What is Regression ?

- Regression analysis is a statistical method that helps us to analyze and understand the relationship between two or more variables of interest.
- The process that is adapted to perform regression analysis helps to understand which factors are important, which factors can be ignored and how they are influencing each other.

Dependent Vs. Independent

- For the regression analysis is be a successful method, we understand the following terms:
- **Dependent Variable:** This is the variable that we are trying to understand or forecast.
- **Independent Variable:** These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable.

Regression Definition – **Why is it called regression?**

- In regression, we normally have one dependent variable and one or more independent variables. Here we try to “regress” the value of dependent variable “Y” with the help of the independent variables. In other words, we are trying to understand, how does the value of ‘Y’ change w.r.t change in ‘X’.

$$Y = f(x)$$

Dependent Variable (GRE Score)

Independent Variable (CGPA)

This is where Regression comes in!

- If we are supposed to find the relationship between two variables, we can apply regression analysis.

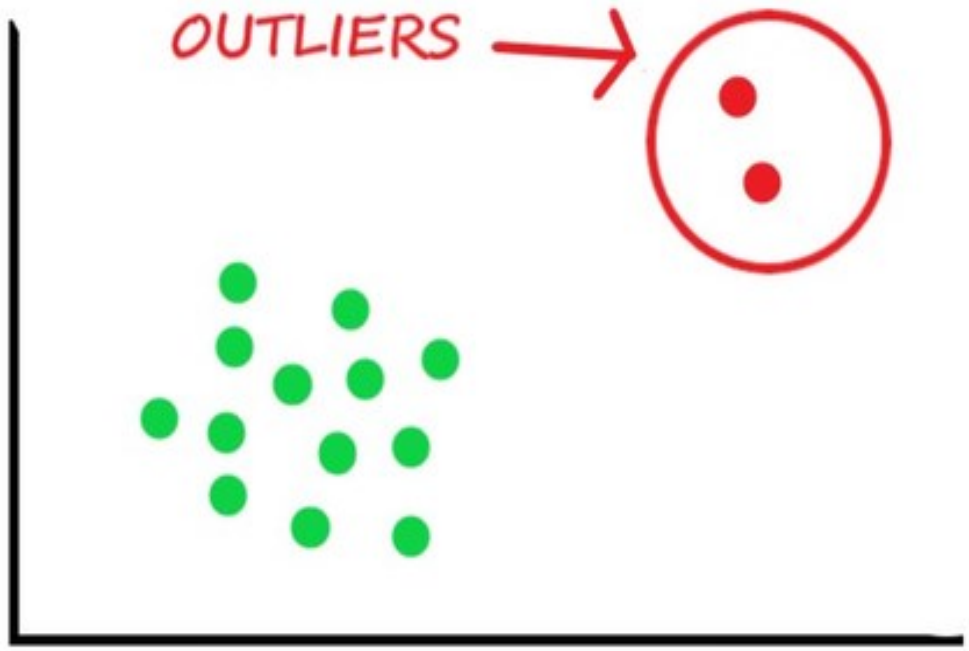
Terminologies used in Regression Analysis

1. Outliers

- Suppose there is an observation in the dataset that has a very high or very low value as compared to the other observations in the data, i.e. it does not belong to the population, such an observation is called an outlier. In simple words, it is an extreme value. An outlier is a problem because many times it hampers the results we get.

Outlier Example 1

Outlier Example 2



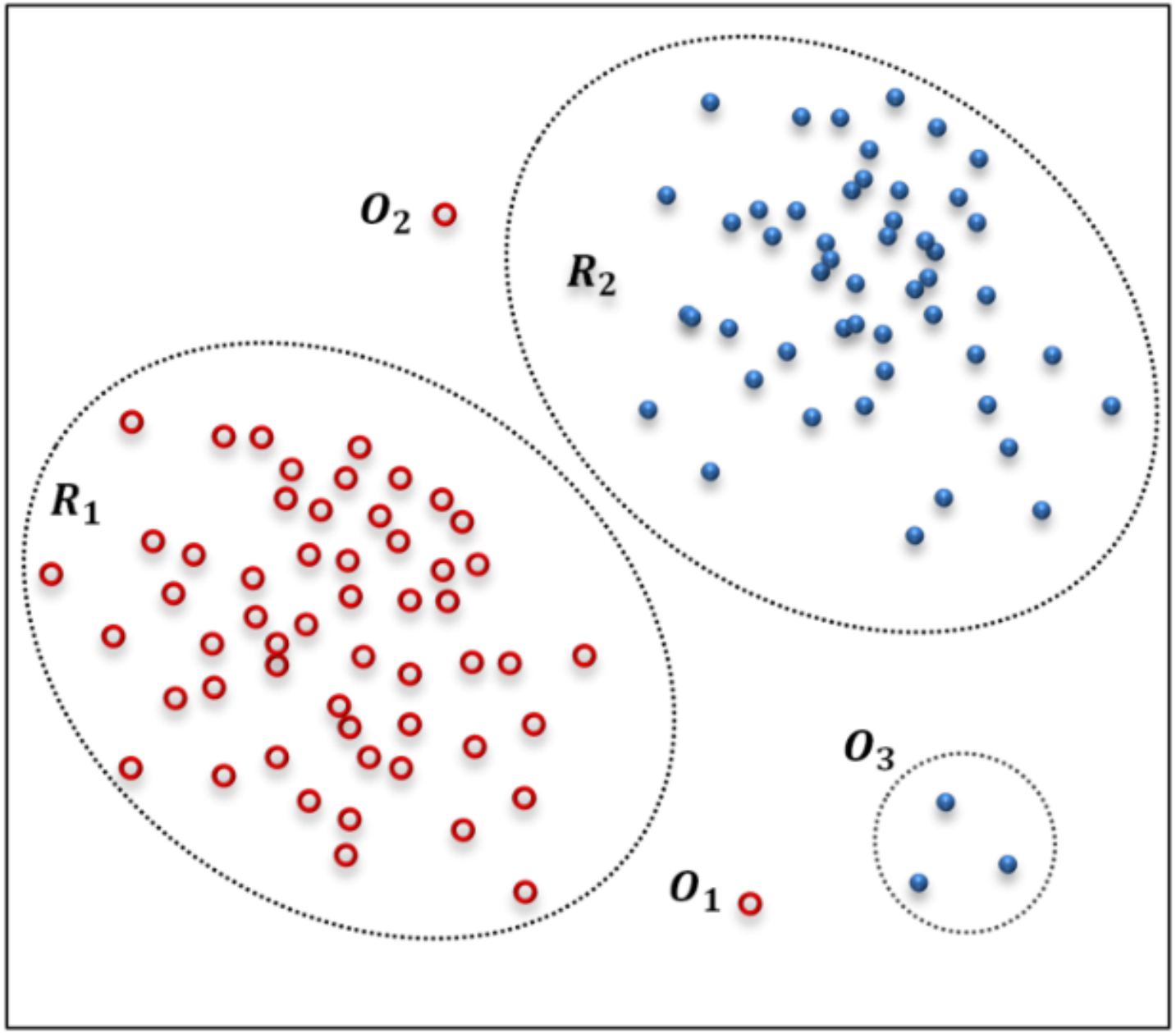


Figure. 1 A simple example of outliers in a 2-D dataset.

Terminologies used in Regression Analysis

2. Multicollinearity

- When the independent variables are highly correlated to each other, then the variables are said to be multicollinear. Many types of regression techniques assume multicollinearity should not be present in the dataset. It is because it causes problems in ranking variables based on its importance, or it makes the job difficult in selecting the most important independent variable

Correlation coefficient

- Correlation shows the strength of a relationship between two variables and is expressed numerically by the correlation coefficient.
- The degree of association is measured by a correlation coefficient, denoted by r. It is sometimes called Pearson's correlation coefficient.
- The correlation coefficient is measured on a scale that varies from + 1 through 0 to – 1.
- Complete correlation between two variables is expressed by either + 1 or -1.
- When one variable increases as the other increases the correlation is positive; when one decreases as the other increases it is negative.
- Complete absence of correlation is represented by 0.

The Formula for Correlation Is

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

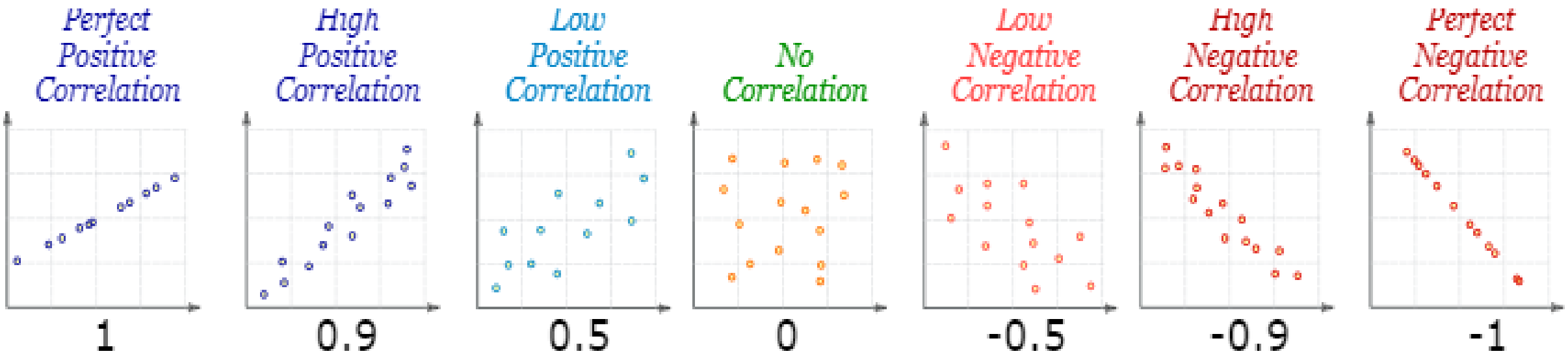
where:

r = the correlation coefficient

$\overline{X}$ = the average of observations of variable X

$\overline{Y}$ = the average of observations of variable Y

A correlation is assumed to be **linear** (following a line).



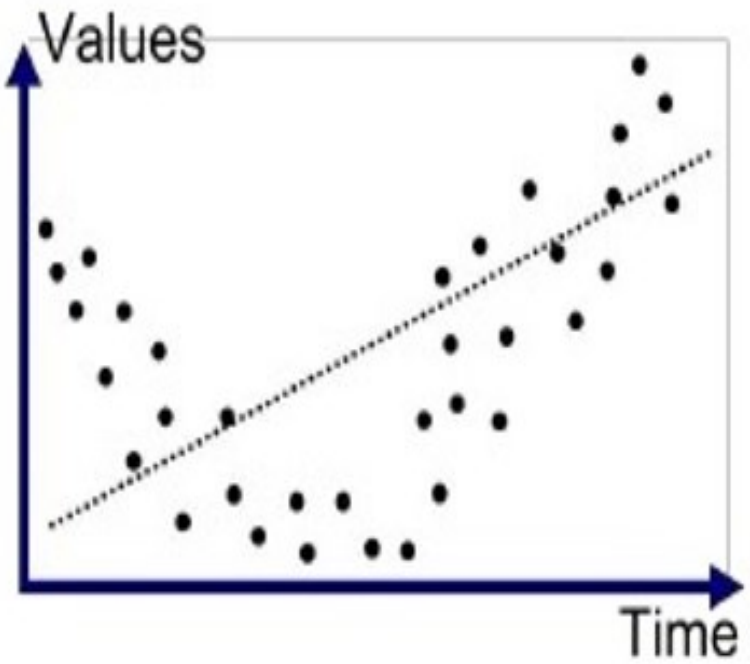
Correlation can have a value:

- 1 is a perfect positive correlation
- 0 is no correlation (the values don't seem linked at all)
- -1 is a perfect negative correlation

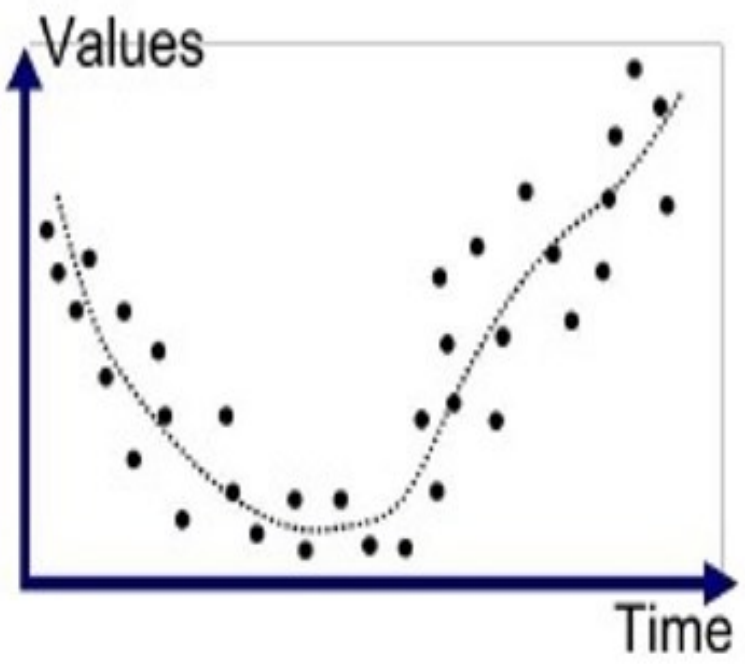
3. Underfit and Overfit

- When we use unnecessary explanatory variables, it might lead to overfitting. Overfitting means that our algorithm works well on the training set but is unable to perform better on the test sets. It is also known as a problem of **high variance**.

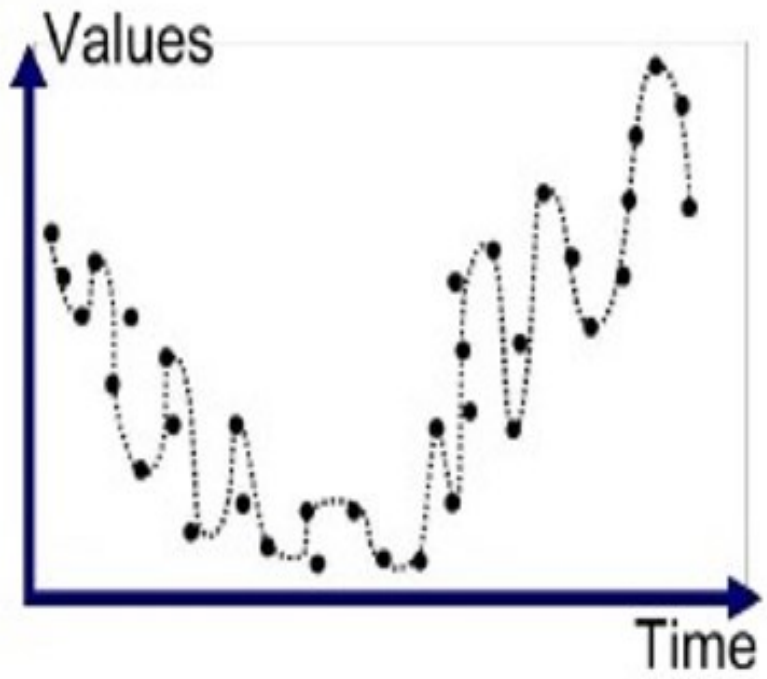
- When our algorithm works so poorly that it is unable to fit even a training set well, then it is said to underfit the data. It is also known as a problem of **high bias**.



Underfitted



Good Fit/Robust



Overfitted

What is Regression?

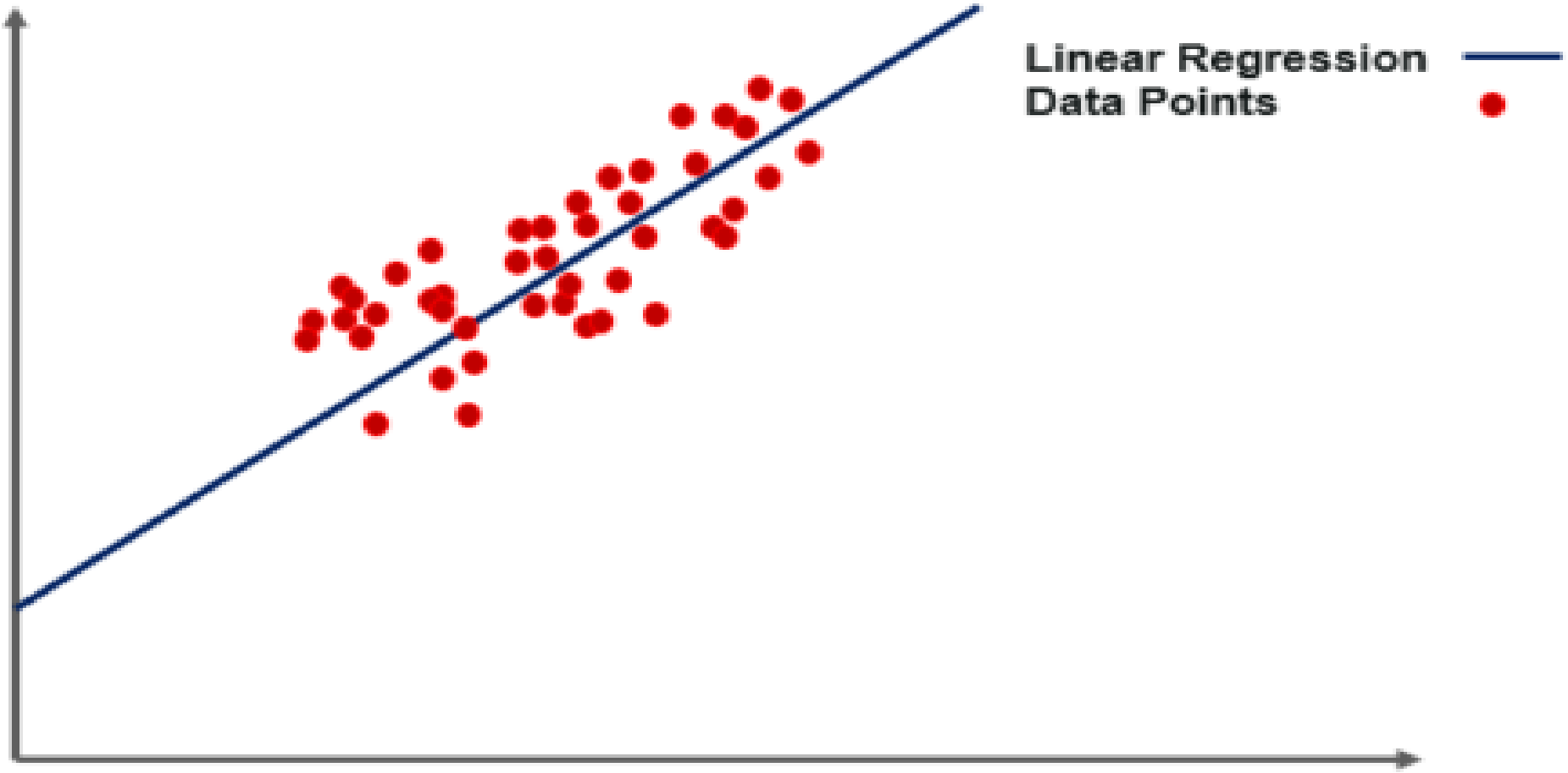
Price	Floor space	Rooms	Lot size	Appartment	Row house	Corner house	Detached
250000	71	4	92	0	1	0	0
209500	98	5	123	0	1	0	0
349500	128	6	114	0	1	0	0
250000	86	4	98	0	1	0	0
419000	173	6	99	0	1	0	0
225000	83	4	67	0	1	0	0
549500	165	6	110	0	1	0	0
240000	71	4	78	0	1	0	0
340000	116	6	115	0	1	0	0

Types Of Regression

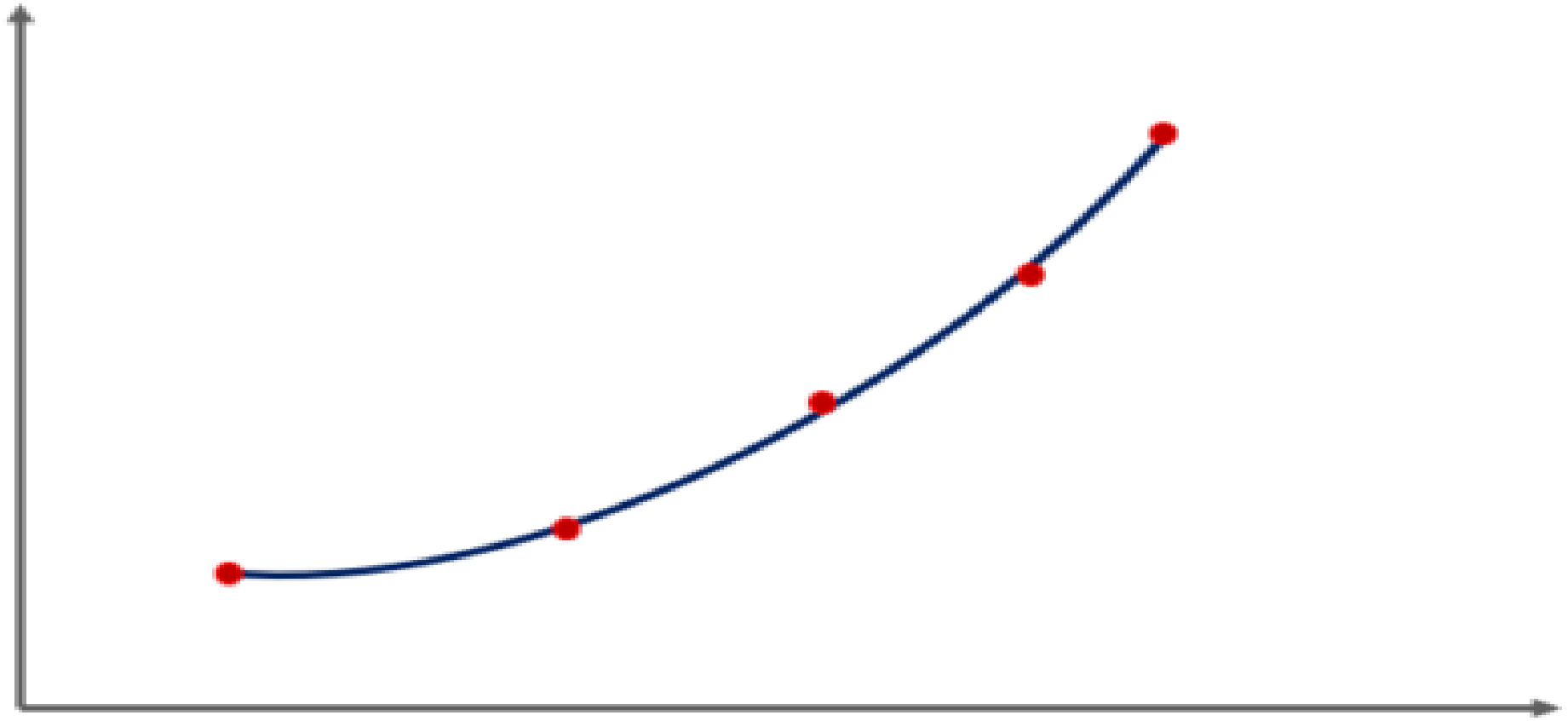
The following are types of regression.

- **Simple Linear Regression**
- **Polynomial Regression**
- **Support Vector Regression**
- **Decision Tree Regression**
- **Random Forest Regression**

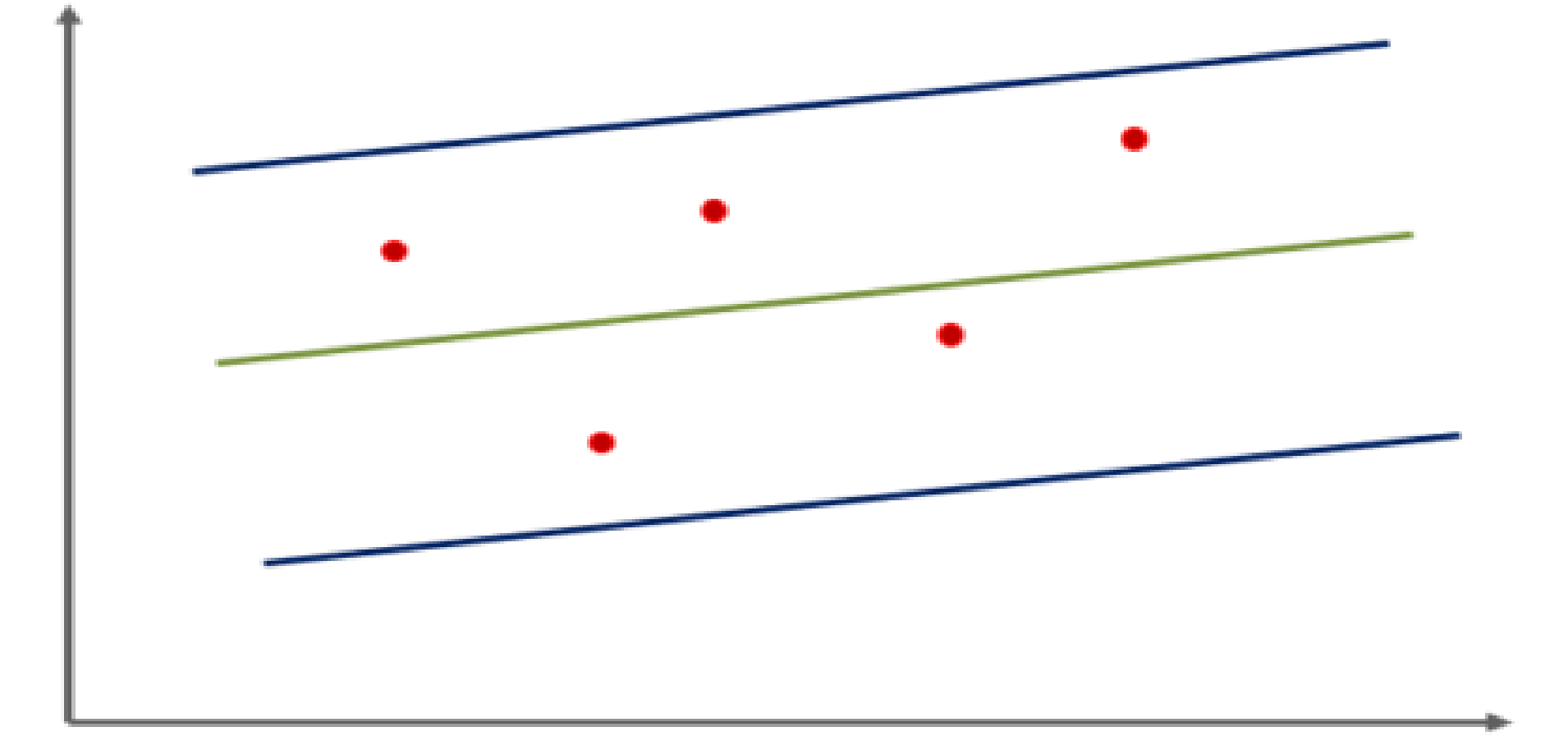
Simple Linear Regression



Polynomial Regression



Support Vector Regression

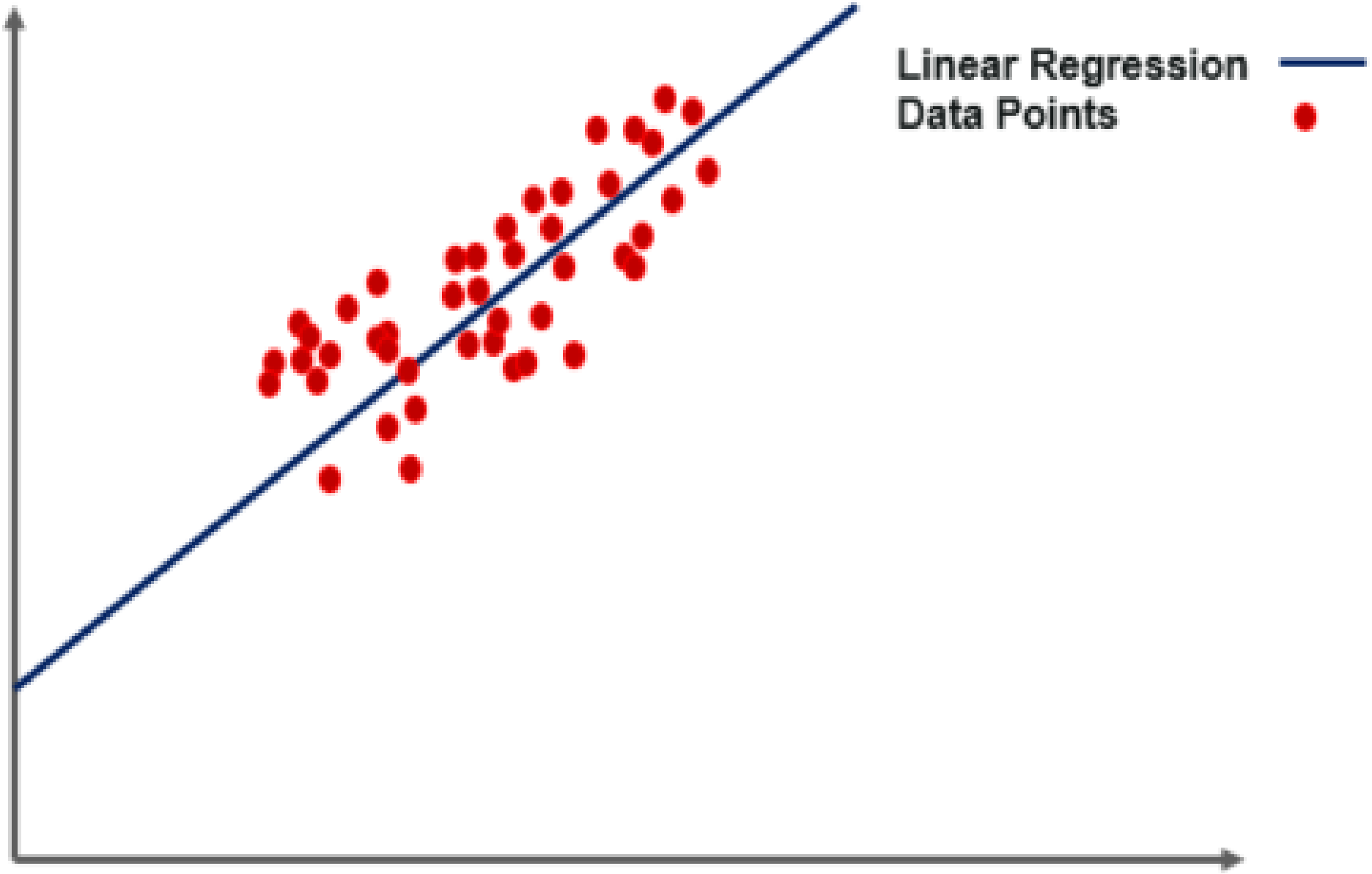


Decision Tree Regression

- A decision tree can be used for both regression and classification.

Random Forest Regression

- In random forest regression, we ensemble the predictions of several decision tree regressions.


Linear Regression
Data Points

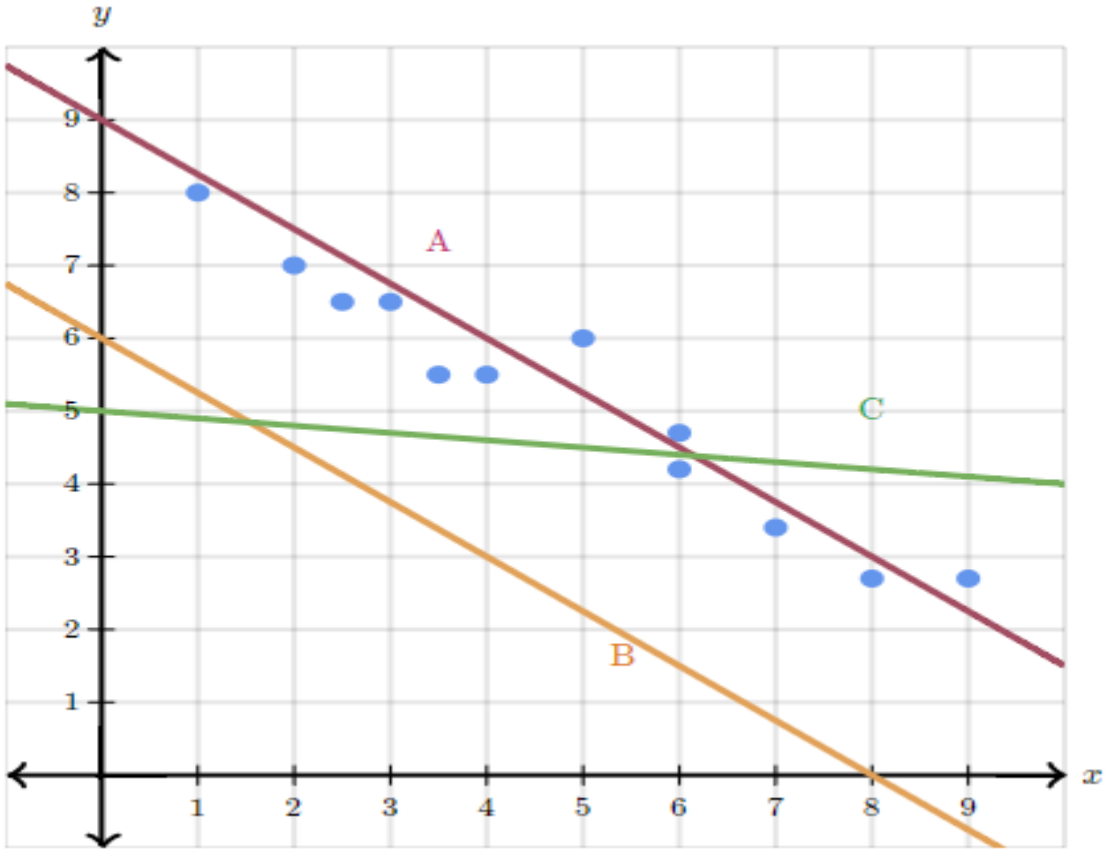
Linear Regression

- When we see a relationship in a scatterplot, we can use a line to summarize the relationship in the data.
- We can also use that line to make predictions in the data. This process is called **linear regression**

Fitting a line to data

- There are more advanced ways to fit a line to data, but in general, we want the line to go through the "middle" of the points.

Which line fits the data graphed below?



Choose 1 answer:

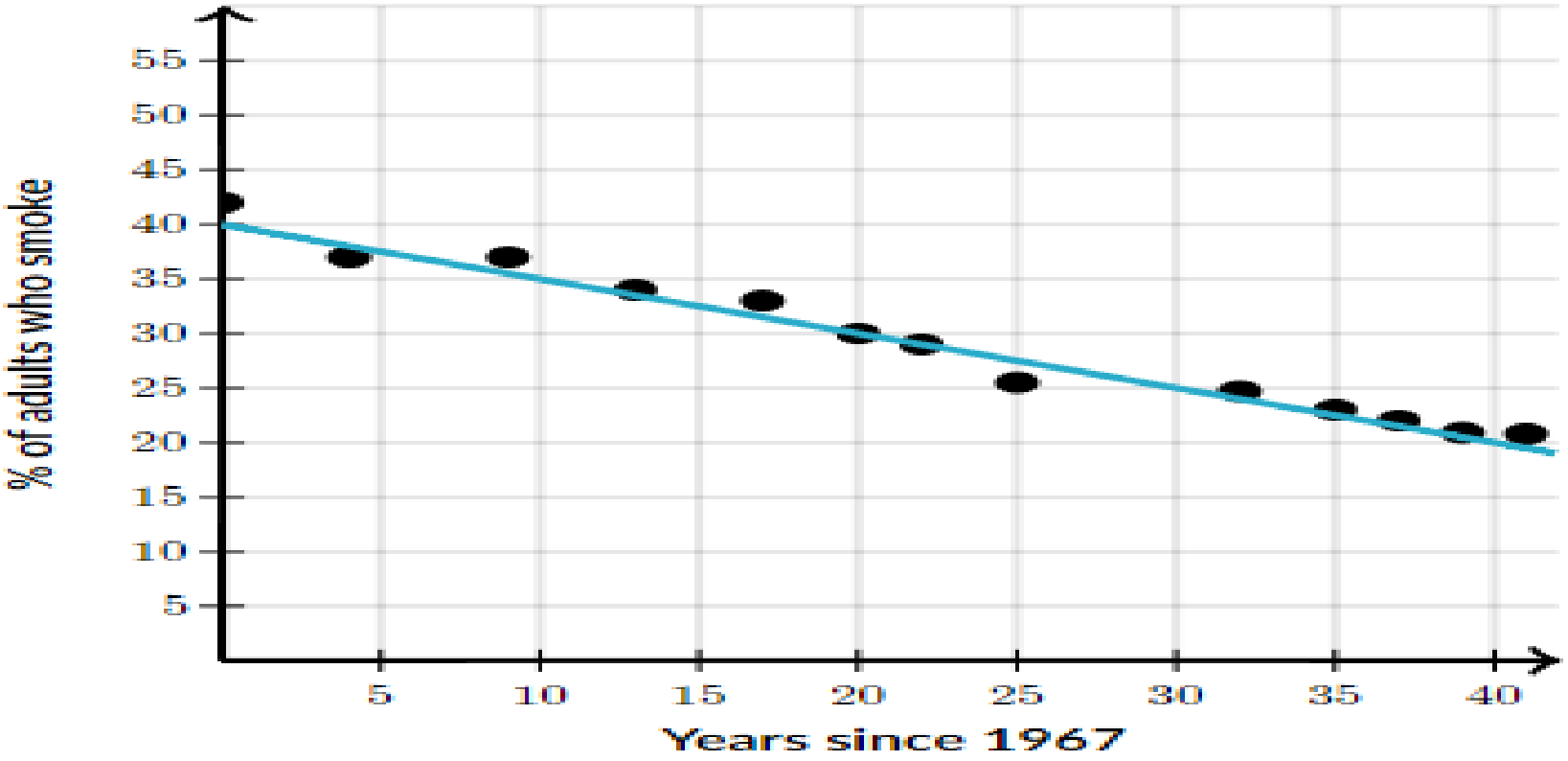
(A) A

(B) B

(C) C

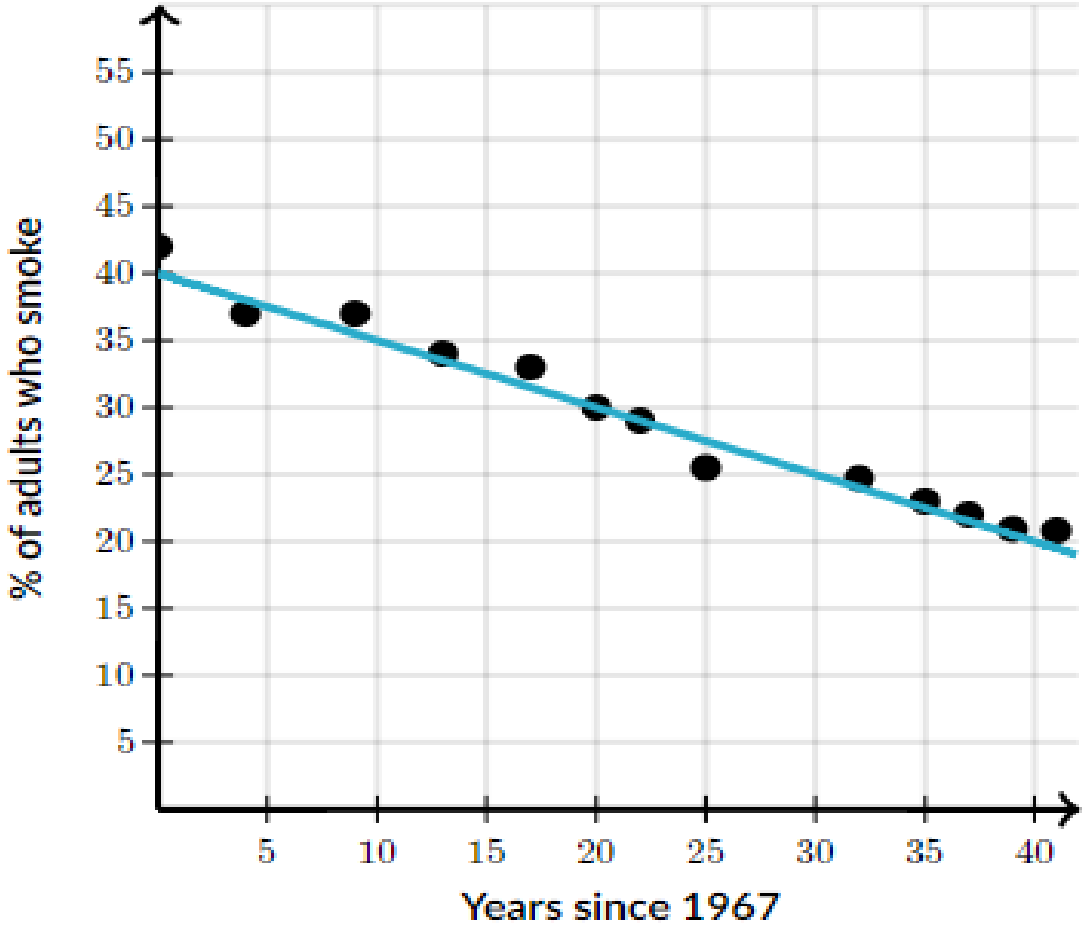
(D) None of the lines fit the data.

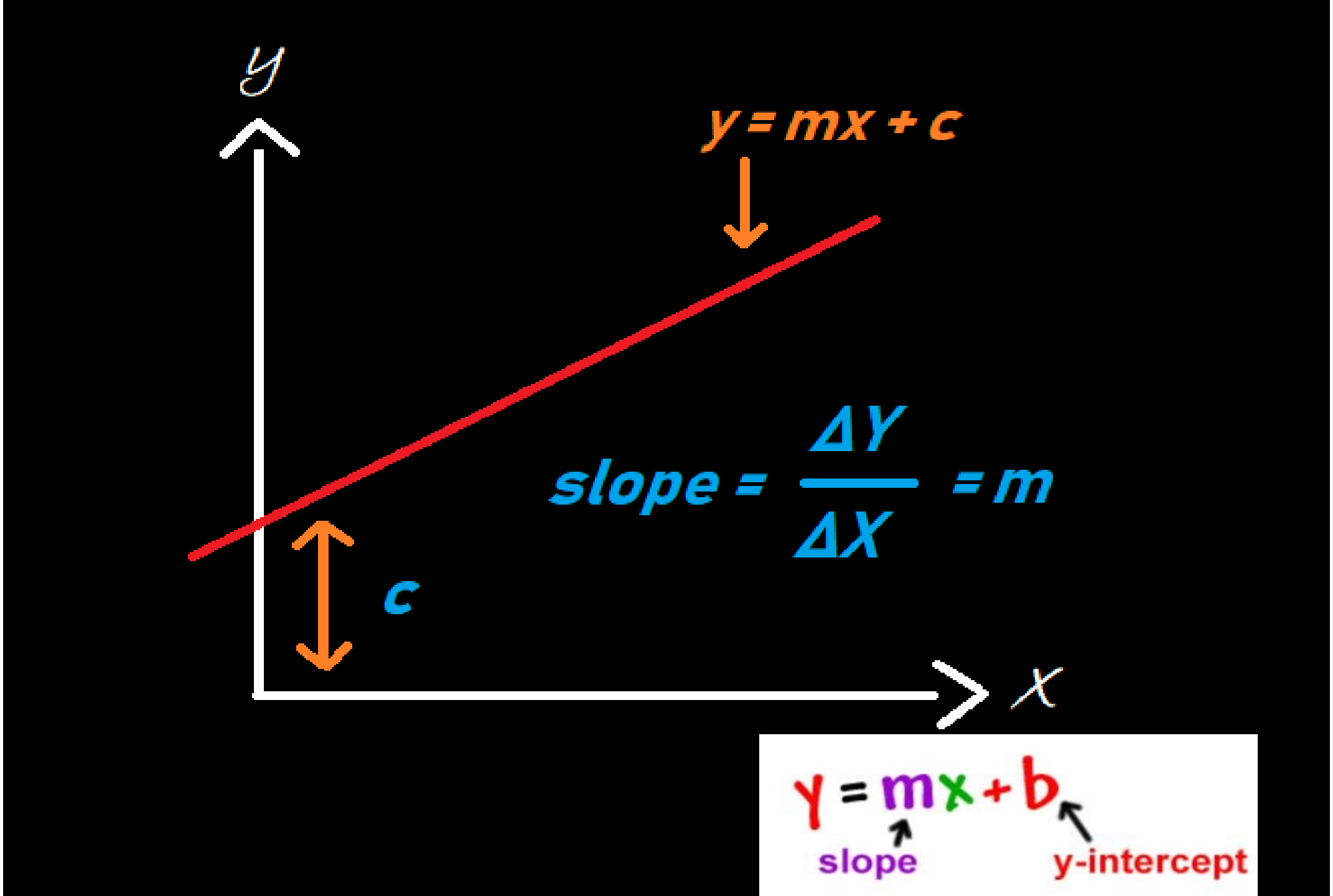
Example: Finding the equation



Example: Finding the equation

The percent of adults who smoke, recorded every few years since 1967, suggests a negative linear association with no outliers. A line was fit to the data to model the relationship.




y
y = mx + c
slope = ΔY / ΔX = m
c
x
y = mx + b
slope
y-intercept

Write a linear equation to describe the given model

Write a linear equation to describe the given model.

Step 1: Find the slope.

This line goes through $(0, 40)$ and $(10, 35)$, so the slope is $\frac{35-40}{10-0} = -\frac{1}{2}$.

Step 2: Find the y-intercept.

We can see that the line passes through $(0, 40)$, so the y-intercept is 40.

Step 3: Write the equation in $y = mx + b$ form.

The equation is $y = -0.5x + 40$

To estimate what percent of adults smoked in 1997, we can plug in 30 for x (since x represents years since 1967):

$$y = -0.5x + 40$$

$$y = (-0.5)(30) + 40$$

$$y = -15 + 40$$

$$y = 25$$

Based on the equation, about 25% of adults smoked in 1997.

Basic Terminology in Classification Algorithms

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. **Eg: Gender classification (Male / Female)**
- **Multi-class classification:** Classification with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. **Eg: An animal can be a cat or dog but not both at the same time.**
- **Multi-label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). **Eg: A news article can be about sports, a person, and location at the same time.**

Applications of Classification Algorithms

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drugs classification
- Facial key points detection

Naïve Bayes Classification

- Based on Bayes theorem
- Assumption
  - Presence of one evidence is independent of other other evidence /feature (naïve)
  - equal contribution to the outcome.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

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

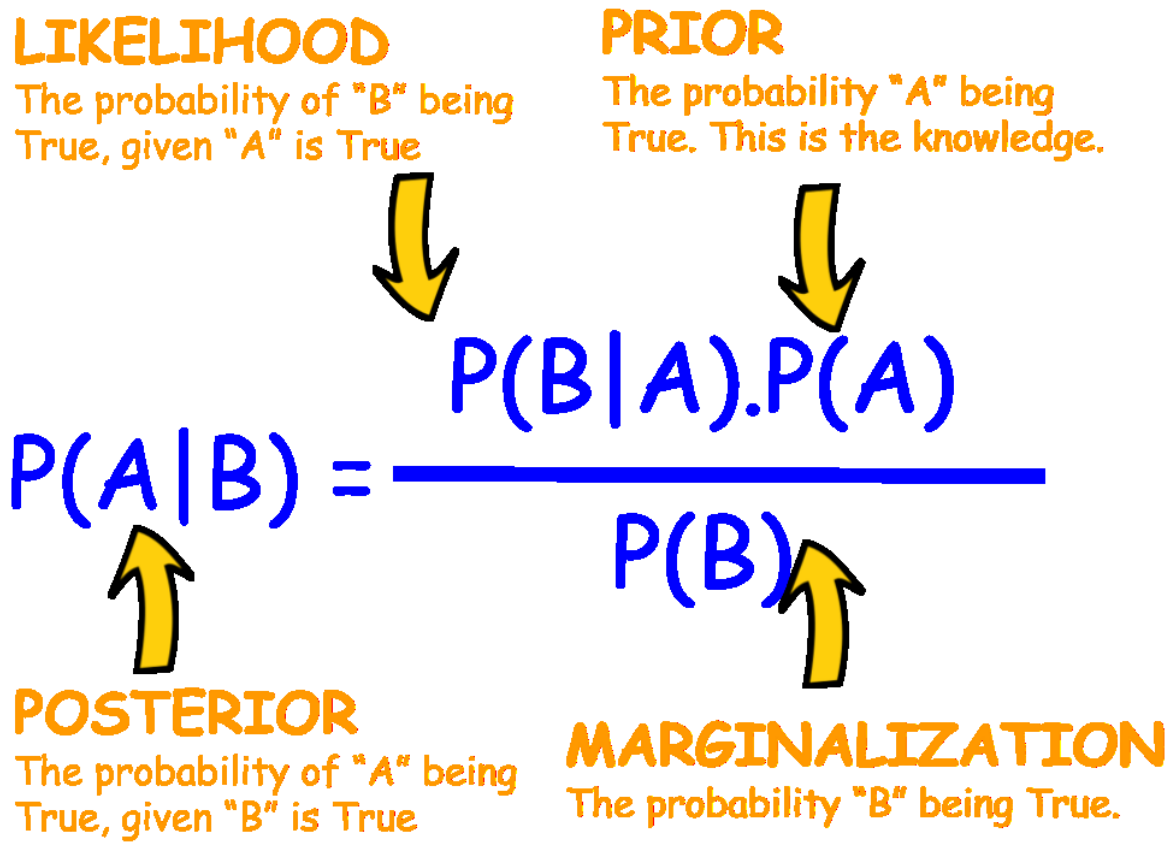
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

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

Working of Naïve Bayes' Classifier:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.


LIKELIHOOD
The probability of "B" being True, given "A" is True
PRIOR
The probability "A" being True. This is the knowledge.
P(A|B) = P(B|A).P(A) / P(B)
POSTERIOR
The probability of "A" being True, given "B" is True
MARGINALIZATION
The probability "B" being True.

- **Problem**: If the weather is sunny, then the Player should play or not?
- **Solution**: To solve this, first consider the below dataset:

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

Outlook

	Yes	No	P(?/yes)	P(?/No)	P()
Sunny	**2**	**3**	**P(Sunny/yes)=2/9**	**P(Sunny/No)=3/5**	**P(Sunny)=5/14**
Overcast	**4**	**0**	**P(Overcast/Yes)=4/9**	**P(Overcast/No)=0**	**P(Overcast)=4/14**
Rainy	**3**	**2**	**P(Rainy/yes)=3/9**	**P(Rainy/No)=2/5**	**P(Rainy)=5/14**
	9	**5**			
	P(Yes) =9/14	**P(No) =5/14**			

Similarily for Temperature

	Yes	**No**	**P(?/yes)**	**P(?/no)**
Hot	**2**	**2**	**2/9**	**2/5**
medium	**4**	**2**	**4/9**	**2/5**
Cool	**3**	**1**	**3/9**	**1/5**
	9	**5**		
	9/14	**5/14**		

Humidity

	Yes	No	P(?/yes)	P(?/no)	P(s)
HIgh	3	4	3/9	4/5	7/14
Normal	6	1	6/9	1/5	7/14
	9	5			
	9/14	5/14			

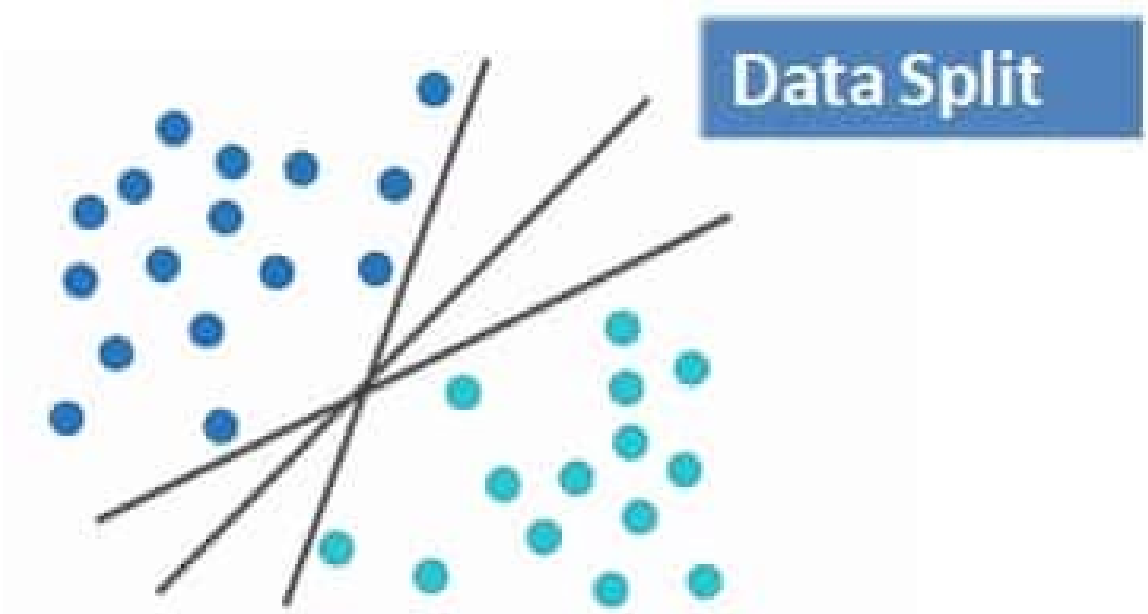
Windy

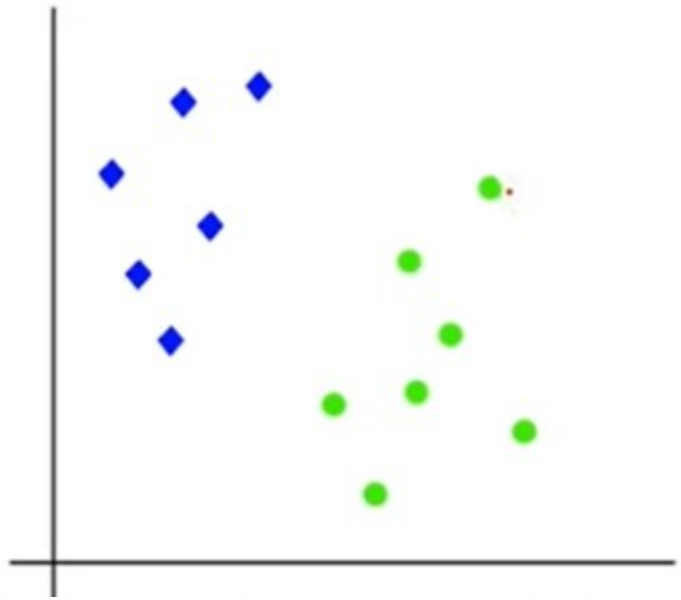
	Yes	No	P(?/yes)	P(?/no)	P()
Weak	6	2	6/9	2/5	8/14
Strong	3	3	3/9	3/5	6/14
	9	5			
	9/14	5/14			

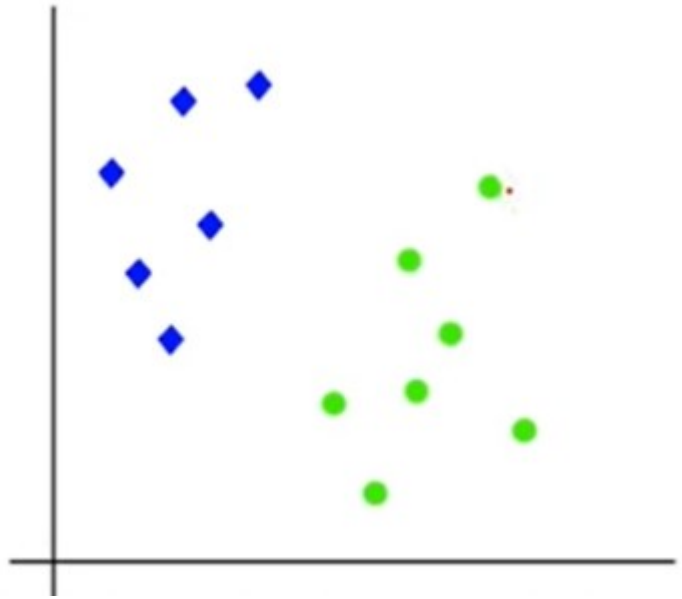
Play	P(yes/no)
Yes	9/14
NO	5/14

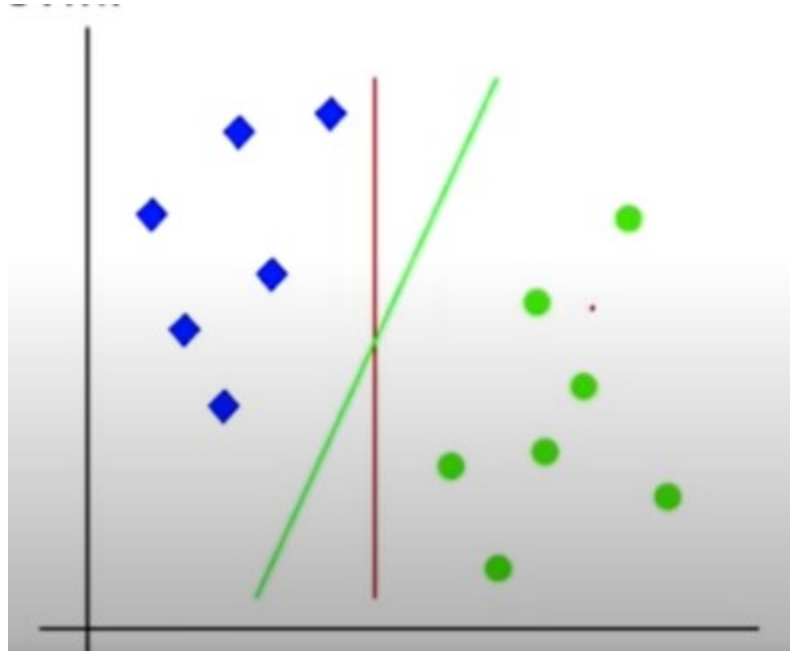
SVM

A **Support Vector Machine (SVM)** is a supervised machine learning algorithm which can be used for both classification and regression problems.



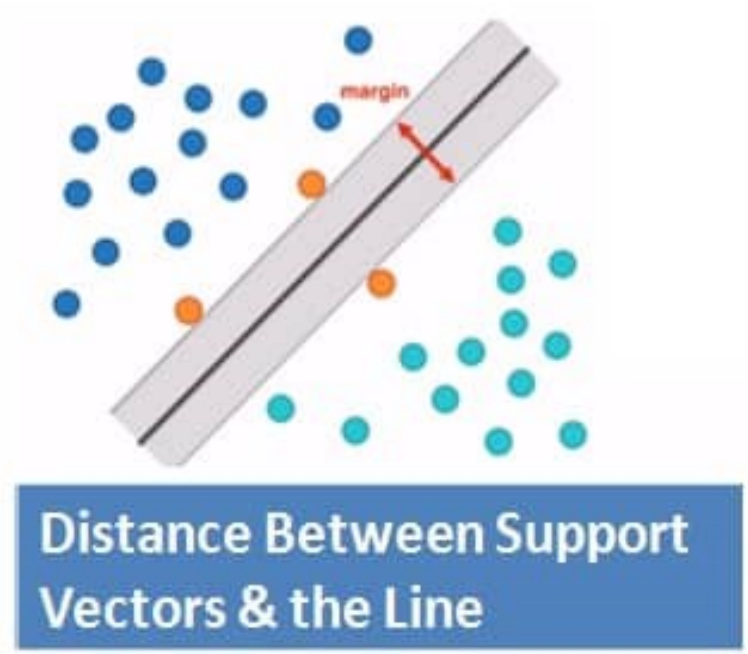


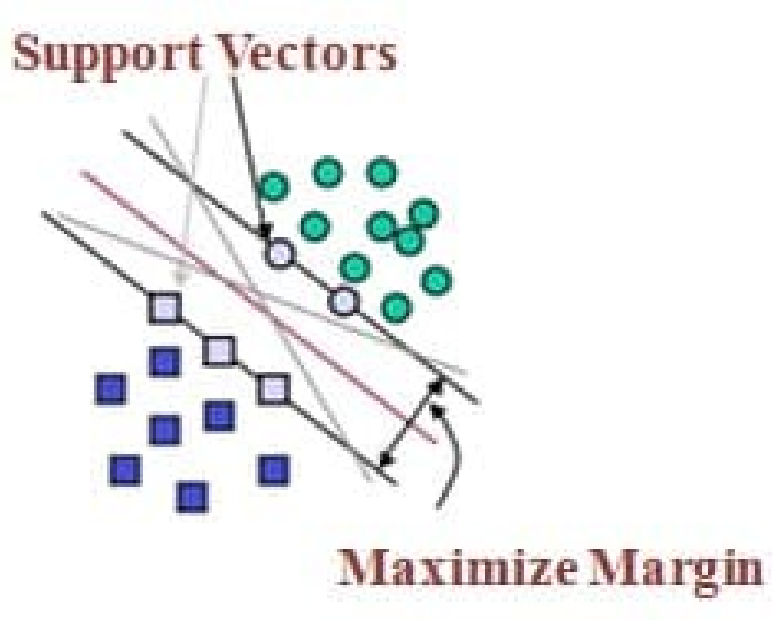




Support Vectors

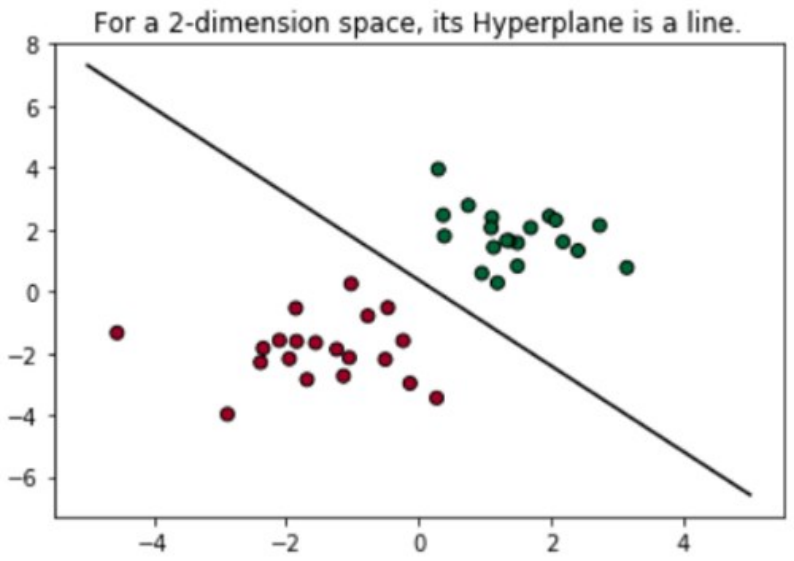
SVMs maximize the margin around the separating hyperplane.
Margins are the (perpendicular) distances between the line and those dots closest to the line.

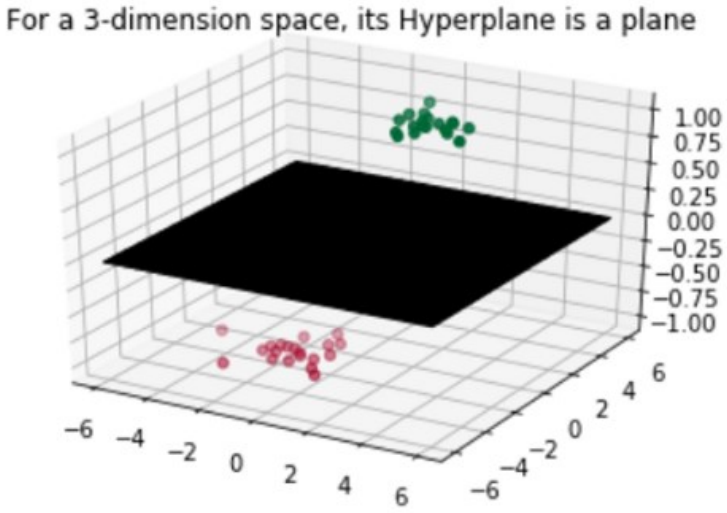




So what is a **Hyperplane**?

- Hyperplane is an (n minus 1)-dimensional subspace for an n-dimensional space
- For a 2-dimension space, its hyperplane will be 1-dimension, which is just a line.
- For a 3-dimension space, its hyperplane will be 2-dimension, which is a plane that slice the cube.



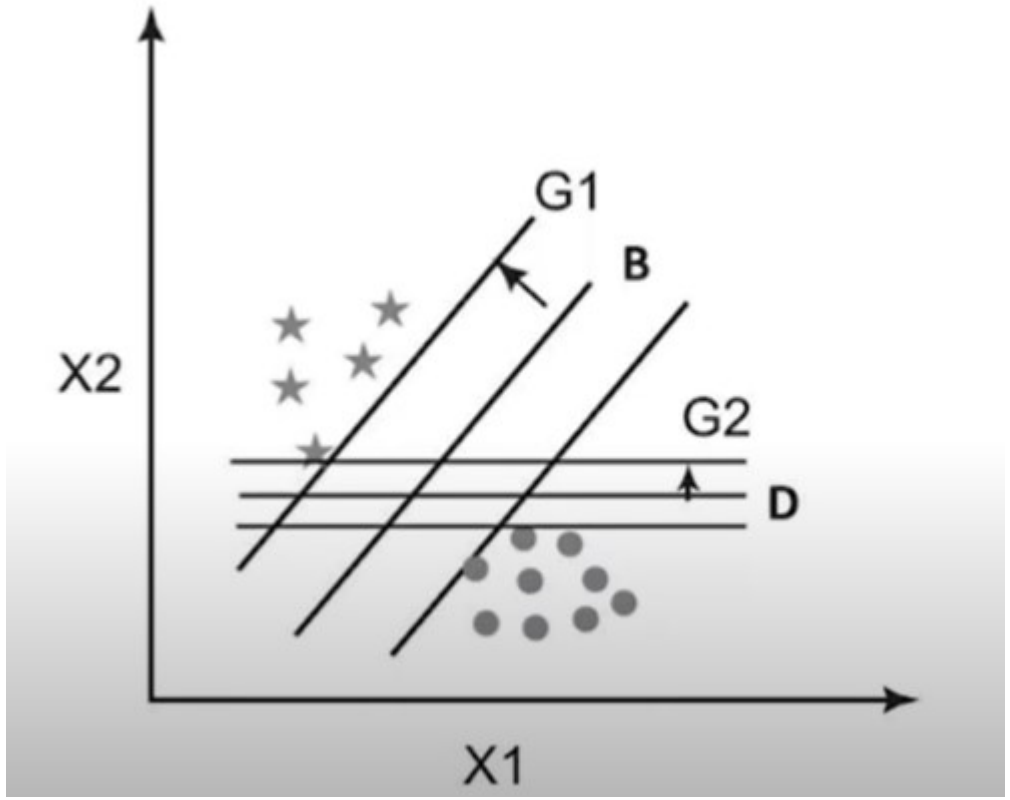


- Any Hyperplane can be written mathematically as below:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \beta_n * x_n = 0$$

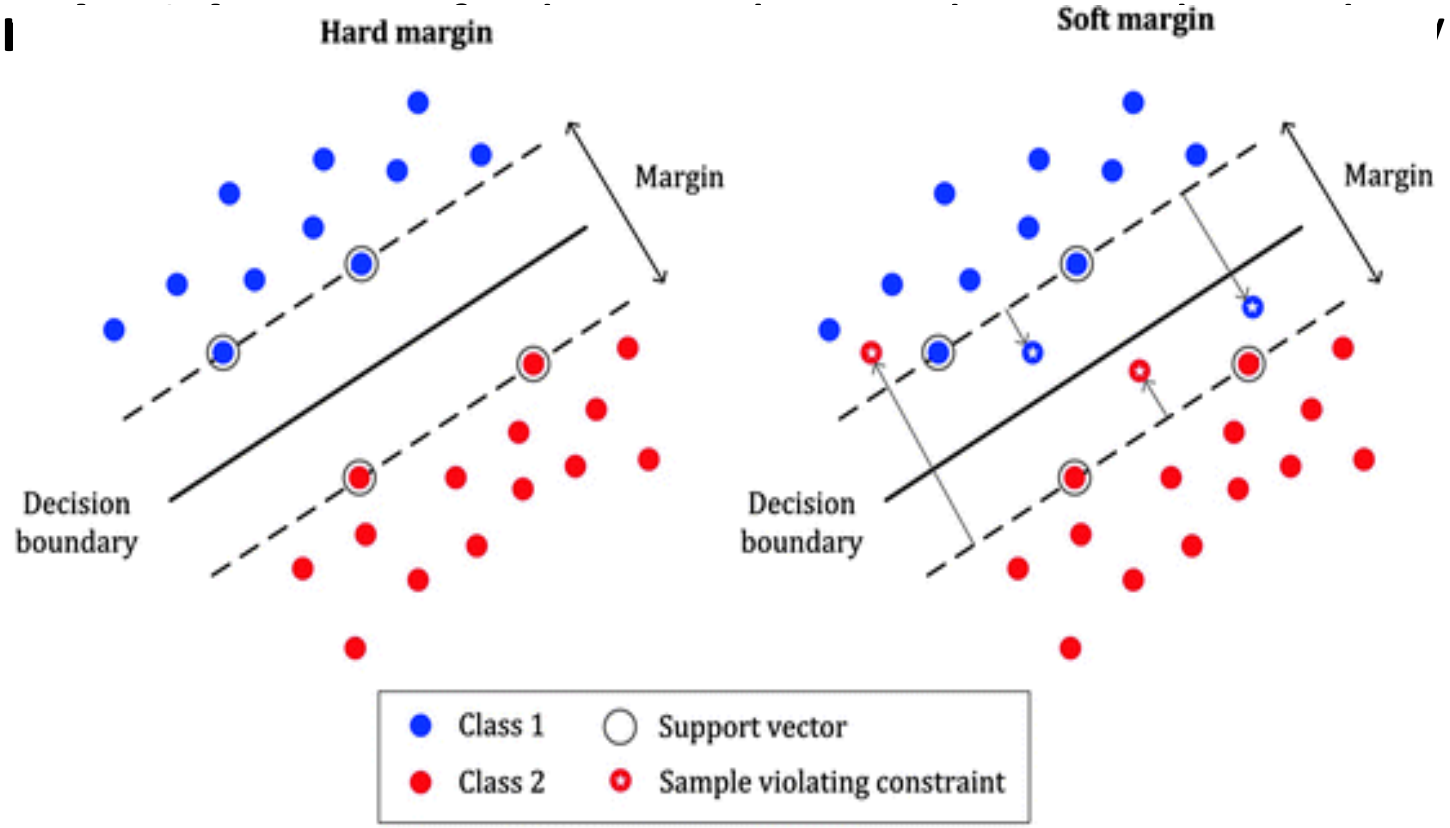
- For a 2-dimensional space, the Hyperplane, which is the line.

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 = 0$$


G1
B
X2
G2
D
X1

Soft Margin Vs. Hard Margin

- **Soft Margin:** try to find a line to separate, but tolerate one or few misclassified dots (e.g. the dots circled in red)
- **Ker**



Comparison with other classifiers

- SVM can be used to separate linear as well as non-linear space (kernel trick)
- Due to maximum margin on both sides it is less likely to result in over-fitting
- Can work with even smaller datasets
- Complexity is linear

- ***Degree of tolerance***
  How much tolerance(soft) we want to give when finding the decision boundary is an important hyper-parameter for the SVM (both linear and nonlinear solutions).
- It is represented as the penalty term — ‘C’.
- The bigger the C, the more penalty SVM gets when it makes misclassification..

we define linear classifier function or Separating classifier hyper-plane as

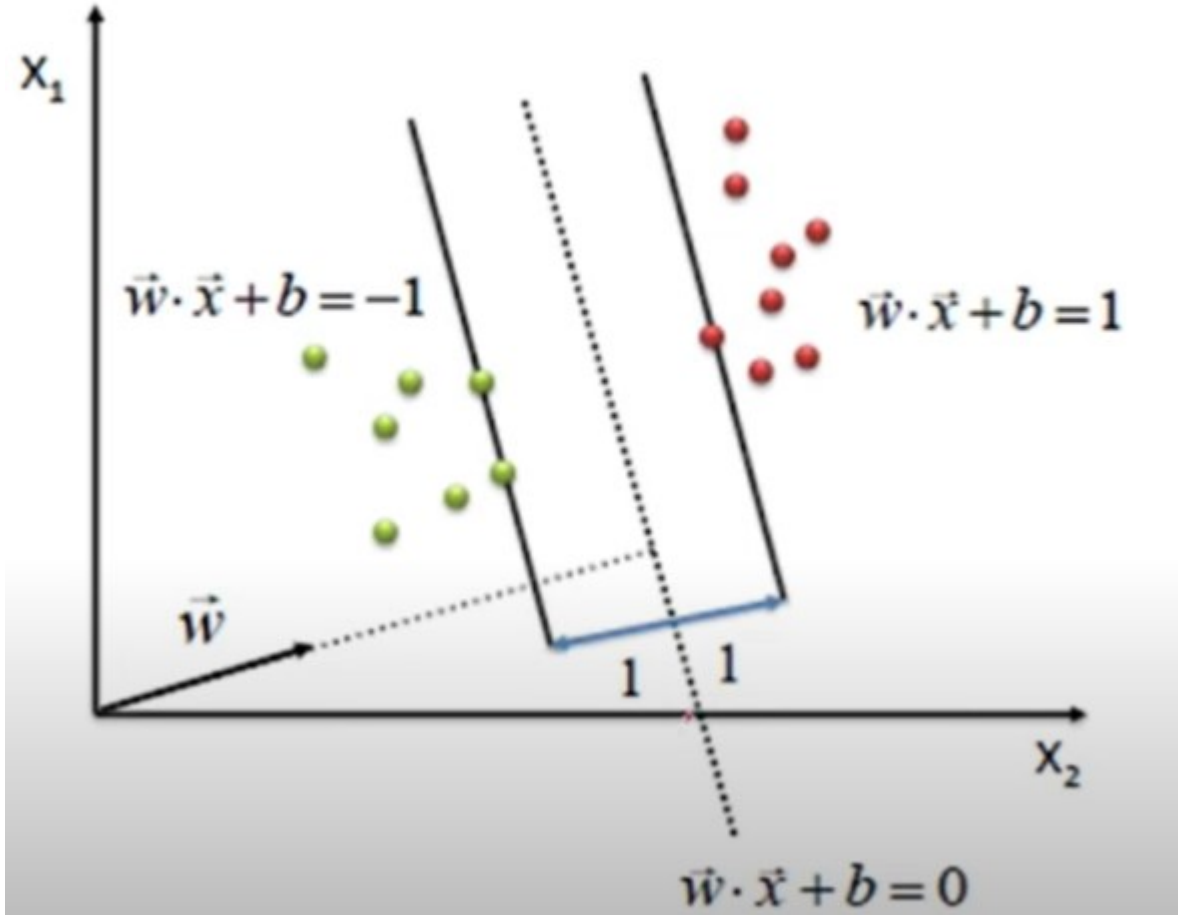
W.T* X + B OR W dot X + B

After training we get optimal values of W & B Then

If for any xi

W.T* X + B>0 then xi lies on +ve side (or class c1)

W.T* X + B<0 then xi lies on -ve side (or class c2)


x1
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
x2
$\vec{w}\cdot\vec{x}+b=0$

- Thus Optimization problem can be framed as

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

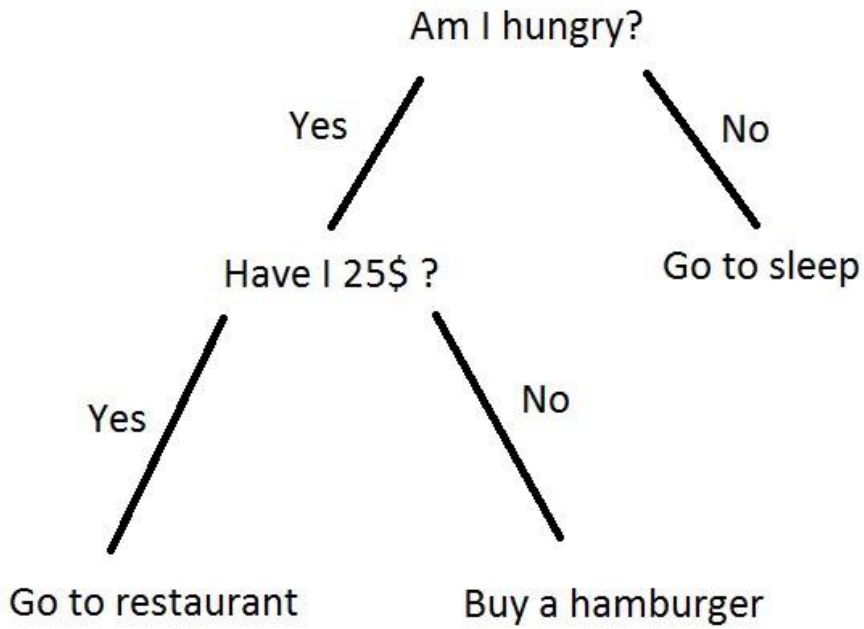
Decision Tree

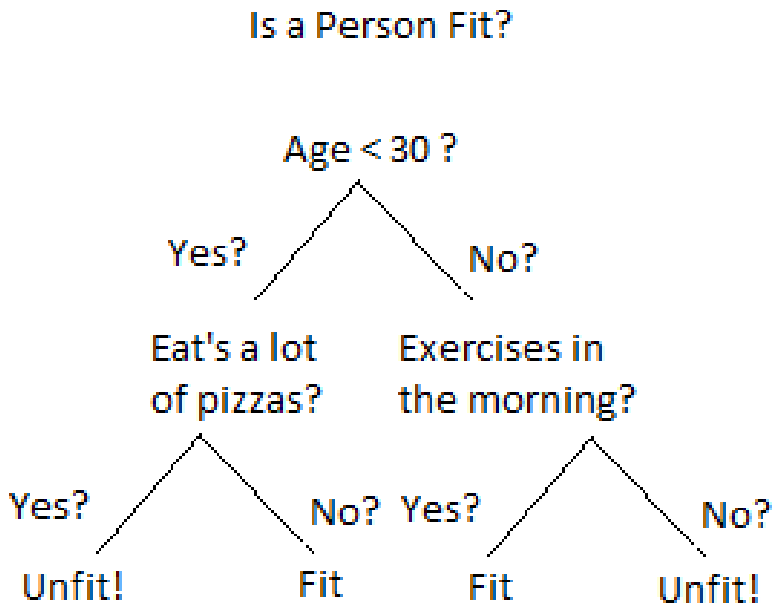
- Decision trees are used for both classification and regression problems
- Decision tree is the most powerful and popular tool for classification and prediction.
- A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.
- simple to understand the data and make some good interpretations.

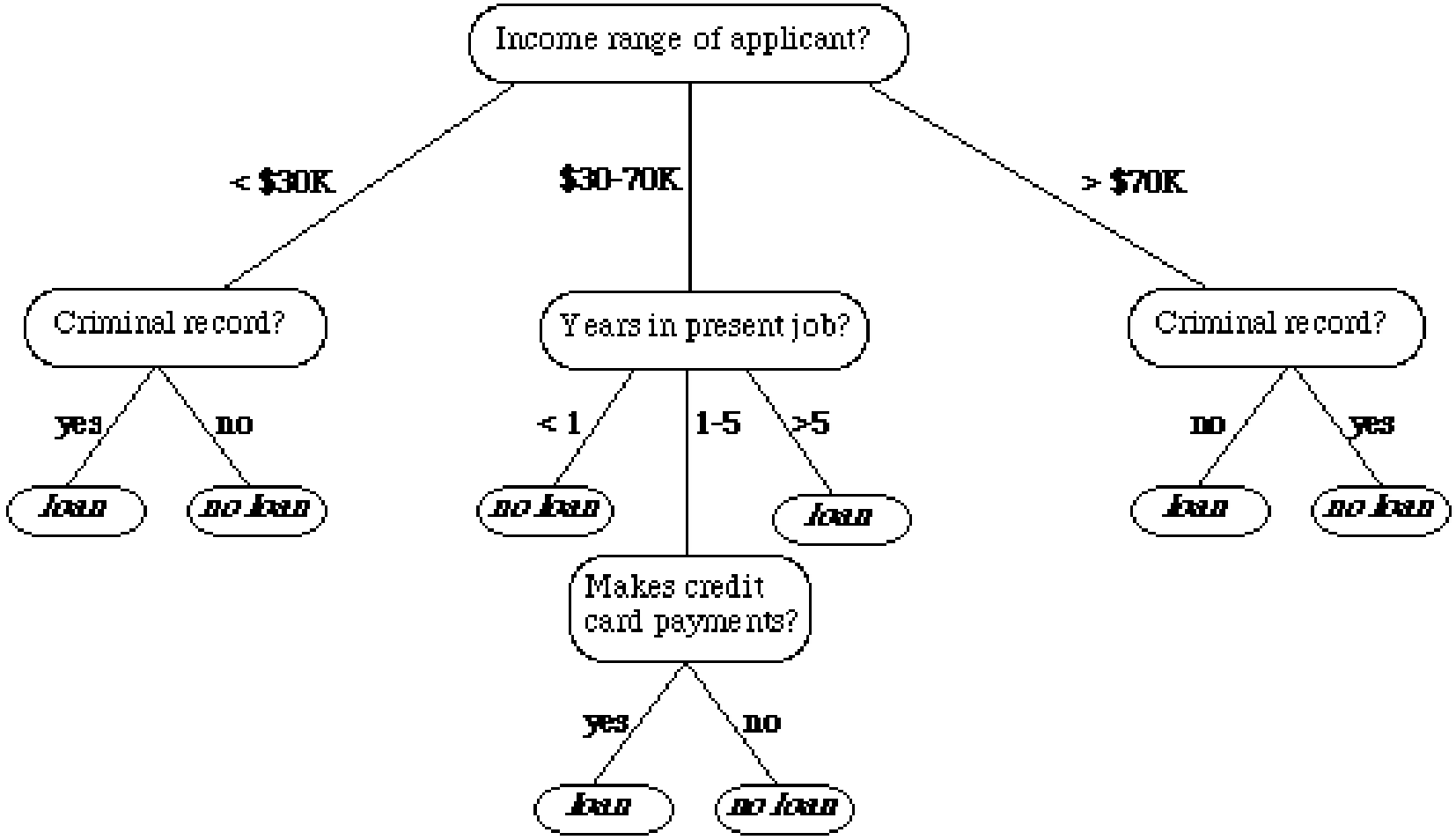
A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).

Tree can be built by using many algorithm but Popular are

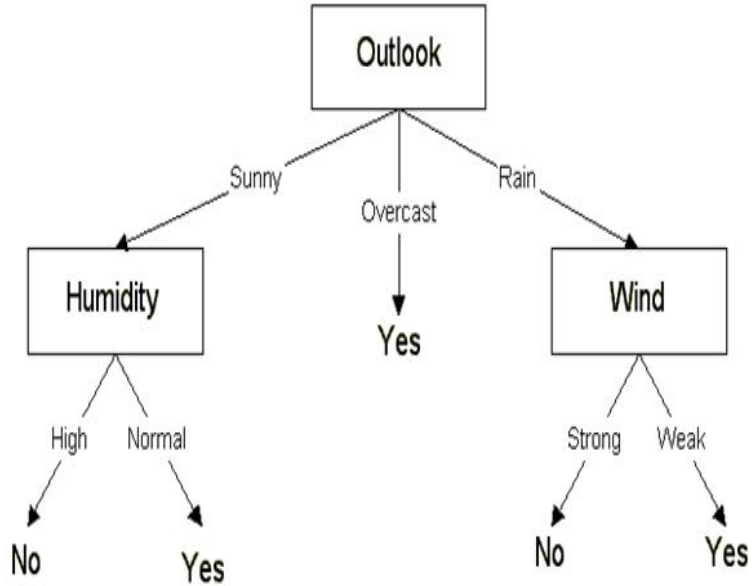
1. ID3 (Iterative Dichotomiser 3) → uses ***Entropy function*** and ***Information gain*** as metrics.
2. CART (Classification and Regression Trees) → uses ***Gini Index(Classification)*** as metric.


Am I hungry?
Yes
No
Have I 25$?
Go to sleep
Yes
No
Go to restaurant
Buy a hamburger


Is a Person Fit?
Age < 30 ?
Yes?
No?
Eat's a lot of pizzas?
Exercises in the morning?
Yes?
No?
Yes?
No?
Unfit!
Fit
Fit
Unfit!


Income range of applicant?
< $30K
$30-70K
> $70K
Criminal record?
Years in present job?
Criminal record?
yes
no
< 1
1-5
>5
no
yes
loan
no loan
loan
no loan
loan
no loan
Makes credit card payments?
yes
no
loan
no loan

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Entropy

- ***Entropy*** *is the measures of* ***impurity, disorder*** *or* ***uncertainty*** *in a bunch of examples.*
- *Entropy controls how a Decision Tree decides to* ***split*** *the data. It actually effects how a* ***Decision Tree*** *draws its boundaries.*

$$Entropy = -\sum p(X) \log p(X)$$

here p(x) is a fraction of examples in a given class

Information Gain

- ***Information gain (IG)*** *measures how much "information" a feature gives us about the class.*
- **Information gain** is the main key that is used by **Decision Tree Algorithms** to construct a Decision Tree.
- **Decision Trees** algorithm will always tries to maximize **Information gain**.
- An **attribute** with highest **Information gain** will tested/split first.

$$\text{Information gain} = \text{entropy (parent)} - [\text{weightes average}] * \text{entropy (children)}$$

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

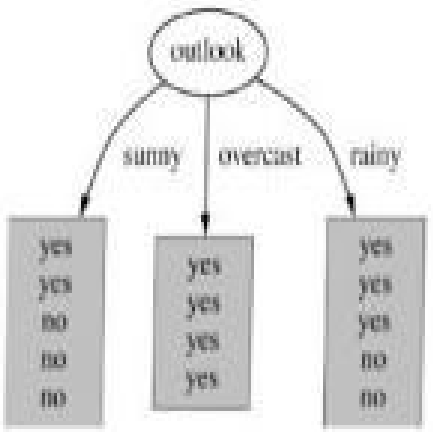
C ={yes,no}

Out of 14 instances, 9 are classified as yes, and 5 as no

pyes = −(9/14)*log2(9/14) = 0.41

pno = −(5/14)*log2(5/14) = 0.53

H (S) = pyes + pno = 0.94



E (Outlook=sunny) = $-\frac{2}{5}\log\left(\frac{2}{5}\right)-\frac{3}{5}\log\left(\frac{3}{5}\right)=0.971$

E (Outlook=overcast) = $-1\log(1)-0\log(0)=0$

E (Outlook=rainy) = $-\frac{3}{5}\log\left(\frac{3}{5}\right)-\frac{2}{5}\log\left(\frac{2}{5}\right)=0.971$

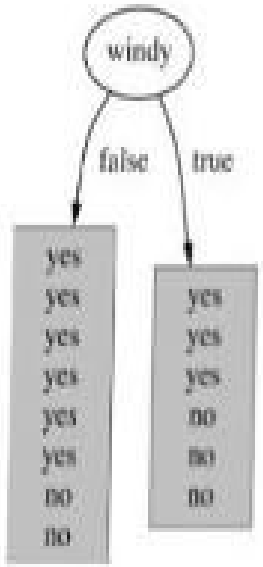
} H(S,Outlook)

Average Entropy information for Outlook

I (Outlook) = $\frac{5}{14}*0.971+\frac{4}{14}*0+\frac{5}{14}*0.971=0.693$

} $\sum_{t\in T} p(t)H(t)$

Gain (Outlook) = E(S) – I (outlook) = 0.94-.693 = 0.247 ⇨ $IG(A,S)=H(S)-\sum_{t\in T} p(t)H(t)$



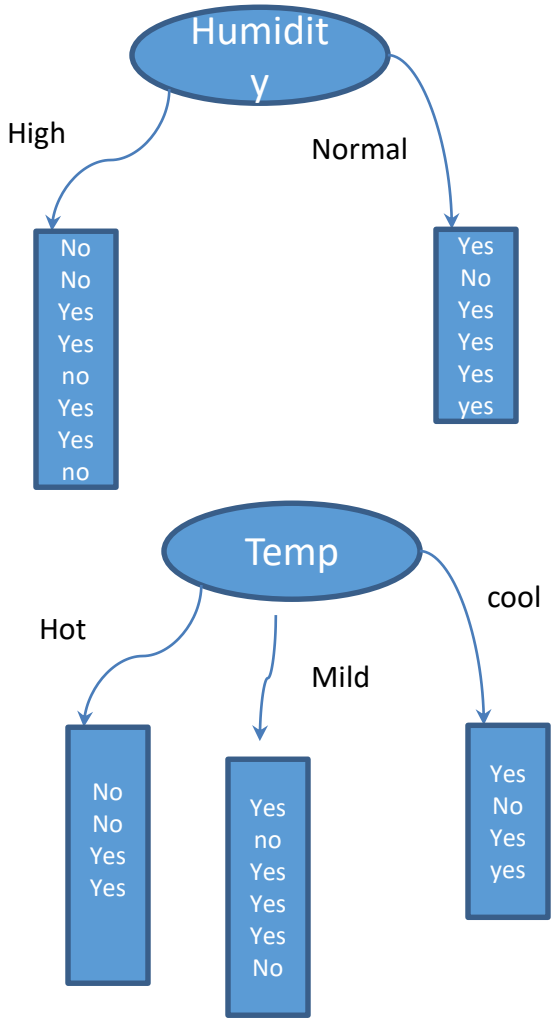
E (Windy=false) = $-\frac{6}{8}\log\left(\frac{6}{8}\right)-\frac{2}{8}\log\left(\frac{2}{8}\right)=0.811$

E (Windy=true) = $-\frac{3}{6}\log\left(\frac{3}{6}\right)-\frac{3}{6}\log\left(\frac{3}{6}\right)=1$

Average entropy information for Windy

I (Windy) = $\frac{8}{14}*0.811+\frac{6}{14}*1=0.892$

Gain (Windy) = E(S) – *I* (Windy) = 0.94-0.892=0.048



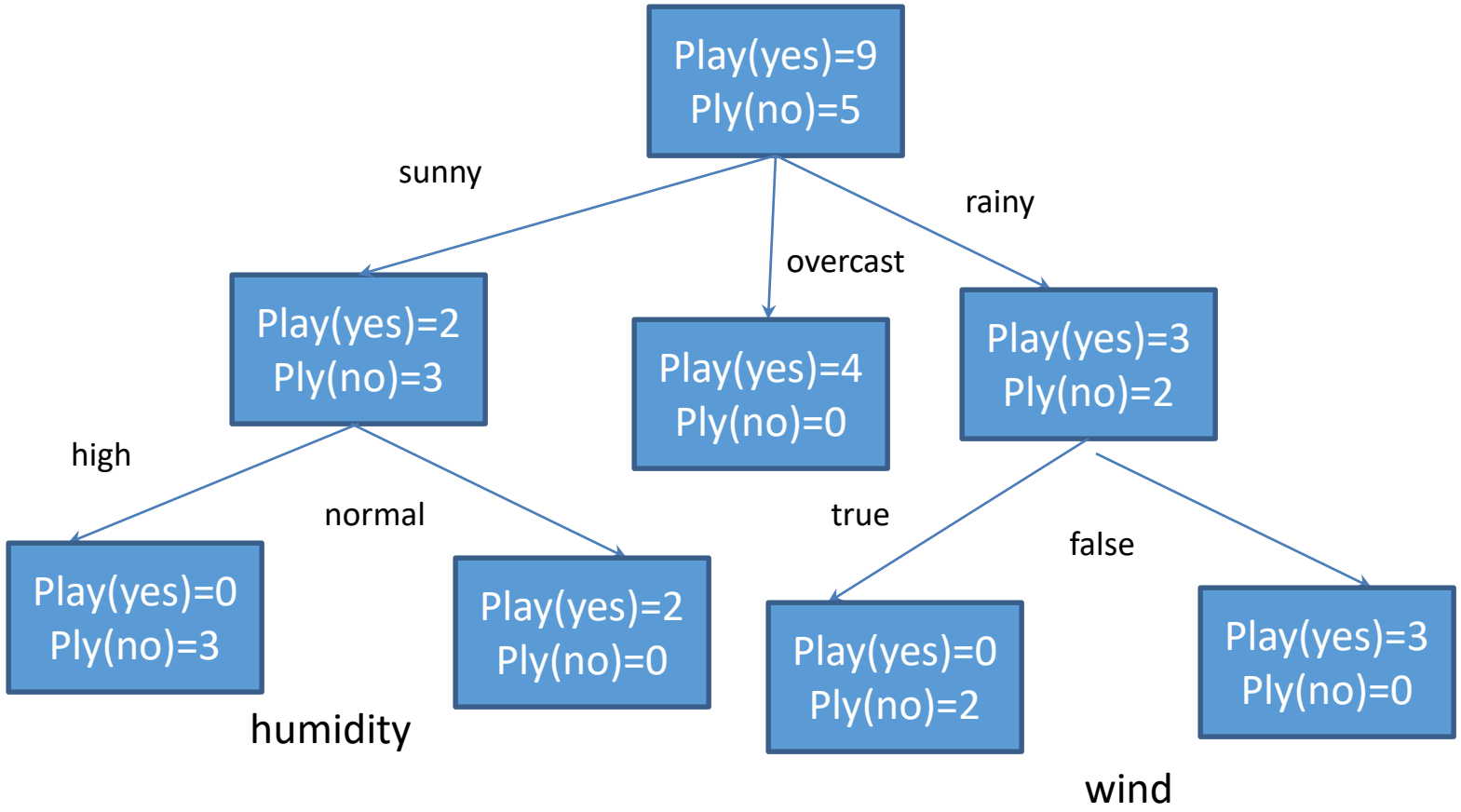
E(Humidity,high)=-4/8*log(4/8)-4/8log(4/8)=x
E(humidity,normal)=-5/6*log5/6-1/6*log(1/6)=y
Average entropy of humidity (z)
Z=8/14*x+6/14*y
Gain(humidity)=.94-z=p
E(temp, High)=-2/4*log(2/4)-2/4*log(2/4)=s
E(temp, mild)=-4/6*log(4/6)-2/6*log(2/6)=t
E(temp, cool)=-3/4*log(3/4)-1/4*log(1/4)=u
Average entropy of Temp
Q=4/14*s+6/14*t+4/14*u
Gain(temp)=.94-Q

$IG(S, Outlook) = 0.246$

$IG(S, Temperature) = 0.029$

$IG(S, Humidity) = 0.151$

$IG(S, Wind) = 0.048$ (Previous example)


Play(yes)=9
Ply(no)=5
sunny
overcast
rainy
Play(yes)=2
Ply(no)=3
Play(yes)=4
Ply(no)=0
Play(yes)=3
Ply(no)=2
high
normal
true
false
Play(yes)=0
Ply(no)=3
Play(yes)=2
Ply(no)=0
Play(yes)=0
Ply(no)=2
Play(yes)=3
Ply(no)=0
humidity
wind

CART

- Classification and Regression Tree(CART)
- Gini index is a metric for classification tasks in CART.
- It stores sum of squared probabilities of each class.

$$Gini = 1 - \Sigma (Pi)^2 \text{ for } i=1 \text{ to number of classes}$$

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$

Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$

Gini(Outlook=Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$

Then, we will calculate weighted sum of gini indexes for outlook feature.

Gini(Outlook) = $(5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$
Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$
Gini(Outlook=Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$
Then, we will calculate weighted sum of gini indexes for outlook feature.
Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = 0.342

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

Temperature

Gini(Temp=Hot) = $1 – (2/4)^2 – (2/4)^2 = 0.5$

Gini(Temp=Cool) = $1 – (3/4)^2 – (1/4)^2 = 1 – 0.5625 – 0.0625 = 0.375$

Gini(Temp=Mild) = $1 – (4/6)^2 – (2/6)^2 = 1 – 0.444 – 0.111 = 0.445$

We'll calculate weighted sum of gini index for temperature feature

Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = 0.439

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

Humidity

Humidity is a binary class feature. It can be high or normal.

$$Gini(Humidity=High) = 1 - (3/7)^2 - (4/7)^2$$
$$= 1 - 0.183 - 0.326$$
$$= 0.489$$

$$Gini(Humidity=Normal) = 1 - (6/7)^2 - (1/7)^2$$
$$= 1 - 0.734 - 0.02$$
$$= 0.244$$

Weighted sum for humidity feature will be calculated next

$$Gini(Humidity) = (7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$$

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

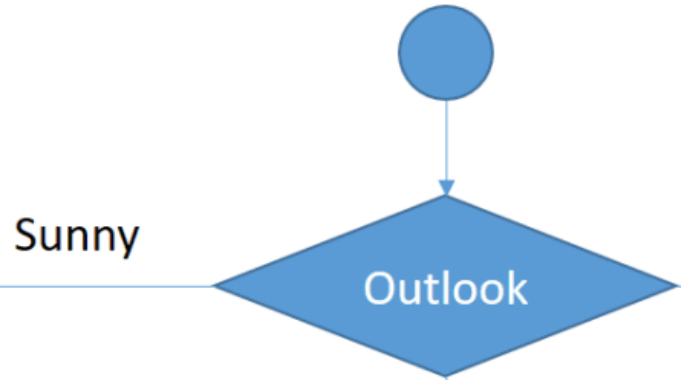
Wind

$$Gini(Wind=Weak) = 1 - (6/8)^2 - (2/8)^2$$
$$= 1 - 0.5625 - 0.062 = 0.375$$

$$Gini(Wind=Strong) = 1 - (3/6)^2 - (3/6)^2$$
$$= 1 - 0.25 - 0.25 = 0.5$$

$$Gini(Wind) = (8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$$

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini of temperature for sunny outlook

Gini(Outlook=Sunny and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$
Gini(Outlook=Sunny and Temp.=Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$
Gini(Outlook=Sunny and Temp.=Mild) = $1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$
Gini(Outlook=Sunny and Temp.) = $(2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2$

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

Gini of wind for sunny outlook

Gini(Outlook=Sunny and Wind=Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.266$

Gini(Outlook=Sunny and Wind=Strong) = $1- (1/2)^2 - (1/2)^2 = 0.2$

Gini(Outlook=Sunny and Wind) = $(3/5)x0.266 + (2/5)x0.2 = 0.466$

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini of humidity for sunny outlook

Gini(Outlook=Sunny and Humidity=High) = $1 - (0/3)^2 - (3/3)^2 = 0$

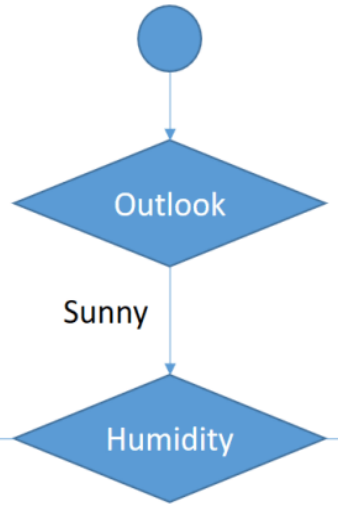
Gini(Outlook=Sunny and Humidity=Normal) = $1 - (2/2)^2 - (0/2)^2 = 0$

Gini(Outlook=Sunny and Humidity) = $(3/5)x0 + (2/5)x0 = 0$

Decision for sunny outlook

We've calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value. We'll put humidity check at the extension of sunny outlook.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466



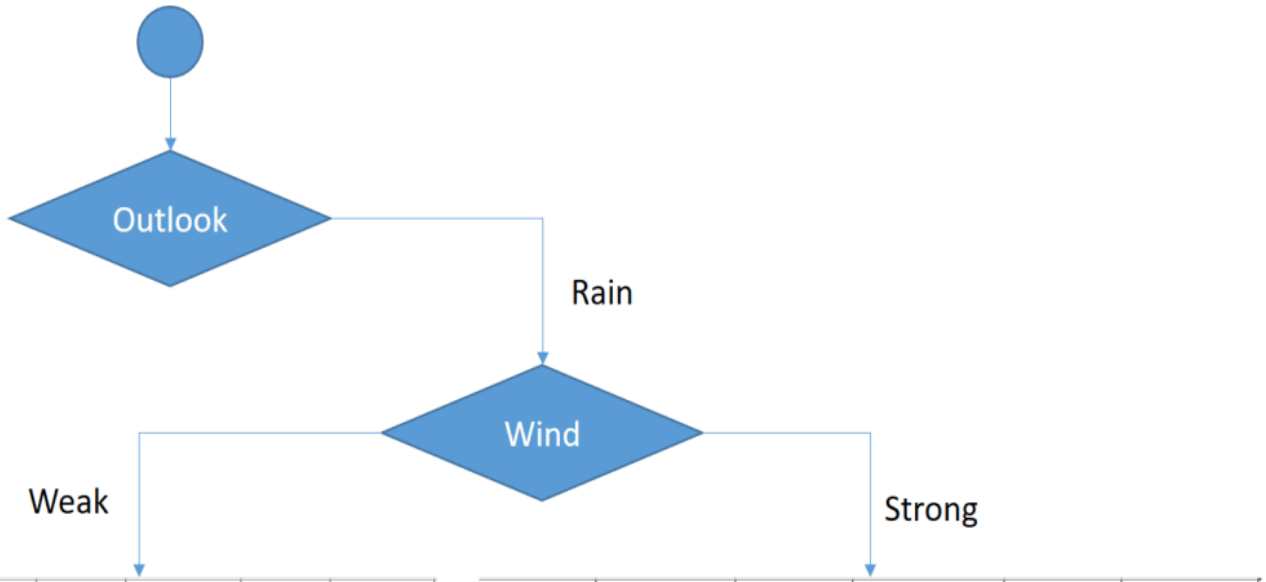
Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Decision for rain outlook

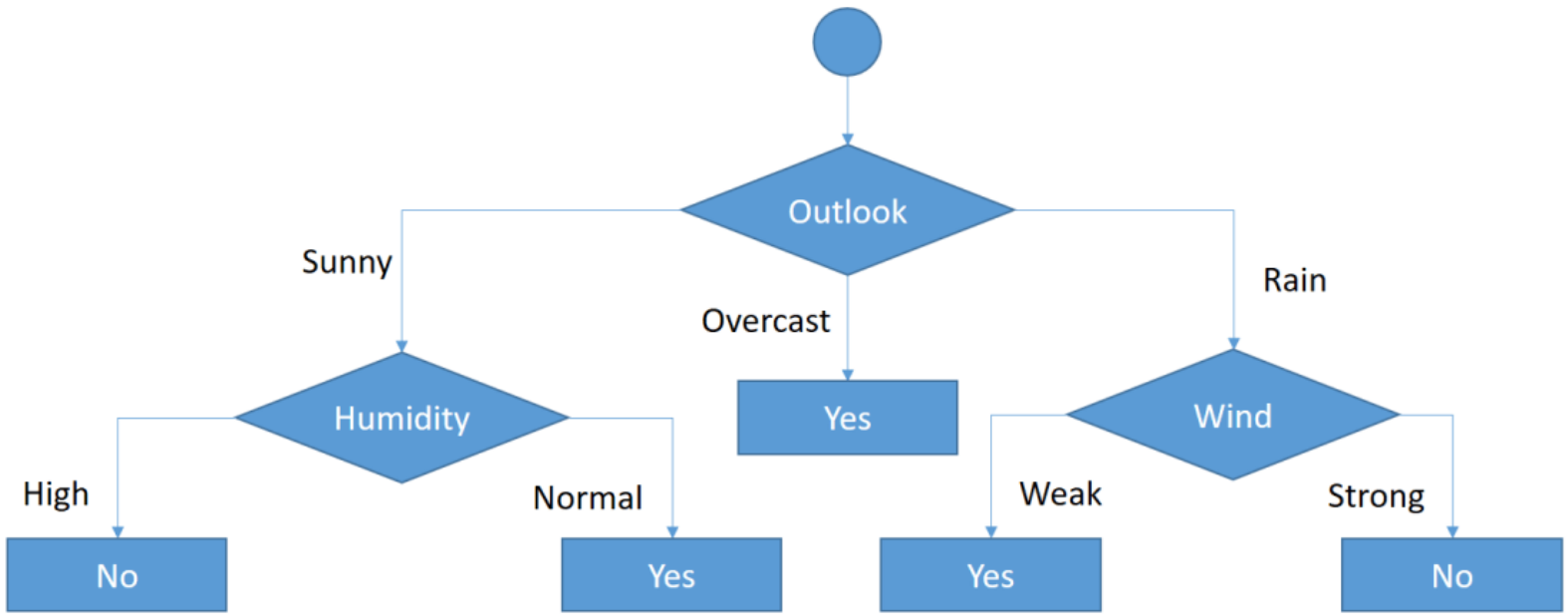
The winner is wind feature for rain outlook because it has the minimum gini index score in features.
Put the wind feature for rain outlook branch and monitor the new sub data sets.

Temperature	0.466
Humidity	0.466
Wind	0



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Weak
Strong
No
Yes
Yes
No

Decision Tree – Regression

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed

The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

Predictors

Target

Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	46
Sunny	Mild	High	False	45
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	46
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

Decision Tree Algorithm

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with *Standard Deviation Reduction*.

Standard Deviation

$$\textbf{Standard Deviation} = S = \sqrt{\frac{\sum(x - \bar{x})^2}{n}}$$

$$\textbf{Coeffeicient of Variation} = CV = \frac{S}{\bar{x}} * 100\%$$

a) Standard deviation for **one** attribute:

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$\boldsymbol{Count} = n = 14$$

$$\boldsymbol{Average} = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$\boldsymbol{Standard\ Deviation} = S = \sqrt{\frac{\sum(x - \bar{x})^2}{n}} = 9.32$$

$$\boldsymbol{Coeffeicient\ of\ Variation} = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

- Standard Deviation (**S**) is for tree building (branching).
- Coefficient of Deviation (**CV**) is used to decide when to stop branching. We can use Count (**n**) as well.
- Average (**Avg**) is the value in the leaf nodes.

b) Standard deviation for **two** attributes (target and predictor):

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)*__S__(Sunny) + **P**(Overcast)*__S__(Overcast) + **P**(Rainy)*__S__(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

Standard Deviation Reduction

The standard deviation reduction is based on the decrease in standard deviation after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest standard deviation reduction (i.e., the most homogeneous branches).

Step 1: The standard deviation of the target is calculated.

Standard deviation (Hours Played) = 9.32

Step 2:
The dataset is then split on the different attributes. The standard deviation for each branch is calculated. The resulting standard deviation is subtracted from the standard deviation before the split. The result is the standard deviation reduction.

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)*__S__(Sunny) + **P**(Overcast)*__S__(Overcast) + **P**(Rainy)*__S__(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

$$SDR(T,X) = S(T) - S(T,X)$$

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR= 0.48		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

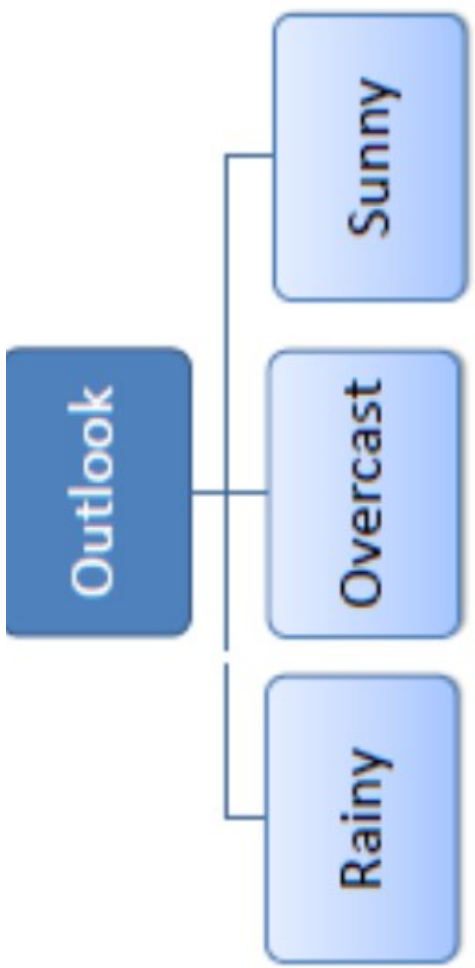
		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Step 3: The attribute with the largest standard deviation reduction is chosen for the decision node.

★		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

Step 4a:

The dataset is divided based on the values of the selected attribute. This process is run recursively on the non-leaf branches, until all data is processed.



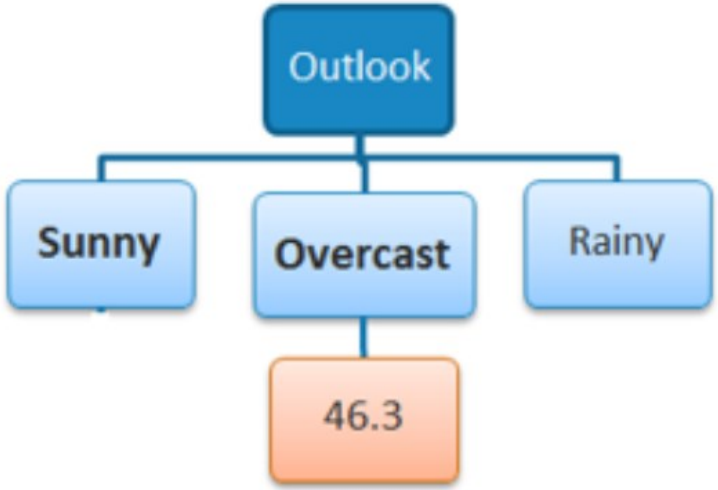
Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

In practice, we need some termination criteria. For example, when coefficient of deviation (CV) for a branch becomes smaller than a certain threshold (e.g., 10%) and/or when too few instances (n) remain in the branch (e.g., 3).

Step 4b: "Overcast" subset does not need any further splitting because its CV (8%) is less than the threshold (10%). The related leaf node gets the average of the "Overcast" subset.

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Step 4c: However, the "Sunny" branch has an CV (28%) more than the threshold (10%) which needs further splitting. We select "Temp" as the best best node after "Outlook" because it has the largest SDR.

Outlook - Sunny

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	**45**
Cool	Normal	FALSE	**52**
Cool	Normal	TRUE	**23**
Mild	Normal	FALSE	**46**
Mild	High	TRUE	**30**
			S = 10.87
			AVG = 39.2
			CV = 28%

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

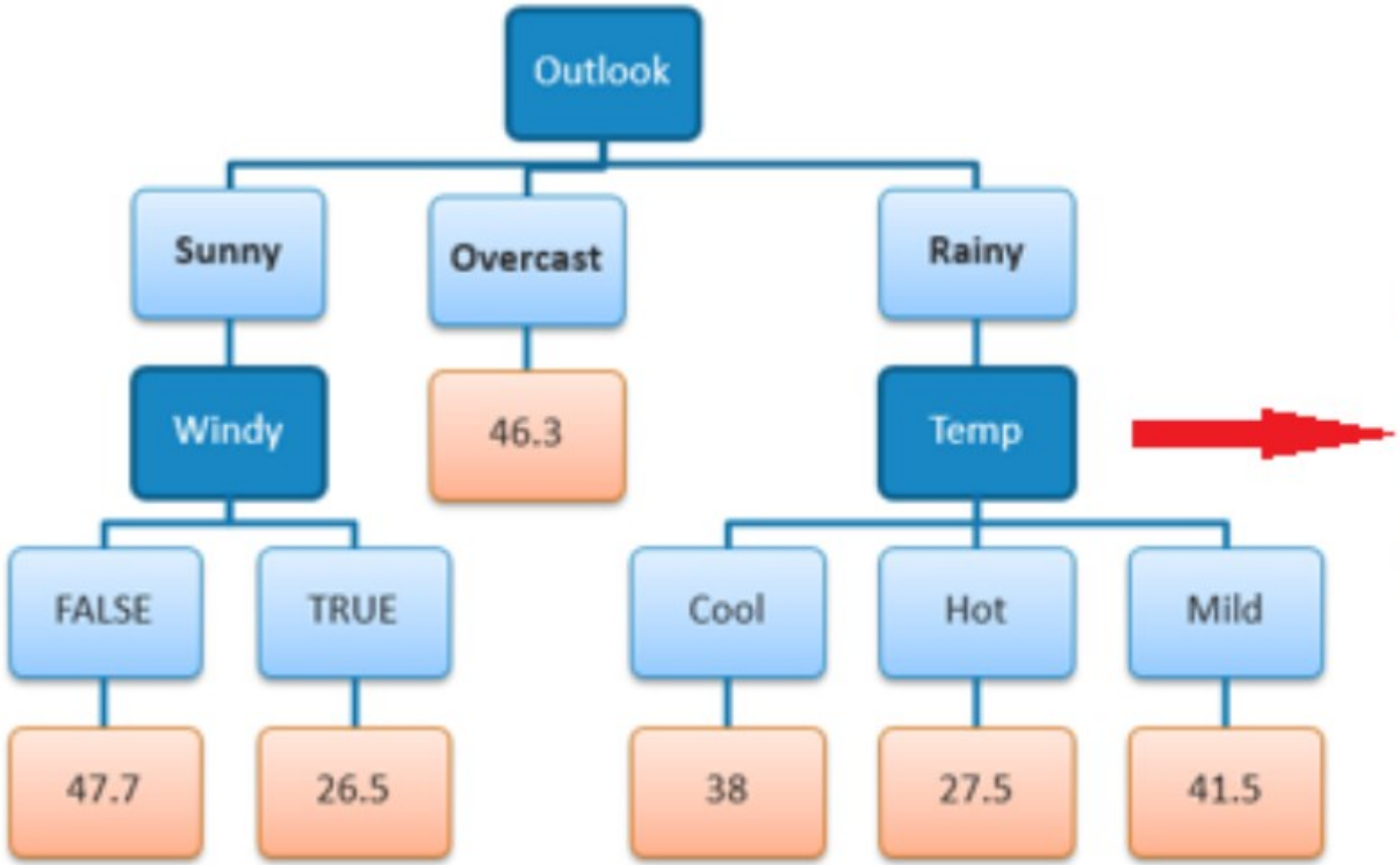
		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

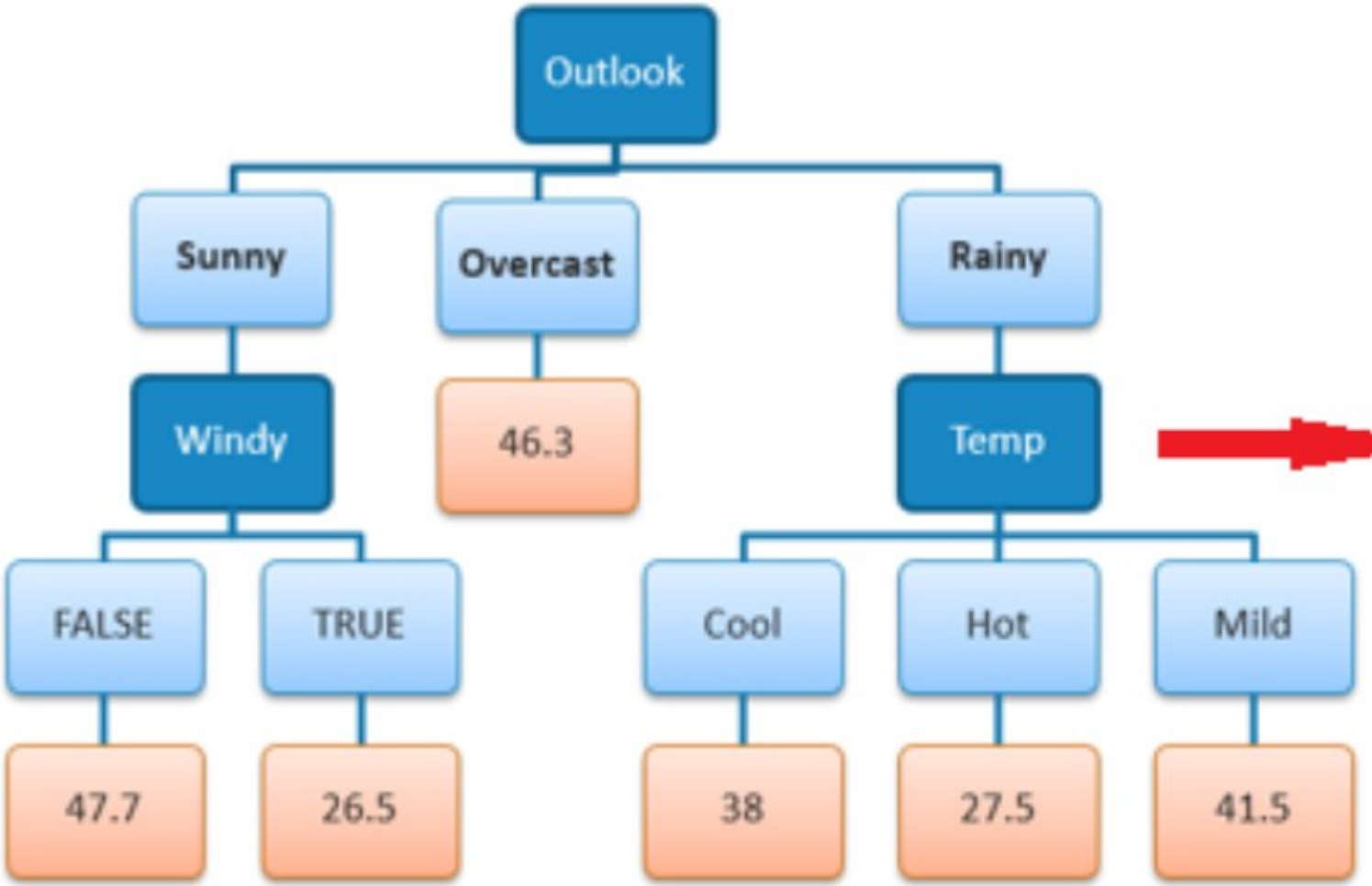
		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

Because the number of data points for all three branches (Cool, Hot and Mild) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.



Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48



Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48

When the number of instances is more than one at a *leaf node* we calculate the *average* as the final value for the target.

In machine learning, the terms training data, testing data, and validation data refer to different subsets of a dataset used for various stages in the development and evaluation of a model. These subsets play crucial roles in training and assessing the performance of machine learning models. Here's a brief explanation of each:

Training Data:

- **Purpose:** This is the subset of data used to train the machine learning model.
- **Usage:** During the training phase, the model learns patterns and relationships within the training data.
- **Size:** The training dataset is typically the largest portion of the overall dataset.

Testing Data (or Test Data):

- **Purpose:** This is a separate subset of data used to evaluate the performance of the model after it has been trained.
- **Usage:** The model has never seen the testing data during training, so its ability to generalize to new, unseen examples is assessed using the testing data.
- **Size:** The testing dataset is held out until the model has been trained to avoid biasing the evaluation.

Validation Data:

•**Purpose:** This is another subset of data used to fine-tune the model during the training phase.

•**Usage:** It helps to assess the model's performance on data it has not seen during training and provides a basis for adjusting hyperparameters to improve generalization.

•**Size:** The validation dataset is used to make decisions about the model's architecture or hyperparameters, and it is not used in the final evaluation of the model's performance.

The typical split among these subsets can vary, but a common practice is to use a large portion (e.g., 70-80%) for training, a smaller portion (e.g., 10-15%) for testing, and the rest for validation. The exact split depends on factors such as the size of the dataset and the specific requirements of the machine learning task. The key idea is to ensure that the model is trained on diverse data, tested on unseen data, and validated to improve its generalization capabilities.

Validation data plays a crucial role in the training process of a machine learning model, particularly in the context of fine-tuning and improving its performance. Here are more details about validation data:

Purpose of Validation Data:

•**Fine-Tuning Hyperparameters:** During the training phase, a machine learning model has various hyperparameters (e.g., learning rate, regularization strength) that need to be set. The goal is to find the combination of hyperparameter values that results in the best model performance. The validation dataset is used to evaluate the model's performance for different hyperparameter settings, helping to choose the best configuration.

•**Preventing Overfitting:** Overfitting occurs when a model performs well on the training data but fails to generalize to new, unseen data. The validation data helps monitor the model's performance on examples it hasn't seen during training. If the model performs well on the training data but poorly on the validation data, it may be overfitting. Adjustments can then be made to prevent overfitting, such as reducing model complexity or applying regularization techniques.

Size of Validation Data:

The size of the validation dataset is crucial. It should be large enough to provide a representative sample of the data but not so large that it significantly reduces the amount of data available for training. Common splits include 80% for training, 10% for validation, and 10% for testing.

validation data is used to fine-tune a model during training, making adjustments to hyperparameters and preventing overfitting. It helps ensure that the model generalizes well to new, unseen data by providing an unbiased evaluation throughout the training process.

Cross Validation

- in a real-life scenario, the model will be tested for its efficiency and accuracy with an altogether different and unique data set.
- Under those circumstances, you'd want your model to be efficient enough or at least to be at par with the same efficiency that it shows for the training set.
- Basically this testing is known as cross-validation in Machine Learning so that it is fit to work with any model in the future.

Cross Validation

- Cross validation is a technique for assessing how the statistical analysis generalizes to an independent data set.
- It is a technique for evaluating machine learning models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data.
- Using cross-validation, there are high chances that **we can detect over-fitting with ease.**

Types Of Cross-Validation

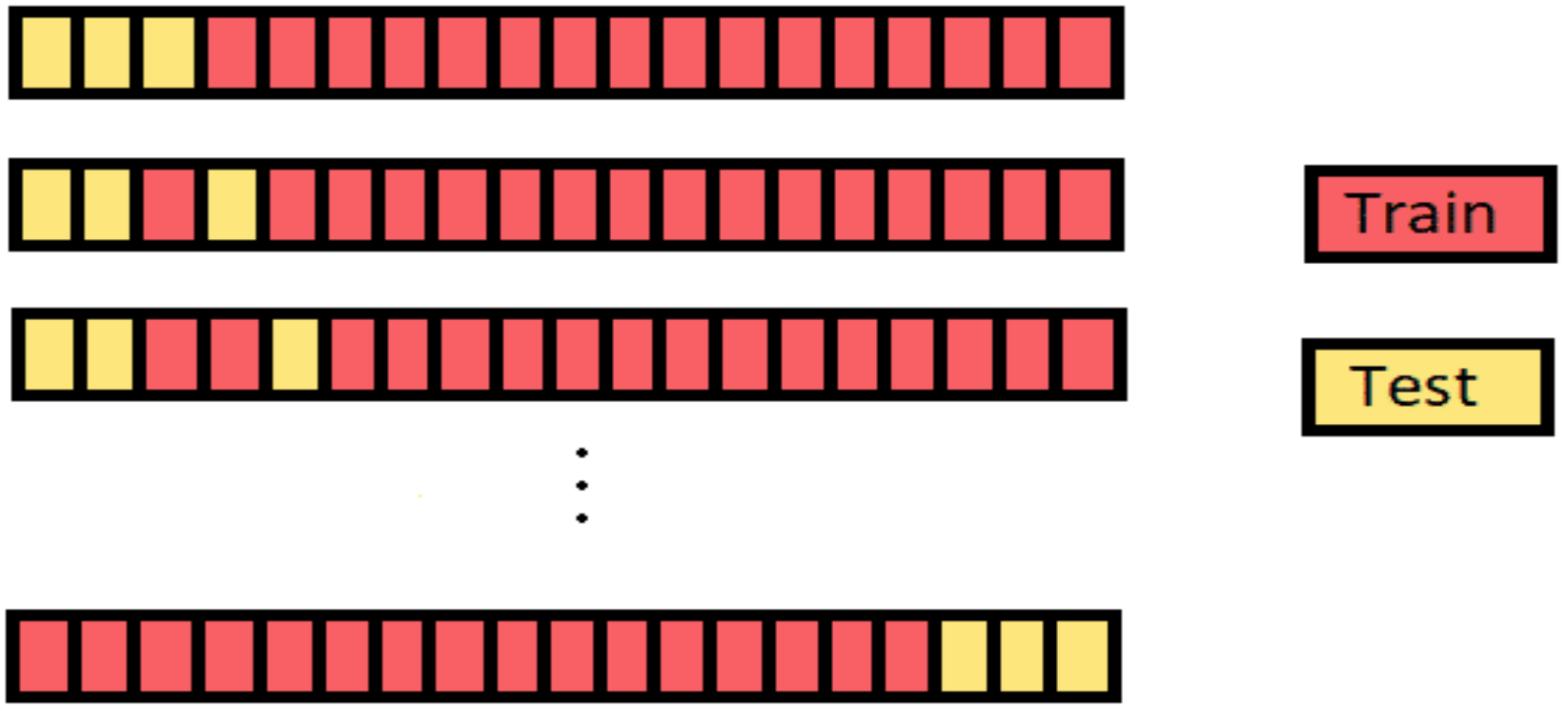
There are two types of cross-validation techniques in Machine Learning.

- **Exhaustive Cross-Validation** – This method basically involves testing the model in all possible ways, it is done by dividing the original data set into training and validation sets. Example: Leave-p-out Cross-Validation, Leave-one-out Cross-validation.
- **Non-Exhaustive Cross-Validation** – In this method, the original data set is not separated into all the possible permutations and combinations. Example: K-fold Cross-Validation, Holdout Method.

Various Types of cross validation

- There are several **cross validation techniques** such as :-
- 1. Leave One-out Cross Validation
  2. Leave P-out Cross Validation
  3. K-Fold Cross Validation
  4. Stratified K-Fold Cross Validation
  5. Holdout Method

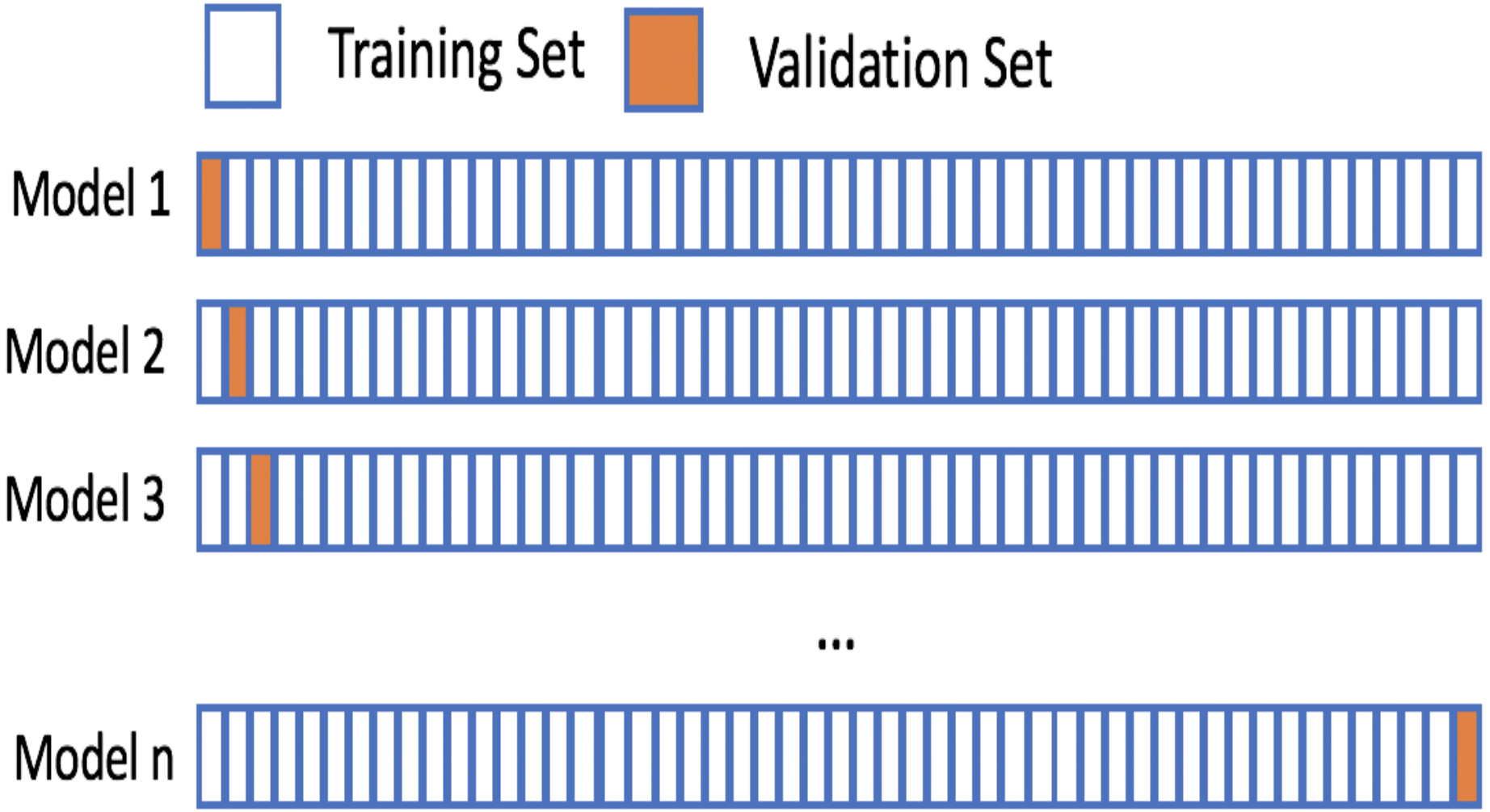
Leave P Out



Leave P out

- In this approach, **p** data points are left out of the training data. Let's say there are **m** data points in the data set, then **m-p** data points are used for the training phase. And the **p** data points are kept as the validation set.
- This technique is rather exhaustive because the above process is repeated for all the possible combinations in the original data set. To check the overall effectiveness of the model, the error is averaged for all the trials.
- It becomes computationally infeasible since the model needs to train and validate for all possible combinations and for a considerably large **p**.

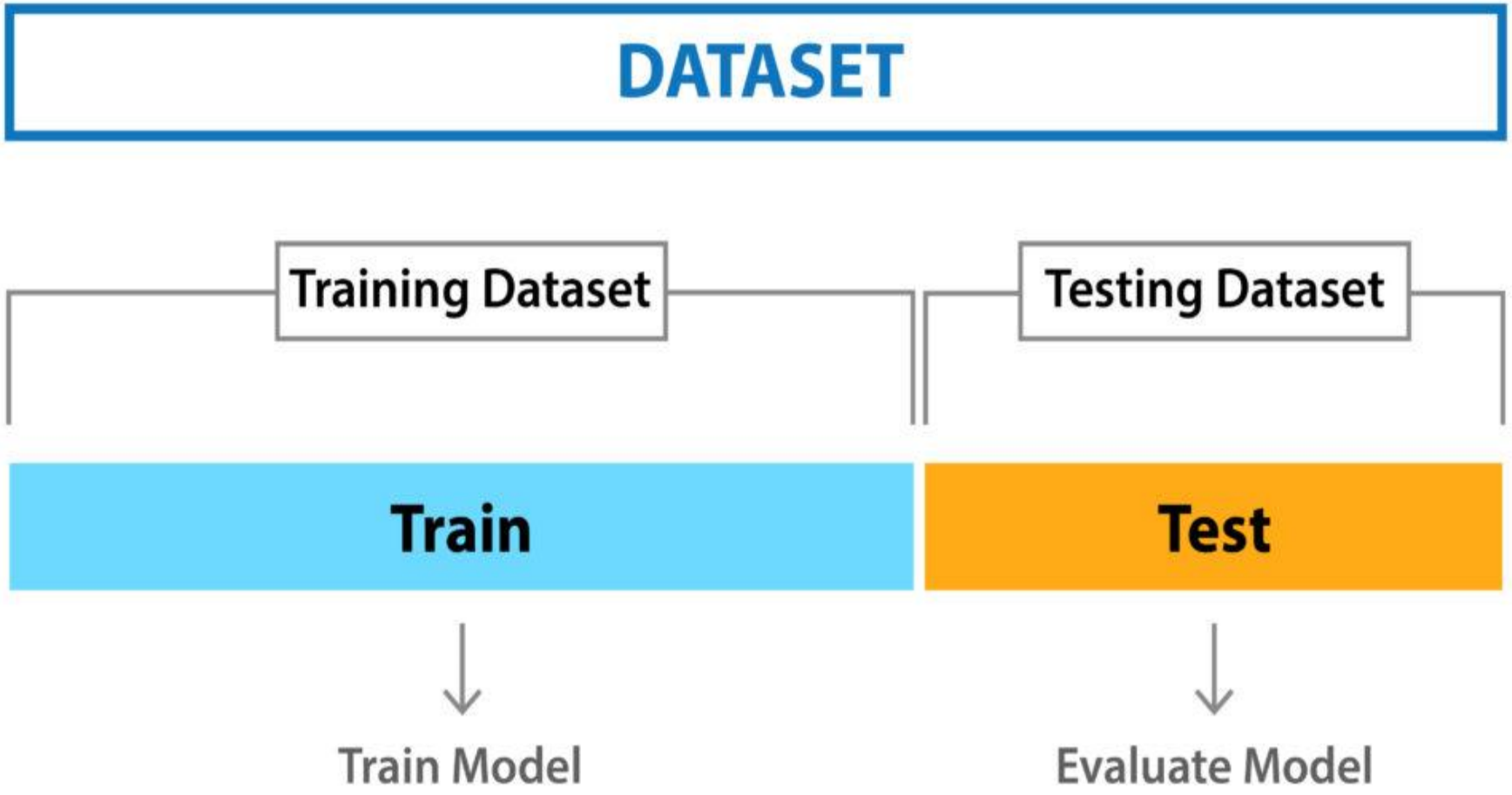
Leave 1 out



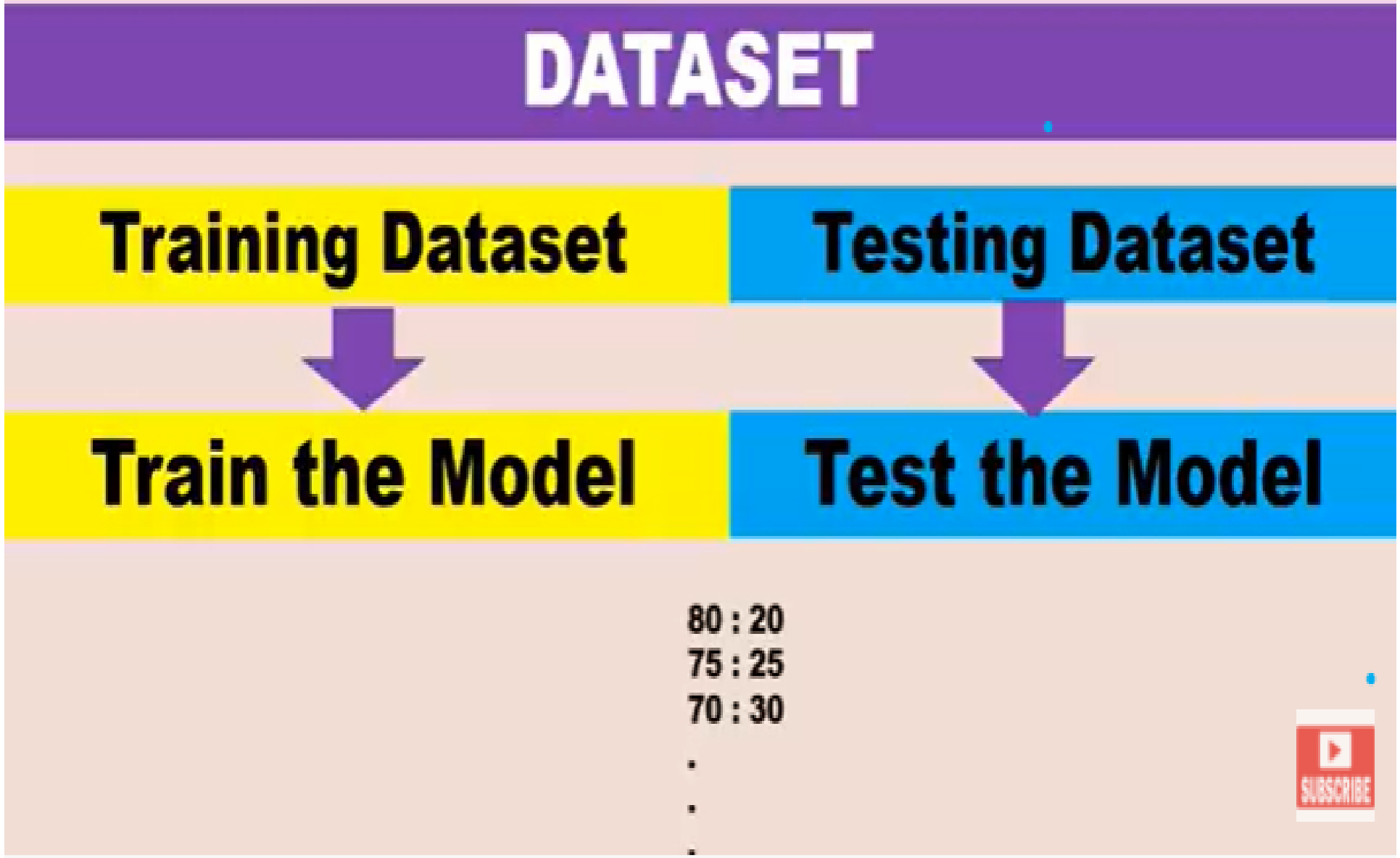
Leave 1 Out

- This method of Cross-validation is similar to Leave-p-out Cross-validation but the only difference is that in this case **p = 1**. It actually saves a lot of time which is a big advantage.
- Although If the sample data is too large, it can still take a lot of time. But it would still be quicker than the Leave-p-out cross-validation method.

Hold Out Method



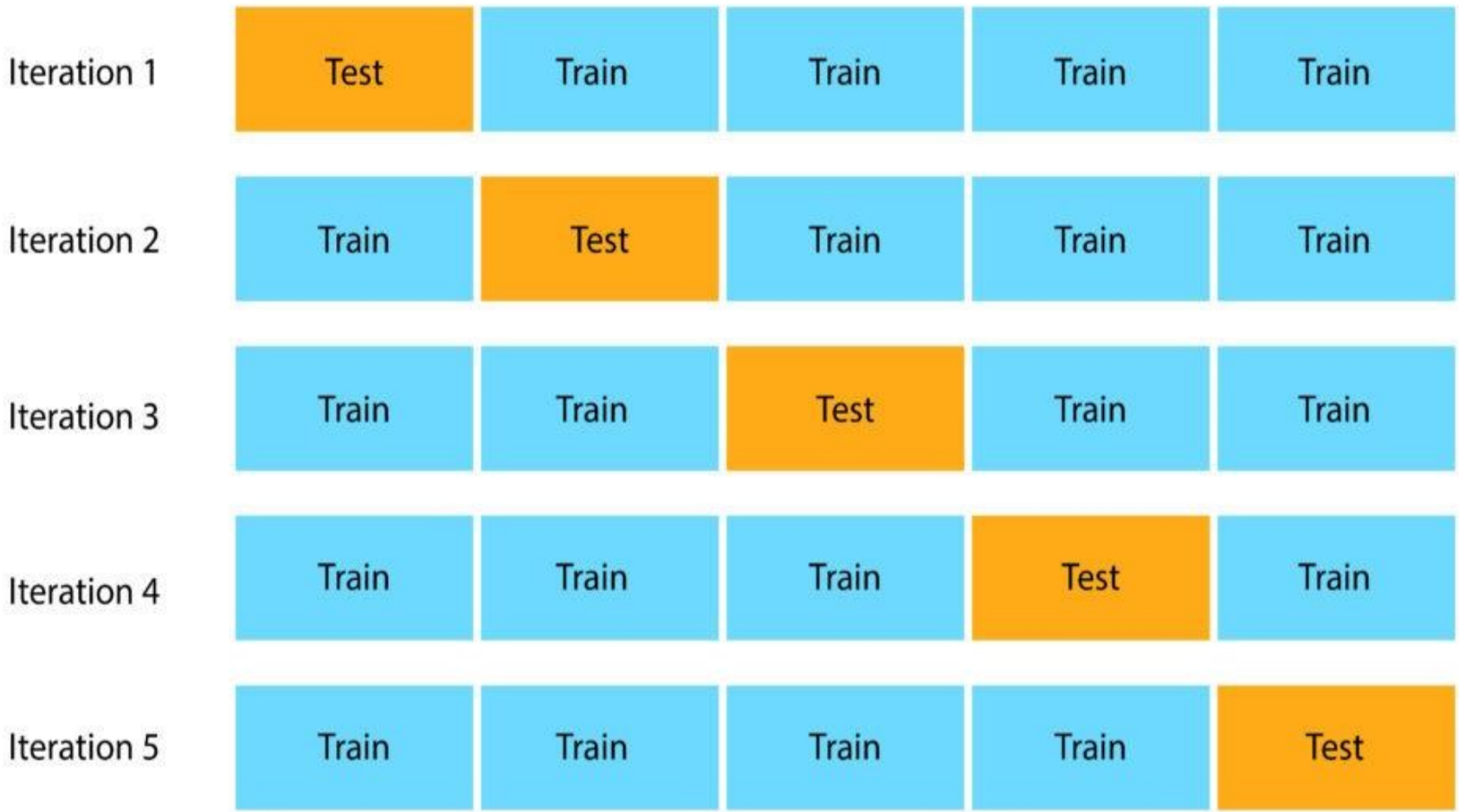
HOLD-OUT METHOD



Hold Out Method

- This is a quite basic and simple approach in which we divide our entire dataset into two parts viz- training data and testing data.
- As the name, we train the model on training data and then evaluate on the testing set.
- Usually, the size of training data is set more than twice that of testing data, so the data is split in the ratio of 70:30 or 80:20.
- In this approach, the data is first shuffled randomly before splitting.
- As the model is trained on a different combination of data points, the model can give different results every time we train it, and this can be a cause of instability.
- Also, we can never assure that the train set we picked is representative of the whole dataset.

K fold cross validation



- K-fold cross validation is one way to improve the holdout method. This method guarantees that the score of our model does not depend on the way we picked the train and test set.
- The data set is divided into k number of subsets and the holdout method is repeated k number of times. Let us go through this in steps:
- Randomly split your entire dataset into k number of folds (subsets)
- For each fold in your dataset, build your model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for kth fold
- Repeat this until each of the k-folds has served as the test set
- The average of your k recorded accuracy is called the cross-validation accuracy and will serve as your performance metric for the model.

Limitations Of Cross-Validation

The following are a few limitations faced by Cross-Validation:

- In an ideal situation, Cross-Validation will produce optimum results. But in case of **inconsistent data**, the results may vary drastically. It is quite uncertain what kind of data will be encountered by the model.

- Predictive modeling often requires an **evolution in terms of data**, this can pretty much change the training and the validation sets drastically.

- The results may **vary depending upon the features of the data set**. Let us say we make a predictive model to detect an ailment in a person and we train it with a specific set of population. It may vary with the general population causing inconsistency and reduced efficiency.
-

Cross-Validation Applications

- With the overpowering applications to prevent a Machine Learning model from Overfitting and Underfitting, there are several other applications of Cross-Validation listed below:
- We can use it to compare the performances of a set of predictive modeling procedures.
- Cross-Validation excels in the field of medical research.
- It can be used in the meta-analysis since a lot of data analysts are already using cross-validation.

During the treatment of cancer patients, the doctor needs to be very careful about which patients need to be given chemotherapy. Which metric should we use in order to decide the patients who should be given chemotherapy?

a. Precision
b. Recall

You have generated data from a 3-degree polynomial with some noise. What do you expect of the model that was trained on this data using a 5-degree polynomial as function class?

a. Low bias, high variance
b. High bias, low variance.
c. Low bias, low variance.
d. High bias, low variance.

- a
- b
- c
- d

Which of the following is true for a decision tree?

a. A decision tree is an example of a linear classifier.
b. The entropy of a node typically decreases as we go down a decision tree.
c. Entropy is a measure of purity.
d. An attribute with lower mutual information should be preferred to other attributes.

- a
- b
- c
- d

What is the naive assumption in a Naive Bayes Classifier?

a. All the classes are independent of each other
b. All the features of a class are independent of each other
c. The most probable feature for a class is the most important feature to be considered for classification
d. All the features of a class are conditionally dependent on each other.

- a
- b
- c
- d