Naive Bayes Classifier for Text and Sentiment Classification

Dr. Sambit Praharaj, Assistant Professor (II), KIIT

1 Bayes Theorem for Text Classification

Naive Bayes is based on Bayes' theorem:

$$P(c \mid d) = \frac{P(d \mid c)\, P(c)}{P(d)}$$

where:

- c is the class (Positive, Negative)
- d is the document (sentence)

2 Why $P(d)$ Is Ignored During Classification

The term $P(d)$ represents the probability of observing the document. It does **not depend on the class** c.

During classification, we compare posterior probabilities across classes:

$$P(c_1 \mid d) \quad \text{vs} \quad P(c_2 \mid d)$$

Since $P(d)$ is the same for all classes, it cancels out:

$$\arg\max_c \frac{P(d \mid c)P(c)}{P(d)} = \arg\max_c P(d \mid c)P(c)$$

Hence, $P(d)$ is treated as a normalization constant and ignored. In many explanations, this simplification is informally written as $P(d) = 1$ to indicate that it does not affect class comparison.

3 Why Word-Level Modeling Is Used Only for Likelihood

3.1 Prior Probability

The prior probability represents the likelihood of a class before observing any text:

$$P(c) = \frac{\text{number of documents in class } c}{\text{total documents}}$$

It depends only on class frequencies and therefore does **not involve word-level modeling**.

3.2 Likelihood Probability

The likelihood models how a document is generated given a class:

$$P(d \mid c) = P(w_1, w_2, \ldots, w_n \mid c)$$

Using the Naive Bayes conditional independence assumption:

$$P(d \mid c) \approx \prod_{i=1}^{n} P(w_i \mid c)$$

Word-level modeling is applied here because documents are composed of words, and sentiment information is carried by individual words.

4 Training Corpus

Sentence	**Class**
I love this movie	Positive
This movie is bad	Negative
I hate this film	Negative

5 Vocabulary Construction

All unique words in the corpus:

$$V = \{i, love, this, movie, is, bad, hate, film\}$$

$$|V| = 8$$

6 Prior Probabilities

Total documents = 3

$$P(\text{Positive}) = \frac{1}{3}, \quad P(\text{Negative}) = \frac{2}{3}$$

7 Word Counts Per Class

7.1 Positive Class

Sentence: *I love this movie*

- i: 1
- love: 1
- this: 1
- movie: 1

$$N_{\text{pos}} = 4$$

7.2 Negative Class

Sentences:

- This movie is bad
- I hate this film

- this: 1
- movie: 1
- is: 1
- bad: 1
- i: 1
- hate: 1
- film: 1

$$N_{\text{neg}} = 7$$

8 Likelihood Estimation with Laplace Smoothing

$$P(w \mid c) = \frac{\text{count}(w, c) + 1}{N_c + |V|}$$

9 Case 1: "Movie was great"

9.1 Tokenization

$$\{\text{movie, was, great}\}$$

Words "was" and "great" are unseen in the training corpus.

9.2 Likelihoods

Positive Class

$$P(movie|Pos) = \frac{2}{12},\ P(was|Pos) = \frac{1}{12},\ P(great|Pos) = \frac{1}{12}$$

Negative Class

$$P(movie|Neg) = \frac{2}{15},\ P(was|Neg) = \frac{1}{15},\ P(great|Neg) = \frac{1}{15}$$

9.3 Posterior Probabilities

Positive

$$P(Pos|d) \propto \frac{1}{3} \times \frac{2}{12} \times \frac{1}{12} \times \frac{1}{12} = \frac{2}{5184}$$

Negative

$$P(Neg|d) \propto \frac{2}{3} \times \frac{2}{15} \times \frac{1}{15} \times \frac{1}{15} = \frac{4}{10125}$$

9.4 Prediction

$$\boxed{\text{Negative}}$$

10 Case 2: “Movie was very good”

10.1 Tokenization

$$\{\text{movie, was, very, good}\}$$

10.2 Likelihoods

Positive Class

$$P(movie|Pos) = \frac{2}{12},\ P(was|Pos) = \frac{1}{12},\ P(very|Pos) = \frac{1}{12},\ P(good|Pos) = \frac{1}{12}$$

Negative Class

$$P(movie|Neg) = \frac{2}{15},\ P(was|Neg) = \frac{1}{15},\ P(very|Neg) = \frac{1}{15},\ P(good|Neg) = \frac{1}{15}$$

10.3 Posterior Probabilities

Positive

$$P(Pos|d) \propto \frac{1}{3} \times \frac{2}{12} \times \left(\frac{1}{12}\right)^3 = \frac{2}{62208}$$

Negative

$$P(Neg|d) \propto \frac{2}{3} \times \frac{2}{15} \times \left(\frac{1}{15}\right)^3 = \frac{4}{253125}$$

10.4 Prediction

$$\boxed{\text{Positive}}$$

11 Key Takeaways

- $P(d)$ is a normalization constant and does not affect class comparison.
- Prior probability depends only on class distribution.
- Word-level modeling is applied only to likelihood.
- Laplace smoothing handles unseen words.
- Naive Bayes relies on word frequencies, not semantic meaning.

12 Conclusion

Naive Bayes separates class frequency information (prior) from word evidence (likelihood) while ignoring the document probability term that does not influence classification. This makes it simple, efficient, and effective for sentiment classification.