

Naive Bayes

David Robinson

Naive Bayes

Naive Bayes is a classification algorithm based on probability.

$$P(Y = y|X = (x_1, x_2, \dots, x_n))$$

Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- $P(A|B)$ is **posterior probability**: probability of hypothesis A in the event of B .
- $P(B|A)$ is **likelihood probability**: probability of evidence B given that the hypothesis A is true.
- $P(A)$ is **prior probability**: probability of the hypothesis before observing the evidence.
- $P(B)$ is **marginal probability**: probability of evidence.

Independence in Probability

Two events are **independent** if the occurrence of one event does not affect the other.

$$P(A \cap B) = P(A) \cdot P(B)$$

Conditional Independence in Naive Bayes

In Naive Bayes, features are assumed to be independent of each other, given the class.

$$P(X_1, x_2, \dots, X_n|Y = y) = P(X_1|Y = y) \cdot P(X_2|Y = y) \cdot \dots \cdot P(X_n|Y = y)$$

By assuming conditional independence, the joint probability can be decomposed into the product of individual conditional probabilities for each feature, reducing the time complexity from $O(2^n)$ to $O(n)$.

Training Phase

1. Estimate the prior probabilities for each class from the training data.

$$P(y = y)$$

2. Estimate the conditional probabilities for each feature and each class label. For continuous features, a normal distribution is built from the mean and variance from the training data. For categorical features, the probabilities are based on the frequency of each feature value in the training data.

$$P(x_i|Y = y)$$

Prediction Phase

1. For a new instance with features, compute the posterior probability for each class.
2. Select the class that maximizes this posterior probability.

Smoothing

If a category was not observed in the training dataset, the Naive Bayes classifier assigns a zero probability to that category. The solution is to add a small value to the count of each feature combination.

Advantages of Naive Bayes Classifier

1. Fast and Efficient: Quick and easy to use for classification tasks
2. Performs Well with Limited Data: When the independence assumption holds, Naive Bayes can outperform more complex models like logistic regression
3. Effective with Categorical Data: Tends to perform better with categorical input variables as it does not rely heavily on numerical distribution assumptions
4. Handles High-Dimensional Data: Can efficiently handle high-dimensional datasets, making it ideal for tasks like text classification (e.g. spam detection)

Disadvantages of Naive Bayes Classifier

1. Assumption of Independent Predictors: In real-world scenarios, it's rare for features to be completely independent
2. Zero Frequency Problem: If a categorical variable in the test dataset contains a category not present in the training dataset, the model assigns a zero probability and fails to make a prediction
3. Sensitivity to Numerical Assumptions: Often assumes that the features follow a normal distribution even when they may not

Types of Naive Bayes Algorithms

1. Gaussian Naive Bayes: assumes that the features follow a normal distribution
2. Multinomial Naive Bayes: assumes that the data follows a multinomial distribution, which is good for discrete count data
3. Bernoulli Naive Bayes: assumes that the features are binary, meaning each predictor represents whether a specific attribute is present or absent