Naive Bayes Classification

A Comprehensive Guide with Worked Examples and Comparisons

Example: Classifying Email as 'Spam' or 'Not Spam'

Suppose we want to classify an email based on whether the words **"free"** and **"win"** appear. We'll use Bernoulli Naive Bayes where features are binary (word present or not).

Training Data Summary

Class	P(Class)	P(free=1 | Class)	P(win=1 | Class)
Spam	0.4	0.6	0.7
Not Spam	0.6	0.1	0.05

Email to Classify

The email contains both words: **free = 1** and **win = 1**.

Computing Posterior Probabilities

For Spam:

$P(\text{Spam} \mid \text{free}=1, \text{win}=1) \propto P(\text{Spam}) \times P(\text{free}=1 \mid \text{Spam}) \times P(\text{win}=1 \mid \text{Spam})$

$= 0.4 \times 0.6 \times 0.7 =$ **0.168**

For Not Spam:

$P(\text{Not Spam} \mid \text{free}=1, \text{win}=1) \propto P(\text{Not Spam}) \times P(\text{free}=1 \mid \text{Not Spam}) \times P(\text{win}=1 \mid \text{Not Spam})$

$= 0.6 \times 0.1 \times 0.05 =$ **0.003**

Prediction: Since the Spam probability (0.168) ■ Not Spam probability (0.003), the email is classified as **spam**.

Why Use Naive Bayes Instead of LDA or QDA?

LDA (Linear Discriminant Analysis) and QDA (Quadratic Discriminant Analysis) are also probabilistic classifiers, but they make different assumptions about the data. Here's a comprehensive comparison:

1. Feature Independence vs. Covariance Structure

Naive Bayes: Assumes features are **independent** given the class. Ignores covariances entirely, resulting in a much simpler model.

LDA: Assumes each class has a multivariate Gaussian distribution with a **shared covariance matrix** across all classes. Decision boundary is linear.

QDA: Same Gaussian assumption, but each class gets its **own covariance matrix**. Decision boundary is quadratic.

✓ **Why NB might be preferred:** When the covariance structure is complex, high-dimensional, or hard to estimate, Naive Bayes avoids needing covariance estimates entirely.

2. Performance in High Dimensions

Naive Bayes: Works extremely well with tens of thousands of sparse features (e.g., word counts in text). Parameters grow *linearly* with the number of features—very scalable. Covariance estimation is unnecessary and would be unstable.

LDA/QDA: Require estimating covariance matrices (LDA: one shared matrix; QDA: one per class). In high dimensions (p), this requires $O(p^2)$ parameters—often impossible with limited data.

✓ **NB dominates** when data is high-dimensional and sparse (NLP, bag-of-words).

3. Robustness to Model Misspecification

Even though the independence assumption is false for real data, Naive Bayes often performs surprisingly well due to the independence assumption simplifying likelihoods and reducing variance (simpler model = lower estimation error).

LDA/QDA can suffer badly if data is not actually Gaussian or covariances differ in complicated ways.

✓ **NB is more robust** when Gaussian assumptions are violated.

4. Small Training Data

Naive Bayes requires very little data to fit—only class priors and feature conditional probabilities (univariate).

LDA/QDA need means and covariances (many parameters!). Especially QDA needs large samples per class; otherwise covariance matrices become singular.

✓ **NB is better** for small datasets or imbalanced cases.

5. Computational Efficiency

Model	Training Cost	Prediction Cost
Naive Bayes	Very fast	Very fast
LDA	Moderate (invert shared covariance)	Fast
QDA	High (one covariance per class)	Moderate

✓ **NB is the fastest,** making it ideal for real-time or large-scale applications.

Summary: When to Prefer Naive Bayes

Use Naive Bayes instead of LDA/QDA when:

- Features are **high-dimensional** or **sparse** (e.g., text classification)
- You have **small datasets**
- You want **simple**, **fast**, and **robust** performance
- Gaussian assumptions of LDA/QDA don't hold
- Covariance estimation would be unreliable

Key Takeaway: Naive Bayes excels in high-dimensional, sparse data scenarios (like text classification) where its simplicity, speed, and low data requirements outweigh its naive independence assumption. Despite this assumption being violated in practice, it often achieves competitive or superior performance compared to more complex methods.