# Naive Bayes: Classification Algorithm

Naive Bayes is a classification algorithm based on Bayes' Theorem, with an assumption of independence among predictors, and it can be very effective in many situations, especially for text classification problems such as spam detection and sentiment analysis.

**Bayes' Theorem**

Bayes' Theorem provides a way of calculating the posterior probability, P(C|X): from P(C), P(X), and P(X|C):

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(C|X)}$$

Where:

- P(C|X) is the posterior probability of class C given predictor X.
- P(C) is the prior probability of class C.
- P(X|C) is the likelihood, which is the probability of predictor X given class C.
- P(X) is the prior probability of predictor X.

## Naive Assumption

The "naive" part of Naive Bayes comes from the assumption that the predictors (features) are independent. This means that the presence or absence of a particular feature does not affect the presence or absence of any other feature.

## Types of Naive Bayes Classifiers

There are three main types of Naive Bayes classifiers, which differ in how they handle the data distribution of features:

1. **Gaussian Naive Bayes**: Assumes that the continuous features follow a Gaussian (normal) distribution.
2. **Multinomial Naive Bayes**: Used for discrete counts, such as word frequencies in text classification.
3. **Bernoulli Naive Bayes**: Used for binary/Boolean features.

## Steps to Implement Naive Bayes

1. **Prepare the Data**: Split the data into training and testing sets.
2. **Calculate Probabilities**:
    - Compute the prior probability for each class.
    - Compute the likelihood of each feature given each class.
3. **Apply Bayes' Theorem**: Use the prior and likelihood to compute the posterior probability for each class.
4. **Make Predictions**: Choose the class with the highest posterior probability.