Name: Apoorv Kashyap

INTERN ID: MST03-004

Naïve Bayes Algorithm

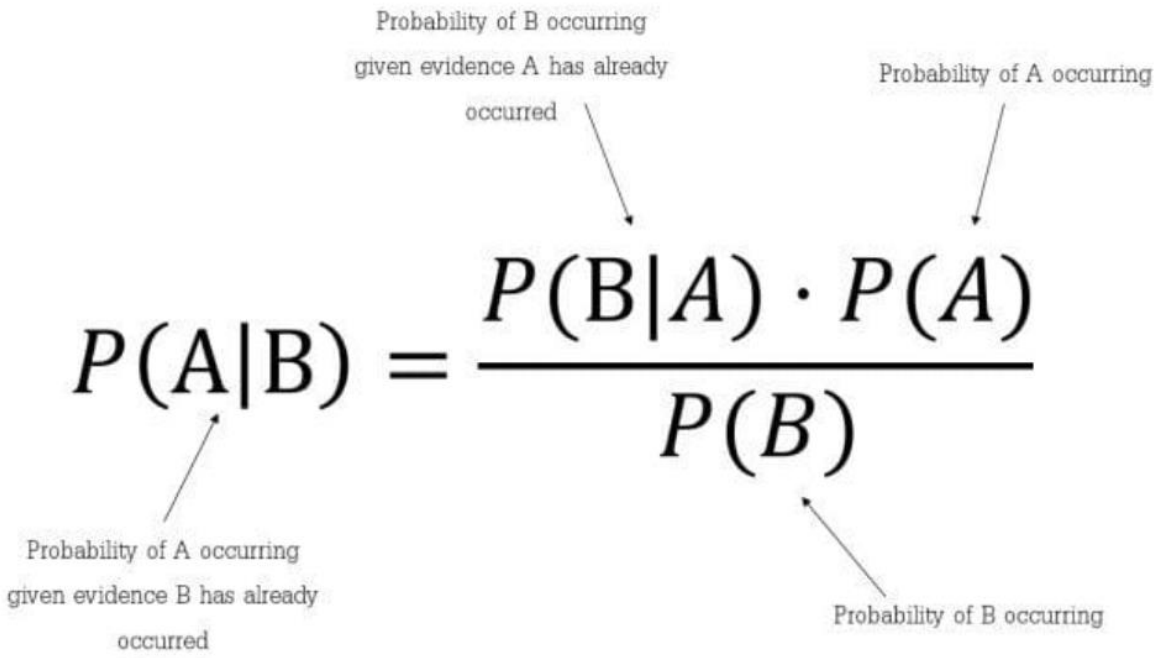
Naive Bayes is a probabilistic machine learning algorithm based on Bayes' Theorem, used primarily for classification tasks. It assumes that the features in a dataset are independent of each other, which is a strong assumption and often not true in reality, but it works well in practice for many applications.

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.

The formula is: —



Which tells us: how often A happens *given that B happens*, written **P(A|B)** also called posterior probability, When we know: how often B happens *given that A happens*, written **P(B|A)** and how likely A is on its own, written **P(A)** and how likely B is on its own, written **P(B).**

Let us apply Bayes theorem to our coin example. Here, we have two coins, and the first two probabilities of getting two heads and at least one tail are computed directly from the sample space.

Now in this sample space, let A be the event that the second coin is head, and B be the event that the first coin is tails. Again, we reversed it because we want to know what the second event is going to be.

We're going to focus on A, and we write that out as a probability of A given B:

Probability = P(A|B)

= [P(B|A) * P(A)] / P(B)

= [P(First coin being tail given the second coin is the head) * P(Second coin being

head)] / P(First coin being tail)

= [(1/2) * (1/2)] / (1/2)

= 1/2 = 0.5

Bayes theorem calculates the conditional probability of the occurrence of an event based on prior knowledge of conditions that might be related to the event.

In simpler terms, Bayes' Theorem is a way of finding a probability when we know certain other probabilities.

Types of Naïve Bayes Classifiers

1. Multinomial Naïve Bayes Classifier

Feature vectors represent the frequencies with which certain events

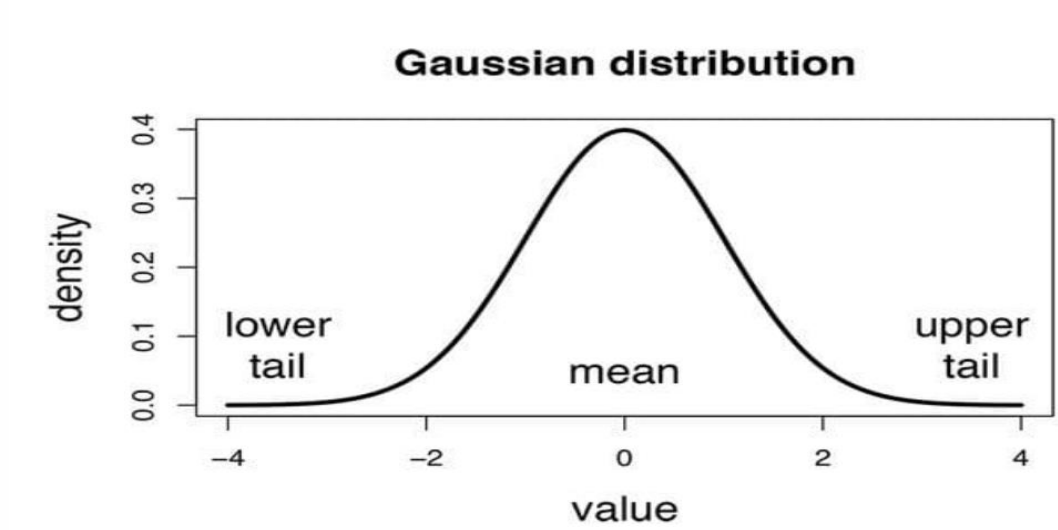
have been generated by a **multinomial distribution**. This is the event model typically used for document classification.

2. Bernoulli Naïve Bayes Classifier:

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).

3. Gaussian Naïve Bayes Classifier:

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



Where is Naive Bayes Used?

You can use Naive Bayes for the following things:

Face Recognition

As a classifier, it is used to identify the faces or its other features, like nose, mouth, eyes, etc.

Weather Prediction

It can be used to predict if the weather will be good or bad.

Medical Diagnosis

Doctors can diagnose patients by using the information that the classifier provides. Healthcare professionals can use Naive Bayes to indicate if a patient is at high risk for certain diseases and conditions, such as heart disease, cancer, and other ailments.

News Classification

With the help of a Naive Bayes classifier, Google News recognizes whether the news is political, world news, and so on.

Advantages of Naive Bayes Classifier

The following are some of the benefits of the Naive Bayes classifier:

- It is simple and easy to implement
- It doesn't require as much training data
- It handles both continuous and discrete data
- It is highly scalable with the number of predictors and data points
- It is fast and can be used to make real-time predictions
- It is not sensitive to irrelevant features

Disadvantages

- Naive Bayes assumes that all predictors (or features) are independent, rarely happening in real life. This limits the applicability of this algorithm in real-world use cases.
- This algorithm faces the 'zero-frequency problem' where it assigns zero probability to a categorical variable whose category in the test data set wasn't available in the training dataset. It would be best if you used a smoothing technique to overcome this issue.

- Its estimations can be wrong in some cases, so you shouldn’t take its probability outputs very seriously.