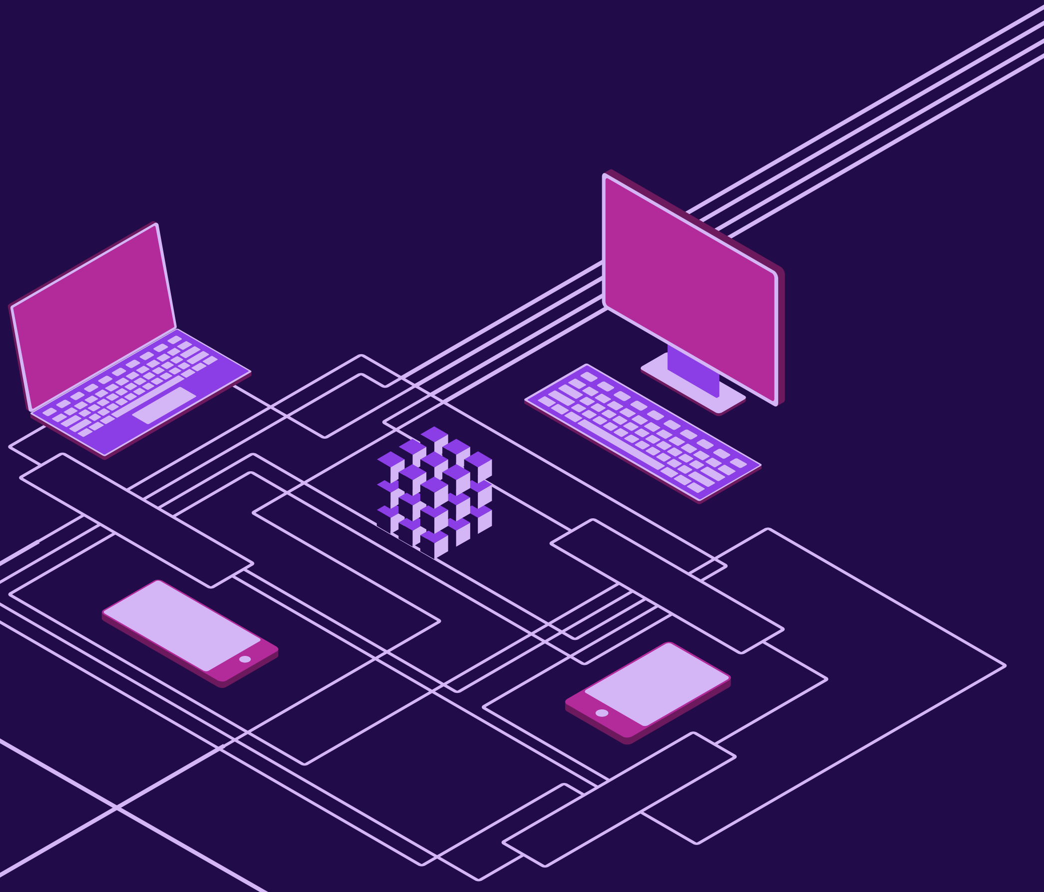
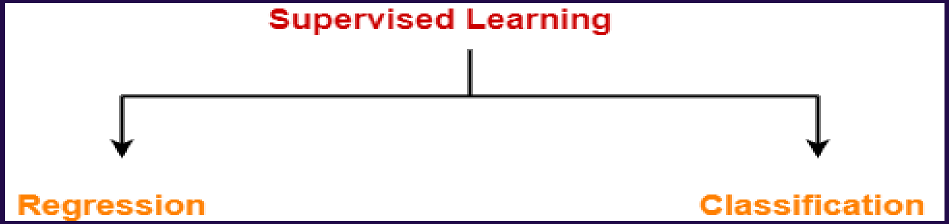
MACHINE LEARNING ALGORITHM
Unit-II

Dr. Gopal Sakarkar
Department of AI, GHRCE, Nagpur

YouTube Channel on Machine Learning Algorithms
https://tinyurl.com/GopalMachineLearningAlgorithms



Linear regression



Before we start Linear Regression
we have to perform cleaning and initial data analysis by

- Look at the summary of numerical variables.
- See the distribution of variables
- Look for possible correlation
- Explore any possible outliers
- Look for data errors with data sanity.
- Make sure data types are correct.

Country		Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40		Yes
France	35	58000	Yes
Spain		52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

Linear regression

Regression gives us simply the linear relationship of two or more variables within a dataset.

We have a dependent variable (or predictor variable) and has a relationship with independent variable (response variable).

Linear relationship between variables means that when the value of one or more independent variables will change (increase or decrease), the value of dependent variable will also change accordingly (increase or decrease).

Mathematically the relationship can be represented with the help of following equation –

$$Y=b+ mX$$

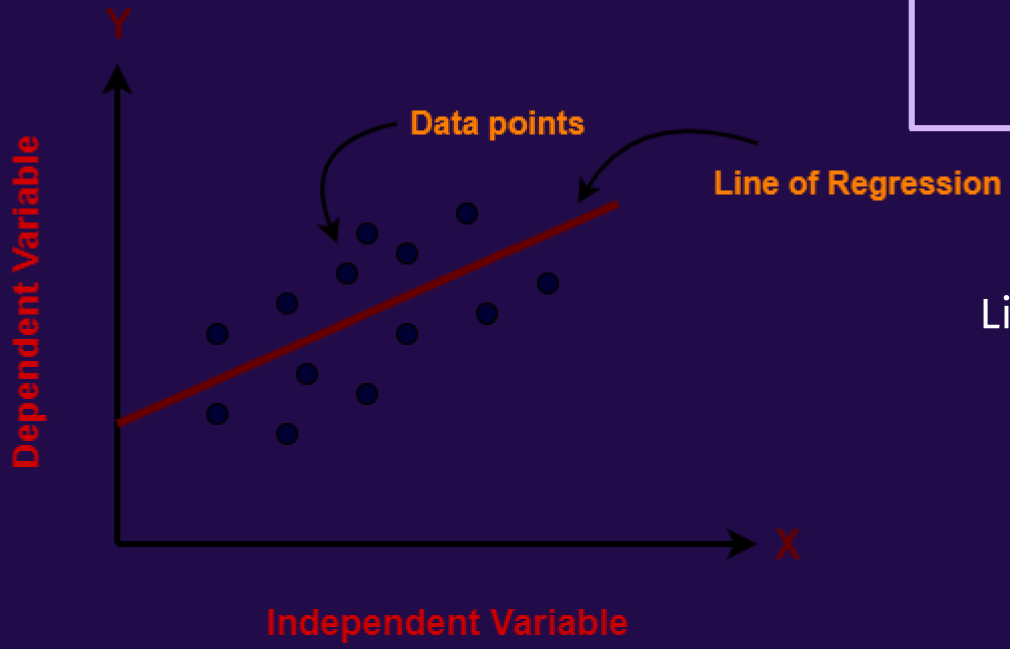
Here, Y is the dependent variable we are trying to predict.

X is the independent variable we are using to make predictions.

m is the slop of the regression line which represents the effect X has on Y

b is a constant.

Country		Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40	56400	Yes
France	35	58000	Yes
Spain	50	52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes



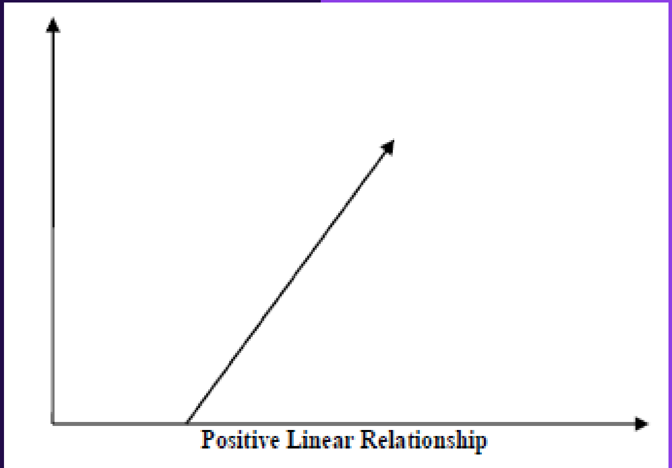
Linear regression

Linear regression model represents the linear relationship between a dependent variable and independent variable(s) via a sloped straight line.

Type of Linear regression

Positive Linear Relationship

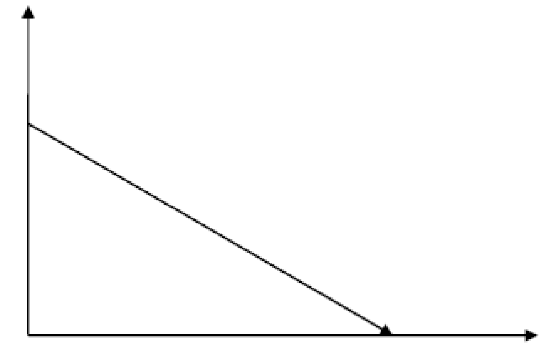
A linear relationship will be called positive if both independent and dependent variable increases. It can be understood with the help of following graph –



Type of Linear regression

Negative Linear relationship

A linear relationship will be called negative if independent increases and dependent variable decreases. It can be understood with the help of following graph –



Negative Linear Relationship

Simple Linear Regression(SLR)

- It is the most basic version of linear regression, which predicts a response using a single feature. The assumption in SLR is that the two variables are linearly related.
- In simple linear regression, the dependent variable depends only on a single independent variable.

  For simple linear regression, the form of the model is-

$$Y = \beta_0 + \beta_1 X$$

Here,

Y is a dependent variable.
X is an independent variable.
β_0 and β_1 are the regression coefficients.
β_0 is the intercept point
β_1 is the slope of line

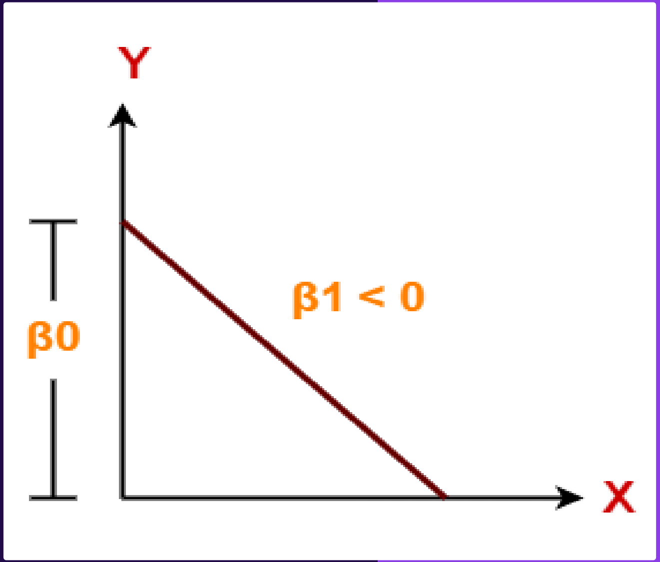
Type of Linear regression

Simple Linear Regression

- There are following 3 cases possible-

Case-01: $\beta 1 < 0$

- It indicates that variable X has negative impact on Y.
- If X increases, Y will decrease and vice-versa.

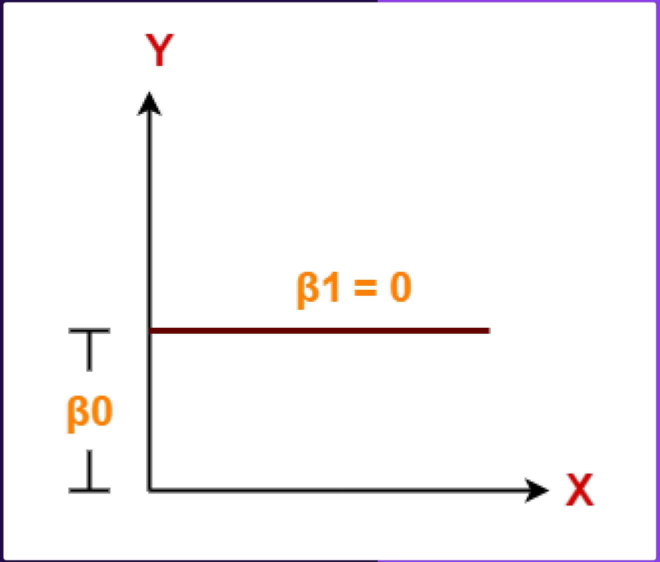


Type of Linear regression

Simple Linear Regression

Case-02: $\beta 1 = 0$

- It indicates that variable X has no impact on Y.
- If X changes, there will be no change in Y

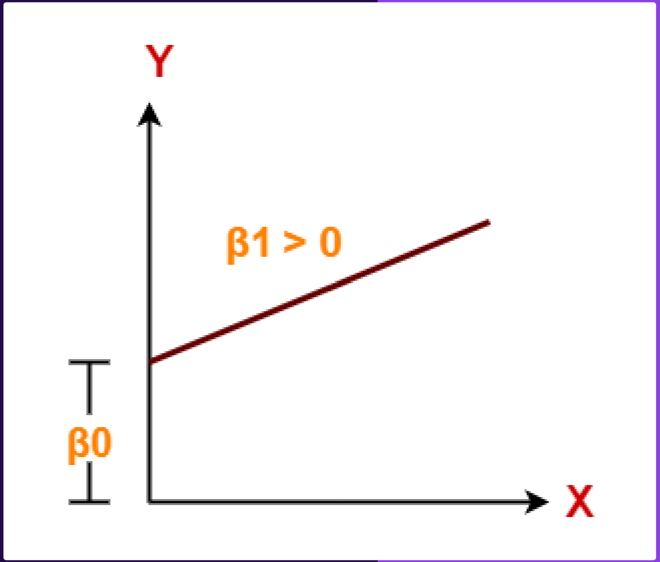


Type of Linear regression

Simple Linear Regression

Case-03: $\beta1 > 0$

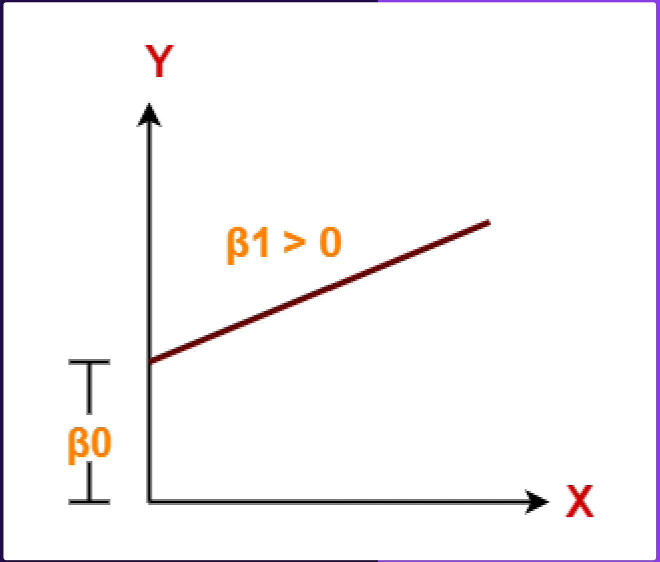
- It indicates that variable X has positive impact on Y.
- If X increases, Y will increase and vice-versa.



Type of Linear regression

Simple Linear Regression

Exp: the weight of a person is depend on height of a person.



Multiple Linear Regression-

In multiple linear regression, the dependent variable depends on more than one independent variables.

For multiple linear regression, the form of the model is-

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + \ldots\ldots + \beta nXn$$

Here,

Y is a dependent variable.
X1, X2, ..., Xn are independent variables.
β0, β1,..., βn are the regression coefficients.
βj (1<=j<=n) is the slope or weight that specifies the factor by which Xj has an impact on Y.

Multiple Linear Regression

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + \ldots\ldots + \beta nXn$$

Exp: Price of Flat depend on size of flat, floor , location, module kitchen etc.

$$Y= 0.9+ 1.2 \; . \; X1+ 2 \, . \, X2 + 4. \, X3 + 1 \, . \, X4$$

It is indication that which factor is more important for predicting price of flat (Y).

Let , X3 is most important factor for this prediction , so keeping the regression coefficients value 4 for X3.

Linear regression

- Given an input x compute an output y
- For example:

  - Predict height from age

  - Predict house price from house area

  - Predict distance from wall from sensors

Linear regression

- Relationship Between Variables Is a Linear Function

Population Y-Intercept

Population Slope

Random Error

$$Y = \beta_0 + \beta_1 x_1 + \epsilon$$

Linear regression

If the target variable is a continuous numeric variable (100–2000), then use a regression algorithm.

You can predict a continuous dependent variable from a number of independent variables.

Sq. area
Location
No. of bedrooms

$$y = w * x + b$$

This shows the relationship between price (y) and sq. area (x), where price is a number from a defined range.

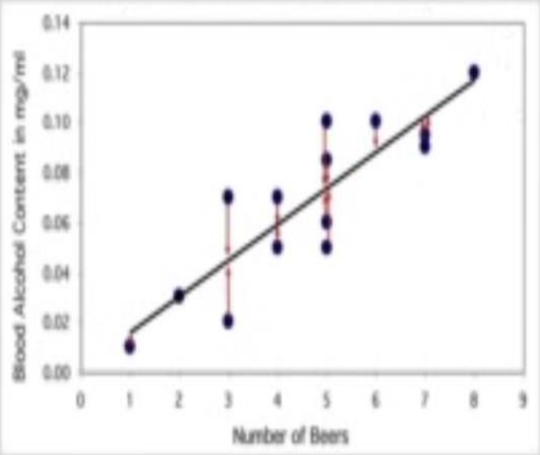
On the basis of size of house to predict selling price of a house

Linear regression

Linear regression

The regression line

The least-squares regression line is the unique line such that the sum of the squared vertical (y) distances between the data points and the line is the smallest possible.



Linear regression

Linear Regression

$$h(x) = \sum_{i=0}^{n} \beta_i x_i$$

To learn the parameters θ (β_i) ?

- Make h(**x**) close to *y*, for the available *training examples.*
- Define a cost function $J(\theta)$

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h(x)^{(i)} - (y)^{(i)})^2$$

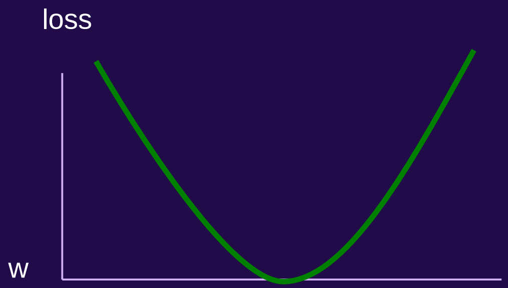
- Find θ that minimizes J(θ).

Finding the Minimum

You're blindfolded, but you can see out of the bottom of the blindfold to the ground right by your feet. I drop you off somewhere and tell you that you're in a convex shaped valley and escape is at the bottom/minimum.

How do you get out?

Finding the minimum



How can we do this for a function?

One approach: gradient descent

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).

Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.

Gradient Descent

- Think of a large bowl like what you would eat cereal out of or store fruit in. This bowl is a plot of the cost function (f).
- A random position on the surface of the bowl is the cost of the current values of the coefficients (cost).
- The bottom of the bowl is the cost of the best set of coefficients, the minimum of the function.
- The goal is to continue to try different values for the coefficients, evaluate their cost and select new coefficients that have a slightly better (lower) cost.
- Repeating this process enough times will lead to the bottom of the bowl and you will know the values of the coefficients that result in the minimum cost.

One approach: gradient descent

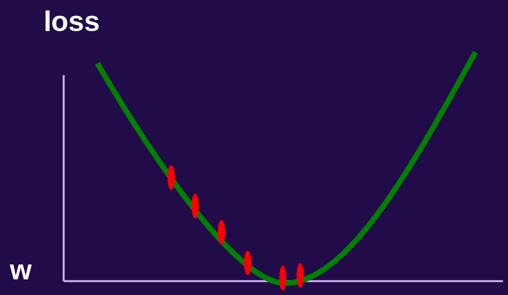
Partial derivatives give us the slope
(i.e. direction to move) in that dimension

One approach: gradient descent

Partial derivatives give us the slope (i.e. direction to move) in that dimension

Approach:

- pick a starting point (w)
- repeat:
  - pick a dimension
  - move a small amount in that dimension towards decreasing loss (using the derivative)

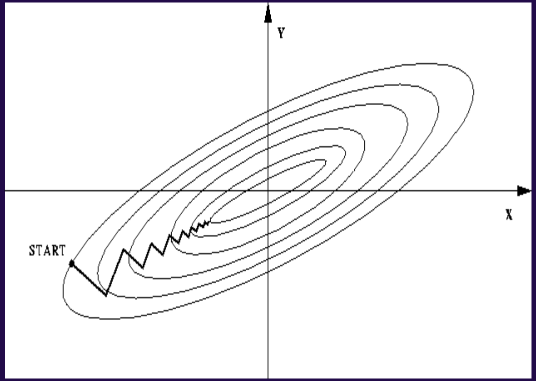


One approach: gradient descent

Partial derivatives give us the slope (i.e. direction to move) in that dimension

Approach:

- pick a starting point (w)
- repeat:
  - pick a dimension
  - move a small amount in that dimension towards decreasing loss (using the derivative)



Gradient Descent Procedure

- The procedure starts off with initial values for the coefficient or coefficients for the function. These could be 0.0 or a small random value.

coefficient = 0.0

- The cost of the coefficients is evaluated by plugging them into the function and calculating the cost.

cost = f(coefficient)

or

cost = evaluate(f(coefficient))

Gradient Descent Procedure

- The derivative of the cost is calculated.
- The derivative is a concept from calculus and refers to the slope of the function at a given point.
- We need to know the slope so that we know the direction (sign) to move the coefficient values in order to get a lower cost on the next iteration.

delta = derivative(cost)

Gradient Descent Procedure

- Now that we know from the derivative which direction is downhill, we can now update the coefficient values.

- A learning rate parameter (alpha) must be specified that controls how much the coefficients can change on each update.

coefficient = coefficient – (alpha * delta)

- This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough

Gradient descent

- pick a starting point (w)
- repeat until loss doesn't decrease in all dimensions:
  - pick a dimension
  - move a small amount in that dimension towards decreasing loss (using the derivative)

$$w_j = w_j - h \frac{d}{dw_j} loss(w)$$

What does this do?

Gradient descent

- pick a starting point (w)
- repeat until loss doesn't decrease in all dimensions:
  - pick a dimension
  - move a small amount in that dimension towards decreasing loss (using the derivative)

$$w_j = w_j - h \frac{d}{dw_j} loss(w)$$

learning rate (how much we want to move in the error direction, often this will change over time)

Some math's

$$\frac{d}{dw_j} loss$$

$$= \frac{d}{dw_j} \sum_{i=1}^{n} \exp(-y_i(w \cdot x_i + b))$$

$$= \sum_{i=1}^{n} \exp(-y_i(w \cdot x_i + b)) \frac{d}{dw_j} - y_i(w \cdot x_i + b)$$

$$= \sum_{i=1}^{n} -y_i x_{ij} \exp(-y_i(w \cdot x_i + b))$$

Gradient descent

- pick a starting point (w)
- repeat until loss doesn't decrease in all dimensions:
  - pick a dimension
  - move a small amount in that dimension towards decreasing loss (using the derivative)

$$w_j = w_j + \eta \sum_{i=1}^{n} y_i x_{ij} \exp(-y_i(w \cdot x_i + b))$$

What is this doing?

Summary: Gradient descent

Gradient descent minimization algorithm

- require that our loss function is convex
- make small updates towards lower losses
- Gradient descent is a simple optimization procedure that you can use with many machine learning algorithms.
- Batch gradient descent refers to calculating the derivative from all training data before calculating an update.
- Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately.

Gradient descent

```
import matplotlib.pyplot as plt
import numpy as np

# original data set
X = [1, 2, 3]
y = [1, 2, 3]

# slope of best_fit_1 is 0.5
# slope of best_fit_2 is 1.0
# slope of best_fit_3 is 1.5

hyps = [0.5, 1.0, 1.5]

# multiply the original X values by the theta
# to produce hypothesis values for each X
def multiply_matrix(mat, theta):
    mutated= []
    for i in range(len(mat)):
        mutated.append(mat[i] * theta)
    return mutated
```

Gradient descent

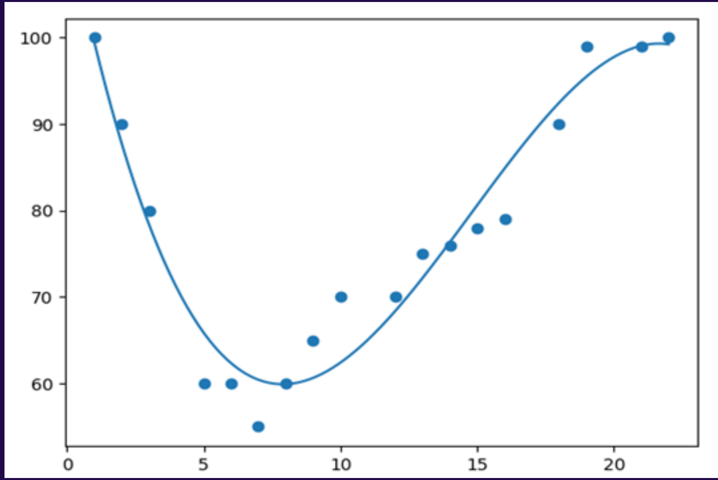
```
# calculate cost by looping each sample
# subtract hyp(x) from y
# square the result
# sum them all together
def calc_cost(m, X, y):
    total = 0
    for i in range(m):
        squared_error = (y[i] - X[i]) ** 2
        total += squared_error

    return total * (1 / (2*m))

# calculate cost for each hypothesis
for i in range(len(hyps)):
    hyp_values = multiply_matrix(X, hyps[i])

print("Cost for ", hyps[i], " is ", calc_cost(len(X), y, hyp_values))
```

Polynomial Regression

- If your linear regression model cannot model the relationship between the target variable and the predictor variable?
- In other words, what if they don't have a linear relationship?
- Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.
- Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an nth degree polynomial.

Polynomial Regression

- Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x)



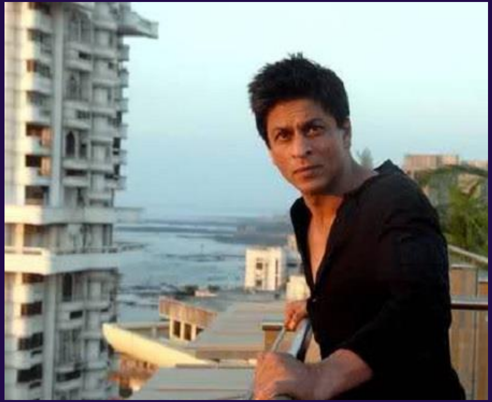
Decision trees

- In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.
- Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions.
- Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms.
- The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome.
- The example of a binary tree for predicting whether a person is fit or unfit providing various information like , eating habits and exercise habits, is given below –

Waiting outside the house to get an autograph.

Which days does he come out to enjoy sports?

41



- Sky condition
- Humidity
- Temperature
- Wind
- Water
- Forecast

- Attributes of a day: takes on values

Learning Task

- We want to make a hypothesis about the day on which SRK comes out..
  - in the form of a boolean function on the attributes of the day.
- Find the right hypothesis/function from historical data

Training Examples for EnjoySport

	Sky	Temp	Humid	Wind	Water	Forecst		EnjoySpt
c(	Sunny	Warm	Normal	Strong	Warm	Same	)=1	Yes
c(	Sunny	Warm	High	Strong	Warm	Same	)=1	Yes
c(	Rainy	Cold	High	Strong	Warm	Change	)=0	No
c(	Sunny	Warm	High	Strong	Cool	Change	)=1	Yes

- Negative and positive learning examples
- Concept learning:

  c is the target concept

  - Deriving a Boolean function from training examples
    - Many “hypothetical” boolean functions
      - Hypotheses; find h such that h = c.
  - Other more complex examples:
    - Non-boolean functions
- Generate hypotheses for concept from TE’s

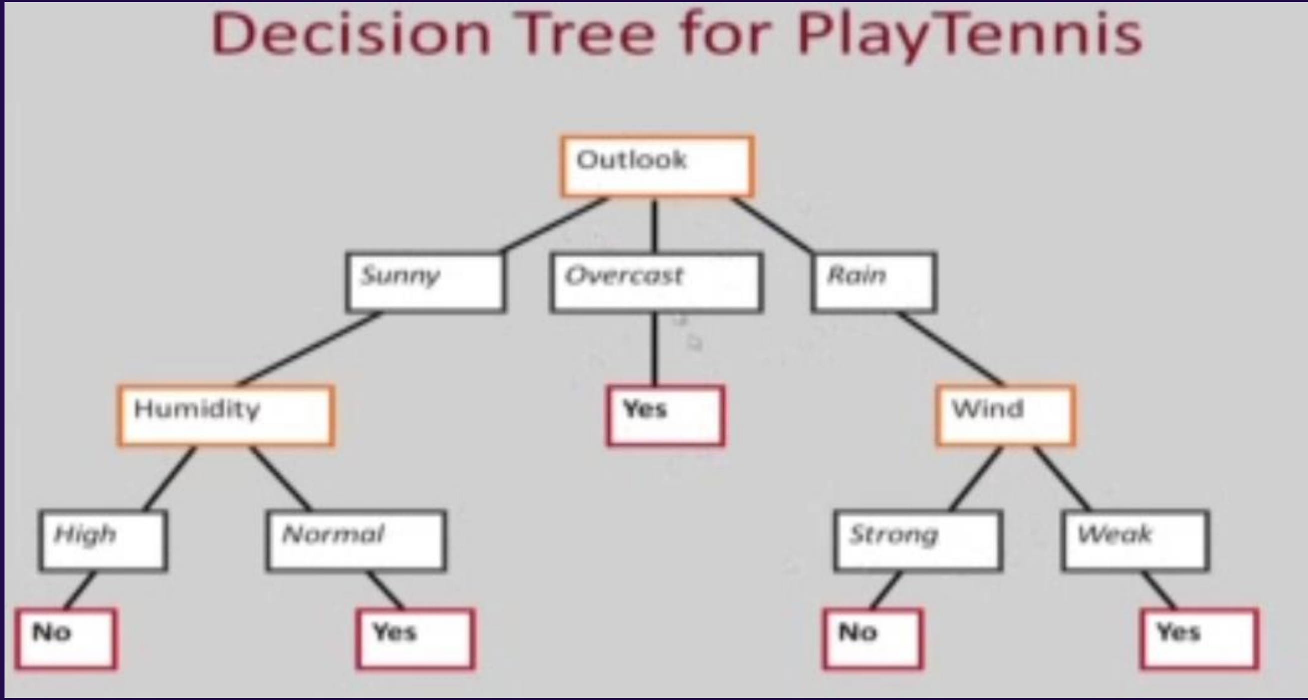
Implementing Decision Tree Algorithm

Decision Tree for PlayTennis

- Attributes and their values:
  - Outlook: *Sunny, Overcast, Rain*
  - Humidity: *High, Normal*
  - Wind: *Strong, Weak*
  - Temperature: *Hot, Mild, Cool*

- Target concept - Play Tennis: *Yes, No*

Implementing Decision Tree Algorithm



Data Set

Sr. No	Age	Competition	Type	Profit
1	Old	Yes	SW	Down
2	Old	No	SW	Down
3	Old	No	HW	Down
4	Mid	Yes	SW	Down
5	Mid	Yes	HW	Down
6	Mid	No	HW	Up
7	Mid	No	SW	Up
8	New	Yes	SW	Up
9	New	No	HW	Up
10	new	No	SW	Up

Decision Tree

Before we started to design Decision tree, following four steps are important

1. To find the Target / class attribute (Profit)
2. To Find the Information Gain of Target attribute
3. To find the Entropy (for deciding root of tree)
4. At the end find the Gain of each attribute

Decision Tree

- Now, Find the Information Gain of target attribute

$$IG = -\left[\frac{P}{P+N} \log_2 \left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2 \left(\frac{N}{P+N}\right) \right]$$

where P=down , N= Up

- Then find the Entropy of given attribute

$$E(A) = \sum_{i=1}^{n} \frac{P_i + N_i}{P+N} \times I(P_i, N_i)$$

i.e. Information gain of Attribute X Probability of that attribute

- Finally , find the Gain for all attribute (here for 3 attributes), those Gain will be greatest , we should called it as Root of Tree.

Gain =IG- E(A)

Decision Tree

Now, Find Information Gain of target attribute

$$IG = -\left[\frac{5}{10}\log_2 (5/10) + \frac{5}{10}\log_2 (5/10)\right]$$

$$= -\left[0.5 \log_2 . 2^{-1} + 0.5 \log_2 . 2^{-1}\right]$$

$$= -\left[0.5 \times (-1 \log_2 . 2) + 0.5 \times (-1 \log_2 . 2)\right]$$

$$= -\left[-0.5 - 0.5\right] \qquad = -\left[-1\right]$$

Sr. No	Age	Competition	Type	Profit
1	Old	Yes	SW	Down
2	Old	No	SW	Down
3	Old	No	HW	Down
4	Mid	Yes	SW	Down
5	Mid	Yes	HW	Down
6	Mid	No	HW	Up
7	Mid	No	SW	Up
8	New	Yes	SW	Up
9	New	No	HW	Up
10	New	No	SW	Up

Decision Tree

Now, Find the Entropy of each attributes
Lets , start with attribute

$$I(old) = -\left[\left(\frac{3}{3}\right) . \log_2 \left(\frac{3}{3}\right) + \frac{0}{3} . \log_2 \left(\frac{0}{3}\right) \right]$$

$= - [0]$

$= 0 \times$ (probability of old) 3/10

$= 0 \times 3/10 = \mathbf{0}$

$$I(mid) = -\left[\left(\frac{2}{4}\right) . \log_2 \left(\frac{2}{4}\right) + \frac{2}{4} . \log_2 \left(\frac{2}{4}\right) \right]$$

$= 1 \times$ (probability of mid) 4/10

$= 1 \times 0.4 = \mathbf{0.4}$

$$I(new) = -\left[\left(\frac{0}{3}\right) . \log_2 \left(\frac{0}{3}\right) + \frac{3}{3} . \log_2 \left(\frac{3}{3}\right) \right]$$

$= 0 \times$ (probability of new) 3/10

$= 0 \times 0.3 = \mathbf{0}$

E (Age) = 0 + 0.4+ 0
= 0.4

Age =

	Down	UP
Old	3	0
Mid	2	2
New	0	3

Sr .No	Age	Competition	Type	Profit
1	Old	Yes	SW	Down
2	Old	No	SW	Down
3	Old	No	HW	Down
4	Mid	Yes	SW	Down
5	Mid	Yes	HW	Down
6	Mid	No	HW	Up
7	Mid	No	SW	Up
8	New	Yes	SW	Up
9	New	No	HW	Up
10	New	No	SW	Up

Decision Tree

Now, find the Gain of all attribute

Where,

Gain (Age) = IG- E(Age)
= 1- 0.4
= 0.6

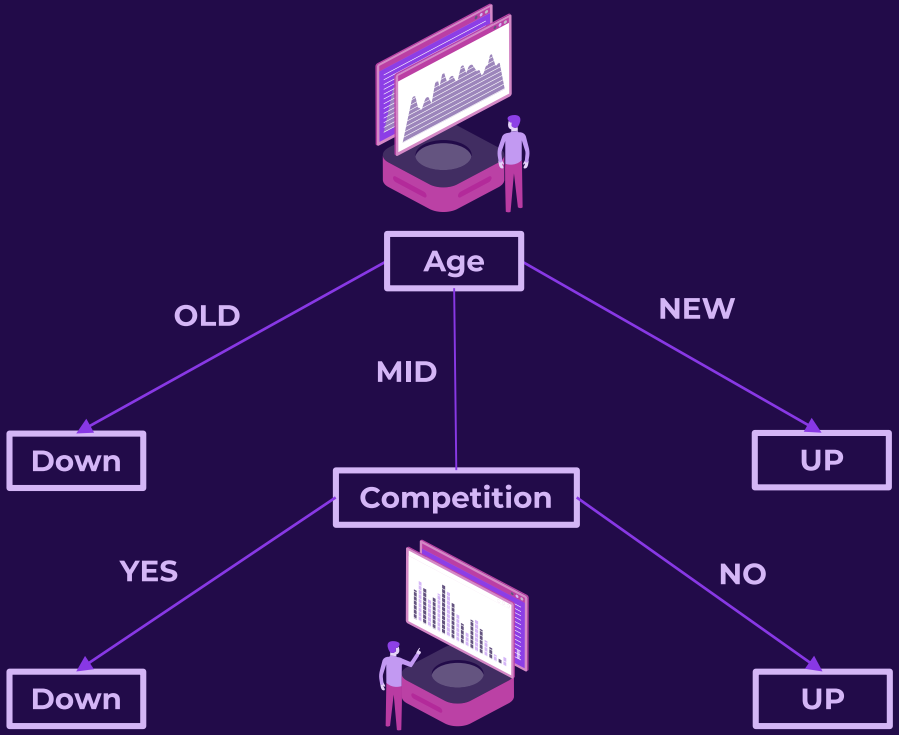
Gain(Competition) = IG- E(Competition)
= 0.124

Gain(Type) = IG- E(Type)
= 0

	Down	UP
Old	3	0
Mid	2	2
New	0	3

Sr.No	Age	Competition	Type	Profit
1	Old	Yes	SW	Down
2	Old	No	SW	Down
3	Old	No	HW	Down
4	Mid	Yes	SW	Down
5	Mid	Yes	HW	Down
6	Mid	No	HW	Up
7	Mid	No	SW	Up
8	New	Yes	SW	Up
9	New	No	HW	Up
10	New	No	SW	Up

Decision Tree



Sr.No	Age	Competition	Type	Profit
1	Old	Yes	SW	Down
2	Old	No	SW	Down
3	Old	No	HW	Down
4	Mid	Yes	SW	Down
5	Mid	Yes	HW	Down
6	Mid	No	HW	Up
7	Mid	No	SW	Up
8	New	Yes	SW	Up
9	New	No	HW	Up
10	New	No	SW	Up

Decision Tree

Income	Gender	Marital Status	Buys
High	Male	Single	No
High	Male	Married	No
High	Male	Single	Yes
Medium	Male	Single	Yes
Low	Female	Single	Yes
Low	Female	Married	No
Low	Female	Married	Yes
Medium	Male	Single	No
Low	Female	Married	Yes
Medium	Female	Single	Yes
Medium	Female	Married	Yes
Medium	Male	Married	Yes
High	Female	Single	Yes
Medium	Male	Married	No

Over fitting

01

Over Fitting

Too MUCH DATA given to Machine so that It become CONFUSED in things!

02

Under Fitting

so LESS DATA given to Machine that it NOT ABLE to Understand Things.

Over fitting

01

Over Fitting

Too MUCH DATA given to Machine so that It become CONFUSED in things!

“OverFitting" : A hypothesis h is said to overfed the training data , if there is another hypothesis h’ , such that h’ has more error then h on training data but, h’ has less error than h on test data.

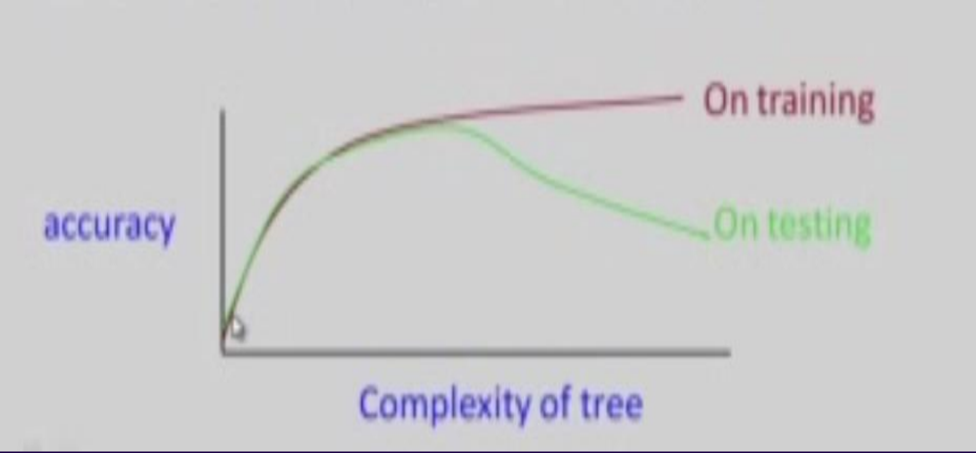
Exp: If we have small decision tree and it has higher error in training data and lower error on test data compare to larger decision tree , which has smaller error in training data and higher error on test data , then we say that overftting has occurred

Over fitting

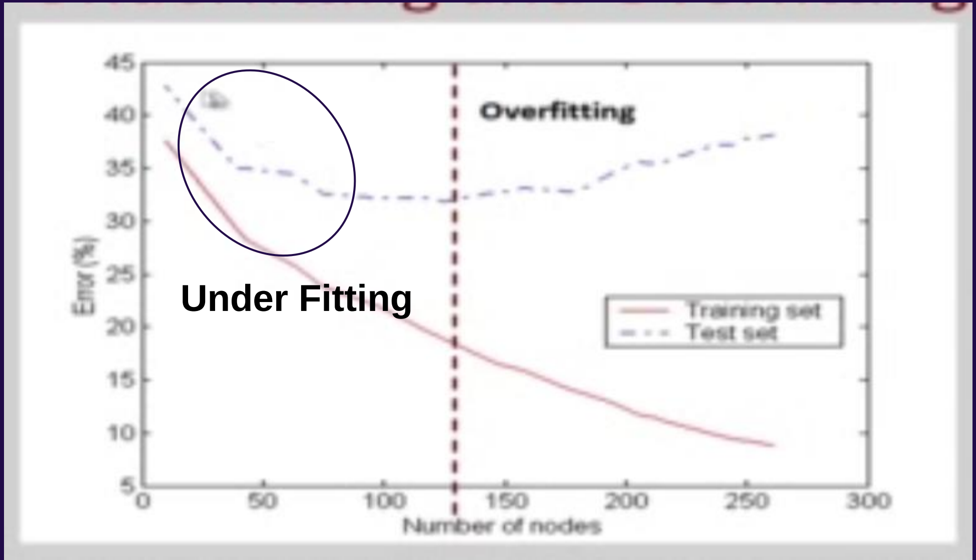
01

Over Fitting

Too MUCH DATA given to Machine so that It become CONFUSED in things!

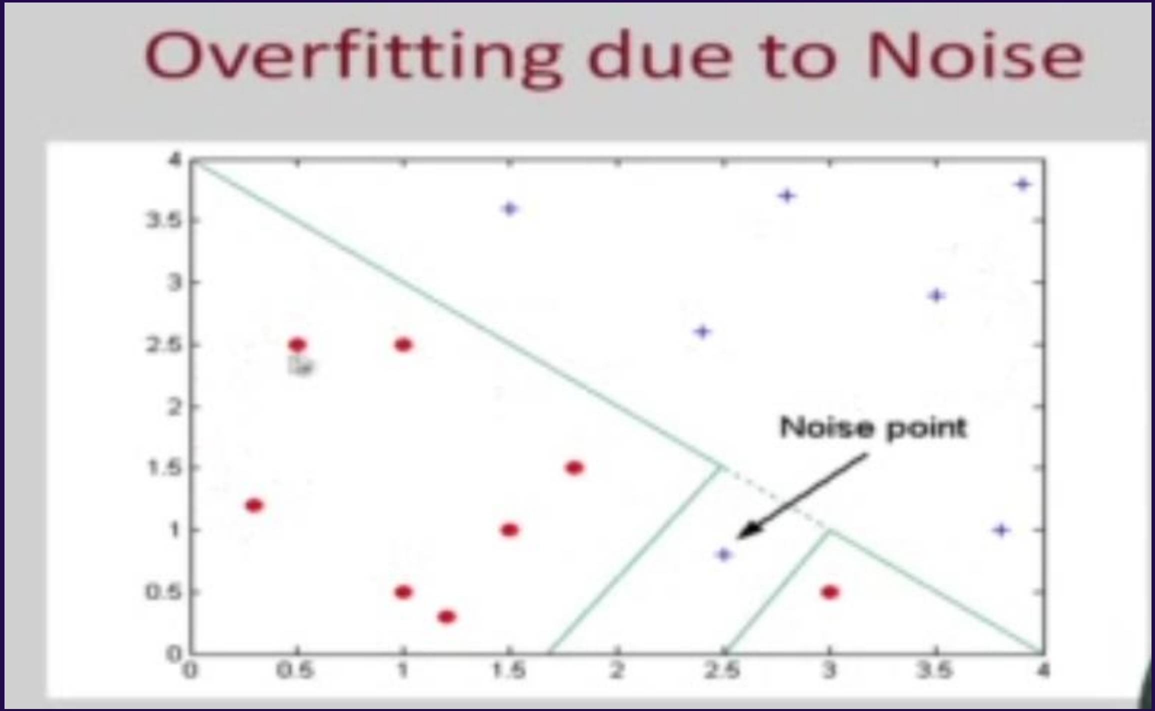


Over fitting



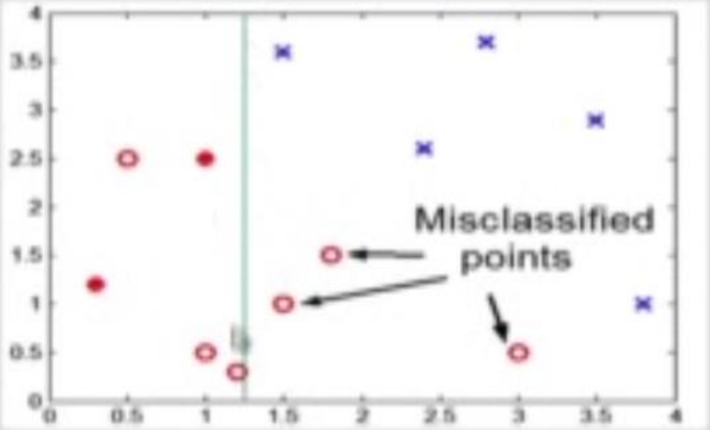
Under Fitting: when model is too simple, both training and test errors are large

Over fitting



Over fitting

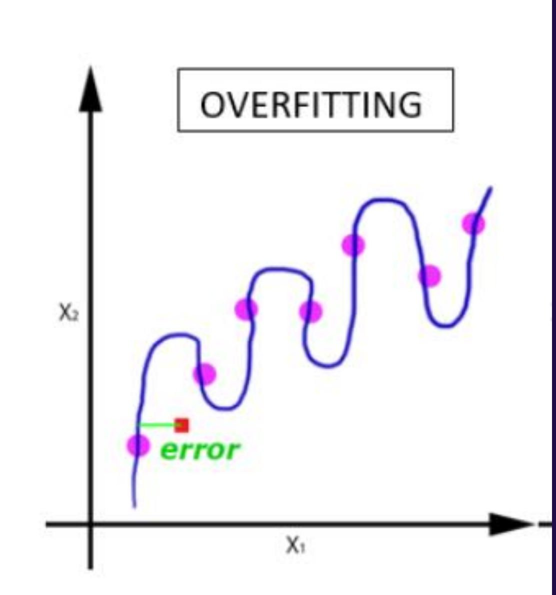
Overfitting due to Insufficient Examples



Lack of data points makes it difficult to predict correctly the class labels of that region

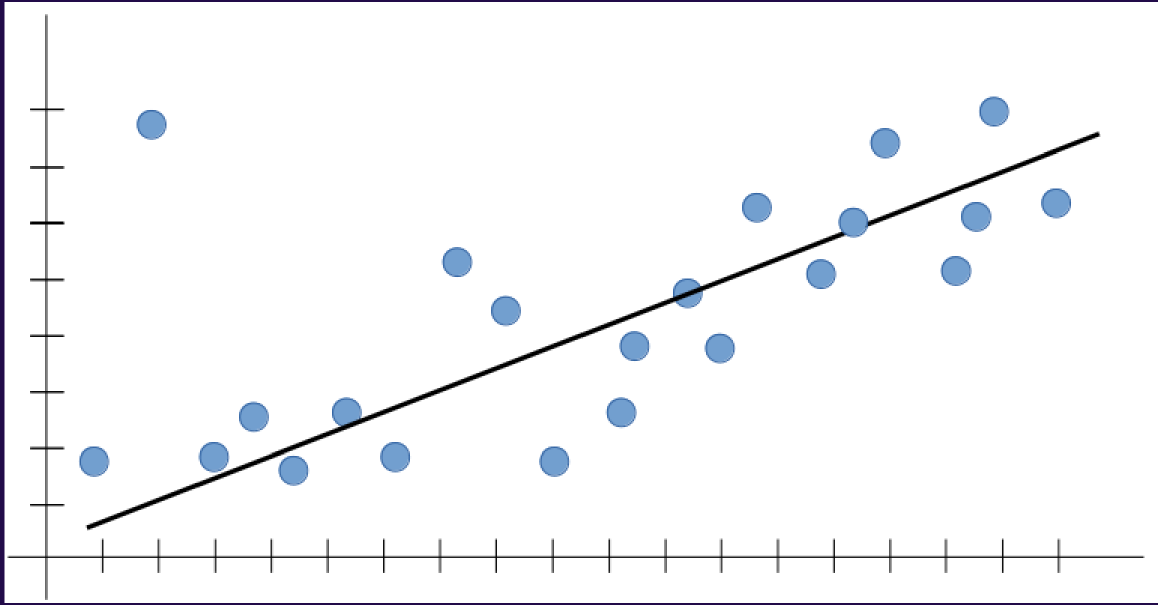
Over fitting

Over fitting : When model is so complex. Here , your model is trying to cover every data points of output on X , Y plot.



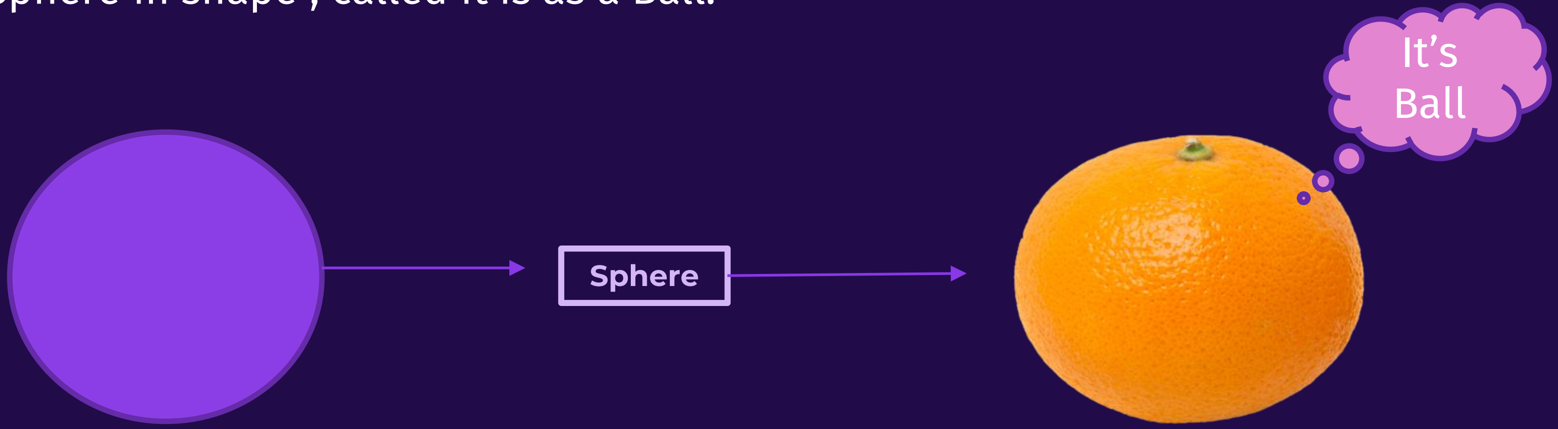
Under fitting

Underfitting : When model is so simple . Here , your model is trying to cover very few data points of output on X , Y plot.



Under fitting

Example :Let us consider that , you have to train your model that if any object looks Sphere in shape , called it is as a Ball.



Here , we are providing only one attribute to identify the object , i.e. Shape = Sphere

Over fitting

Example :Let us consider that , you have provide large number of attributes like , Sphere, Play, Not Eat, Radius=5 cm.

Sphere

Play

Not Eat

Radius = 5 cm

Here , we are providing lots of attributes to identify the object.

Over fitting

AI camera mistakes linesman's bald head for ball and follows it through match

During a football match in Scotland, an artificial intelligence (AI) camera continuously tracked a linesman's bald head mistaking it for the ball. A video of the gaffe has gone viral on social media. The commentator had to repeatedly apologise as the camera kept on mistaking the linesman's head for the ball.

Over fitting

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records

Over fitting

Avoid Overfitting

- How can we avoid overfitting a decision tree?
  - Prepruning: Stop growing when data split not statistically significant
  - Postpruning: Grow full tree then remove nodes
- Methods for evaluating subtrees to prune:
  - Minimum description length (MDL):

    Minimize: size(tree) + size(misclassifications(tree))
  - Cross-validation

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

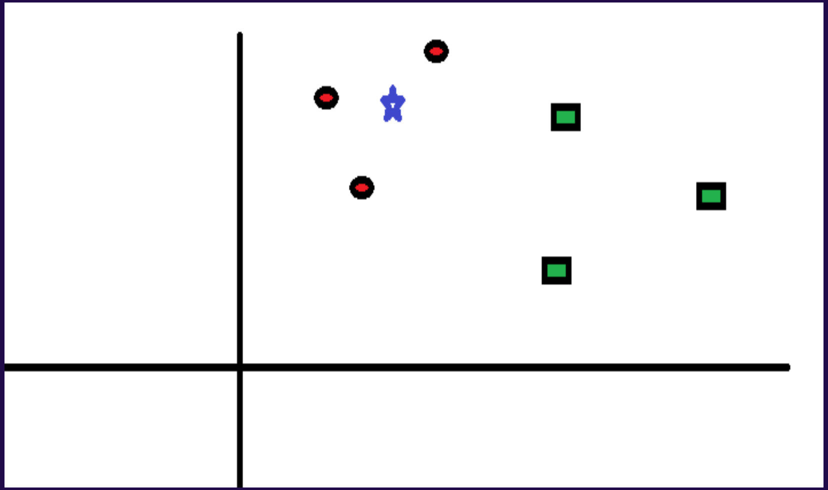
- KNN algorithm can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

- KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or avers the labels (in the case of regression).

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

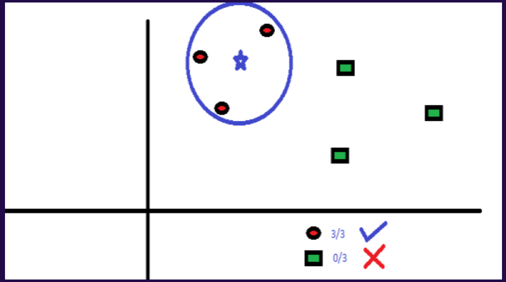
Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS)



Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

- Let, to find out the class of the blue star (BS).
- BS can either be RC or GS and nothing else. The “K” is KNN algorithm is the nearest neighbor we wish to take the vote from.
- Let’s say K = 3. Hence, we will now make a circle with BS as the center just as big as to enclose only three datapoints on the plane.
- Refer to the following diagram for more details:



- The three closest points to BS is all RC.
- Hence, with a good confidence level, we can say that the BS should belong to the class RC.
- Here, the choice became very obvious as all three votes from the closest neighbor went to RC

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

- Let, us take an another example
- Query : X=(Math=6, Comp Sci=8) , Is students Pass or Fail ?
- Here , we take K=3 any random value of K to find out nearest neighbors
- To find the distance between these values , we use Euclidean Distance

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

$$d = \sqrt{|X01 - XA1|^2 + |X02 - XA2|^2}$$

Where , Xo is observed value
Xa is actual value

Sr.No	Math	Comp Sci	Result
1	4	3	Fail
2	6	7	Pass
3	7	8	Pass
4	5	5	Fail
5	8	8	Pass
X	6	8	????

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

$$d = \sqrt{|X01 - XA1|^2 + |X02 - XA2|^2}$$

$$d1 = \sqrt{|6-4|^2 + |8-3|^2} = \sqrt{29} = 5.38$$

$$d2 = \sqrt{|6-6|^2 + |8-7|^2} = 1$$

$$d3 = \sqrt{|6-7|^2 + |8-8|^2} = 1$$

$$d4 = \sqrt{|6-5|^2 + |8-5|^2} = \sqrt{10} = 3.16$$

$$d5 = \sqrt{|6-8|^2 + |8-8|^2} = 2$$

Sr.No	Math	Comp Sci	Result
1	4	3	Fail
2	6	7	Pass
3	7	8	Pass
4	5	5	Fail
5	8	8	Pass
X	6	8	????

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

Three NN are (1,1,2,)

Sr. No	Math	Comp Sci	Result
2	6	7	Pass
3	7	8	Pass
5	8	8	Pass

$3\ Pass\ \ and\ 0\ Fail$

3> 0

Sr.No	Math	Comp Sci	Result
1	4	3	Fail
2	6	7	Pass
3	7	8	Pass
4	5	5	Fail
5	8	8	Pass
X	**6**	**8**	**Pass**

Instance Based Learning / Lazy Algorithm

K Nearest Neighbor Algorithm

Temp(X) in C	Humidity (Y) %	Rain Condition
27.8	76	Yes
28.2	76	Yes
28.7	80	No
28.6	81.6	Yes
27.7	89.4	Yes
30.5	89.9	No
26.7	81.4	Yes
25.9	85	No
36	90	No
31.8	88	Yes
35.7	70	No

Using KNN algorithm find the Rain Condition, Let K=3
When Temp: 29.6 C and Humidity: 78 %

KNN

01 Why do we need KNN?

02 What is KNN?

03 How do we choose the factor ‘K’?

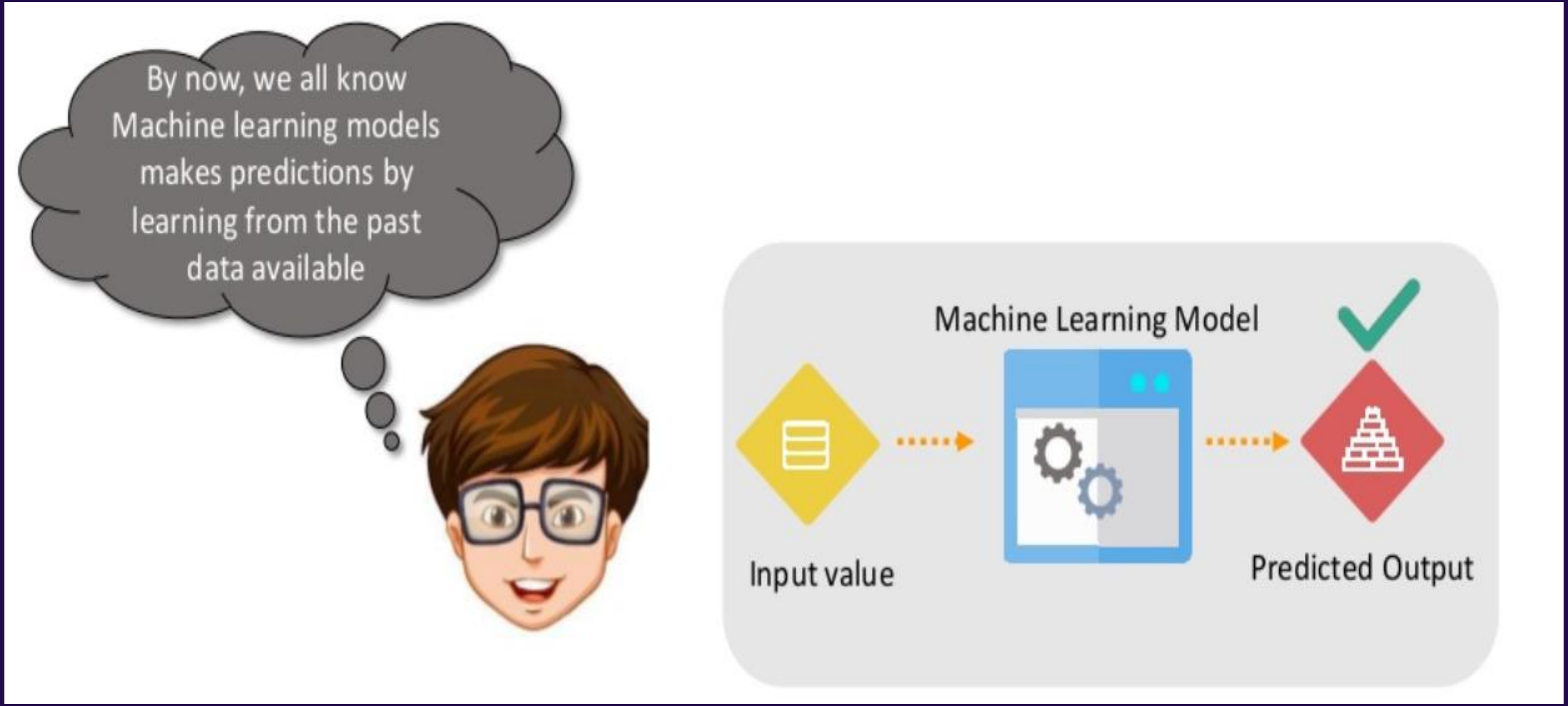
04 When do we use KNN?

05 How does KNN Algorithm work?

06 Use Case: Predict whether a person will need diabetes or not

Why KNN?

Why KNN?

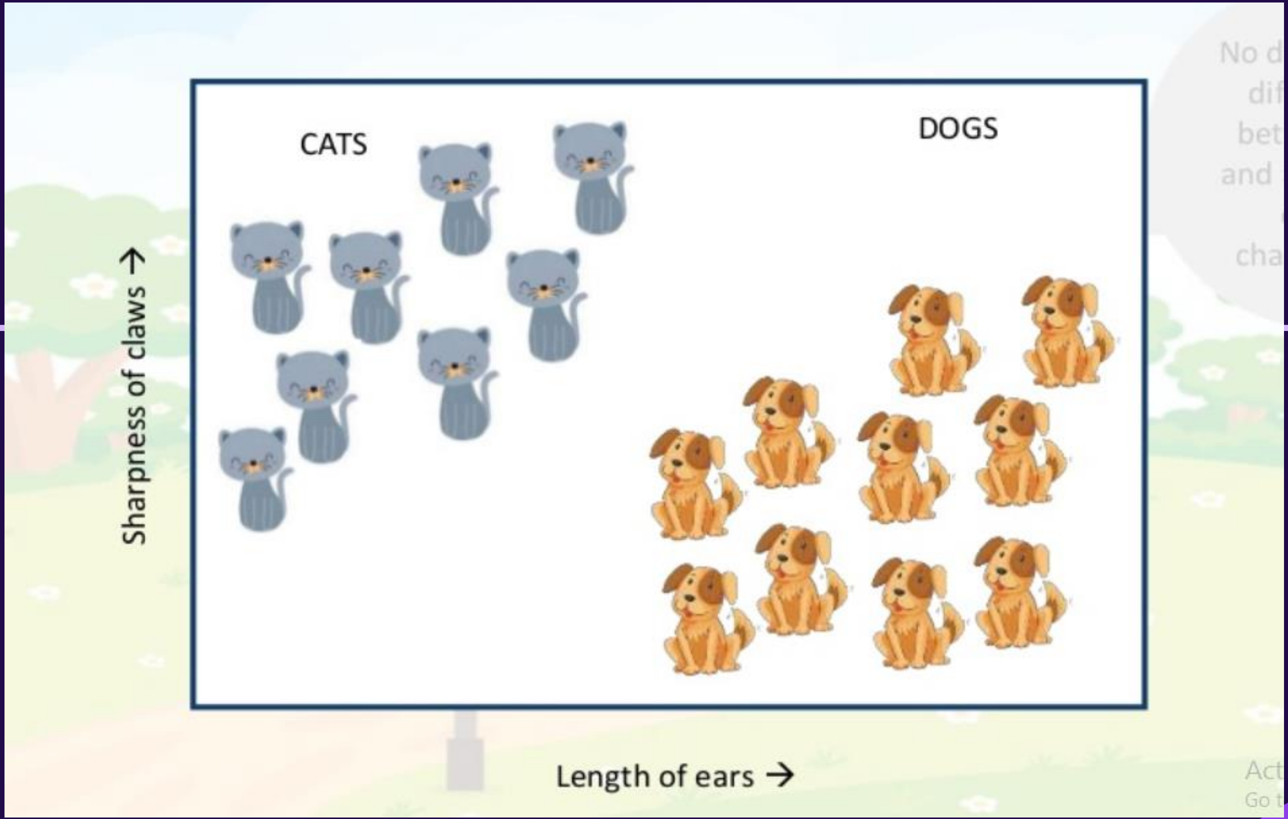


By now, we all know Machine learning models makes predictions by learning from the past data available
Machine Learning Model
Input value
Predicted Output

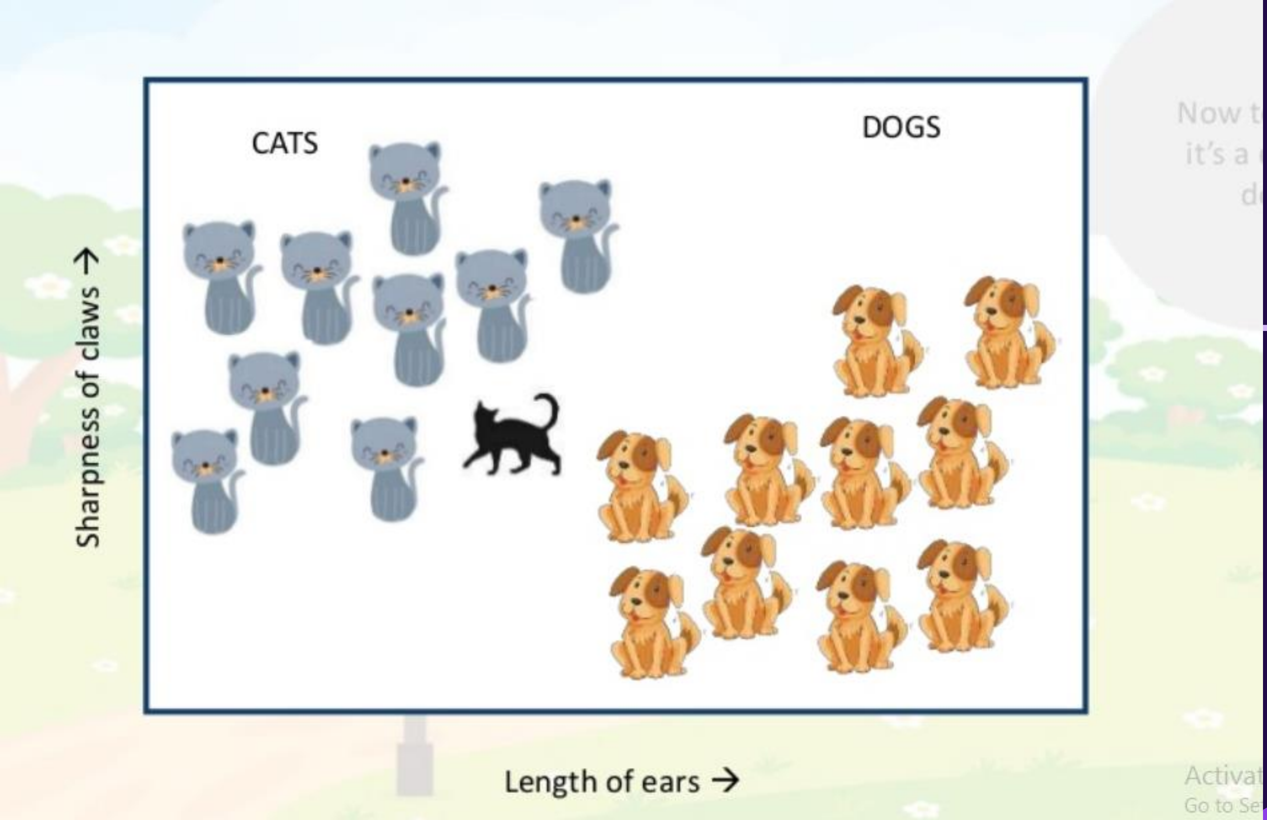

Is that a dog?


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

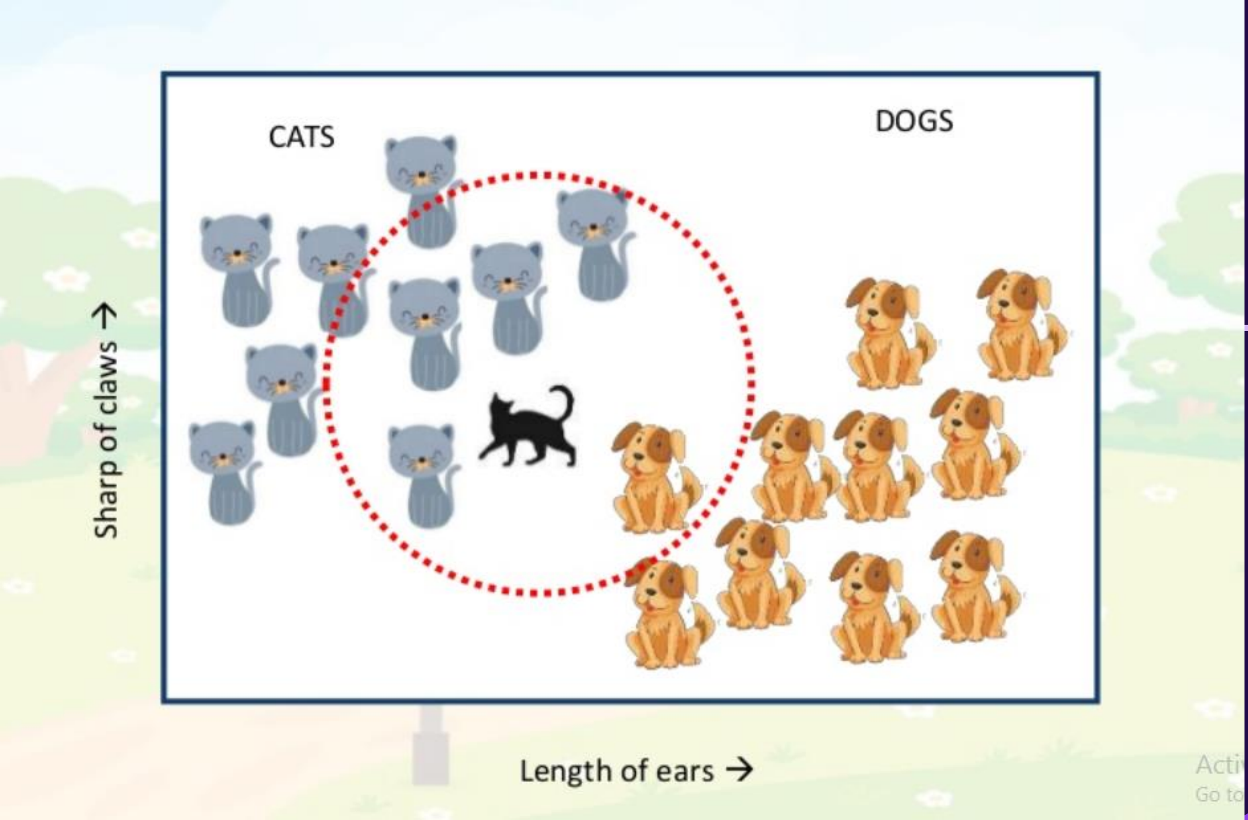
CATS	DOGS
	
Sharp Claws, uses to climb	Dull Claws
Smaller length of ears	Bigger length of ears
Meows and purrs	Barks
Doesn't love to play around	Loves to run around


CATS
DOGS
Sharpness of claws →
Length of ears →

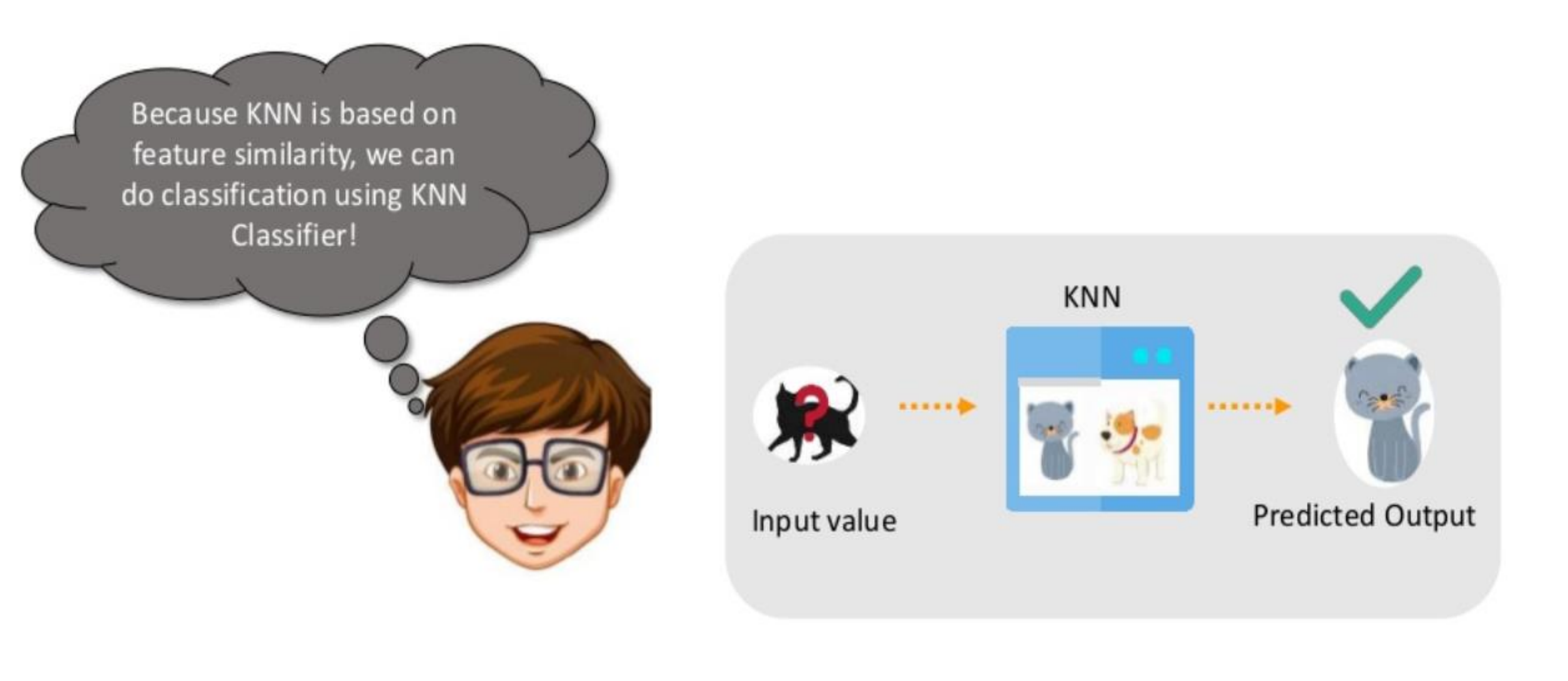

Now tell me if it
is a cat or a dog?


CATS
DOGS
Sharpness of claws →
Length of ears →


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


CATS
DOGS
Sharp of claws →
Length of ears →

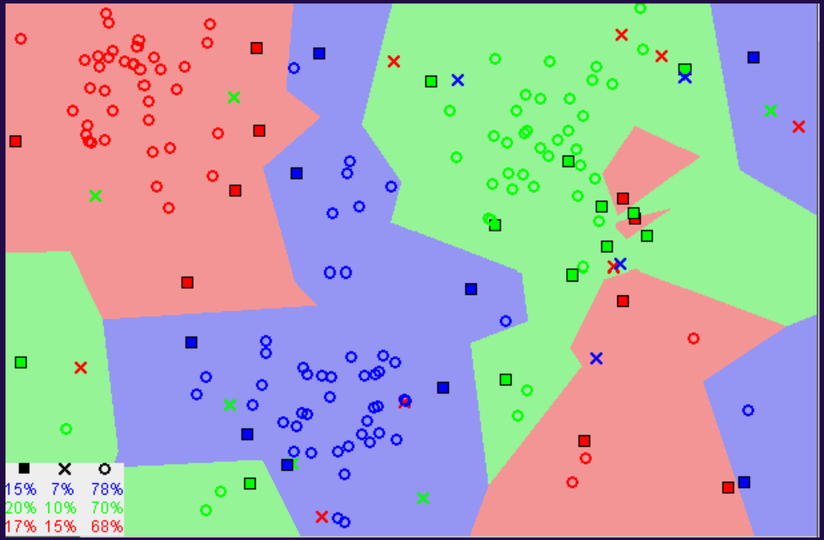
Why KNN?


Because KNN is based on feature similarity, we can do classification using KNN Classifier!
KNN
Input value
Predicted Output

What is KNN?

What is KNN?

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.



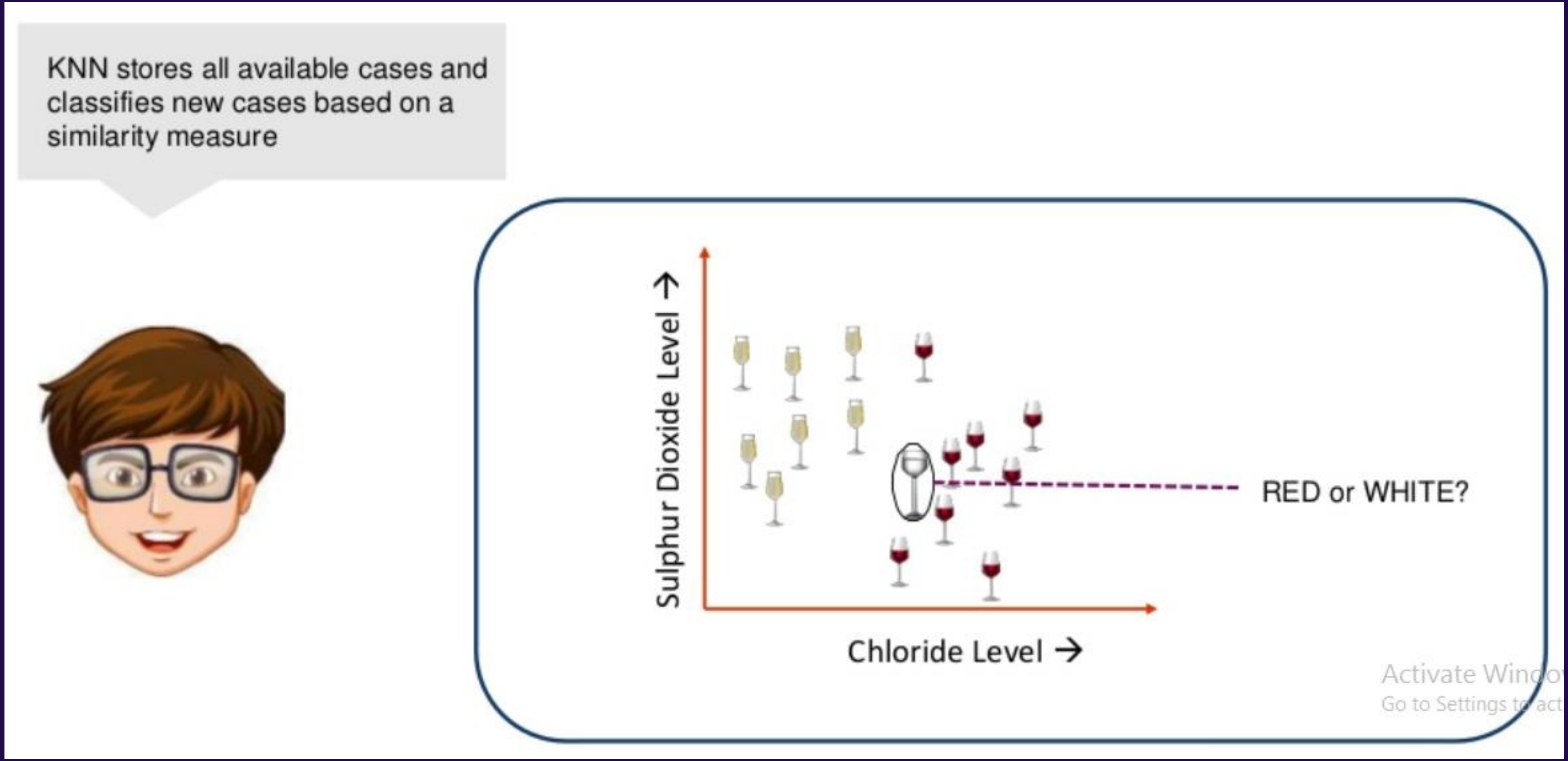
What is KNN Algorithm?

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

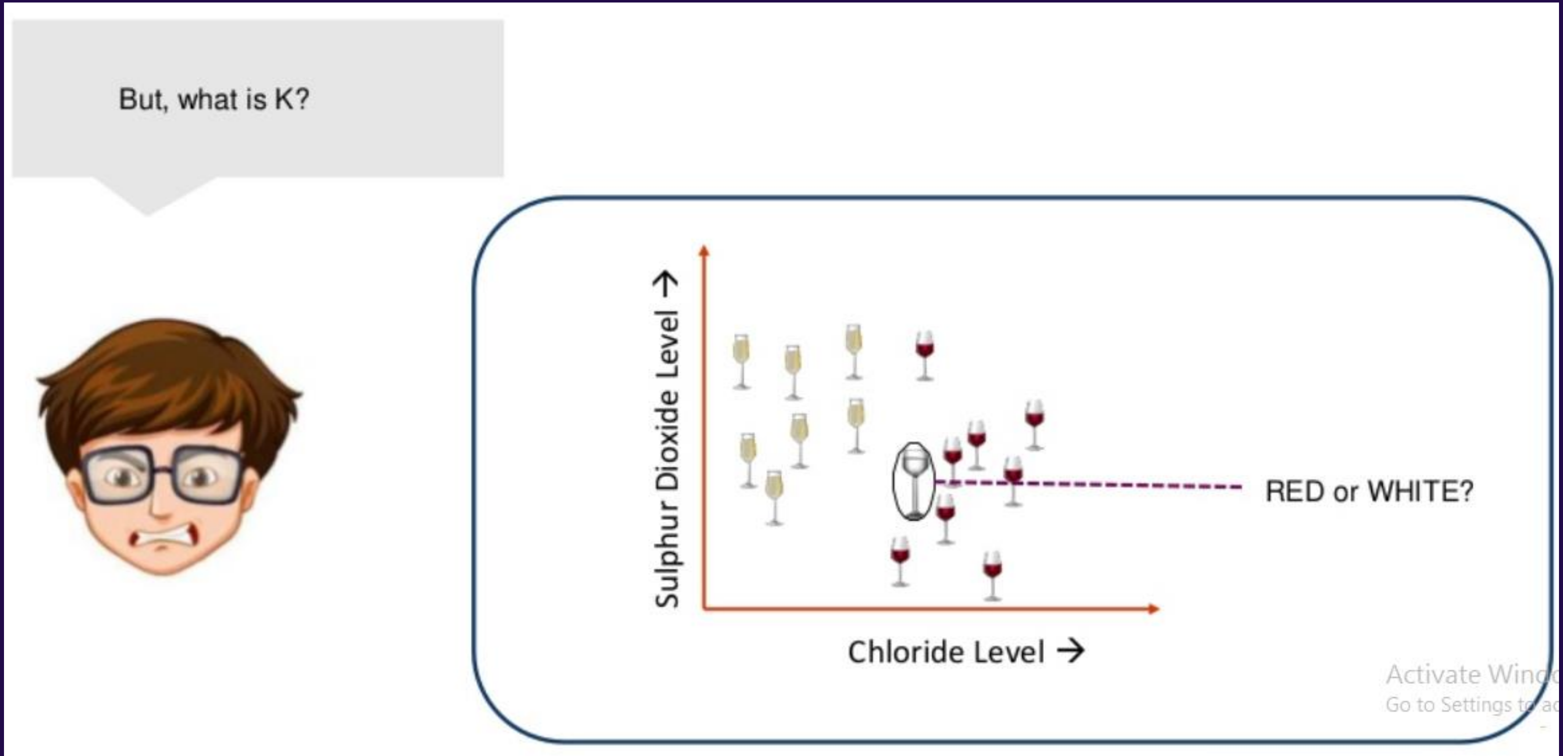
Classification

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

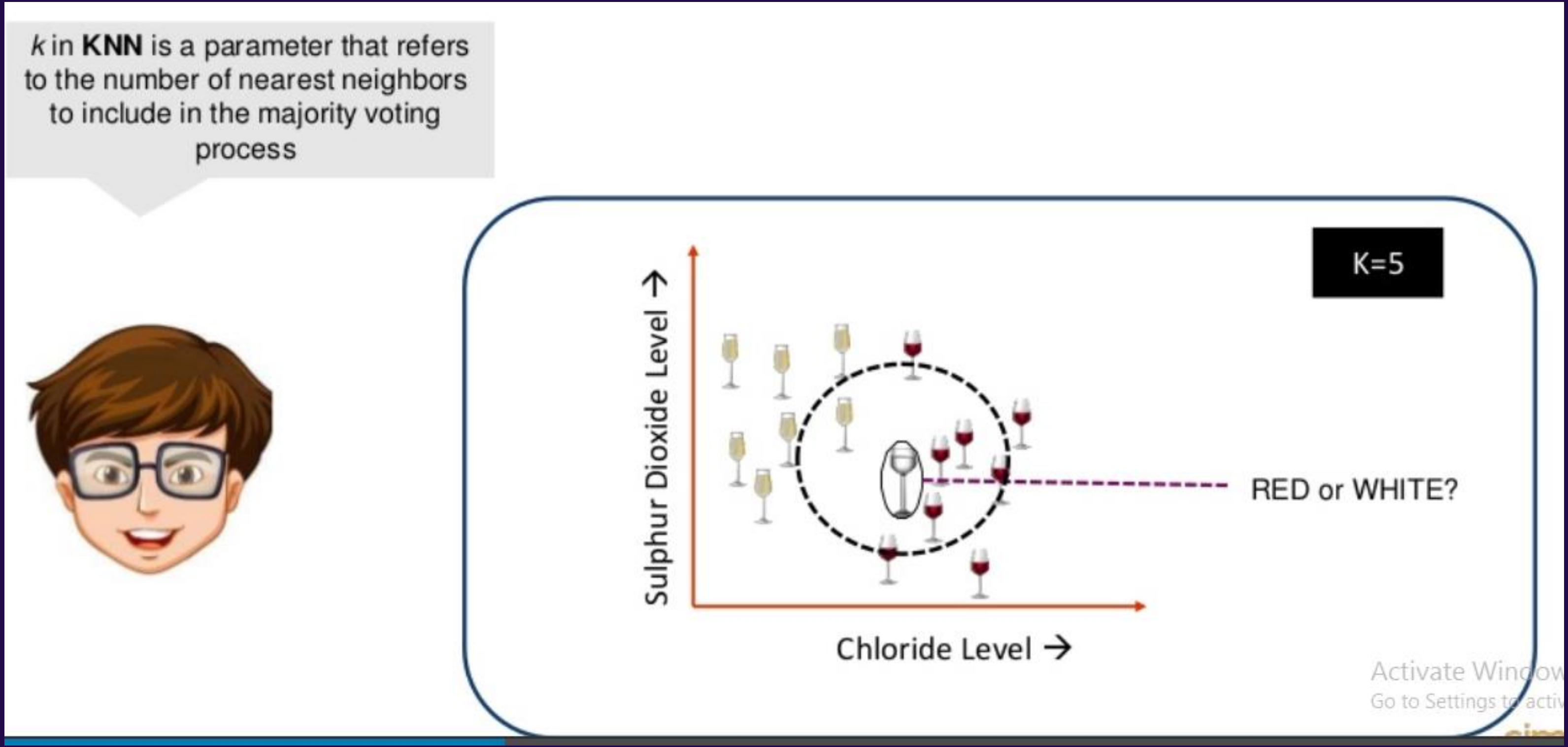
What is KNN Algorithm?



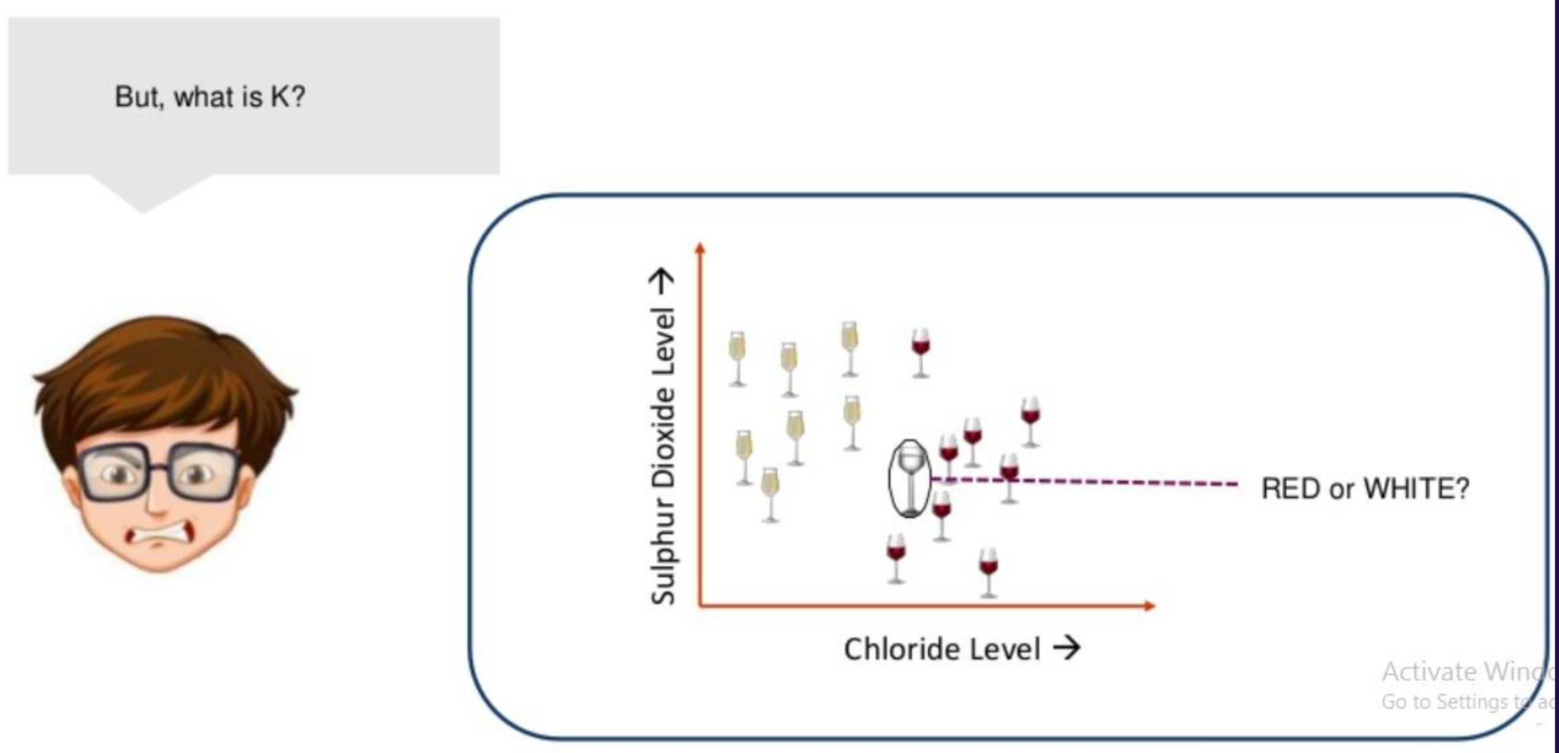
What is KNN Algorithm?



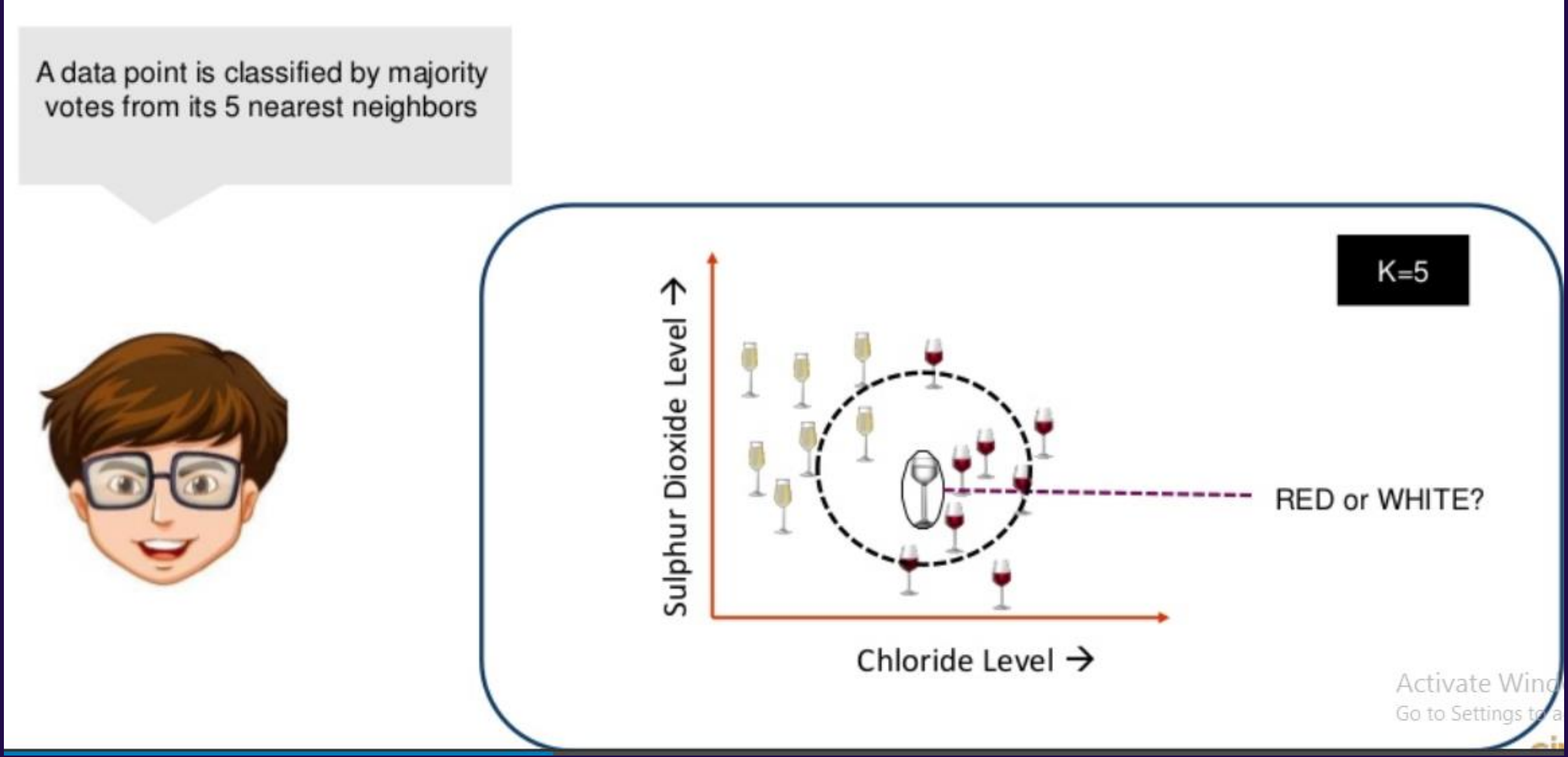
What is KNN Algorithm?



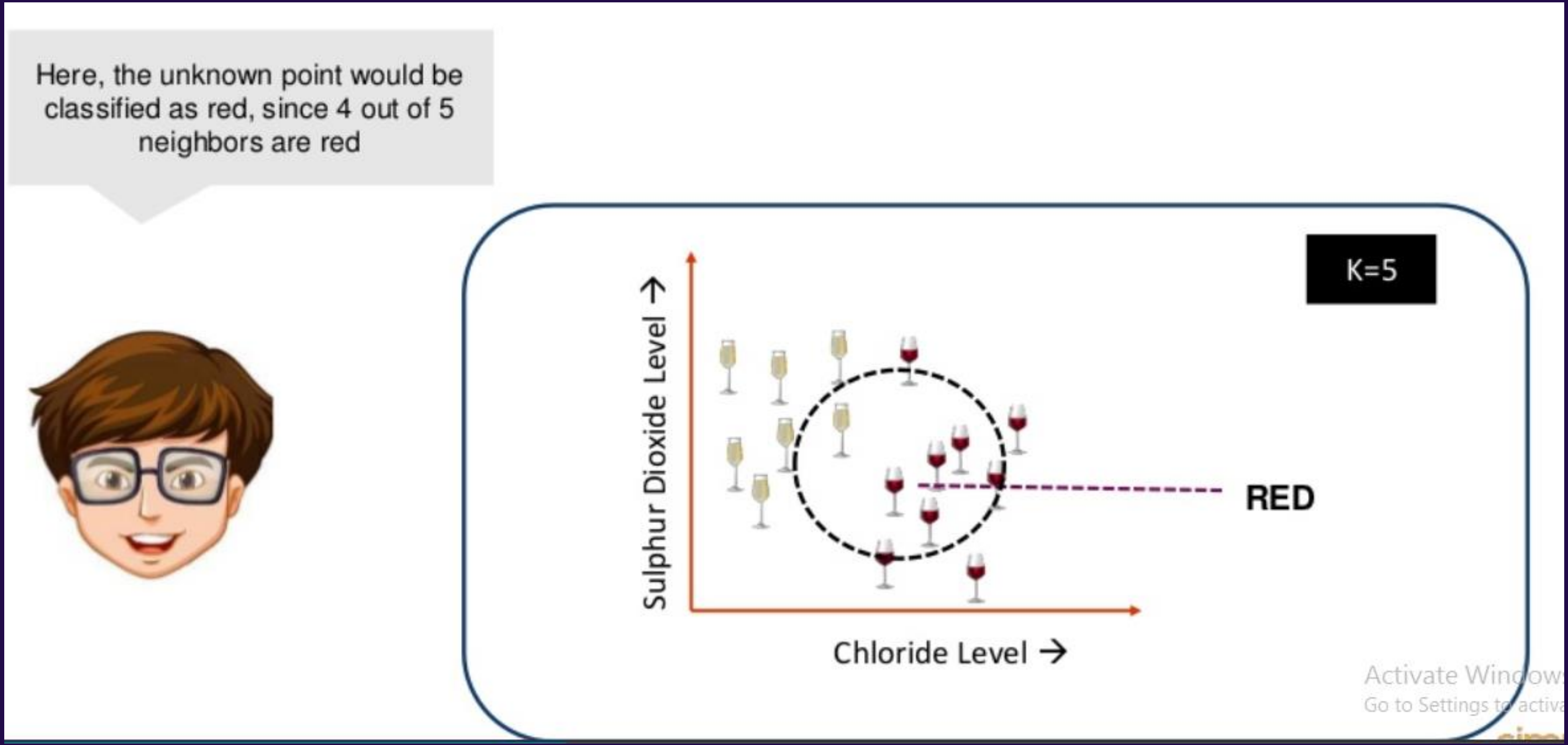
What is KNN Algorithm?



What is KNN Algorithm?



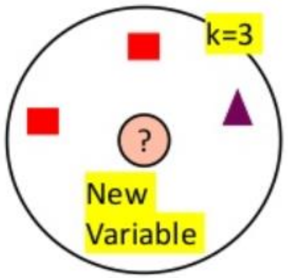
What is KNN Algorithm?



How do we choose 'k'?

How do we choose ’k’?

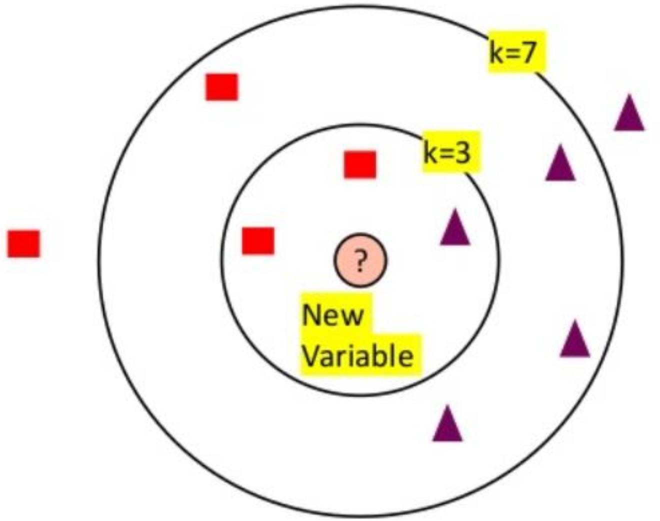
KNN Algorithm is based on **feature similarity**: Choosing the right value of *k* is a process called parameter tuning, and is important for better accuracy



So at k=3, we can classify ‘?’ as

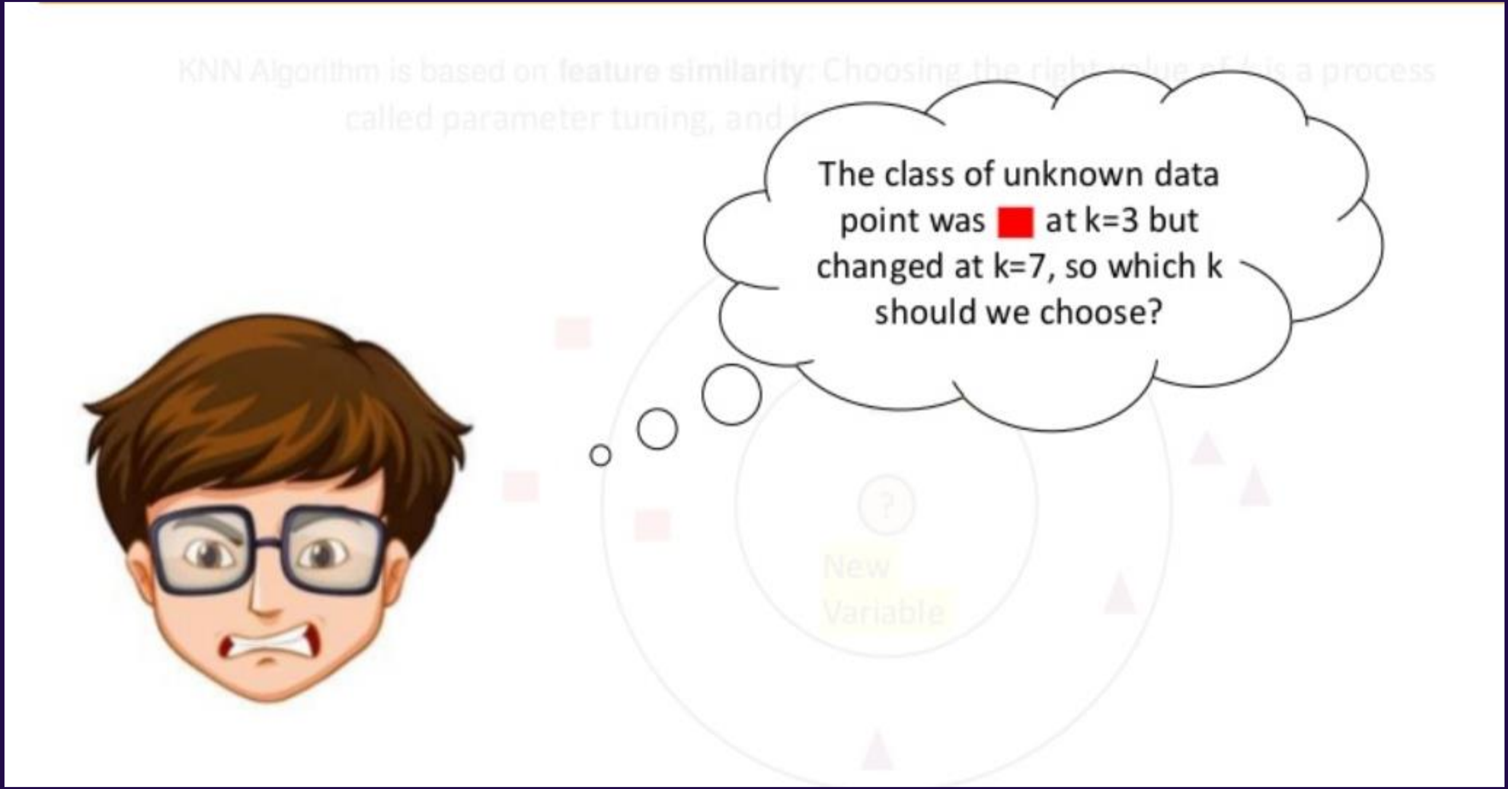
How do we choose 'k'?

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

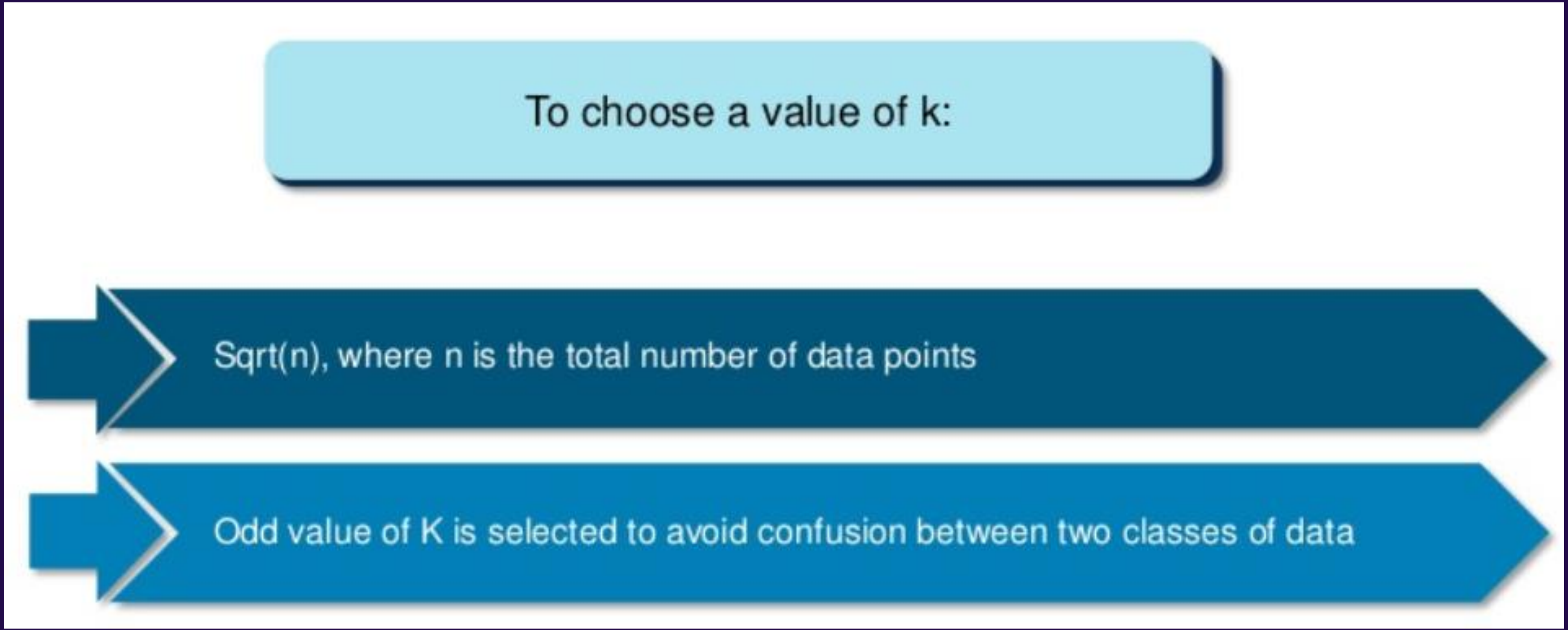


But at k=7, we classify '?' as ▲

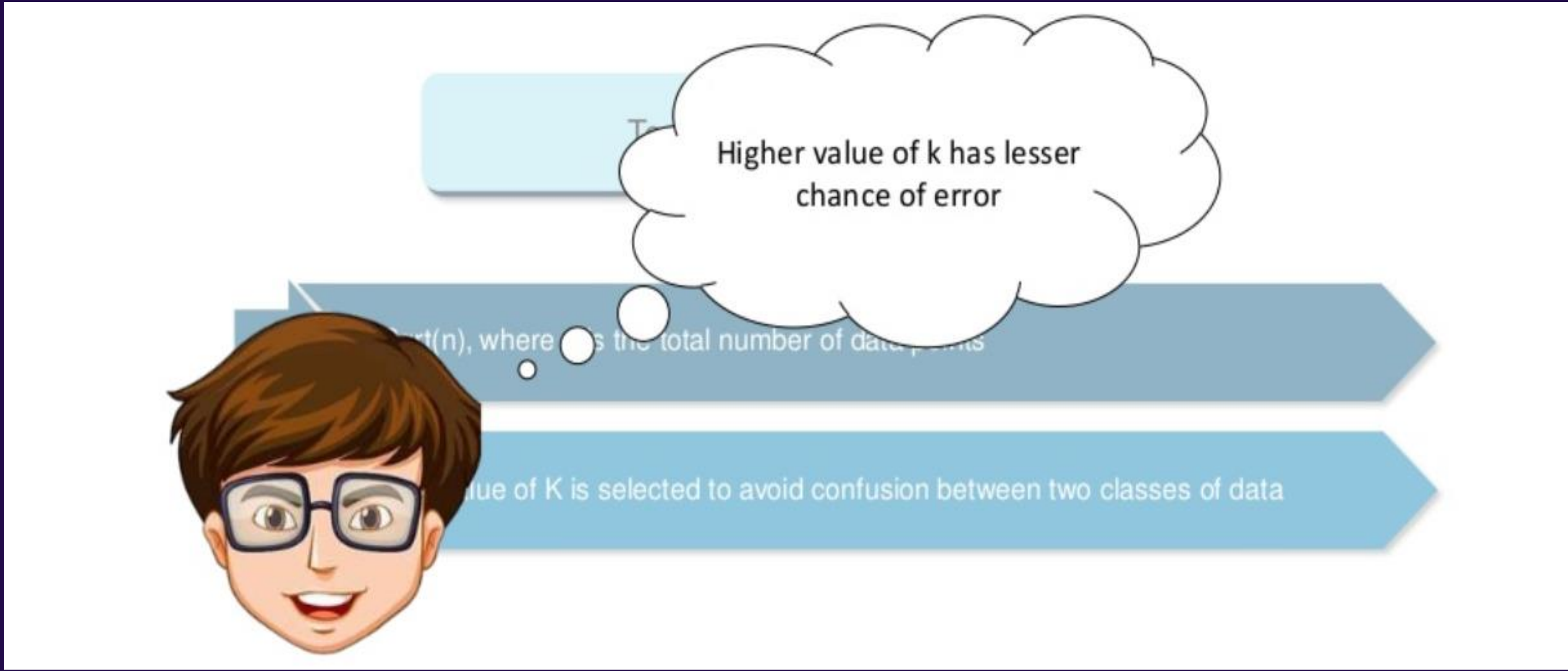
How do we choose ’k’?



How do we choose ’k’?

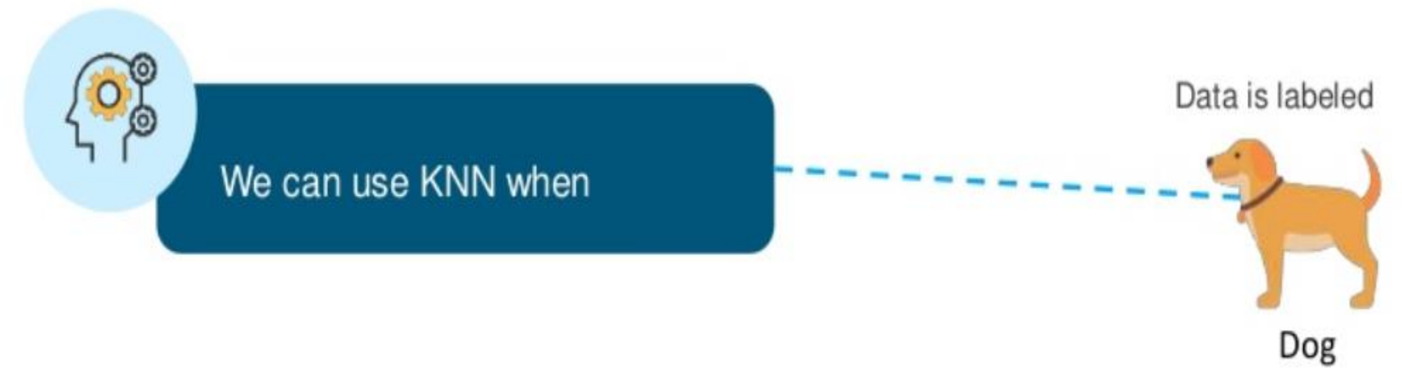


How do we choose 'k'?

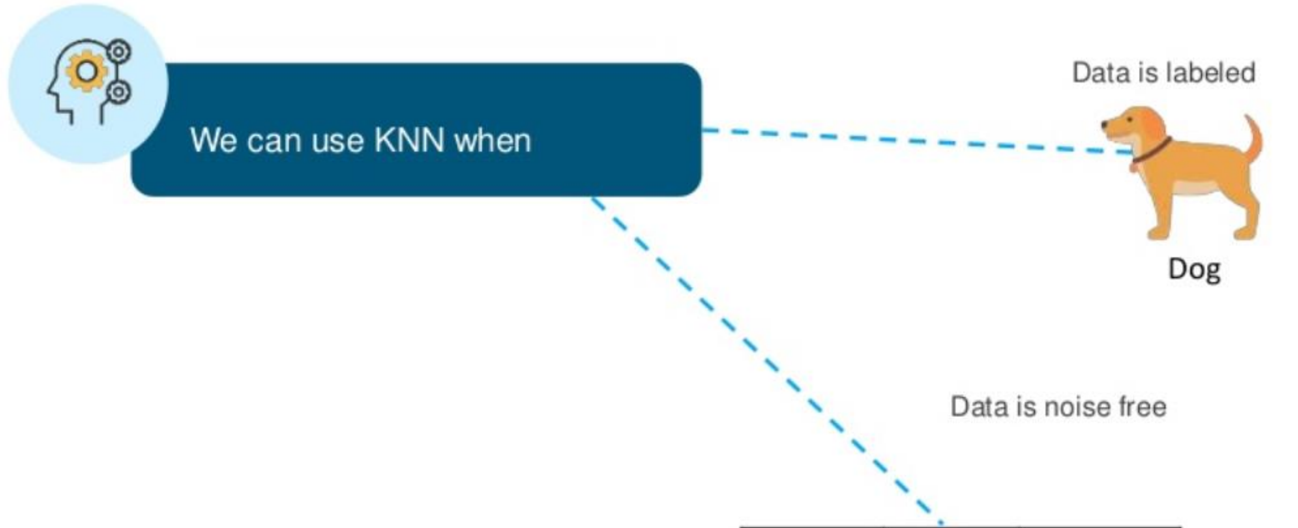


When do we use KNN?

When do we use KNN?



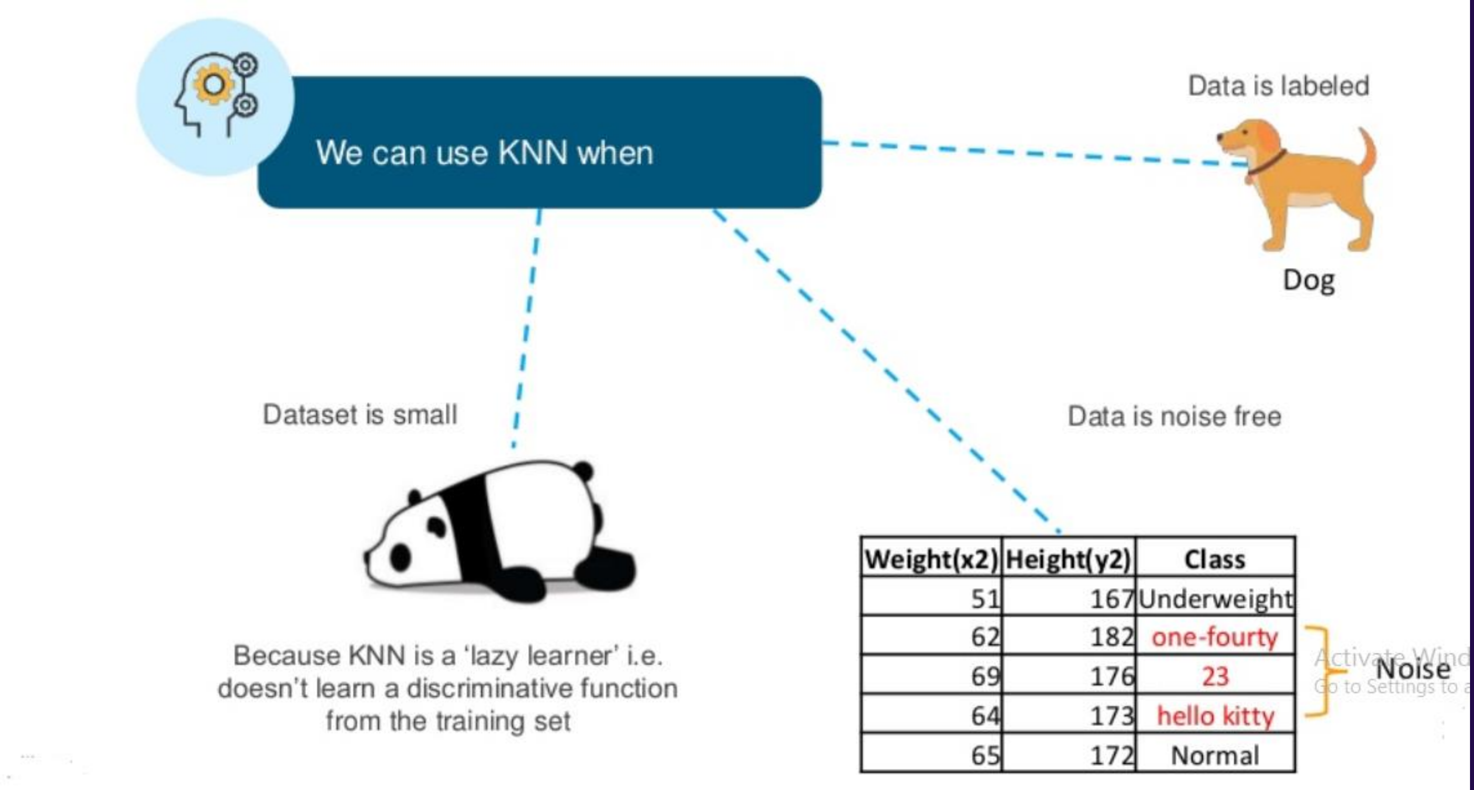
When do we use KNN?



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?



How does KNN Algorithm work?

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**	**?**

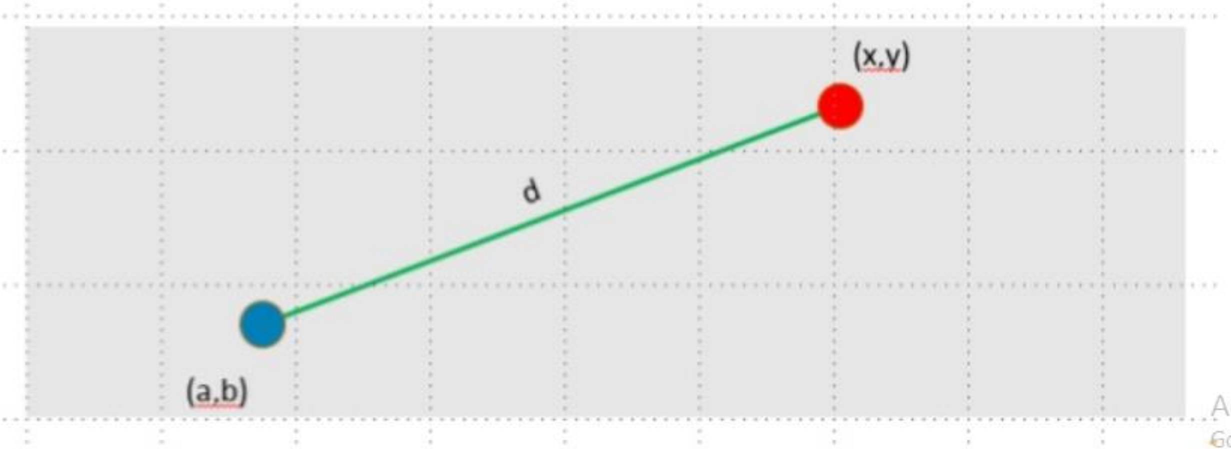
How does KNN Algorithm work?

To find the nearest neighbors, we will calculate Euclidean distance

How does KNN Algorithm work?

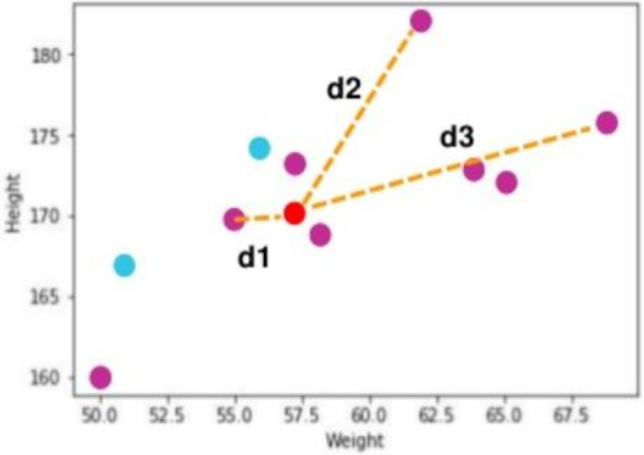
According to the **Euclidean distance** formula, the **distance** between two points in the plane with coordinates (x, y) and (a, b) is given by:

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



How does KNN Algorithm work?

Let's calculate it to understand clearly:



● Unknown data point

dist(**d1**)= √(170-167)² + (57-51)² ~= 6.7

dist(**d2**)= √(170-182)² + (57-62)² ~= 13

dist(**d3**)= √(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

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?

How does KNN Algorithm work?

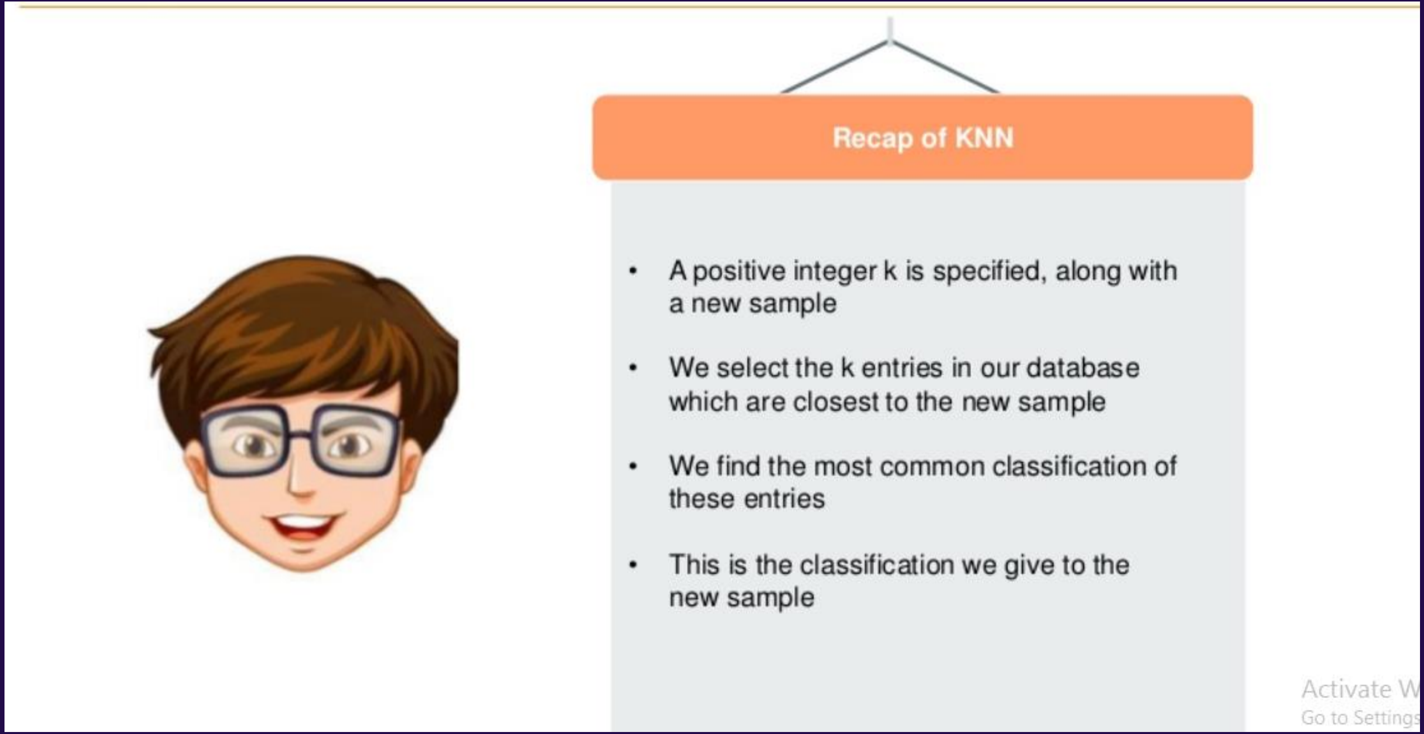
Class	Euclidean Distance
Underweight	6.7
Normal	13
Normal	13.4
Normal	7.6
Normal	8.2
Underweight	4.1
Normal	1.4
Normal	3
Normal	2

k = 3

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

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

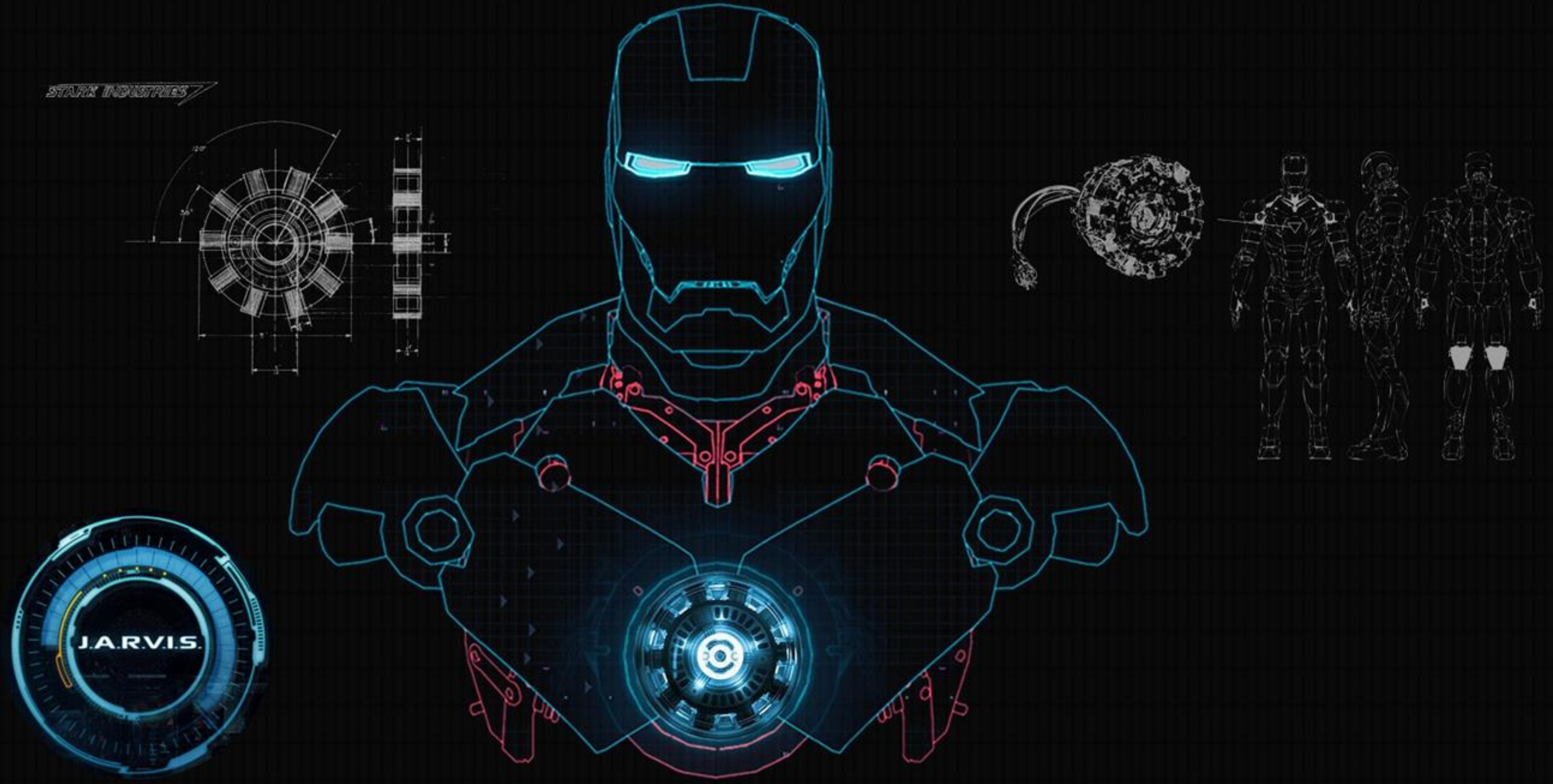
Recap of KNN


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
Activate W
Go to Settings

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



Thank You