Naive Bayes is a fundamental classification technique in machine learning, known for its simplicity and effectiveness. It is based on applying Bayes' theorem with strong independence assumptions between features. Despite its simplicity, Naive Bayes often performs well in a variety of applications, particularly in text classification.

Bayes' Theorem

Bayes' theorem is a mathematical formula used for calculating conditional probabilities. It provides a way to update the probability estimate for a hypothesis as more evidence or information becomes available.

The theorem is expressed as:

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

Where:

- P(C∣X)P(C|X)P(C∣X) is the posterior probability of class CCC given feature vector XXX.
- P(X∣C)P(X|C)P(X∣C) is the likelihood of feature vector XXX given class CCC.
- P(C)P(C)P(C) is the prior probability of class CCC.
- P(X)P(X)P(X) is the probability of feature vector XXX.

Naive Bayes Assumption

The "naive" aspect of Naive Bayes classifiers stems from the assumption that the features are conditionally independent given the class label. This simplifies the computation of the likelihood P(X∣C)P(X|C)P(X∣C) as follows:

P(X∣C)=P(x1,x2,…,xn∣C)≈∏i=1nP(xi∣C)P(X|C) = P(x_1, x_2, \ldots, x_n | C) \approx \prod_{i=1}^{n} P(x_i|C)P(X∣C)=P(x1,x2,…,xn∣C)≈∏i=1nP(xi∣C)

Classification Rule

Given a feature vector X=(x1,x2,…,xn)X = (x_1, x_2, \ldots, x_n)X=(x1,x2,…,xn), the Naive Bayes classifier predicts the class CCC that maximizes the posterior probability:

C^=argmaxCP(C∣X)\hat{C} = \arg\max_{C} P(C|X)C^=argmaxCP(C∣X)

Using Bayes' theorem and the independence assumption, this can be rewritten as:

C^=argmaxCP(C)⋅∏i=1nP(xi∣C)\hat{C} = \arg\max_{C} P(C) \cdot \prod_{i=1}^{n} P(x_i|C)C^=argmaxCP(C)⋅∏i=1nP(xi∣C)

Types of Naive Bayes Classifiers

1. **Gaussian Naive Bayes**: Assumes that the continuous features follow a Gaussian (normal) distribution. P(xi∣C)=12πσC2exp(−(xi−μC)22σC2)P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right)P(xi∣C)=2πσC21exp(−2σC2(xi−μC)2) where μC\mu_CμC and σC\sigma_CσC are the mean and standard deviation of the feature xix_ixi for class CCC.
2. **Multinomial Naive Bayes**: Used for discrete data, often applied in text classification where features represent word counts or frequencies. P(xi∣C)=(P(xi∣C))xie−P(xi∣C)xi!P(x_i|C) = \frac{(P(x_i|C))^{x_i} e^{-P(x_i|C)}}{x_i!}P(xi∣C)=xi!(P(xi∣C))xie−P(xi∣C)
3. **Bernoulli Naive Bayes**: Used for binary/boolean features, considering the presence or absence of a feature. P(xi∣C)=P(xi=1∣C)xi⋅(1−P(xi=1∣C))(1−xi)P(x_i|C) = P(x_i=1|C)^{x_i} \cdot (1 - P(x_i=1|C))^{(1-x_i)}P(xi∣C)=P(xi=1∣C)xi⋅(1−P(xi=1∣C))(1−xi)

Training the Classifier

Training a Naive Bayes classifier involves estimating the parameters P(C)P(C)P(C) and P(xi∣C)P(x_i|C)P(xi∣C) from the training data.

- **Prior Probability P(C)P(C)P(C)**: This is estimated by the relative frequency of each class in the training set.
- **Likelihood P(xi∣C)P(x_i|C)P(xi∣C)**: This depends on the specific type of Naive Bayes classifier being used (Gaussian, Multinomial, Bernoulli).

Making Predictions

To predict the class label for a new instance with feature vector XXX, compute the posterior probability for each class and choose the class with the highest posterior probability:

C^=argmaxC(P(C)⋅∏i=1nP(xi|C))\hat{C} = \arg\max_{C} \left(P(C) \cdot \prod_{i=1}^{n} P(x_i|C) \right)C^=argmaxC(P(C)⋅∏i=1nP(xi∣C))