NAÏVE BAYES THEORM

NAÏVE BAYES THEORM(Introduction):

ML Labs Pvt Ltd

- Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem
- Naive Bayes is a machine learning algorithm for classification problems. It is based on Bayes' probability theorem.
- It is primarily used for text classification which involves high dimensional training data sets. A few examples are spam filtration, sentimental analysis, and classifying news articles.

NAÏVE BAYES THEORM(Introduction):

ML Labs Pvt Ltd

- It is not only known for its simplicity, but also for its effectiveness. It is fast to build models and make predictions with Naive Bayes algorithm.
- Naive Bayes is the first algorithm that should be considered for solving text classification problem.

What is Naive Bayes algorithm?

ML Labs Pvt Ltd

- **Naive Bayes algorithm** is the algorithm that learns the probability of an object with certain features belonging to a particular **group/class**. In short, it is a probabilistic classifier. You must be wondering why is it called so?
- The Naive Bayes algorithm is called "**naive**" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features.

What is Naive Bayes algorithm?

ML Labs Pvt Ltd

- For instance, if you are trying to identify a fruit based on its color, shape, and taste, then an orange colored, spherical, and tangy fruit would most likely be an orange. Even if these features depend on each other or on the presence of the other features, all these properties individually contribute to the probability that this fruit is an orange and that is why it is known as “naive.”

What is Naive Bayes algorithm?

ML Labs Pvt Ltd

- As for the “Bayes” part, it refers to the statistician and philosopher, Thomas Bayes and the theorem named after him, Bayes’ theorem, which is the base for Naive Bayes Algorithm.

EXAMPLES:

ML Labs Pvt Ltd

To start with, let us consider a dataset.
Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit("Yes") or unfit("No") for playing golf.

EXAMPLES: Here is a tabular representation of our dataset.

ML Labs Pvt Ltd

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF	PLAY GOLF
0	Rainy	Hot	High	FALSE	No
1	Rainy	Hot	High	TRUE	No
2	Overcast	Hot	High	FALSE	Yes
3	Sunny	Mild	High	FALSE	Yes
4	Sunny	Cool	Normal	FALSE	Yes
5	Sunny	Cool	Normal	TRUE	No
6	Overcast	Cool	Normal	TRUE	Yes
7	Rainy	Mild	High	FALSE	No
8	Rainy	Cool	Normal	FALSE	Yes
9	Sunny	Mild	Normal	FALSE	Yes
10	Rainy	Mild	Normal	TRUE	Yes
11	Overcast	Mild	High	TRUE	Yes
12	Overcast	Hot	Normal	FALSE	Yes
13	Sunny	Mild	High	TRUE	No

EXAMPLES:

ML Labs Pvt Ltd

The dataset is divided into two parts, namely, **feature matrix** and the **response vector**.

Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of **dependent features.** In above dataset, features are **'Outlook', 'Temperature', 'Humidity'** and **'Windy'.**

Response vector contains the value of class variable(prediction or output) for each row of feature matrix. In above dataset, the class variable name is **'Play golf'.**

Naive Bayes assumption

ML Labs Pvt Ltd

- The fundamental Naive Bayes assumption is that each feature makes an:
  - **independent**
  - **Equal**
- **contribution to the outcome.** With relation to our dataset, this concept can be understood as:

Naive Bayes assumption

ML Labs Pvt Ltd

- We assume that no pair of features are dependent. For example, the temperature being ‘Hot’ has nothing to do with the humidity or the outlook being ‘Rainy’ has no effect on the winds. Hence, the features are assumed to be **independent**.

Naive Bayes assumption

ML Labs Pvt Ltd

- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can’t predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

Naive Bayes assumption

ML Labs Pvt Ltd

With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being ‘Hot’ has nothing to do with the humidity or the outlook being ‘Rainy’ has no effect on the winds. Hence, the features are assumed to be **independent**.

Naive Bayes assumption

ML Labs Pvt Ltd

- Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can’t predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

Naive Bayes assumption(Note):

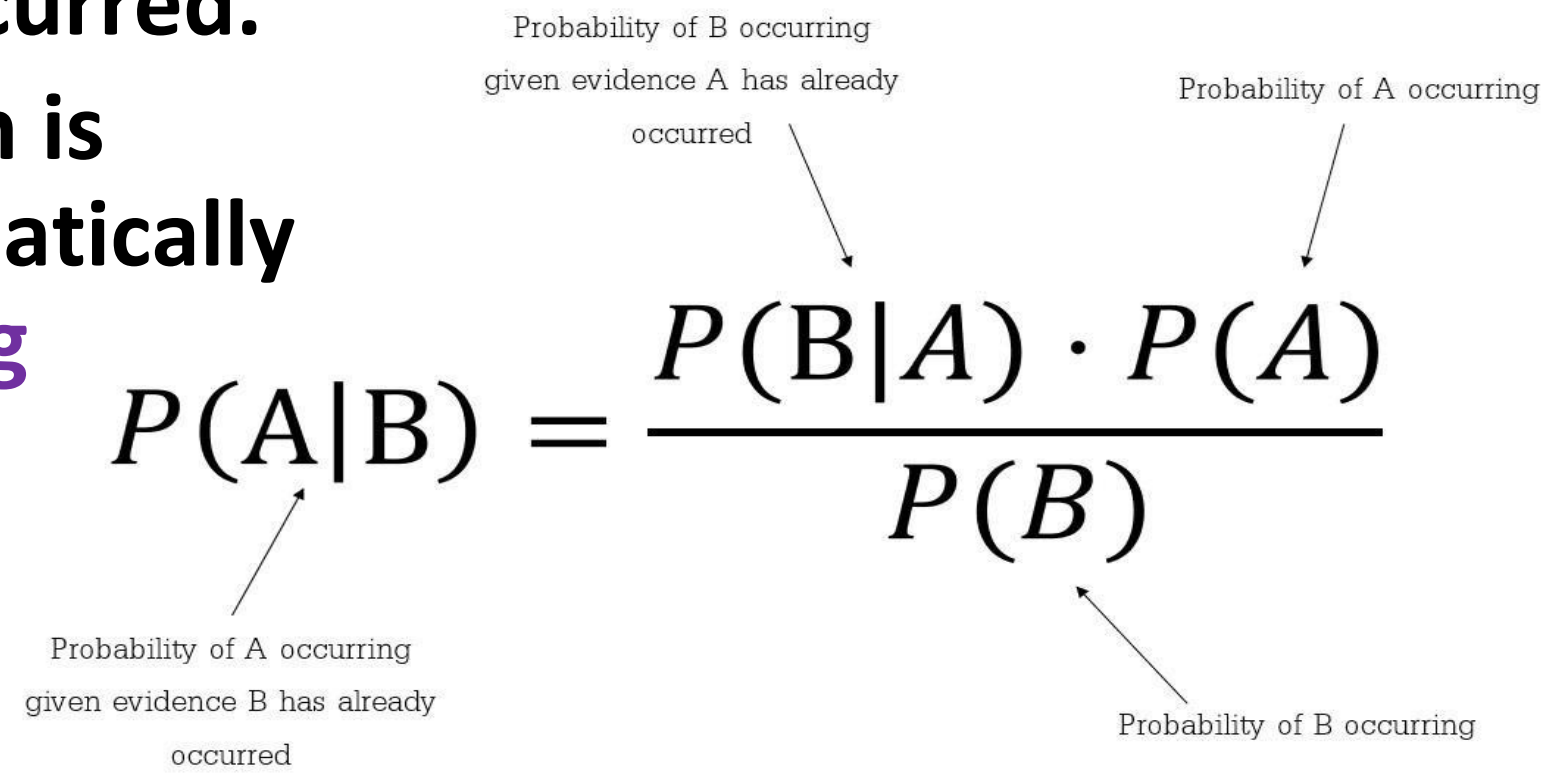
ML Labs Pvt Ltd

- **Important Note**: The assumptions made by Naive Bayes are not generally correct in real-world situations. In-fact, the independence assumption is never correct but often works well in practice.
- Now, before moving to the formula for **Naive Bayes**, it is important to know about **Bayes' theorem.**

Bayes' Theorem

ML Labs Pvt Ltd

- Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred.
- Bayes' theorem is stated mathematically as the following equation:



Bayes' Theorem:

ML Labs Pvt Ltd

where A and B are events and P(B) ?

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e., Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e., **probability of event after evidence is seen.**

Bayes' Theorem:

ML Labs Pvt Ltd

- Just to clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)
  - **X = (Rainy, Hot, High, False)**
  - **y = No**
- So basically, **P(X|y)** here means, the probability of "Not playing golf" given that the weather conditions are "Rainy outlook", "Temperature is hot", "high humidity" and "no wind".

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

Let us test it on a new set of features (let us call it today):


```
today = (Sunny, Hot, Normal, False)
```

So, probability of playing golf is given by:

$$P(Yes|today) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(today)}$$

and probability to not play golf is given by:

$$P(No|today) = \frac{P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)}{P(today)}$$

Since, P(today) is common in both probabilities, we can ignore P(today) and find proportional probabilities as:

$$P(Yes|today) \propto \frac{2}{9}.\frac{2}{9}.\frac{6}{9}.\frac{6}{9}.\frac{9}{14} \approx 0.0141$$

and

$$P(No|today) \propto \frac{3}{5}.\frac{2}{5}.\frac{1}{5}.\frac{2}{5}.\frac{5}{14} \approx 0.0068$$

Now, since

$$P(Yes|today) + P(No|today) = 1$$

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

$$P(Yes|today) = \frac{0.0141}{0.0141+0.0068} = 0.67$$

and

$$P(No|today) = \frac{0.0068}{0.0141+0.0068} = 0.33$$

Since

$$P(Yes|today) > P(No|today)$$

So, prediction that golf would be played is 'Yes'.

The method that we discussed above is applicable for discrete data. In case of continuous data, we need to make some assumptions regarding the distribution of values of each feature. The different naive Bayes

Gaussian Naive Bayes classifier

ML Labs Pvt Ltd

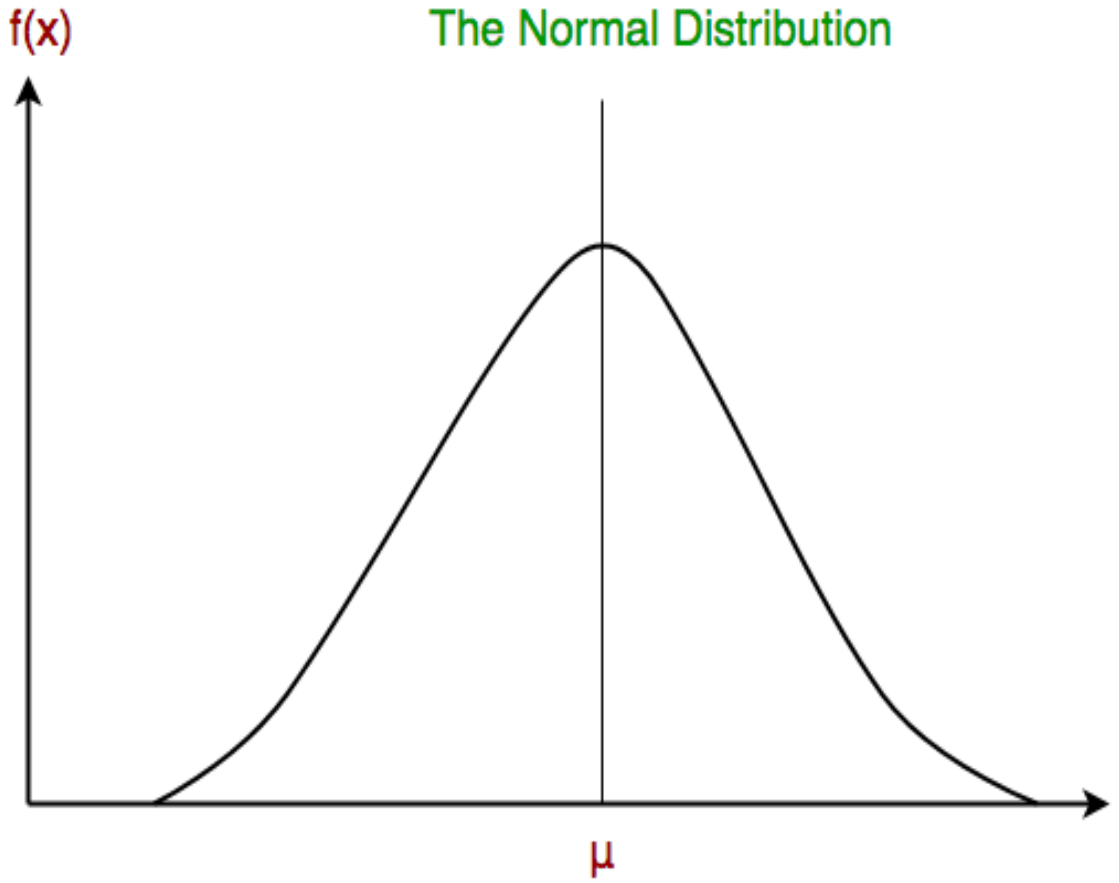
In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**. A Gaussian distribution is also called **Normal distribution**. When plotted, it gives a **bell-shaped** curve which is symmetric about the mean of the feature values as shown below:

Gaussian Naive Bayes classifier

ML Labs Pvt Ltd

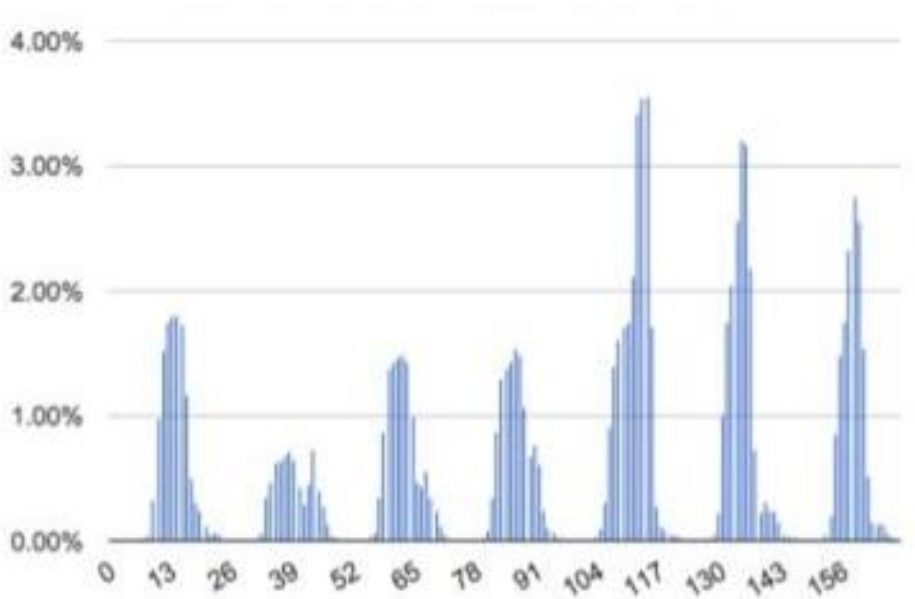
The likelihood of the features is assumed to be Gaussian; hence, conditional probability is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$



Other popular Naive Bayes classifiers are:

- **Multinomial Naive Bayes**: Feature vectors represent the frequencies with which certain events have been generated by a **multinomial distribution**. This is the event model typically used for document classification.



Other popular Naive Bayes classifiers are:

ML Labs Pvt Ltd

- **Bernoulli Naive Bayes**: In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence(i.e., a word occurs in a document or not) features are used rather than term frequencies(i.e., frequency of a word in the document).

Pros and Cons of Naive Bayes algorithm:

ML Labs Pvt Ltd

Every coin has two sides. So, does the Naive Bayes algorithm. It has advantages as well as disadvantages, and they are listed below:

Pros:

- It is a relatively easy algorithm to build and understand.
- It is faster to predict classes using this algorithm than many other classification algorithms.
- It can be easily trained using a small data set.

Pros and Cons of Naive Bayes algorithm:

ML Labs Pvt Ltd

Cons:

- If a given class and a feature have 0 frequency, then the conditional probability estimate for that category will come out as 0. This problem is known as the "**Zero Conditional Probability Problem**." This is a problem because it wipes out all the information in other probabilities too. There are several sample correction techniques to fix this problem such as "**Laplacian Correction**."

Pros and Cons of Naive Bayes algorithm:

ML Labs Pvt Ltd

Cons:

- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data sets in real life.

Applications of Naive Bayes:

ML Labs Pvt Ltd

Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations.

Applications of Naive Bayes:

ML Labs Pvt Ltd

- **Recommendation System**: Naive Bayes classifiers are used in various inferencing systems for making certain recommendations to users out of a list of possible options.
- **Real-Time Prediction**: Naive Bayes is a fast algorithm, which makes it an ideal fit for making predictions in real time.

Applications of Naive Bayes:

- **Multiclass Prediction**: This algorithm is also well-known for its multiclass prediction feature. Here, we can predict the probability of multiple classes of the target variable.
- **Sentiment Analysis**: Naive Bayes is used in sentiment analysis on social networking datasets like Twitter* and Facebook* to identify positive and negative customer sentiments.

Applications of Naive Bayes:

ML Labs Pvt Ltd

- **Text Classification**: Naive Bayes classifiers are frequently used in text classification and provide a high success rate, as compared to other algorithms.
- **Spam Filtering**: Naive Bayes is widely used in spam filtering for identifying spam email.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

An interesting point about naive Bayes is that even when the independence assumption is violated and there are clear, known relationships between attributes, it works decently anyway. There are two reasons that make naive Bayes a very efficient algorithm for classification problems.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

Performance:

- The naive Bayes algorithm gives useful performances despite having correlated variables in the dataset, even though it has a basic assumption of independence among features.
- The reason for this is that in a given dataset, two attributes may depend on each other, but the dependence may distribute evenly in each of the classes.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

Performance:

- In this case, the conditional independence assumption of naive Bayes is violated, but it is still the optimal classifier.
- Further, what eventually affects the classification is the combination of dependencies among all attributes.
- If we just look at two attributes, there may exist strong dependence between them that affects the classification.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

Performance:

- When the dependencies among all attributes work together, however, they may cancel each other out and no longer affect the classification.
- Therefore, we argue that it is the distribution of dependencies among all attributes over classes that affects the classification of naive Bayes, not merely the dependencies themselves.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

Speed:

- The main cause for the fast speed of naive Bayes training is that it converges toward its asymptotic accuracy at a different rate than other methods, like logistic regression, support vector machines, and so on.
- Naive Bayes parameter estimates converge toward their asymptotic values in order of log(n) examples, where n is number of dimensions.

Why is Naive Bayes so Efficient?

ML Labs Pvt Ltd

Speed:

- In contrast, logistic regression parameter estimates converge more slowly, requiring order n examples. It is also observed that in several datasets logistic regression outperforms naive Bayes when many training examples are available in abundance, but naive Bayes outperforms logistic regression when training data is scarce.

ML Labs Pvt Ltd