Bayes Theorem

Naive Bayes is built on Bayes' Theorem:

$$P(C \mid A) = \frac{P(A \mid C)\,P(C)}{P(A)}$$

In words:

- Posterior = "Probability of class C given attributes"
- Likelihood = How likely the attributes are if the class were C
- Prior = How common the class is overall
- Evidence = Just a normalizing constant

This is the mathematical engine Naive Bayes uses for classification.

Bayesian Classifiers (general idea)

The goal in classification is: given a new record with attributes (A_1, A_2, …, A_m), choose the class C that has the highest posterior probability:

$$\max_C P(C \mid A_1, A_2, ..., A_m)$$

So classification becomes ….. predict the class with the largest:

$$P(A_1, ..., A_m \mid C)\,P(C)$$

The "Naive" Assumption

The hard part is $P(A_1 \ldots A_m|C)$ — this is a giant joint distribution. All attributes are conditionally independent given the class. So……

$$P(A_1, ..., A_m \mid C) = \prod_{j=1}^{m} P(A_j \mid C)$$

Why is this naive: because in real life attributes are not independent ……. but the classifier still works surprisingly well in many settings (especially text classification).

Naive Bayes Steps
To classify a new record:

1. Compute P(C) for each class
   (just count how often each class appears in the training data)

2. For each attribute value, compute $P(Aj = aj \mid C)$
   (counts, or Gaussian pdf for continuous)

3. Multiply them: $score(c) = P(C) \cdot \prod_j P(Aj \mid C)$

4. Pick the class with the highest score.

Estimating Probabilities From Data
Naive Bayes needs two types of probabilities:

1. P(C) — priors ….. just count how many rows belong to each class.
Class column: Yes appears 3 times | No appears 7 times ……. P(Yes)=3/10, P(No) = 7/10

2. P(Aj | C) — attribute likelihoods
For categorical attributes (like “Married”, “Single”), you simply count within the class:

- Ex: P(marital status = single | Class = yes)
- For class Yes, look only at the rows with Yes.
- Among those rows, count how many have Marital Status = “Single”.

Continuous Attributes
Naive Bayes handles continuous (numeric) features like Income differently….. two approaches:

1. Binning
   - Break income into ranges like <80k, 80–120k, etc.
   - Treat each bin as a category.
   - Downsides: loses information and bins become correlated.

2. Assume a distribution (usually Gaussian)
   - Assume Aj | C ~ N(μ, σ^2)
   - Estimate:
     - mean from the rows with that class
     - variance from the rows with that class
   - Then compute the Gaussian pdf at the new value (like income = 120k).
     - The pdf becomes P(Aj = a | C)

Multiplying Everything Together

For the test record in the lecture: X = (Refund=No, Status = married, Income=120k)
Compute:

- P(refund = no | C)
- P(status = married | C)
- P(income = 120k | C)
- Multiply those three
- Multiply this by P(C)
- Do this for all classes of C (yes and no) and compare

Ex results:

- P(X | no) = 0.024
- P(C | yes) = 0

Zero-Probability Problem (Laplace Smoothing)
If one conditional probability is zero, the entire product becomes zero, which is bad.

Solution: Laplace estimate

Instead of $$P(A = a \mid C) = \frac{N_{ac}}{N_c}$$ use $$P(A = a \mid C) = \frac{N_{ac} + 1}{N_c + k}$$ where **k** = number of possible attribute values. This avoids the “zero wipes everything out” issue.

Naive Bayes Limitations

- Independence assumption rarely true
- Works very well for text, but not always for correlated features
- Zero-probability problem (fixed by smoothing)
- Works best when attributes are informative and independent-ish