

Project

November 19, 2025

0.0.1 What is Naive Bayes?

Naive Bayes is a **probabilistic classifier** based on Bayes' theorem:

$$P(Y | X) = \frac{P(X | Y) P(Y)}{P(X)}$$

It makes the **conditional independence assumption**, meaning that given the class ($Y = c$), all features (x_1, \dots, x_d) are independent:

$$P(X | Y = c) = \prod_{j=1}^d P(x_j | Y = c)$$

0.0.2 Why “Gaussian”?

For continuous features, Gaussian Naive Bayes assumes each feature follows a **normal (Gaussian) distribution** under each class:

$$x_j | Y = c \sim \mathcal{N}(\mu_{jc}, \sigma_{jc}^2)$$

0.0.3 Steps of the Algorithm

1. Estimate priors:

The prior probability of each class (c) is:

$$P(Y = c) = \frac{\text{count of class } c}{N}$$

2. Estimate mean and variance for every feature (j) and every class (c).

3. Prediction:

Compute the **posterior probability** for each class:

$$P(Y = c | X) \propto P(Y = c) \prod_{j=1}^d P(x_j | Y = c)$$

Then **choose the class with the maximum posterior probability**.

0.0.4 Advantages

- Extremely fast training
- Closed-form parameter estimates
- Works well with small datasets
- No gradient descent required
- Robust to irrelevant features
- Strong baseline classifier

0.0.5 Disadvantages

- Assumes independence — rarely true in real data
- Assumes features follow a normal distribution — often incorrect
- Performs poorly when features are highly correlated
- Decision boundaries are quadratic, which may limit flexibility

0.0.6 Representation

Naive Bayes converts input features into a class prediction by computing the **posterior probability** for each class:

$$P(Y = c \mid X = x_1, \dots, x_d) \propto P(Y = c) \prod_{j=1}^d \mathcal{N}(x_j \mid \mu_{jc}, \sigma_{jc}^2)$$

The classifier predicts the class with the highest posterior.

In practice, we work with **log probabilities** (to avoid numerical underflow):

$$\hat{y} = \arg \max_c \left[\log P(Y = c) + \sum_{j=1}^d \log \mathcal{N}(x_j; \mu_{jc}, \sigma_{jc}^2) \right]$$

We use **log probabilities for numerical stability** because multiplying many small Gaussian likelihoods can lead to extremely tiny numbers that computers cannot represent reliably. Summing logs avoids this problem.

0.0.7 Loss Function

Important: Naive Bayes does **not** minimize a traditional loss like MSE or cross-entropy using gradient descent.

Instead, the model is trained by **maximizing the likelihood** of the data. Equivalently, the loss is the **negative log-likelihood**:

$$L = - \sum_{i=1}^N \log P(y^{(i)} | x^{(i)})$$

Gaussian Naive Bayes maximizes the likelihood under the assumption that each feature is normally distributed for each class.

The parameters come directly from **Maximum Likelihood Estimation (MLE)**:

Mean estimate:

$$\mu_{jc} = \frac{1}{N_c} \sum_{i:y_i=c} x_{ij}$$

Variance estimate:

$$\sigma_{jc}^2 = \frac{1}{N_c} \sum_{i:y_i=c} (x_{ij} - \mu_{jc})^2$$

Because the parameters have **closed-form MLE solutions**,
No gradient descent, No iterative optimization is required during training.

0.0.8 Optimizer

Naive Bayes does **not** use an iterative optimizer such as gradient descent.

Gaussian Naive Bayes has **closed-form Maximum Likelihood Estimates (MLE)** for all parameters.

Prior probabilities:

$$\hat{P}(Y = c) = \frac{N_c}{N}$$

Gaussian parameters for each feature (j) and class (c):

- Mean: μ_{jc}
- Variance: σ_{jc}^2

These are computed **directly from the data** using simple MLE formulas.
No optimization loop is required.

0.0.9 Citations

- Collins, M. (2002). The Naive Bayes model, maximum-likelihood estimation, and the EM algorithm (Technical Report). Columbia University.
- Rish, I. (2001). An empirical study of the naive Bayes classifier. IBM T.J. Watson Research Center.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. Machine Learning, 29(2–3), 131–163.

- Zaidi, N. A., Cerquides, J., Carman, M. J., & Webb, G. I. (2013). Alleviating Naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, 14, 1947-1988.
- Ahmed, M. S., Shahjaman, M., Rana, M. M., & Mollah, M. N. H. (2017). Robustification of Naïve Bayes Classifier and Its Application for Microarray Gene Expression Data Analysis. *BioMed research international*, 2017, 3020627.
- John, G. H., & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers [Preprint]. arXiv.

[]:

Overview of [the name of your ML algorithm] (20 points)

Give an overview of the algorithm and describe its advantages and disadvantages.

Representation: describe how the feature values are converted into a single number prediction.

Loss: describe the metric used to measure the difference between the model's prediction and the target variable.

Optimizer: describe the numerical algorithm used to find the model parameters that minimize the loss given a training set.

Use markdown in the jupyter notebook, add equations to explain math, and use pseudo-code to explain how numerical algorithms work. Use citations and references. Use at least 500 words (excluding equations and pseudo-code).

ta email - I've seen that you already completed the first step which is to choose what ML algo you want to work on, that's great! As you can read in the final project rubric, the next step is to complete the markdown section of the report and make sure everyone understands the math and numerical methods behind the algorithm.

It would be great if you could finish this by the end of this current week, in order to schedule a meeting next week to validate that you are on the right track!

Gaussian Naive Bayes for classification We cover the Naive Bayes algorithm for categorical (binary) features (Chapter 24.0 and 24.1 in the textbook). Gaussian Naive Bayes is an extension of the method to continuous features. You can read more about this algorithm here. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB

https://www-cambridge-org.revproxy.brown.edu/core/services/aop-cambridge-core/content/view/ABD3A52A2171432702023317201AC255/9781107298019c24_p295-308_CBO.pdf/generative_models.pdf