

# Applying Bayes' Theorem in NLP: Sentiment Analysis

## Naive Bayes Classification with a Worked Example

Syed Hamed Raza

UMT Lahore

December 2, 2025

# Outline

- 1 Motivation
- 2 Bayesian Foundations
- 3 Dataset Creation
- 4 Computing Probabilities
- 5 Worked Example: Classifying a Test Sentence
- 6 Conclusion

# Why Naive Bayes for NLP?

- Simple, efficient probabilistic classifier for text.
- Based on Bayes' Theorem with conditional independence.
- Performs well on small datasets and sparse features.
- Strong baseline for sentiment analysis tasks.
- Useful for modeling student feedback in educational settings.

# Bayes' Theorem

$$P(C | d) = \frac{P(d | C) P(C)}{P(d)}$$

- $C$ : class (Positive, Negative)
- $d$ : document (bag of words)
- Goal: choose class with highest posterior probability.

# Naive Independence Assumption

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

- Words assumed conditionally independent.
- Works surprisingly well despite simplification.
- Very fast to train and apply.

# Mini Corpus for Sentiment Analysis

- We build a small labeled dataset:

<b>ID</b>	<b>Sentence</b>	<b>Label</b>
D1	I love this movie	pos
D2	This film is excellent and inspiring	pos
D3	What a terrible and boring movie	neg
D4	I hate this film	neg
D5	An excellent and enjoyable movie	pos
D6	This movie is boring and slow	neg

# Preprocessing Pipeline

- Lowercasing and tokenization.
- Removing stopwords: {this, is, a, and, what, i}.
- Final tokens:
  - D1: love, movie
  - D2: film, excellent, inspiring
  - D3: terrible, boring, movie
  - D4: hate, film
  - D5: an, excellent, enjoyable, movie
  - D6: movie, boring, slow

# Class Priors

- 3 positive documents
- 3 negative documents

$$P(pos) = 0.5 \quad P(neg) = 0.5$$

# Word Counts per Class

## Positive class tokens (9 total):

- love(1), movie(2), film(1), excellent(2), inspiring(1), an(1),  
enjoyable(1)

## Negative class tokens (8 total):

- terrible(1), boring(2), movie(2), hate(1), film(1), slow(1)

# Laplace Smoothing

$$P(w | C) = \frac{\text{count}(w, C) + 1}{N_C + |V|}$$

Vocabulary size:  $|V| = 11$

**Positive denominator:**  $9 + 11 = 20$  **Negative denominator:**  $8 + 11 = 19$

# Test Sentence

- Raw: “This movie is excellent and enjoyable”
- After preprocessing: movie, excellent, enjoyable

# Likelihoods for Positive Class

$$P(\text{movie} \mid pos) = \frac{3}{20} = 0.15$$

$$P(\text{excellent} \mid pos) = \frac{3}{20} = 0.15$$

$$P(\text{enjoyable} \mid pos) = \frac{2}{20} = 0.10$$

$$\begin{aligned} P(pos \mid d) &\propto 0.5 \times 0.15 \times 0.15 \times 0.10 \\ &= 0.001125 \end{aligned}$$

# Likelihoods for Negative Class

$$P(\text{movie} \mid \text{neg}) = \frac{3}{19} = 0.1579$$

$$P(\text{excellent} \mid \text{neg}) = \frac{1}{19} = 0.0526$$

$$P(\text{enjoyable} \mid \text{neg}) = \frac{1}{19} = 0.0526$$

$$P(\text{neg} \mid d) \propto 0.5 \times 0.1579 \times 0.0526 \times 0.0526$$

$$\approx 0.0002187$$

# Final Prediction

$$P(pos \mid d) = 0.001125 > 0.0002187 = P(neg \mid d)$$

## Classification Result

The sentence is predicted as **Positive**.

# Key Takeaways

- Naive Bayes uses Bayes' theorem with independence assumption.
- Works well for text classification tasks.
- Simple to compute, even with tiny datasets.
- Strong baseline for sentiment analysis in educational NLP.