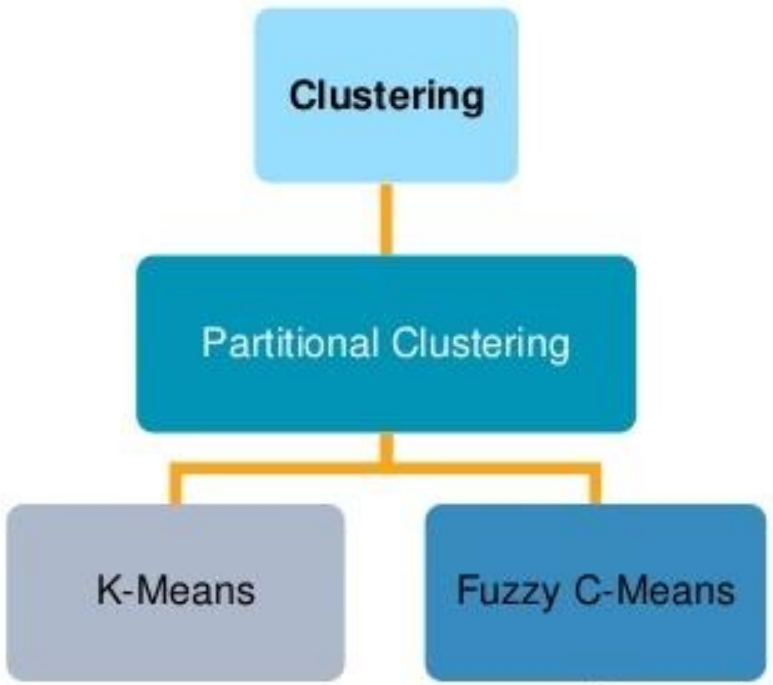
Types of Clustering

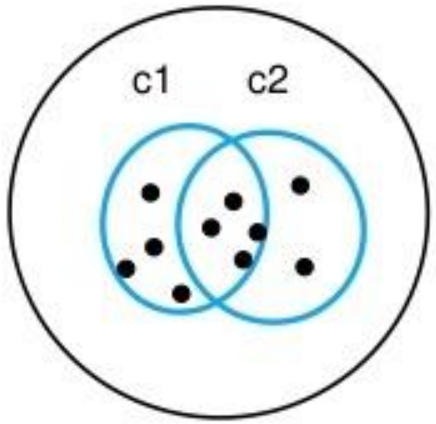




Division of objects into clusters such that each object is in exactly one cluster, not several

Types of Clustering



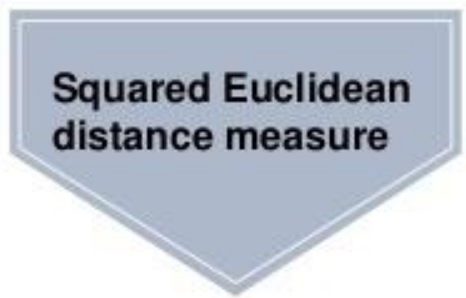


Division of objects into clusters such that each object can belong to multiple clusters

Applications of K-Means Clustering

Distance Measure

Distance measure will determine the similarity between two elements and it will influence the shape of the clusters



Euclidean Distance Measure

01 **Euclidean distance measure**

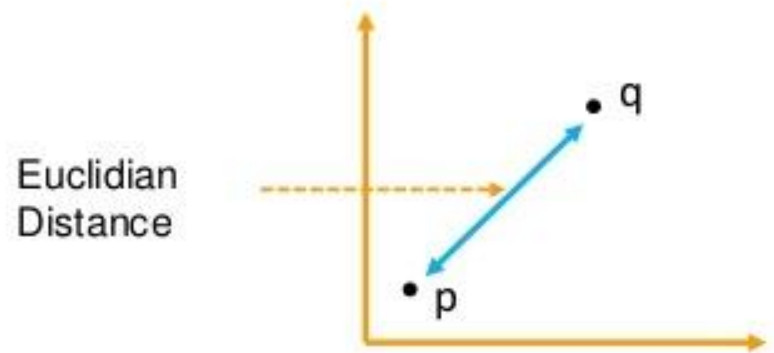
02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

- The Euclidean distance is the "ordinary" straight line
- It is the distance between two points in Euclidean space

$$\mathbf{d}=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$$



Squared Euclidean Distance Measure

01 **Euclidean distance measure**

02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The **Euclidean squared distance** metric uses the same equation as the **Euclidean distance** metric, but does not take the square root.

$$\mathbf{d}= \sum_{i=1}^{n}(q_i - p_i)^2$$

Manhattan Distance Measure

01 **Euclidean distance measure**

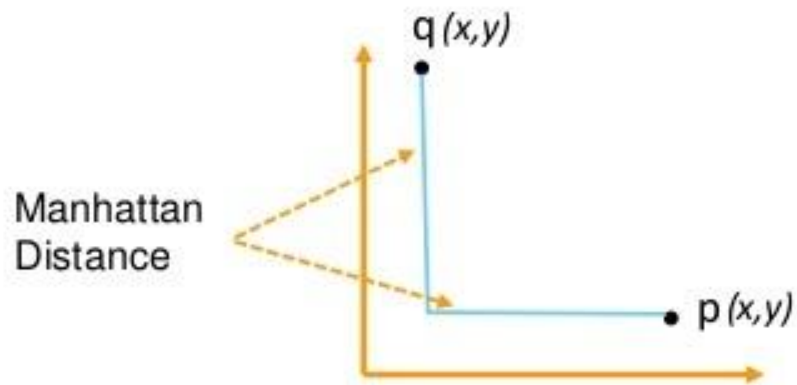
02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The Manhattan distance is the simple sum of the horizontal and vertical components or the distance between two points measured along axes at right angles

$$\mathbf{d}= \sum_{i=1}^{n} | q_x - p_x | + | q_x - p_y |$$



Cosine Distance Measure

01 **Euclidean distance measure**

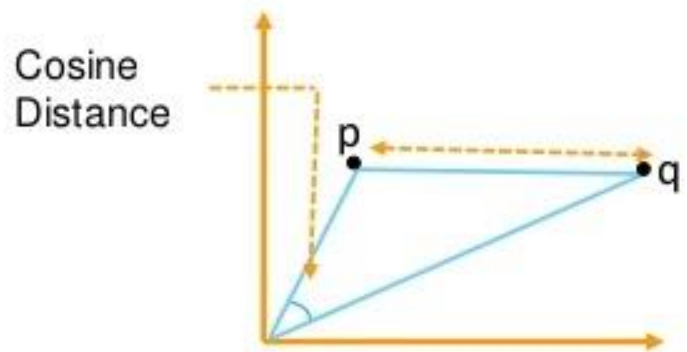
02 Squared euclidean distance measure

03 Manhattan distance measure

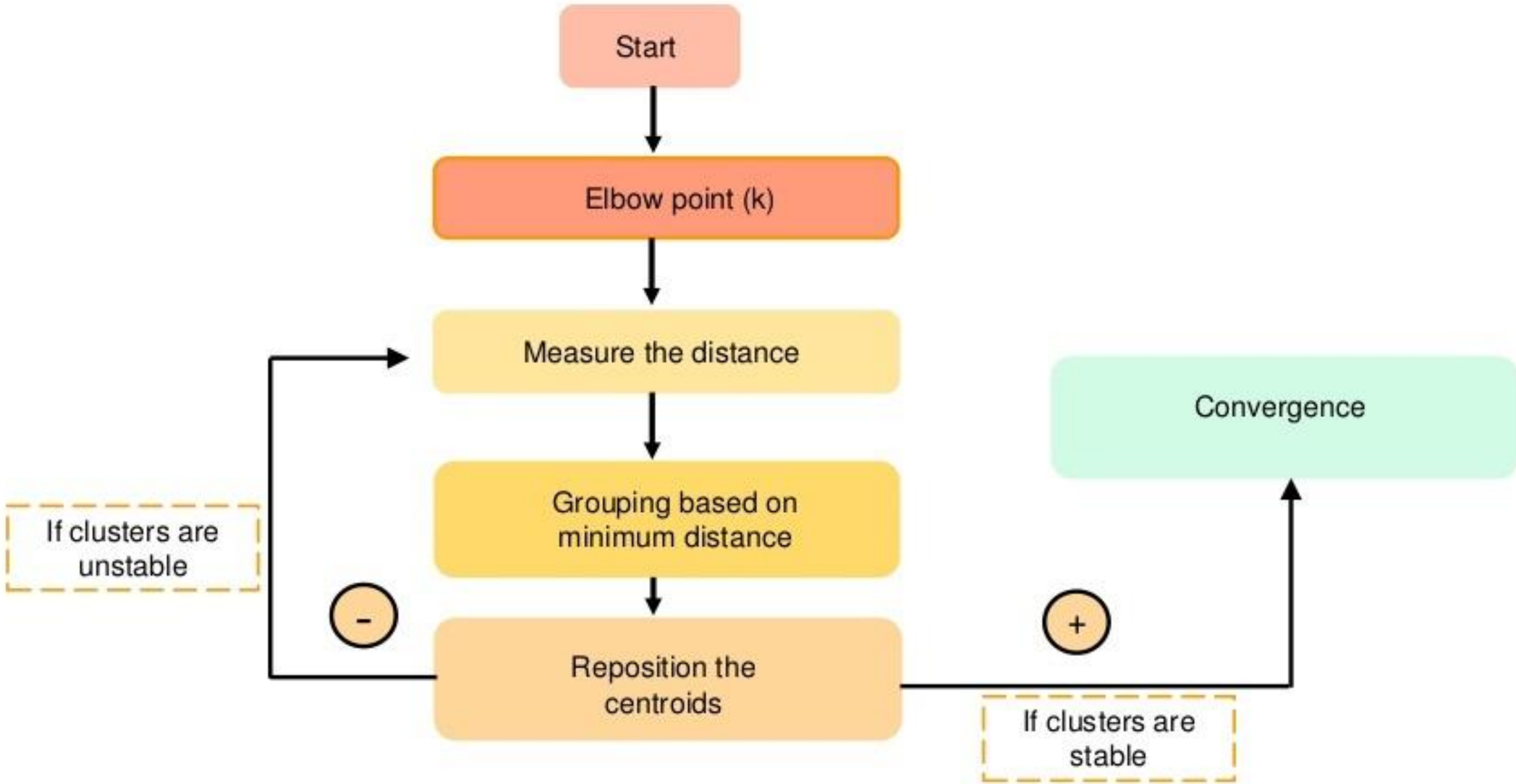
04 Cosine distance measure

The cosine distance similarity measures the angle between the two vectors

$$\mathbf{d}= \frac{\sum_{i=0}^{n-1} q_{i-} p_x}{\sum_{i=0}^{n-1}(q_i)^2 \times \sum_{i=0}^{n-1}(p_i)^2}$$



How does K-Means clustering work?



How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

- Let's say, you have a dataset for a **Grocery shop**

- Now, the important question is, "***how would you choose the optimum number of clusters?***"

How does K-Means clustering work?

Elbow point

Measure the distance

Grouping

Reposition the centroids

Convergence

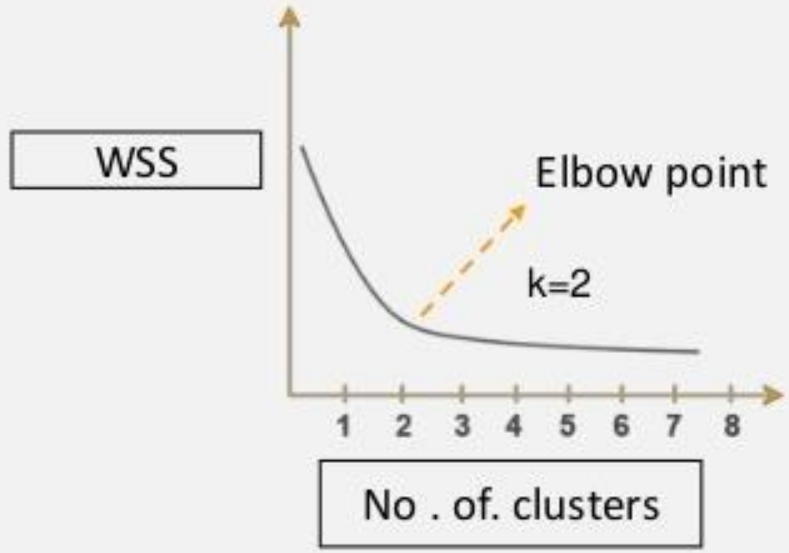
- The best way to do this is by **Elbow method**
- The idea of the elbow method is to run K-Means clustering on the dataset where 'k' is referred as number of clusters
- Within sum of squares (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

Where x_i = data point and c_i = closest point to centroid

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence



- Now, we draw a curve between ***WSS*** (within sum of squares) and the ***number of clusters***
- Here, we can see a very slow change in the value of WSS after k=2, so you should take that elbow point value as the final number of clusters

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 1: The given data points below are assumed as **delivery points**

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

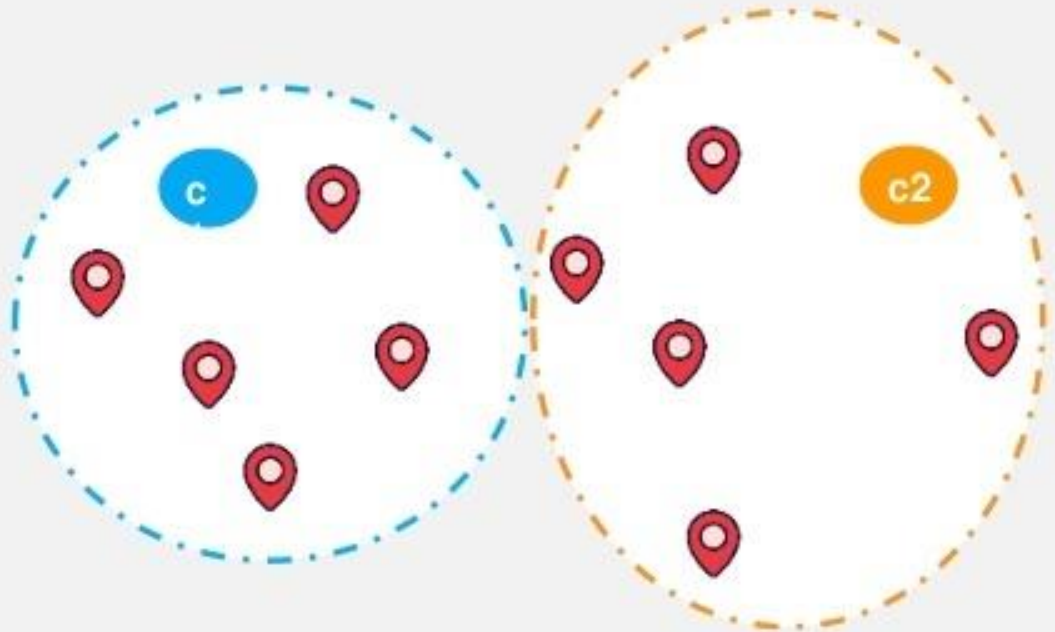
Step 2: We can randomly initialize two points called the cluster centroids,

Euclidean distance is a distance measure used to find out which data point is closest to our centroids

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 3: Based upon the distance from c1 and c2 centroids, the data points will group itself into clusters



How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 4: Compute the centroid of data points inside blue cluster

Step 5: Reposition the centroid of the blue cluster to the new centroid

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 6: Now, compute the centroid of data points inside orange cluster

Step 7: Reposition the centroid of the orange cluster to the new centroid

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 8: Once the clusters become static, K-Means clustering algorithm is said to be converged

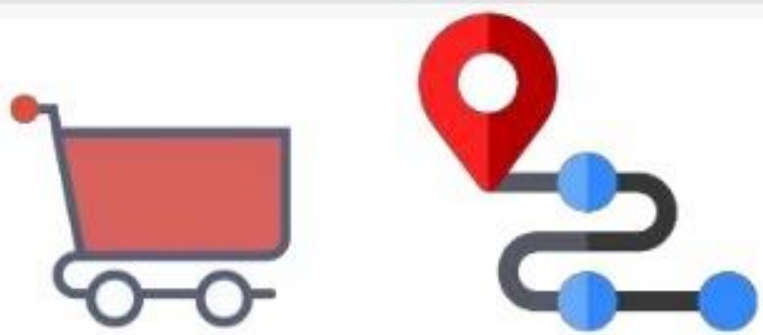
Demo: K-Means Clustering

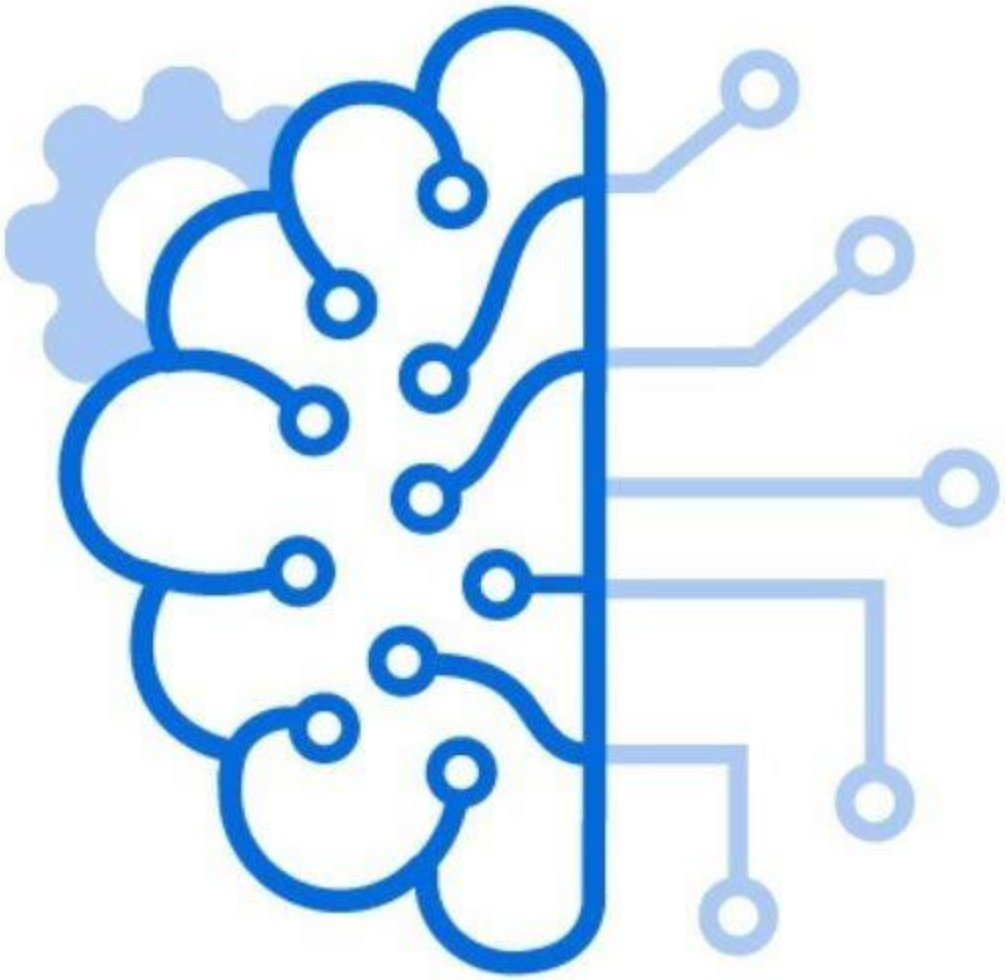
Problem Statement

- Walmart wants to open a chain of stores across Florida and wants to find out optimal store locations to maximize revenue

Solution

- Walmart already has a strong e-commerce presence
- Walmart can use its online customer data to analyze the customer locations along with the monthly sales





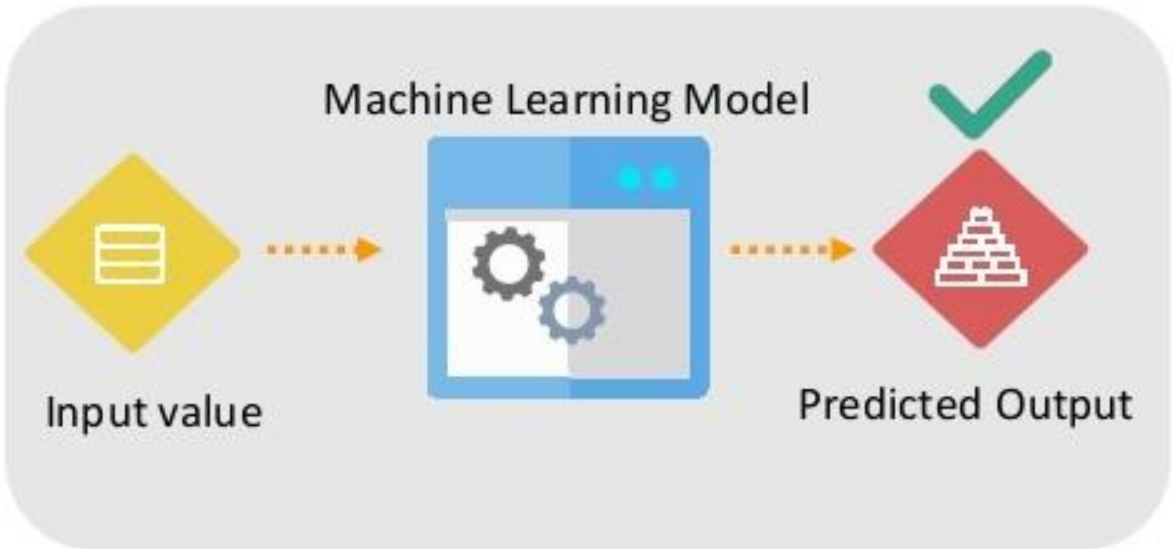
KNN Algorithm

What's in it for you?

- Why do we need KNN?
- What is KNN?
- How do we choose the factor 'K'?
- When do we use KNN?
- How does KNN Algorithm work?
- Use Case: Predict whether a person will have diabetes or not

Why KNN?


By now, we all know
Machine learning models
makes predictions by
learning from the past
data available


Machine Learning Model
Input value
Predicted Output

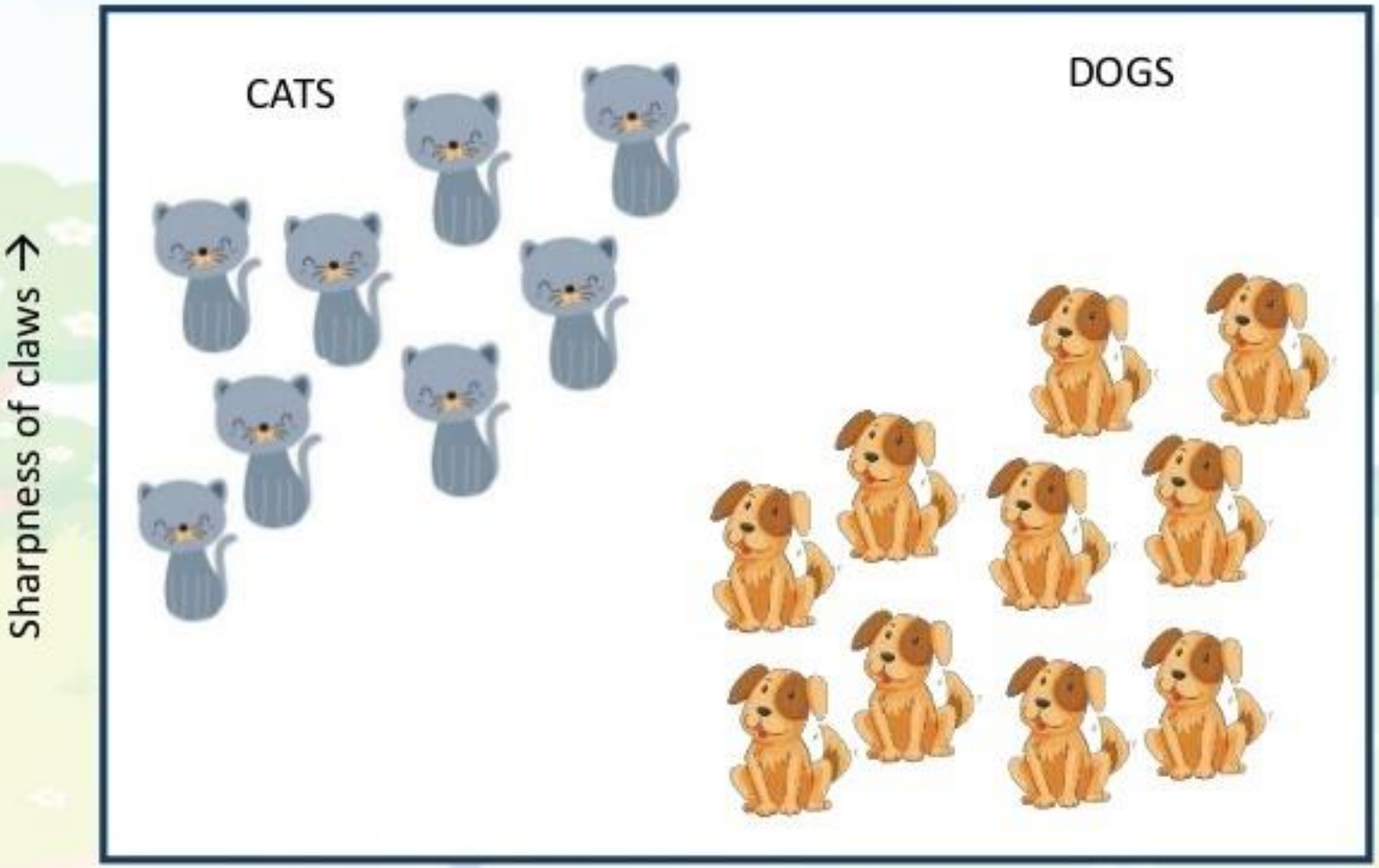

Is that a dog?


No dear, you can differentiate between a cat and a dog based on their characteristics

No dear, you can differentiate between a cat and a dog based on their characteristics

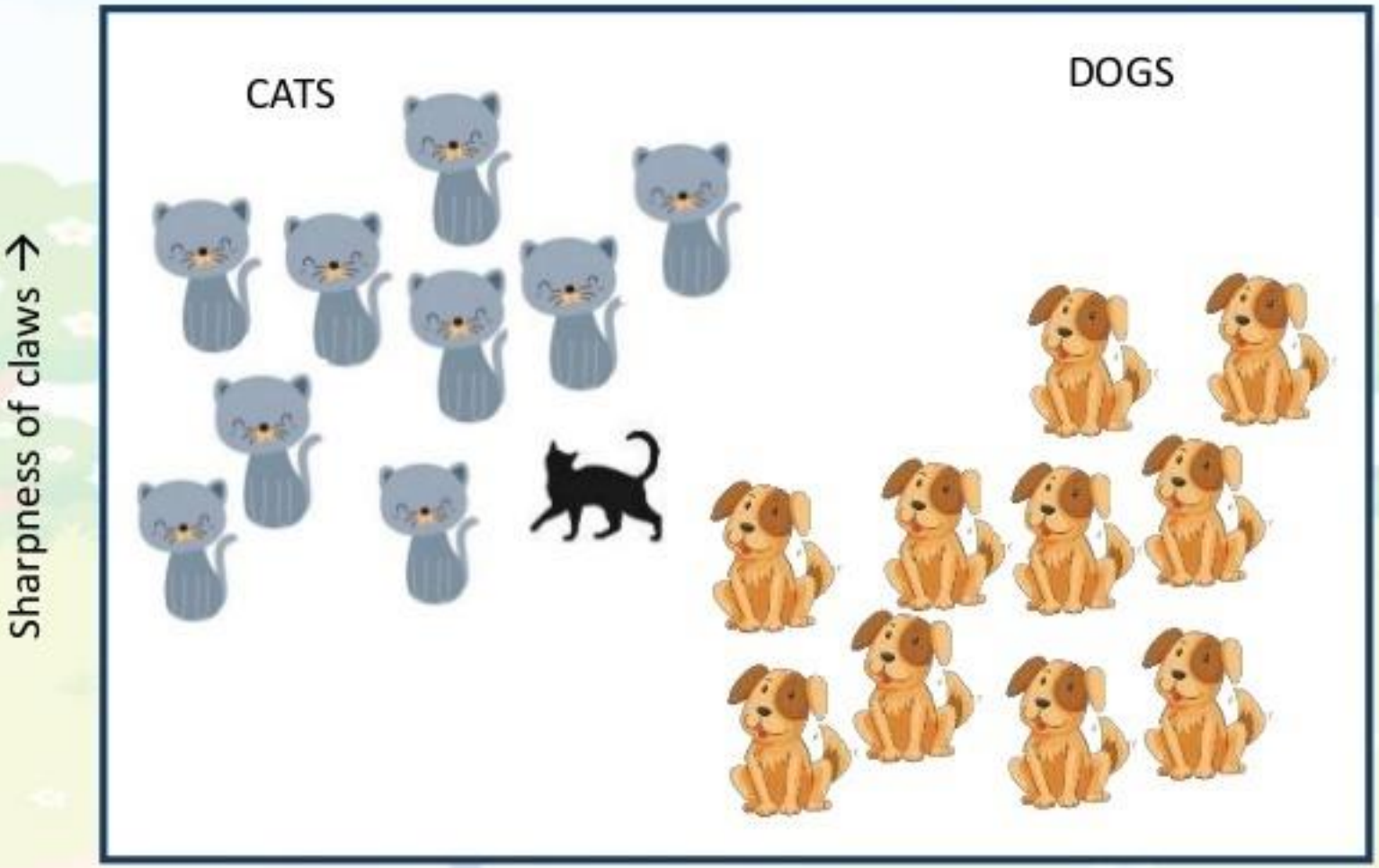
CATS	DOGS
	
Sharp Claws, uses to climb	Dull Claws
Smaller length of ears	Bigger length of ears
Meows and purrs	Barks
Doesn't love to play around	Loves to run around

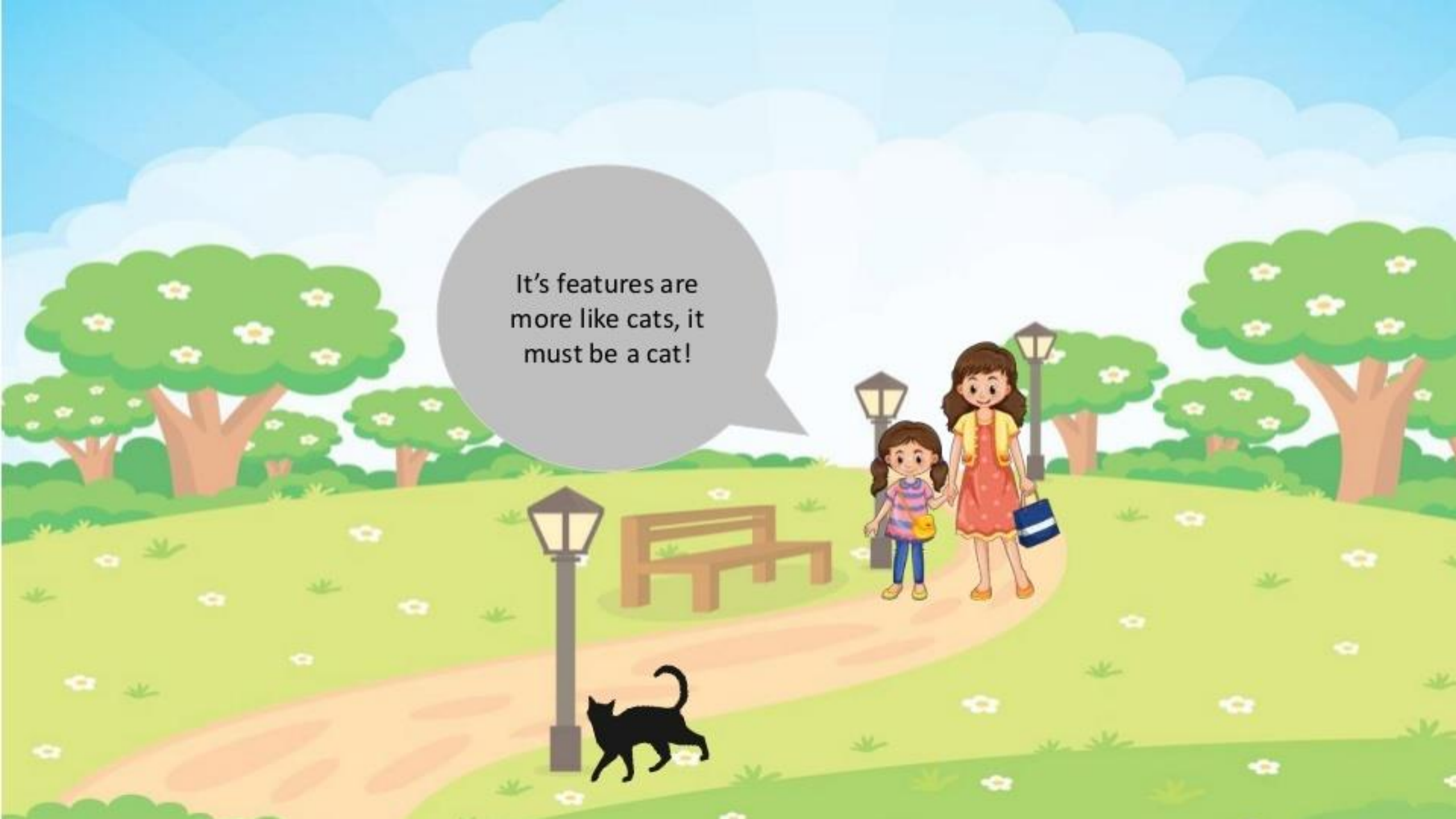
No dear, you can differentiate between a cat and a dog based on their characteristics

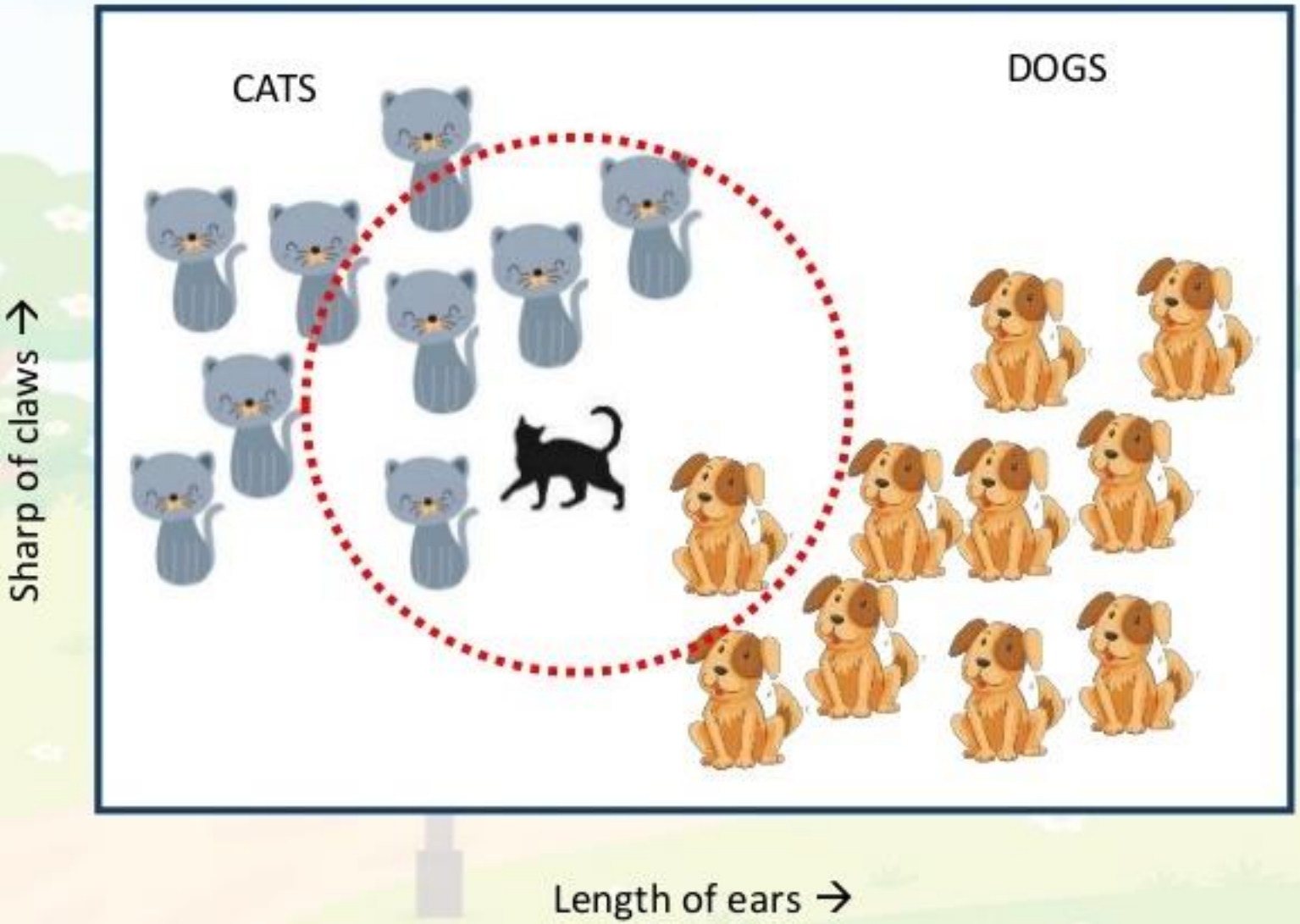



Now tell me if it
is a cat or a dog?

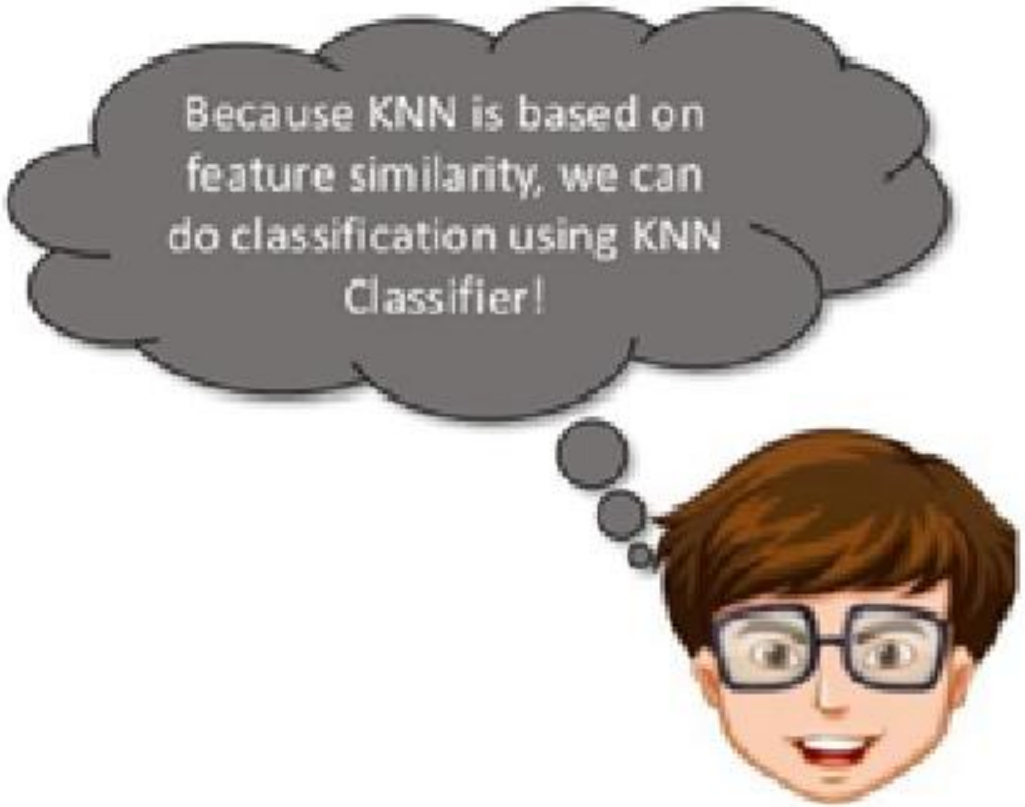
Now tell me if it's a cat or a dog?

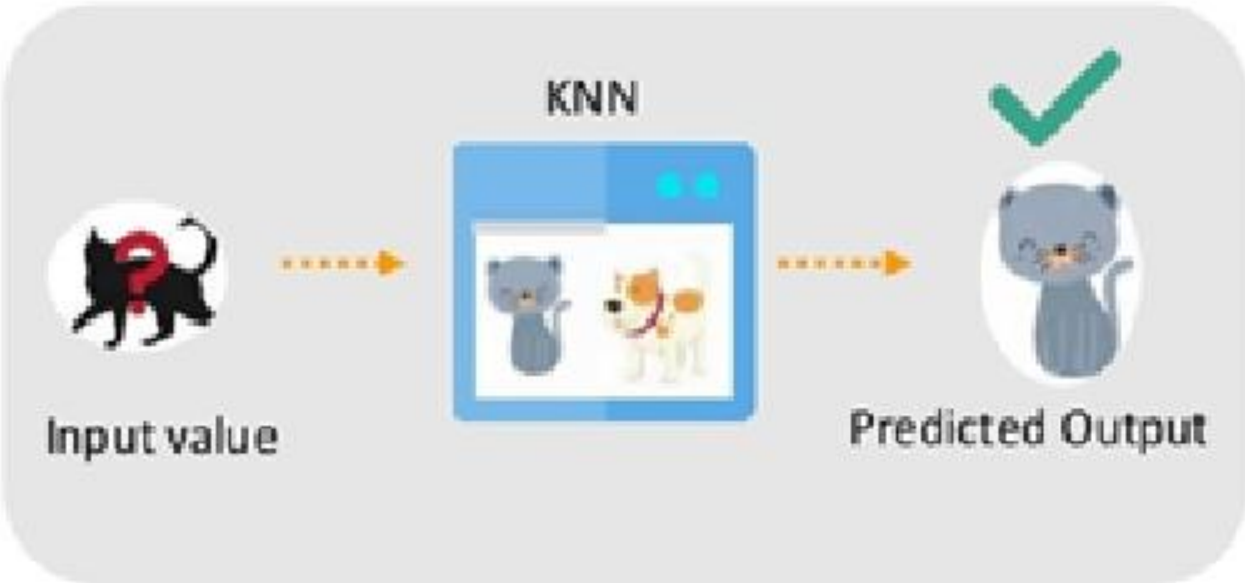



It’s features are more like cats, it must be a cat!


CATS
DOGS
Sharp of claws →
Length of ears →

Why KNN?


Because KNN is based on feature similarity, we can do classification using KNN Classifier!


KNN
Input value
Predicted Output

What is KNN Algorithm?

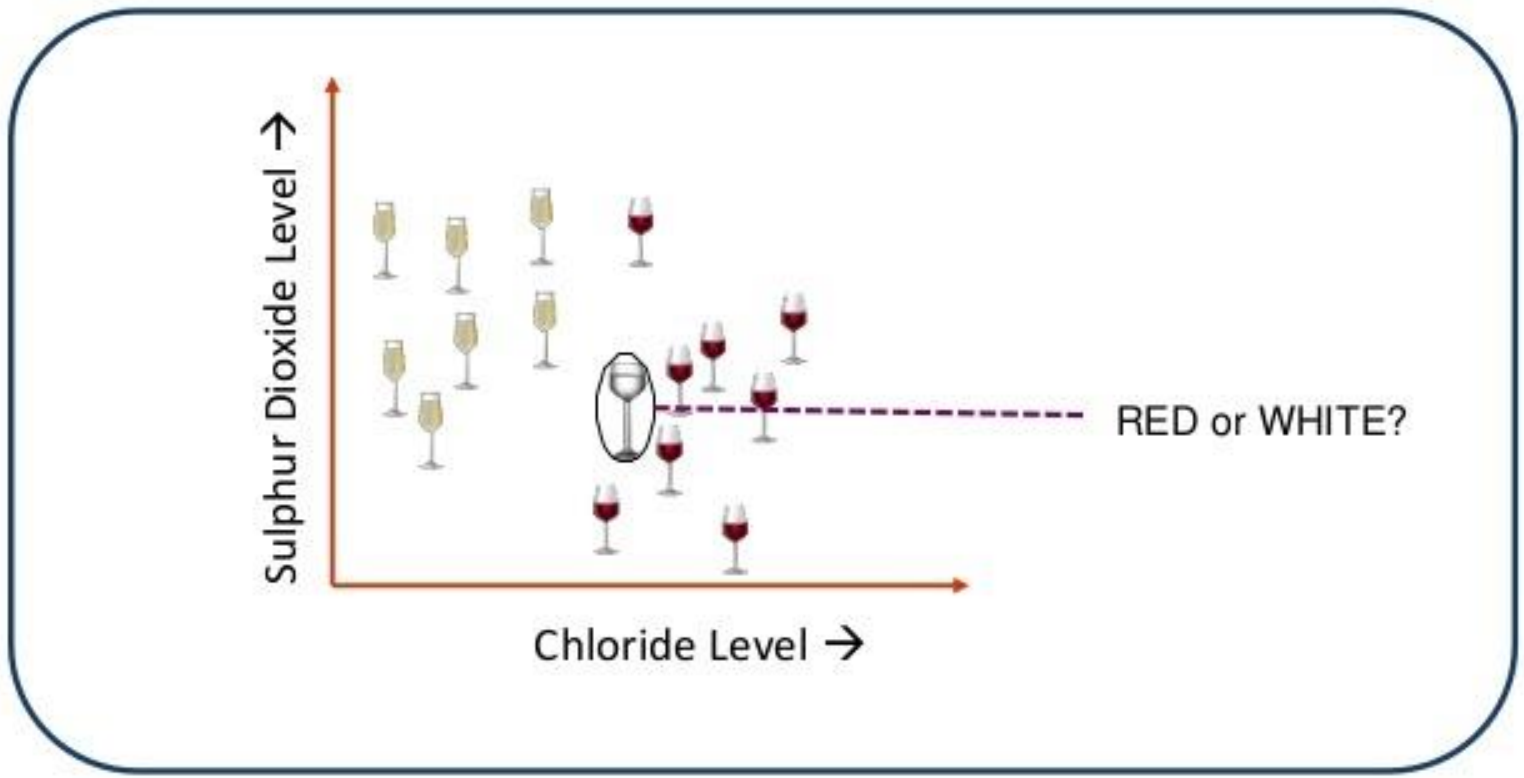
KNN – K Nearest Neighbors, is one of the simplest **Supervised** Machine Learning algorithm mostly used for

Classification

It classifies a data point based on how its neighbors are classified

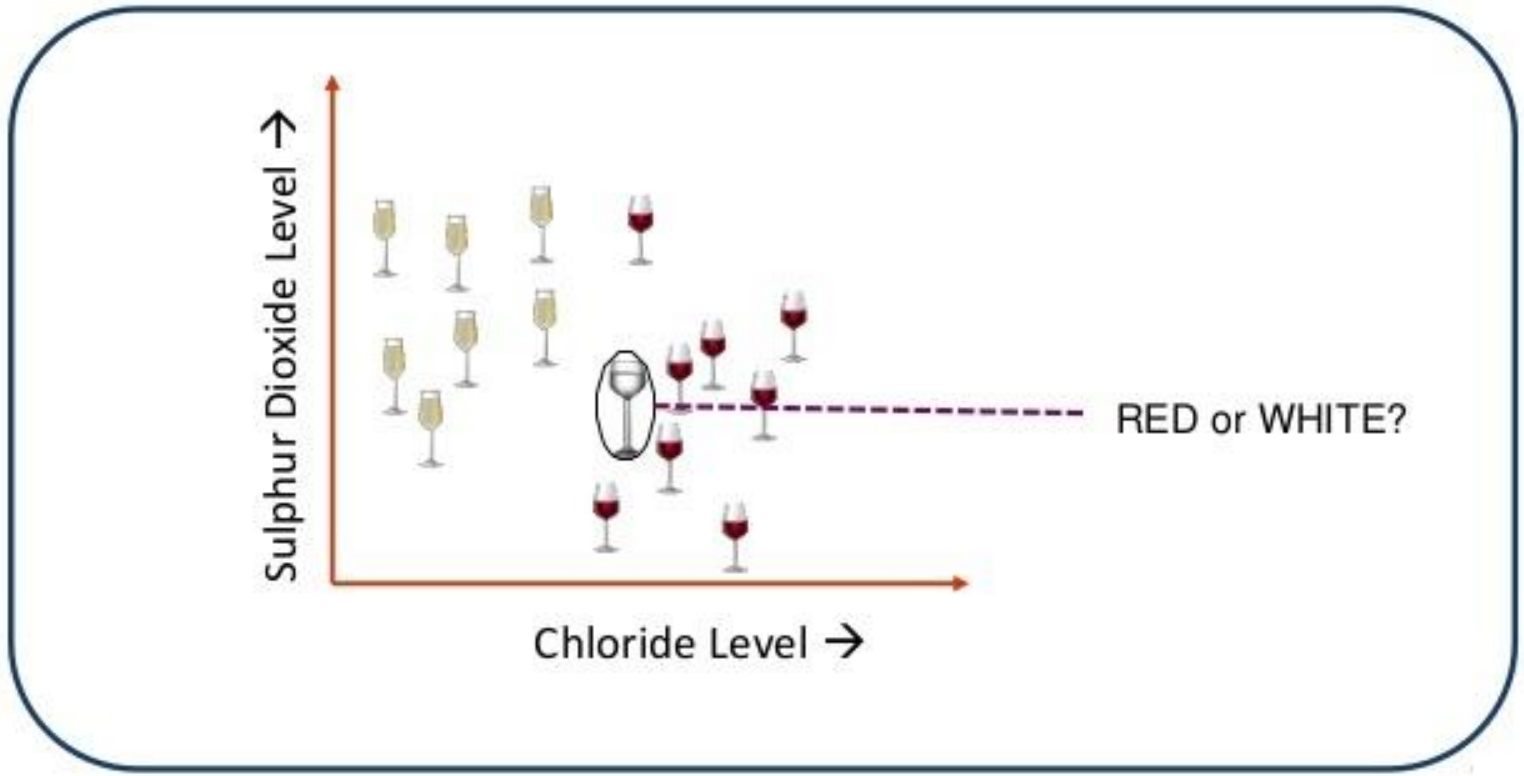
What is KNN Algorithm?

KNN stores all available cases and classifies new cases based on a similarity measure



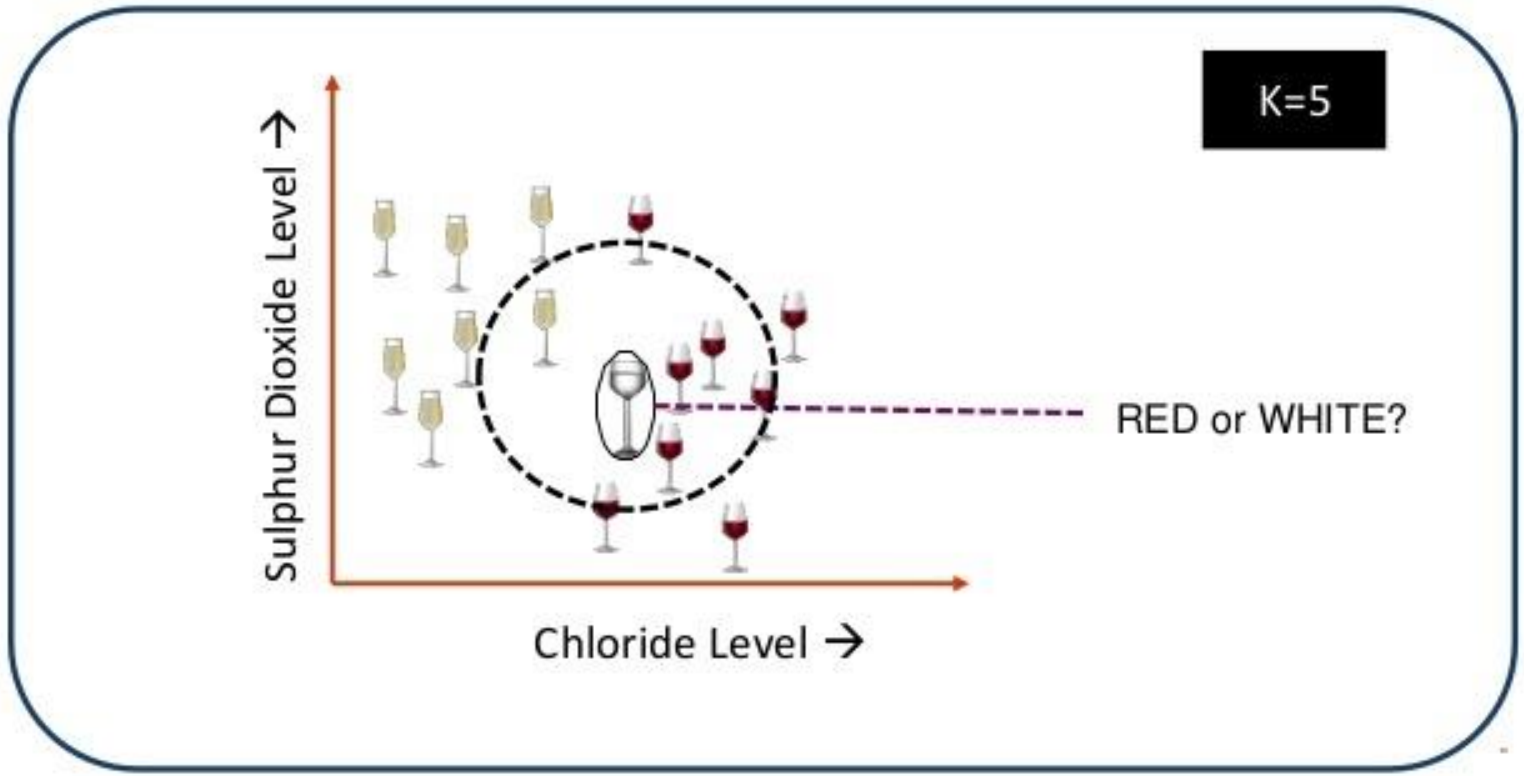
What is KNN Algorithm?

But, what is K?



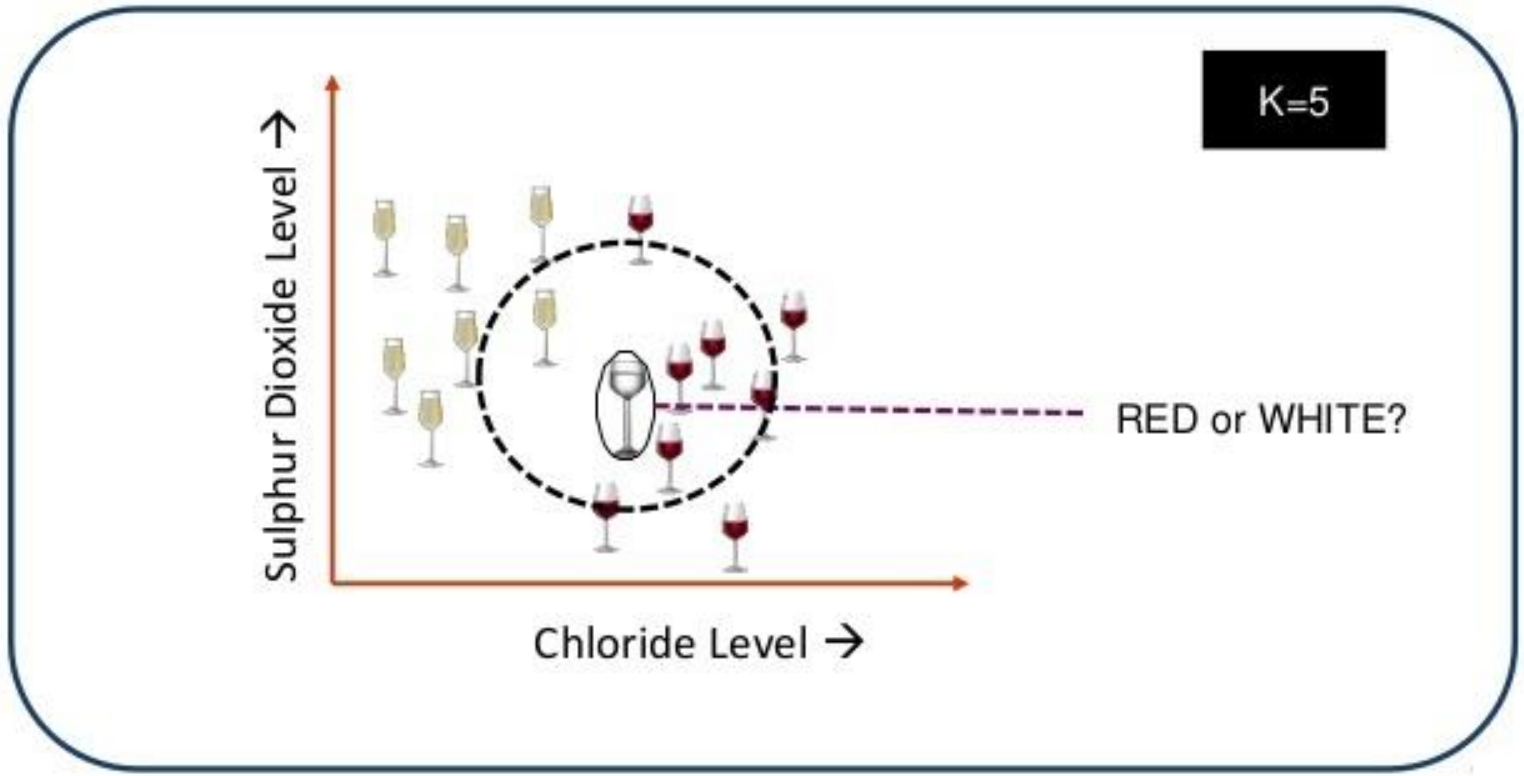
What is KNN Algorithm?

k in **KNN** is a parameter that refers to the number of nearest neighbors to include in the majority voting process



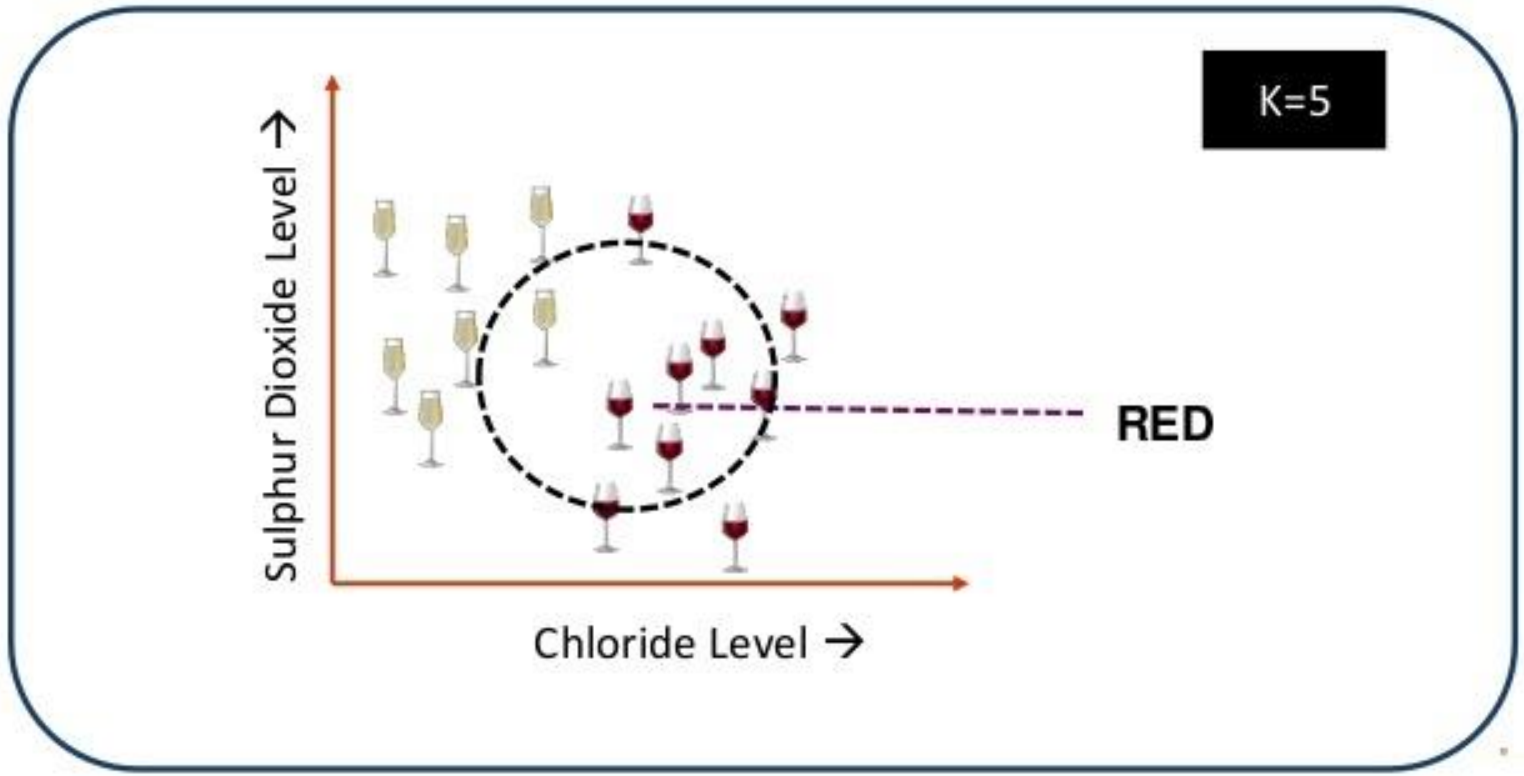
What is KNN Algorithm?

A data point is classified by majority votes from its 5 nearest neighbors



What is KNN Algorithm?

Here, the unknown point would be classified as red, since 4 out of 5 neighbors are red

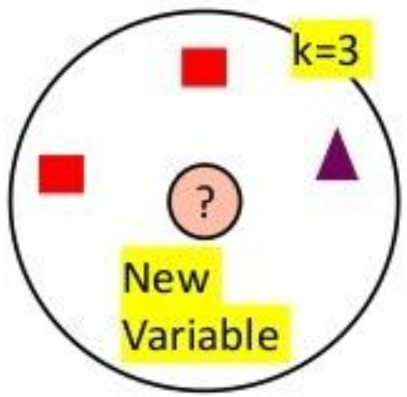


How do we choose the factor ‘k’?

KNN Algorithm is based on **feature similarity**: Choosing the right value of *k* is a process called parameter tuning, and is important for better accuracy

How do we choose the factor ‘k’?

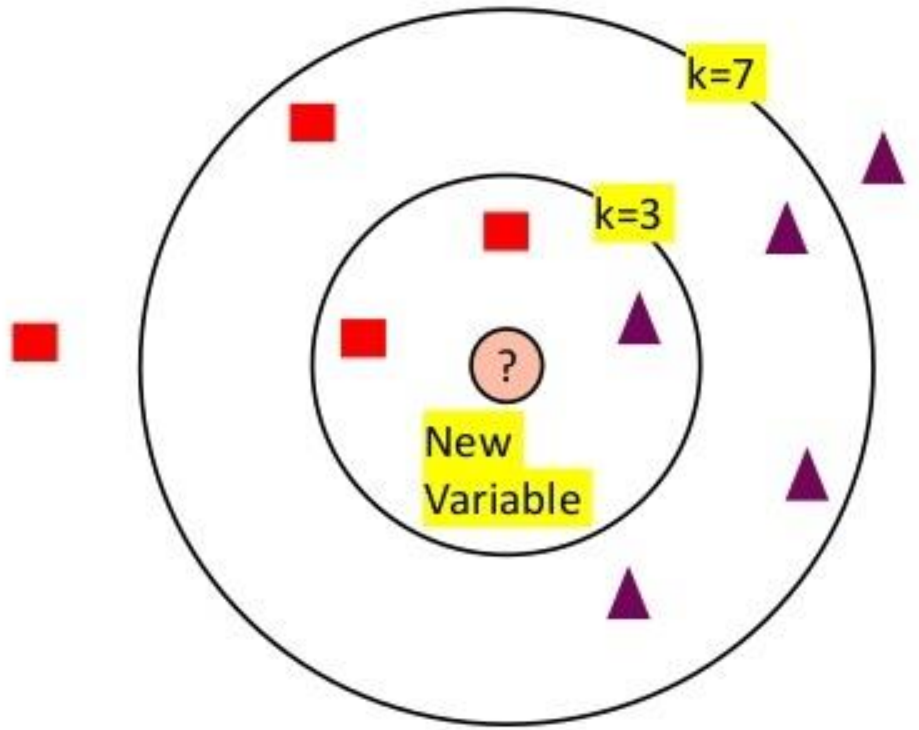
KNN Algorithm is based on **feature similarity**: Choosing the right value of *k* is a process called parameter tuning, and is important for better accuracy



So at k=3, we can classify ‘?’ as 

How do we choose the factor ‘k’?

KNN Algorithm is based on **feature similarity**: Choosing the right value of *k* is a process called parameter tuning, and is important for better accuracy



But at k=7, we classify ‘?’ as

How do we choose the factor ‘k’?

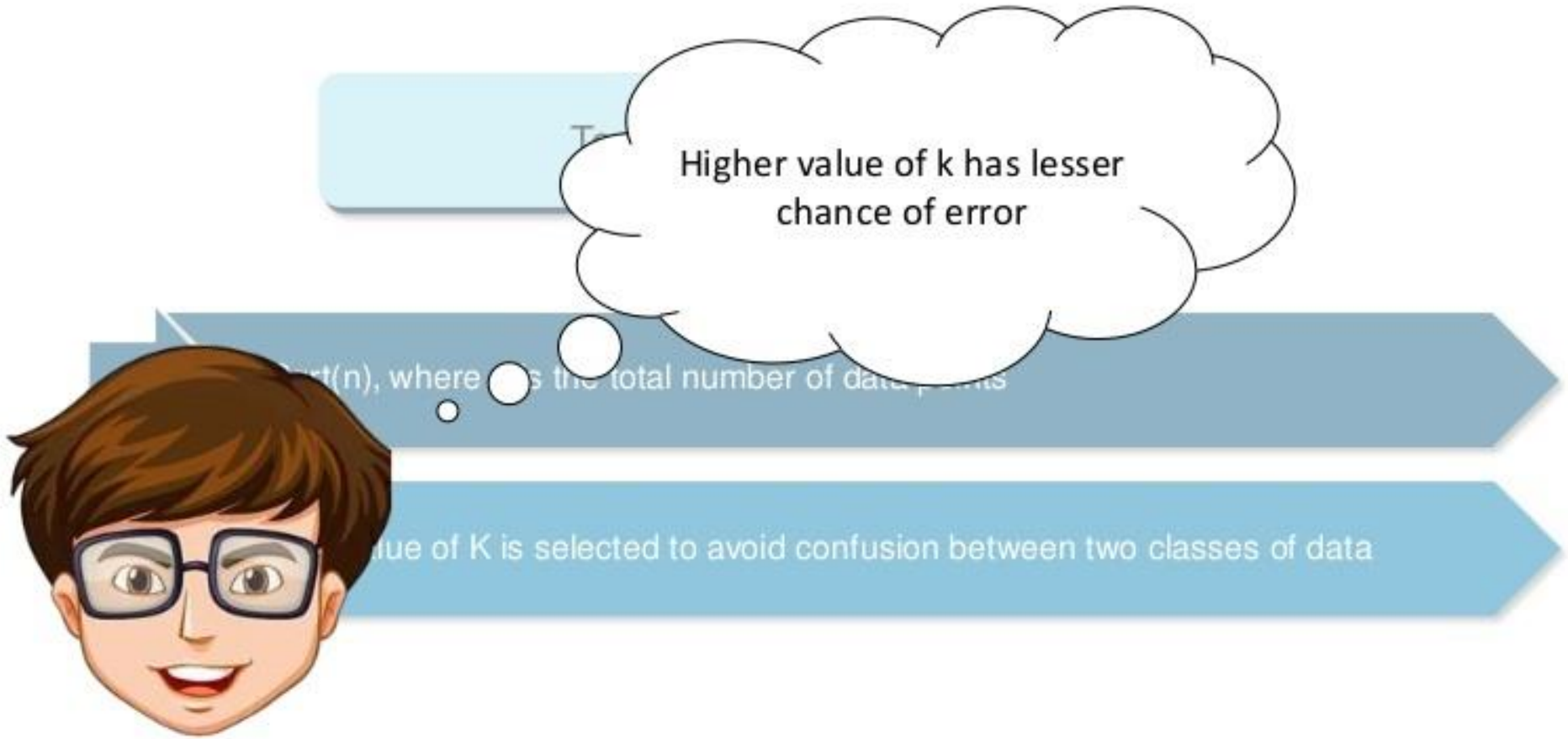
The class of unknown data point was ■ at k=3 but changed at k=7, so which k should we choose?

How do we choose the factor ‘k’?

To choose a value of k:

- Sqrt(n), where n is the total number of data points
- Odd value of K is selected to avoid confusion between two classes of data

How do we choose the factor ‘k’?



When do we use KNN Algorithm?

We can use KNN when

Data is labeled

Dog

When do we use KNN Algorithm?

We can use KNN when

Data is labeled

Dog

Data is noise free

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	one-fourty
69	176	23
64	173	hello kitty
65	172	Normal

Noise

When do we use KNN Algorithm?

We can use KNN when

Data is labeled

Dog

Dataset is small

Because KNN is a 'lazy learner' i.e. doesn't learn a discriminative function from the training set

Data is noise free

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	one-fourty
69	176	23
64	173	hello kitty
65	172	Normal

Noise

How does KNN Algorithm work?

Consider a dataset having two variables: height (cm) & weight (kg) and each point is classified as Normal or Underweight

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	Normal
69	176	Normal
64	173	Normal
65	172	Normal
56	174	Underweight
58	169	Normal
57	173	Normal
55	170	Normal

How does KNN Algorithm work?

On the basis of the given data we have to classify the below set as Normal or Underweight using KNN

57 kg	**170 cm**	**?**

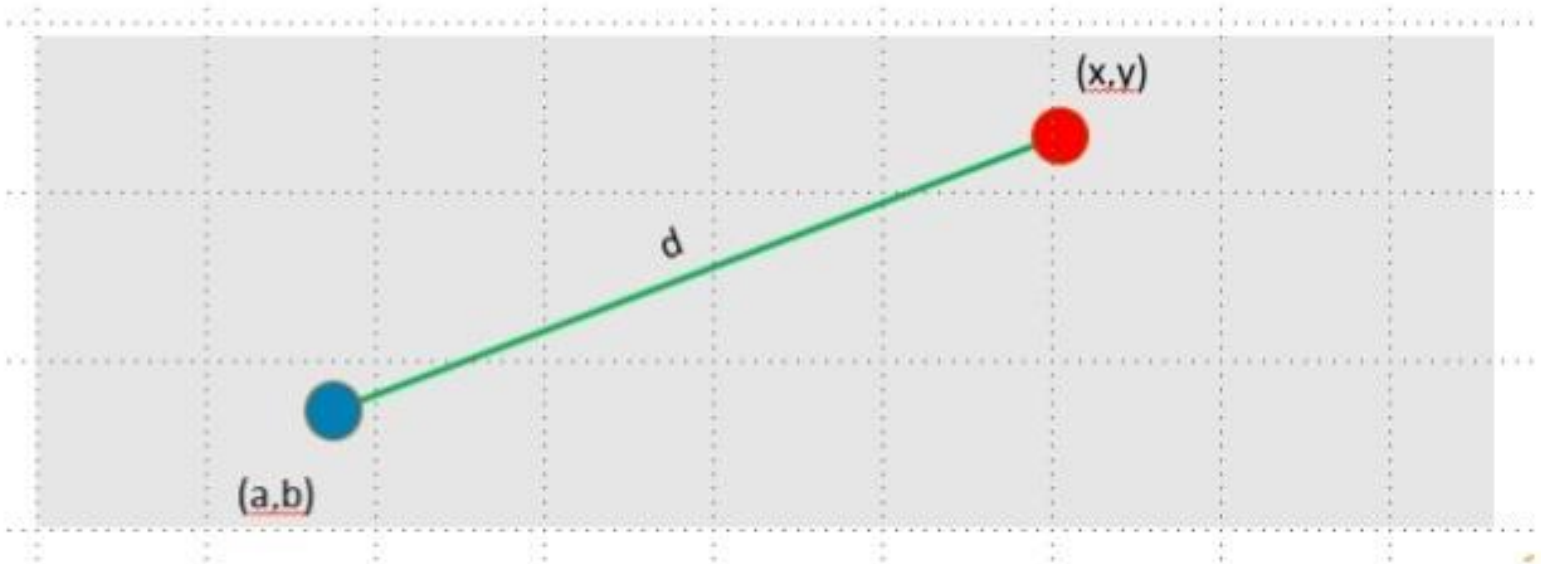
How does KNN Algorithm work?

To find the nearest neighbors, we will calculate Euclidean distance

How does KNN Algorithm work?

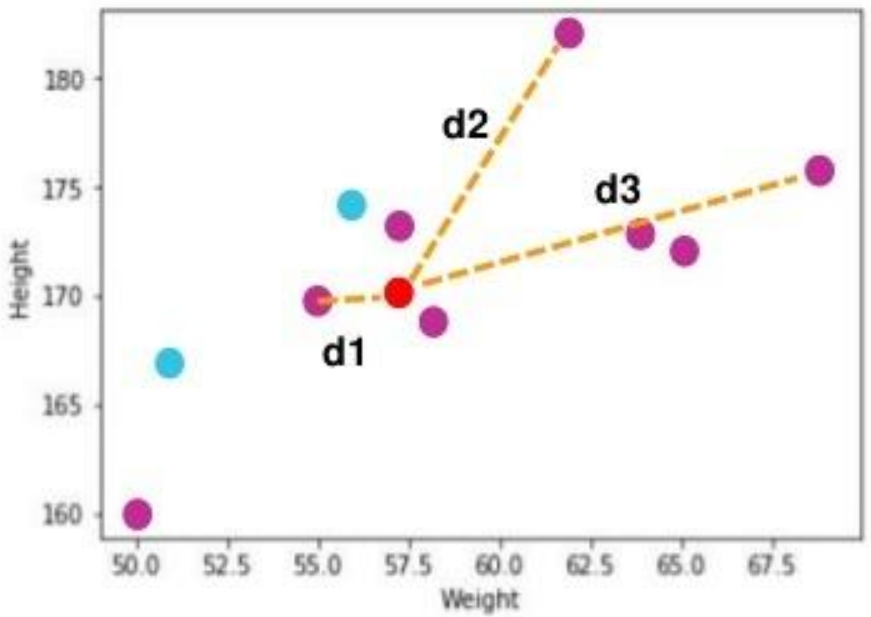
According to the **Euclidean distance** formula, the **distance** between two points in the plane with coordinates (x, y) and (a, b) is given by:

$$dist(d) = \sqrt{(x - a)^2 + (y - b)^2}$$



How does KNN Algorithm work?

Let's calculate it to understand clearly:



dist(**d1**)= $\sqrt{(170-167)^2 + (57-51)^2}$ ~= 6.7

dist(**d2**)= $\sqrt{(170-182)^2 + (57-62)^2}$ ~= 13

dist(**d3**)= $\sqrt{(170-176)^2 + (57-69)^2}$ ~= 13.4

Similarly, we will calculate Euclidean distance of unknown data point from all the points in the dataset

● Unknown data point

How does KNN Algorithm work?

Hence, we have calculated the Euclidean distance of unknown data point from all the points as shown:

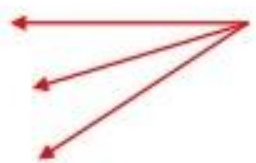
Where (x1, y1) = (57, 170) whose class we have to classify

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

How does KNN Algorithm work?

Now, lets calculate the nearest neighbor at k=3

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2



k = 3

57 kg	170 cm	?

How does KNN Algorithm work?

Now, lets calculate the nearest neighbor at k=3

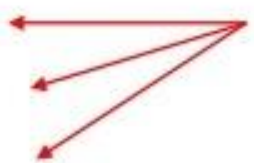
We have n=10,
And sqrt(10)=3.1
Hence, we have taken k=3

Wei			n Distance
			.7
			.4
			7.6
65			3
56	174	derweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

57 kg	170 cm	?

How does KNN Algorithm work?

Class	Euclidean Distance
Underweight	6.7
Normal	13
Normal	13.4
Normal	7.6
Normal	8.2
Underweight	4.1
Normal	1.4
Normal	3
Normal	2



k = 3

So, majority neighbors are pointing towards '*Normal*'

Hence, as per KNN algorithm the class of (57, 170) should be 'Normal'

Recap of KNN

Recap of KNN

- A positive integer k is specified, along with a new sample
- We select the k entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample

KNN - Predict diabetes

Objective: Predict whether a person will be diagnosed with diabetes or not

> We have a dataset of 768 people who were or were not diagnosed with diabetes

KNN - Predict diabetes

Import the required Scikit-learn libraries as shown:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
```

KNN - Predict diabetes

Load the dataset and have a look:

```
dataset = pd.read_csv('../Downloads/diabetes.csv')
```

```
dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

KNN - Predict diabetes

Values of columns like ‘Glucose’, BloodPressure’ cannot be accepted as zeroes because it will affect the outcome

We can replace such values with the mean of the respective column:

```
# Replace zeroes
zero_not_accepted = ['Glucose', 'BloodPressure', 'SkinThickness', 'BMI', 'Insulin']
```

```
for column in zero_not_accepted:
    dataset[column] = dataset[column].replace(0, np.NaN)
    mean = int(dataset[column].mean(skipna=True))
    dataset[column] = dataset[column].replace(np.NaN, mean)
```

KNN - Predict diabetes

Before proceeding further, let's split the dataset into train and test:

```
# split dataset
X = dataset.iloc[:, 0:8]
y = dataset.iloc[:, 8]
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.2)
```

KNN - Predict diabetes

Rule of thumb: Any algorithm that computes distance or assumes normality, **scale your features!**

Feature Scaling:

```
# Feature scaling
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

KNN - Predict diabetes

N_neighbors here is 'K'
p is the power parameter to define the metric used, which is 'Euclidean' in our case

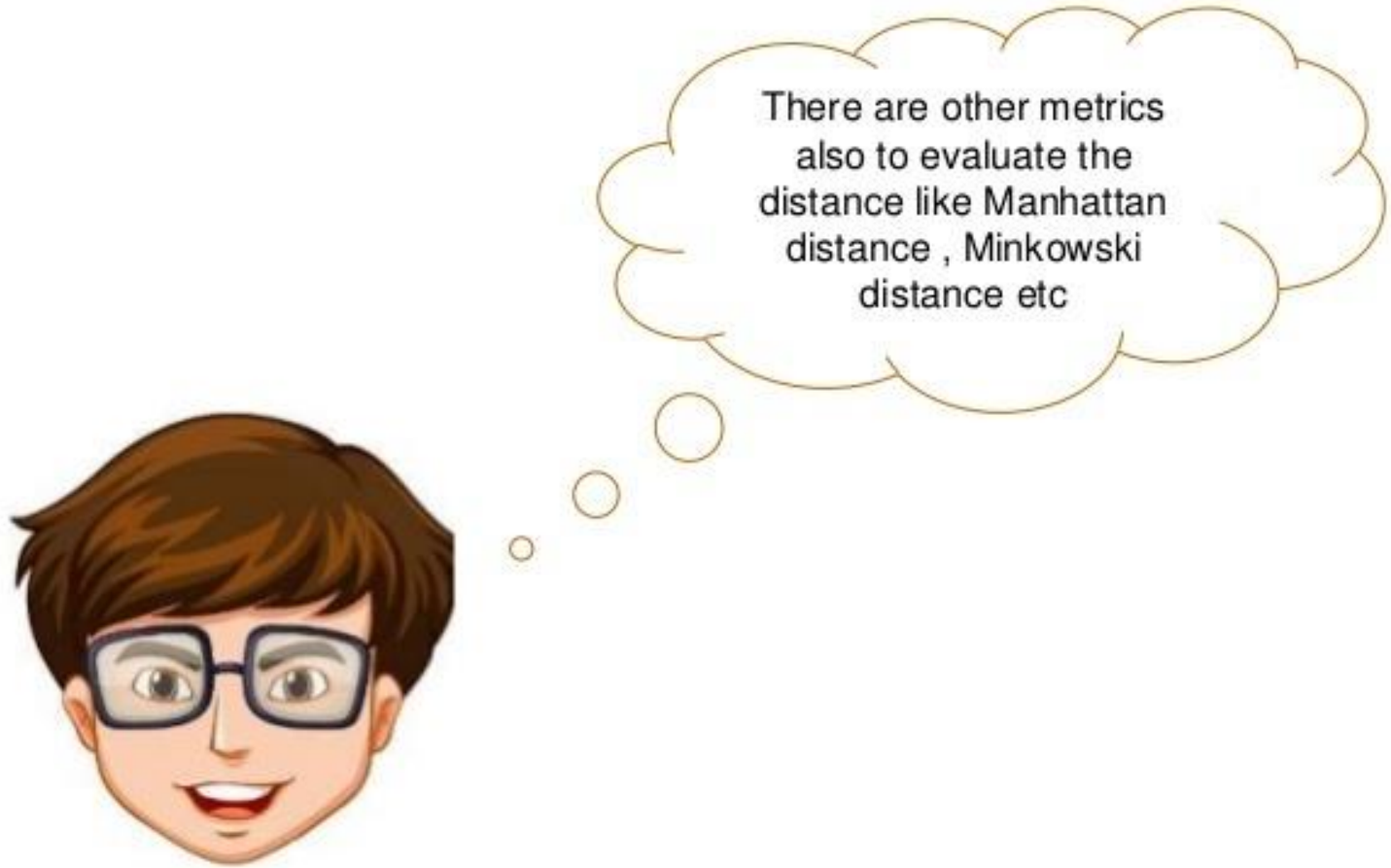
Then define the model using KNeighborsClassifier and fit the train data in the model

```
# Define the model: Init K-NN
classifier = KNeighborsClassifier(n_neighbors=11, p=2, metric='euclidean')
```

```
# Fit Model
classifier.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
           metric_params=None, n_jobs=1, n_neighbors=11, p=2,
           weights='uniform')
```

KNN - Predict diabetes



KNN - Predict diabetes

Let's predict the test results:

```
# Predict the test set results
y_pred = classifier.predict(X_test)
```

```
y_pred
```

```
array([1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      dtype=int64)
```

KNN - Predict diabetes

It's important to evaluate the model, let's use confusion matrix to do that:

```
# Evaluate Model
cm = confusion_matrix(y_test, y_pred)
print (cm)
print(f1_score(y_test, y_pred))
```

```
[[94 13]
 [15 32]]
0.6956521739130436
```

KNN - Predict diabetes

Calculate accuracy of the model:

```
print(accuracy_score(y_test, y_pred))
```

```
0.8181818181818182
```

KNN - Predict diabetes

KNN - Predict diabetes

```
print(accuracy_score(y_test, y_pred))
```

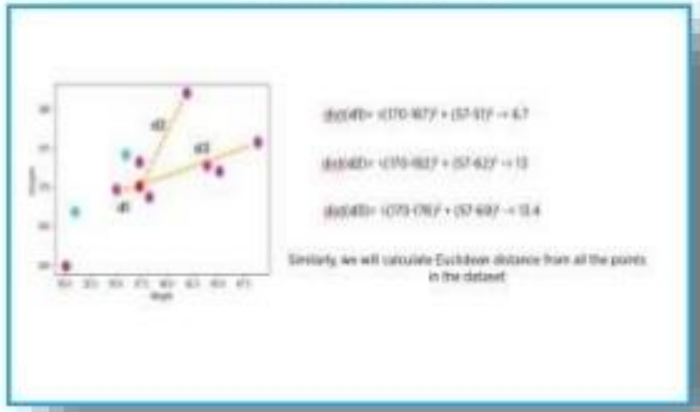
```
0.8181818181818182
```

And accuracy of 80% tells us that it is a pretty fair fit in the model!

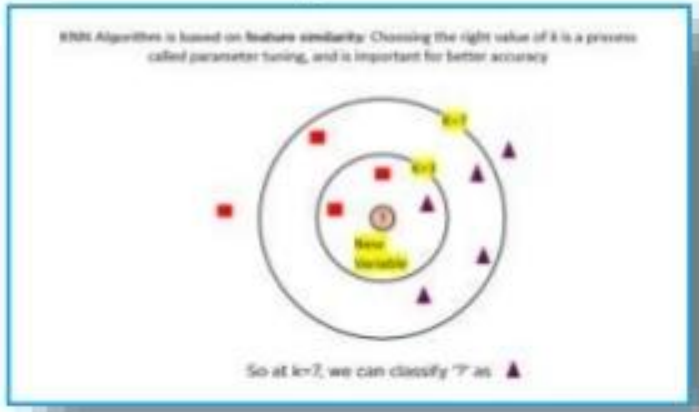
Summary

Why we need knn?

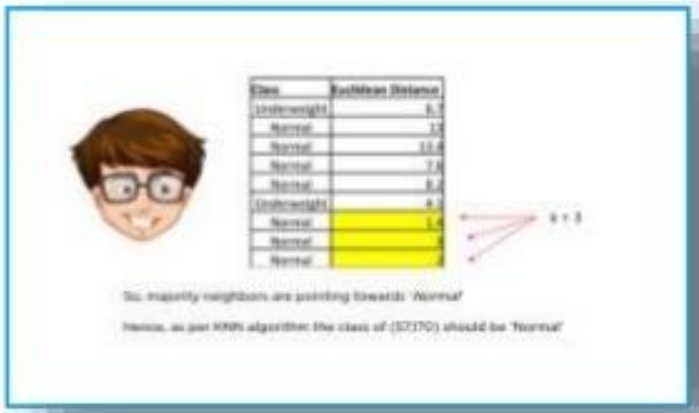
Eucledian distance



Choosing the value of k



How KNN works?



Knn classifier for diabetes prediction

Principal Component Analysis

Principal Component Analysis is basically a statistical procedure to convert a set of observation of possibly correlated variables into a set of values of linearly uncorrelated variables.

Each of the principal components is chosen in such a way so that it would describe most of the still available variance and all these principal components are orthogonal to each other. In all principal components first principal component has maximum variance.

Uses of PCA:

- It is used to find inter-relation between variables in the data.
- It is used to interpret and visualize data.
- As number of variables are decreasing it makes further analysis simpler.
- It's often used to visualize genetic distance and relatedness between populations.

These are basically performed on square symmetric matrix. It can be a pure sums of squares and cross products matrix or Covariance matrix or Correlation matrix. A correlation matrix is used if the individual variance differs much.

Objectives of Principal Component Analysis

•It is basically a non-dependent procedure in which it reduces attribute space from a large number of variables to a smaller number of factors.

•PCA is basically a dimension reduction process but there is no guarantee that the dimension is interpretable.

•Main task in this PCA is to select a subset of variables from a larger set, based on which original variables have the highest correlation with the principal amount.

Principal Axis Method: PCA basically searches a linear combination of variables so that we can extract maximum variance from the variables. Once this process completes it removes it and search for another linear combination which gives an explanation about the maximum proportion of remaining variance which basically leads to orthogonal factors. In this method, we analyze total variance.

Eigenvector: It is a non-zero vector that stays parallel after matrix multiplication. Let's suppose x is eigen vector of dimension r of matrix M with dimension r*r if Mx and x are parallel. Then we need to solve Mx=Ax where both x and A are unknown to get eigen vector and eigen values.
Under Eigen-Vectors we can say that Principal components show both common and unique variance of the variable. Basically, it is variance focused approach seeking to reproduce total variance and correlation with all components. The principal components are basically the linear combinations of the original variables weighted by their contribution to explain the variance in a particular orthogonal dimension.

Eigen Values: It is basically known as characteristic roots. It basically measures the variance in all variables which is accounted for by that factor. The ratio of eigenvalues is the ratio of explanatory importance of the factors with respect to the variables. If the factor is low then it is contributing less in explanation of variables. In simple words, it measures the amount of variance in the total given database accounted by the factor. We can calculate the factor's eigen value as the sum of its squared factor loading for all the variables.

Genetic Algorithms

- Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms.
- Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space.

They are commonly used to generate high-quality solutions for optimization problems and search problems.

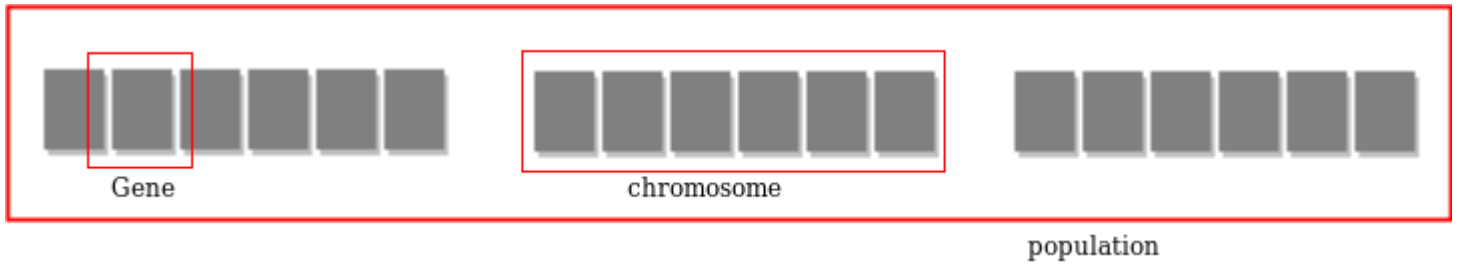
Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. **Each generation consist of a population of individuals** and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

Foundation of Genetic Algorithms

- Individual in population compete for resources and mate
- Those individuals who are successful (fittest) then mate to create more offspring than others
- Genes from “fittest” parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
- Thus each successive generation is more suited for their environment.

Search Space

The population of individuals are maintained within search space. Each individual represent a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).



Fitness Score

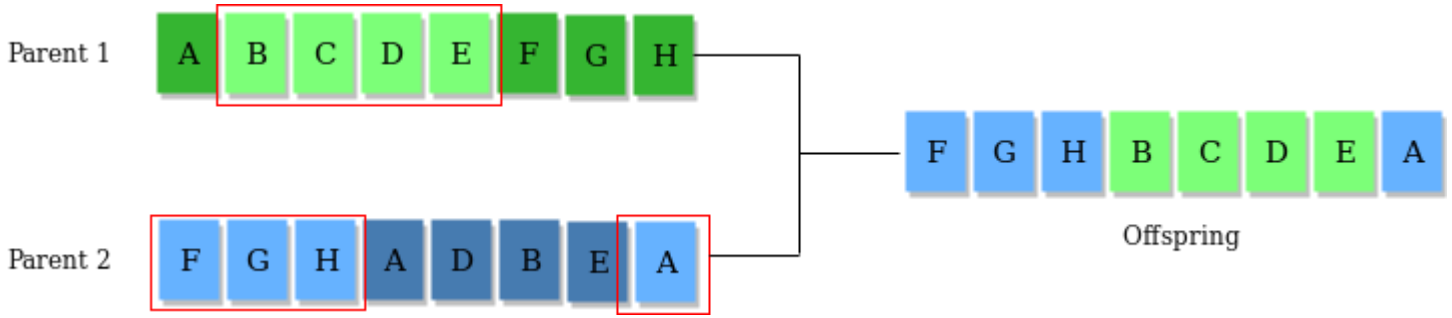
- A Fitness Score is given to each individual which **shows the ability of an individual to "compete"**. The individual having optimal fitness score (or near optimal) are sought.
- The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores.
- The individuals having better fitness scores are given more chance to reproduce than others.
- The individuals with better fitness scores are selected who mate and produce **better offspring** by combining chromosomes of parents.
- The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted.
- It is hoped that over successive generations better solutions will arrive while least fit die.
- Each new generation has on average more "better genes" than the individual (solution) of previous generations. Thus each new generations have better **"partial solutions"** than previous generations.

Operators of Genetic Algorithm

Once the initial generation is created, the algorithm evolve the generation using following operators –

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass there genes to the successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example –



3) Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For example –