Naive Bayes Classifier

Introduction

The **Naive Bayes** classifier is a family of simple yet effective probabilistic classifiers based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Despite this strong assumption, Naive Bayes classifiers have been found to work well in many real-world situations, particularly in text classification and spam filtering.

1. Bayes' Theorem

Bayes' theorem describes the probability of an event based on prior knowledge of conditions that might be related to the event. Mathematically, it is expressed as:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

where:

- $P(A|B)$ is the posterior probability: the probability of event A occurring given that B is true.
- $P(B|A)$ is the likelihood: the probability of event B occurring given that A is true.
- $P(A)$ is the prior probability of event A.
- $P(B)$ is the marginal probability of event B.

2. Naive Bayes Assumption

The "naive" assumption is that the features (predictors) are independent of each other given the class. For a feature vector $X = (x_1, x_2, \ldots, x_n)$, the Naive Bayes classifier assumes:

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C) \tag{2}$$

where C is the class label.

3. Types of Naive Bayes Classifiers

- **Gaussian Naive Bayes**: Assumes that the continuous values associated with each feature are distributed according to a Gaussian (normal) distribution.
- **Multinomial Naive Bayes**: Used for discrete data like word counts in text classification.
- **Bernoulli Naive Bayes**: Used for binary/boolean features, such as in text classification where the feature set could be the presence or absence of a word.

Real-Life Example: Spam Detection

Let's consider a simplified example of a spam email classifier using a Bernoulli Naive Bayes model. The goal is to classify emails as "spam" or "not spam" based on certain keywords.

Example Dataset

Suppose we have the following training data:

Email	Contains "free"	Contains "win"	Contains "offer"	Spam/Not Spam
1	Yes	No	Yes	Spam
2	No	Yes	No	Not Spam
3	Yes	No	No	Not Spam
4	Yes	Yes	Yes	Spam
5	No	Yes	No	Spam

From this dataset, we can compute the following probabilities:

- **Prior Probabilities**:

$$P(\text{Spam}) = \frac{3}{5} = 0.6$$
$$P(\text{Not Spam}) = \frac{2}{5} = 0.4$$

- **Likelihood Probabilities** (assuming word presence or absence is independent given the class):

$$P(\text{free} = \text{Yes}|\text{Spam}) = \frac{2}{3} \approx 0.67$$
$$P(\text{free} = \text{Yes}|\text{Not Spam}) = \frac{1}{2} = 0.5$$
$$P(\text{win} = \text{Yes}|\text{Spam}) = \frac{2}{3} \approx 0.67$$
$$P(\text{win} = \text{Yes}|\text{Not Spam}) = \frac{1}{2} = 0.5$$
$$P(\text{offer} = \text{Yes}|\text{Spam}) = \frac{2}{3} \approx 0.67$$
$$P(\text{offer} = \text{Yes}|\text{Not Spam}) = 0$$

Classifying a New Email

Consider a new email with the following features:

- Contains "free": Yes
- Contains "win": Yes
- Contains "offer": No

We need to calculate the posterior probabilities for both classes (Spam and Not Spam).

Step-by-Step Calculation

1. Posterior Probability for Spam:

$$P(\text{Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) \propto P(\text{free} = \text{Yes}|\text{Spam}) \times P(\text{win} = \text{Yes}|\text{Spam}) \times P(\text{offer} = \text{No}|\text{Spam}) \times P(\text{Spam})$$

$$P(\text{Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) \propto 0.67 \times 0.67 \times 0.33 \times 0.6 = 0.0885$$

2. Posterior Probability for Not Spam:

$$P(\text{Not Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) \propto P(\text{free} = \text{Yes}|\text{Not Spam}) \times P(\text{win} = \text{Yes}|\text{Not Spam}) \times P(\text{offer} = \text{No}|\text{Not Spam}) \times P(\text{Not Spam})$$

$$P(\text{Not Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) \propto 0.5 \times 0.5 \times 1 \times 0.4 = 0.1$$

3. Normalization:
Normalize the probabilities to get a proper probability distribution:

$$P(\text{Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) = \frac{0.0885}{0.0885 + 0.1} = \frac{0.0885}{0.1885} \approx 0.469$$

$$P(\text{Not Spam}|\text{free} = \text{Yes}, \text{win} = \text{Yes}, \text{offer} = \text{No}) = \frac{0.1}{0.0885 + 0.1} = \frac{0.1}{0.1885} \approx 0.531$$

Conclusion

Since $P(\text{Not Spam}) > P(\text{Spam})$, we classify the new email as "Not Spam."

Key Takeaways

- **Simplicity and Efficiency**: Naive Bayes is simple and computationally efficient.
- **Works Well with High-Dimensional Data**: Especially effective in text classification problems.
- **Independence Assumption**: Despite its simplicity, the assumption that features are independent given the class often works well in practice.