Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Naïve Bayes classifier

• Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks.

Bayes Theorem

• Bayes' Theorem is a simple mathematical formula used for calculating conditional probabilities.

• Conditional probability is a measure of the probability of an event occurring given that another event has (by assumption, presumption, assertion, or evidence) occurred.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Here P(Y) is called prior probability which means it is the probability of an event before the evidence P(Y|X) is called the posterior probability i.e., Probability of an event after the evidence is seen.

Assumptions Made by Naïve Bayes

- The fundamental Naïve Bayes assumption is that each feature makes an: independent and equal contribution to the outcome.

Naïve Bayes Example

Consider the car theft problem with attributes Color, Type, Origin, and the target, Stolen can be either Yes or No.

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Concerning our dataset, the concept of assumptions made by the algorithm can be understood as:

- We assume that no pair of features are dependent. For example, the color being ‘Red’ has nothing to do with the Type or the Origin of the car. Hence, the features are assumed to be Independent.
- Secondly, each feature is given the same influence(or importance). For example, knowing the only Color and Type alone can’t predict the outcome perfectly. So none of the attributes are irrelevant and assumed to be contributing Equally to the outcome.

Suppose, we want to classify a (Red SUV Domestic) is getting stolen or not. (Note that there is no example of a Red Domestic SUV in our data set.)

Bayes theorem can be rewritten as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, X is features(Color,Type,Origin in this case)

$$X = (x_1, x_2, x_3, ..., x_n)$$

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

For all entries in the dataset, the denominator does not change, it remains static. Therefore, the denominator can be removed and proportionality can be injected.

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

The posterior probability **P(y|X)** can be calculated first using the Naïve Bayesian equation and the class with the highest posterior probability is the outcome of the prediction.

Color

	Yes	No	P(Yes)	P(No)
Red	3	2	3/5	2/5
Yellow	2	3	2/5	3/5
Total	5	5		

Type

	Yes	No	P(Yes)	P(No)
Sports	4	2	4/5	2/5
SUV	1	3	1/5	3/5
Total	5	5		

Origin

	Yes	No	P(Yes)	P(No)
Domestic	2	3	2/5	3/5
Imported	3	2	3/5	2/5
Total	5	5		

Total

Stolen		P(Yes)/P(No)
Yes	5	5/10
No	5	5/10
Total	10	

We can calculate the posterior probability P(Yes | X) i.e. P(Yes|Red, SUV, Domestic)as :

$$P(Yes|X)=P(Red|Yes)*P(SUV|Yes)*P(Domestic|Yes)*P(Yes)$$
$$=(3/5) * (1/5) * (2/5) * (5/10)$$
$$=0.024$$
$$P(No|X)=P(Red|No)*P(SUV|No)*P(Domestic|No)*P(No)$$
$$=(2/5) * (3/5) * (3/5) * (5/10)$$
$$=0.072$$

As, 0.072 > 0.024 i.e. P(No|X) > P(Yes|X)

It means given the features (RED, SUV, Domestic), gets classified as **'NO'** the car is not stolen.

Instance-based classifier – K- Nearest Neighbour Classifier

Generalization — usually refers to a ML model's ability to perform well on new unseen data rather than just the data that it was trained on.

Most Machine Learning tasks are about making predictions. Having a good performance measure on the training data is good, but insufficient; the true goal is to perform well on new instances.

There are two main approaches to generalization: instance-based learning and model-based learning.

1. Instance-based learning:

(sometimes called memory-based learning) is a family of learning algorithms that, instead of performing explicit generalization, compares new problem instances with instances seen in training, which have been stored in memory.

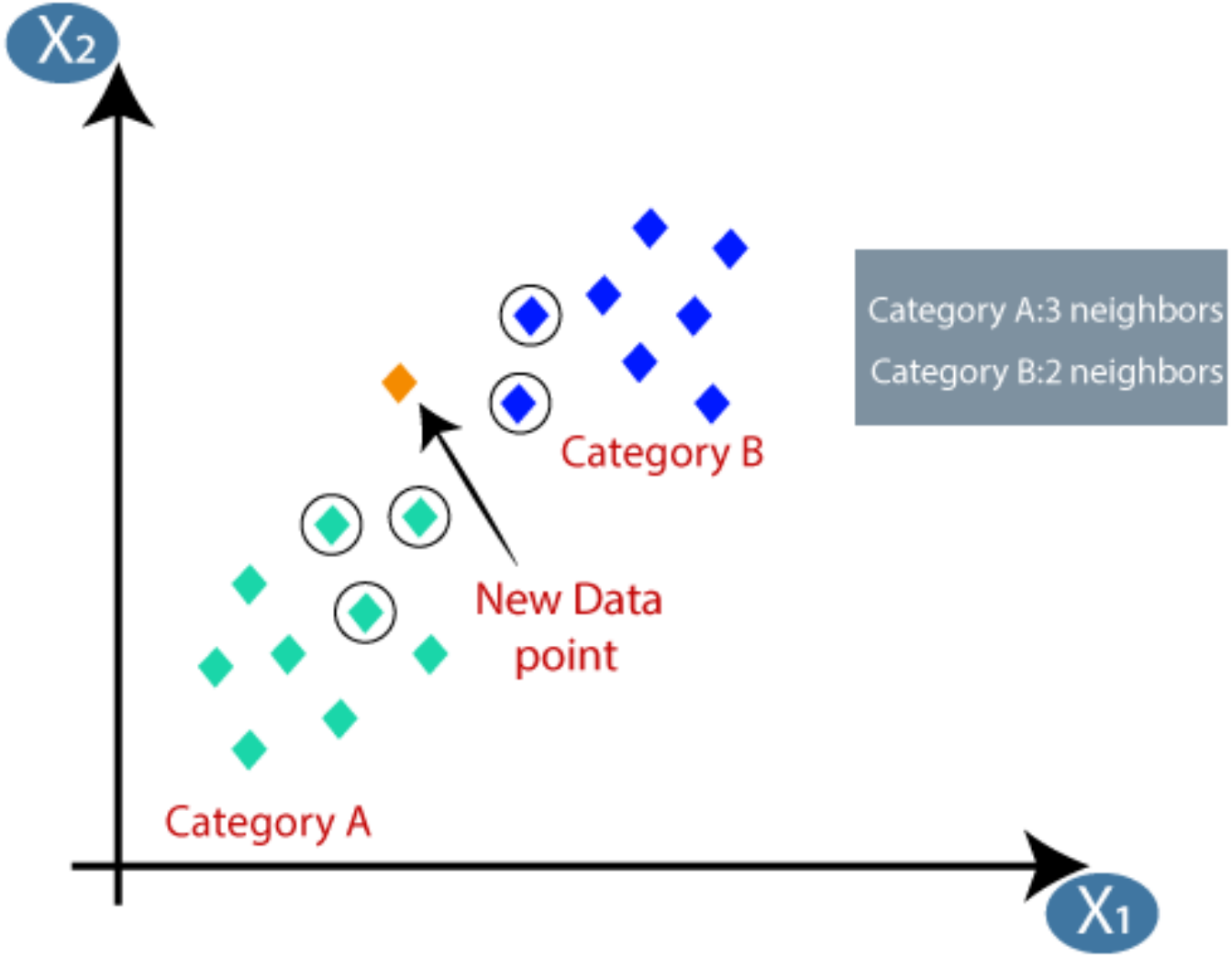
Ex- k-nearest neighbor, decision tree

K-Nearest Neighbor(KNN) Classifier

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

Here is step by step on how to compute K-nearest neighbors KNN algorithm:

1. Determine parameter K = number of nearest neighbors
2. Calculate the distance between the query-instance and all the training samples
3. Sort the distance and determine nearest neighbors based on the K-th minimum distance
4. Gather the category y of the nearest neighbors
5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance


X2
Category A:3 neighbors
Category B:2 neighbors
Category B
New Data point
Category A
X1

Euclidean distance (p=2): This is the most commonly used distance measure, and it is limited to real-valued vectors. Using the below formula, it measures a straight line between the query point and the other point being measured.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Example

We have data from the questionnaires survey (to ask people opinion) and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here is four training samples

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7. Without another expensive survey, can we guess what the classification of this new tissue is?

1. Determine parameter K = number of nearest neighbors

Suppose use K = 3

2. Calculate the distance between the query-instance and all the training samples

Coordinate of query instance is (3, 7), instead of calculating the distance we compute square distance which is faster to calculate (without square root)

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	**Square Distance to query instance (3, 7)**
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

3. *Sort the distance and determine nearest neighbors based on the K-th minimum distance*

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	**Rank minimum distance**	**Y = Category of nearest Neighbor**
7	7	$(7-3)^2+(7-7)^2=16$	**3**	**Bad**
7	4	$(7-3)^2+(4-7)^2=25$	**4**	-
3	4	$(3-3)^2+(4-7)^2=9$	**1**	**Good**
1	4	$(1-3)^2+(4-7)^2=13$	**2**	**Good**

4. *Use simple majority of the category of nearest neighbors as the prediction value of the query instance*

We have 2 good and 1 bad, since 2>1 then we conclude that a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7 is included in **Good** category.

CART Example

Past Trend	Open Interest	Trading Vol.	Return
+ve	Low	High	Up
-ve	High	Low	Down
+ve	Low	High	Up
+ve	High	High	Up
-ve	Low	High	Down
+ve	Low	Low	Down
-ve	High	High	Down
-ve	Low	High	Down
+ve	Low	Low	Down
+ve	High	High	Up

1) Gini Index for past trends :-

+ve =6 (4 up &2 down)

-ve=4 (4 Down)

P(+ve)= 6/10 P(-ve)=4/10

If(past trends= +ve & return =up)

P= 4/6

If(past trends= +ve & return =down)

P=2/6

When +ve So, Gini Index= 1-((4/6)^2+(2/6)^2)=0.45

If(past trends= -ve & return =up)

P= 0

If(past trends= -ve & return =down)

P=4/4=1

When +ve So, Gini Index= 1-((0)^2+(1)^2)=0

There fore,

Gini Index for past trends

= (6/10*0.45+ 4/10*0)

= 0.27

2) **Gini Index for Open Interest:-**

Low=6 (4 up & 2 down)
High=4 (2 up & 2 down)
P(High)= 4/10 P(Low)=6/10
If(Open Interest = high & return =up)
P= 2/4
If(Open Interest = high& return =down)
P=2/4
When High So, Gini Index= 1-((2/4)^2+(2/4)^2)=0.5
If(Open Interest = low & return =up)
P= 2/6
If(Open Interest = low & return =down)
P=4/6
When Low So, Gini Index= 1-((2/6)^2+(4/6)^2)=0.45

There fore,

Gini Index for Open Interest
= (4/10*0.5+ 6/10*0.45)
= 0.47

- 3) **Gini Index for Open Interest**:-

Low=6 (4 up & 2 down)

High=4 (2 up & 2 down)

P(High)= 4/10 P(Low)=6/10

If(Open Interest = high & return =up)

P= 2/4

If(Open Interest = high& return =down)

P=2/4

When High So, Gini Index= 1-((2/4)^2+(2/4)^2)=0.5

If(Open Interest = low & return =up)

P= 2/6

If(Open Interest = low & return =down)

P=4/6

When Low So, Gini Index= 1-((2/6)^2+(4/6)^2)=0.45

There fore,

Gini Index for Open Interest

= (4/10*0.5+ 6/10*0.45)

= 0.47

4) **Gini Index for Trading Volume**:-

Low=3 (3 down)

High=7 (4 up & 3 down)

P(High)= 7/10 P(Low)=3/10

If(Trading Volume = high & return =up)

P= 4/7

If(Trading Volume = high& return =down)

P=3/7

When High So, Gini Index= 1-((4/7)^2+(3/7)^2)=0.49

If(Trading Volume = low & return = down)

P= 3/3=1

When Low So, Gini Index= 1-1=0

There fore,

Gini Index for Trading Volume =7/10*0.49+3/10*0

=0.34

- Past trend has lowest Gini Index so chosen as a Root node.
- Repeat the same procedure to determine sub-nodes of decision tree.