Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India [*]

January 2025

Naïve Bayes Classifier

1 Probability and Related Concepts

1.1 Probability

Probability is a measure of the likelihood of an event occurring. It is defined mathematically as:

$$P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes}}$$

Example: If a fair coin is flipped, the probability of getting heads is:

$$P(\text{Heads}) = \frac{1}{2}$$

1.2 Joint Probability

Joint Probability refers to the probability of two events, A and B, happening simultaneously. It is represented as $P(A \cap B)$ or $P(A, B)$.

Example-1: Consider rolling two dice. The probability of getting a 3 on the first die (A) and a 4 on the second die (B) is:

$$P(A \cap B) = P(A) \times P(B) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$$

Example-2: Picking Multi-Color Balls

Ball Color	Weight (Light/Heavy)
Green	Light
Green	Heavy
Green	Light
Blue	Heavy
Red	Heavy
Green	Heavy
Blue	Light
Green	Light
Red	Light

[*]Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

1.3 Joint Probability Calculation

Let's calculate the joint probability of drawing a green ball and having it be light.

$$P(\text{Green} \cap \text{Light}) = P(\text{Green}) \cdot P(\text{Light}|\text{Green})$$

1.3.1 Calculate Prior Probability of Green Ball

There are 9 balls in total, with 5 green balls. Thus, the prior probability of picking a green ball is:

$$P(\text{Green}) = \frac{5}{9}$$

1.3.2 Calculate Conditional Probability of Light Given Red

There are 3 red light balls out of 5 Green balls, so the conditional probability of the ball being light, given that it is red, is:

$$P(\text{Light}|\text{Green}) = \frac{3}{5}$$

1.3.3 Calculate Joint Probability of Green and Light

Now we can calculate the joint probability:

$$P(\text{Green} \cap \text{Light}) = P(\text{Green}) \cdot P(\text{Light}|\text{Green}) = \frac{5}{9} \cdot \frac{3}{5} = \frac{1}{3}$$

Interpretation

The joint probability of picking a green ball and it being light is $\frac{1}{3}$, which means that 1 out of the 3 balls in the dataset are red and light (for easy understanding 3 out of 9 balls).

1.4 Bayes' Theorem

Bayes' theorem describes the relationship between conditional probabilities. It states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where:
- $P(A|B)$: Probability of A given B (posterior probability).
- $P(B|A)$: Probability of B given A (likelihood).
- $P(A)$: Probability of A (prior probability).
- $P(B)$: Probability of B (marginal probability).

Relationship between Bayes' Theorem and Joint Probability: From the definition of conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{and} \quad P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Combining these gives:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

1.5 Naive Bayes Classifier

The **Naive Bayes classifier** is a probabilistic machine learning algorithm based on Bayes' theorem. It assumes that the features are conditionally independent given the class label.

Why is it called "Naive"? The classifier is called *naive* because it assumes the independence of features, which is often not true in real-world data.

Equation: For a given class C_k and features $X = \{x_1, x_2, \ldots, x_n\}$, the probability of the class is:

$$P(C_k|X) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{P(X)}$$

Since $P(X)$ is constant for all classes, it is sufficient to maximize:

$$P(C_k|X) \propto P(C_k)\prod_{i=1}^{n} P(x_i|C_k)$$

1.6 Pseudo-Code for Naive Bayes Classifier

Algorithm 1 Naive Bayes Classifier

Require: Training dataset with features and labels
Ensure: Predicted label for a given test sample

```
Calculate prior probabilities P(C_k) for each class C_k
for each feature x_i do
    Compute the likelihood P(x_i|C_k) for each class C_k
end for
For a test sample X = {x_1, x_2, ..., x_n}:
for each class C_k do
    Compute posterior probability P(C_k|X) ∝ P(C_k) ∏_{i=1}^{n} P(x_i|C_k)
end for
Assign the class C_k with the highest posterior probability to the test sample
```

2 Toy Problem

Dataset:

Feature (Weather)	Feature (Temperature)	Play (Yes/No)
Sunny	*Hot*	*No*
Sunny	*Mild*	*Yes*
Rainy	*Cool*	*Yes*
Overcast	*Mild*	*Yes*
Rainy	*Hot*	*No*

2.1 Step-by-Step Solution

2.1.1 Calculate Prior Probabilities

The total number of instances is 5. Among them: - 3 instances correspond to Play=Yes, - 2 instances correspond to Play=No.

$$P(\text{Play=Yes}) = \frac{3}{5} = 0.6, \quad P(\text{Play=No}) = \frac{2}{5} = 0.4$$

2.1.2 Compute Likelihoods for Each Feature

For Weather:

$$P(\text{Sunny}|\text{Play=Yes}) = \frac{1}{3}, \quad P(\text{Sunny}|\text{Play=No}) = \frac{1}{2}$$

$$P(\text{Rainy}|\text{Play=Yes}) = \frac{1}{3}, \quad P(\text{Rainy}|\text{Play=No}) = \frac{1}{2}$$

$$P(\text{Overcast}|\text{Play=Yes}) = \frac{1}{3}, \quad P(\text{Overcast}|\text{Play=No}) = 0$$

For Temperature:

$$P(\text{Hot}|\text{Play=Yes}) = \frac{1}{3}, \quad P(\text{Hot}|\text{Play=No}) = \frac{1}{2}$$

$$P(\text{Mild}|\text{Play=Yes}) = \frac{2}{3}, \quad P(\text{Mild}|\text{Play=No}) = \frac{1}{2}$$

$$P(\text{Cool}|\text{Play=Yes}) = \frac{1}{3}, \quad P(\text{Cool}|\text{Play=No}) = 0$$

2.1.3 For the Test Instance (Weather=Sunny, Temperature=Mild)

Posterior Probability for Play=Yes:

$$P(\text{Play=Yes}|\text{Test}) \propto P(\text{Play=Yes}) \cdot P(\text{Sunny}|\text{Yes}) \cdot P(\text{Mild}|\text{Yes})$$

$$P(\text{Play=Yes}|\text{Test}) \propto 0.6 \cdot \frac{1}{3} \cdot \frac{2}{3} = 0.6 \cdot 0.333 \cdot 0.666 = 0.1332$$

Posterior Probability for Play=No:

$$P(\text{Play=No}|\text{Test}) \propto P(\text{Play=No}) \cdot P(\text{Sunny}|\text{No}) \cdot P(\text{Mild}|\text{No})$$

$$P(\text{Play=No}|\text{Test}) \propto 0.4 \cdot \frac{1}{2} \cdot \frac{1}{2} = 0.4 \cdot 0.5 \cdot 0.5 = 0.1$$

2.1.4 Decision

Since $P(\text{Play=Yes}|\text{Test}) = 0.1332 > P(\text{Play=No}|\text{Test}) = 0.1$, we classify the test instance as:

$$\text{Play = Yes}$$

3 Merits and Demerits of Naive Bayes Classifier

- **Merits:**
  - Simple to implement and computationally efficient.
  - Works well with small datasets and high-dimensional data.
  - Handles both continuous and discrete data.
- **Demerits:**
  - Assumes independence of features, which is rarely true in real-world data.
  - Performs poorly when features are highly correlated.
  - May produce biased results if the dataset has imbalanced classes.