*Naïve Bayes*

* Naive Bayes is probabilistic classifier based on Bayes Theorem. It’s called “naïve” because it assumes independence between the features given the class label, which is rarely true in real-world data.
* P(Y/X) = P(X/Y) x P(Y)/ P(X).
* Where, P(Y/X) = Posterior probability of class Y given input features X.
* P(X/Y) = Likelihood of observing features X given class Y.
* P(Y) = Prior probability of class Y.
* P(X) = Probability of input features X (act as a normalizing constant).
* We assume feature independence, so we can break P(X/Y) into a product of individual probabilities.

|  |  |  |
| --- | --- | --- |
| Type | Used When | Assumes |
| Gaussian Naïve Bayes | Features are continuous | Features follow a normal (Gaussian Distribution) |
| Multinomial Naïve Bayes | Features are counts | Feature counts follow a multinomial distribution |
| Bernoulli Naïve Bayes | Features are Binary | Features follow a Bernoulli distribution |

* Advantages:
* Simple and fast to train.
* Works well with high-dimensional data (like text).
* Handles both binary and multiclass classification.
* Limitations:
* Assumes feature independence, which is often unrealistic.
* Can perform poorly when features are highly correlated.
* Common Applications:
* Spam detection
* Sentiment Analysis
* News Article Categorization
* Medical Diagnosis

Types:

* Gaussian Naïve Bayes
* Used when features are continuous/numeric.
* Assumes: Feature values follow a normal (gaussian) distribution.
* Example:
* Classifying whether a person is fit or unfit based on their body temperature.

|  |  |
| --- | --- |
| Temperature (F) | Fever |
| 98.6 | No |
| 101.2 | Yes |
| 99.1 | No |
| 102.4 | Yes |

* Here, temperature is continuous.
* Gaussian Naïve Bayes assumes the values for each class follow a normal (bell-curve distribution).
* It uses the mean and standard deviation of temperatures for each class to compute probabilities.
* Multinomial Naïve Bayes:
* Use when: Features are counts or frequencies – most often for text data.
* Example: Classifying emails as spam or not, based on word frequencies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Email | “free” | “win” | “meeting” | Class |
| 1 | 2 | 1 | 0 | Spam |
| 2 | 0 | 0 | 3 | Not Spam |
| 3 | 1 | 2 | 0 | Spam |

* Features are how many times each word appears.
* Multinomial NB learns the probability of each word appearing in spam vs not spam.
* It multiplies those word probabilities together per class and picks the highest score.
* Bernoulli Naïve Bayes
* Use When: Features are Binary (0 or 1) – indicating presence or absence of something.
* Example: Same spam classification, but now you just check if a word is present:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Email | “free” | “win” | “meeting” | Class |
| 1 | 1 | 1 | 0 | Spam |
| 2 | 0 | 0 | 1 | Not Spam |
| 3 | 1 | 1 | 0 | Spam |

* It only cares whether a word is present or not, not how many times.
* Bernoulli NB calculates the probability of presence/absence of each feature per class