Naive Bayes Classification Example

1 Toy Dataset: Weather and Golf Play

Table 1: Weather conditions and golf play decisions

Day	Outlook	Temperature	Humidity	Windy	Play Golf
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

2 Classification Question

Given the following weather conditions:

- Outlook = Sunny
- Temperature = Cool
- Humidity = High
- Windy = True

Should we play golf (*Play Golf* = Yes or No)?

3 Solution Approach

We will solve this using Naive Bayes classification by:

1. Calculating prior probabilities $P(\text{Yes})$ and $P(\text{No})$
2. Computing conditional probabilities for each feature:

$$\begin{gathered} P(\text{Outlook} = \text{Sunny} \mid \text{Yes}) \\ P(\text{Temperature} = \text{Cool} \mid \text{Yes}) \\ \vdots \\ P(\text{Windy} = \text{True} \mid \text{No}) \end{gathered}$$

3. Calculating the joint probabilities:

$$P(\text{Yes} \mid \mathbf{X}) \propto P(\text{Yes}) \times \prod_{i=1}^{n} P(X_i \mid \text{Yes})$$
$$P(\text{No} \mid \mathbf{X}) \propto P(\text{No}) \times \prod_{i=1}^{n} P(X_i \mid \text{No})$$

4. Making the prediction by comparing both probabilities.

Naive Bayes

Naive Bayes is a classification algorithm that uses probability to predict which category an instance belongs to, assuming that all features are conditionally independent. (change in one feature doesn't not affect the other features).

Algorithm (Binary classification)

1. Calculate $P(\text{class } 0 \mid X_1, X_2, \ldots X_i)$
2. Calculate $P(\text{class } 1 \mid X_1, X_2, \ldots, X_i)$
3. If $P(\text{class } 0 \mid X_1, X_2, X_i) > P(\text{class } 1 \mid X_1, X_2, X_i)$:

    return "class 0"

else:

    return "class 1"

Why feature independence assumption helps?

→ If the features are not independent, it would become a full Bayesian classifier.

In such case, we had to compute:

$$P(\text{class } 0 \mid X_1, X_2, \ldots X_i) = P(\text{class } 0) \cdot P(X_1, X_2, \ldots X_i \mid \text{class } 0)$$

The joint probability $P(X_1, X_2, \ldots X_i \mid \text{class } 0)$ requires estimating all possible feature interactions/combinations (a^n, where n = number of features, a = number of classes). For example, for 10 features in a binary classification task, we have to compute $2^{10} = 1024$ combinations (computationally inefficient)

→ BUT !! When the features are conditionally independent:

$$P(\text{class } 0 \mid X_1, X_2, \ldots X_i) = \frac{\cancel{P(X_1 \mid \text{class } 0) \cdot P(X_2 \mid \text{class } 0) \cdot P(X_i \mid 0)}}{}$$

$$= \frac{P(X_1 \mid \text{class } 0) \cdot P(X_2 \mid \text{class } 0) \cdot P(X_i \mid \text{class } 0) \cdot P(\text{class } 0)}{P(X_1, X_2, X_3)}$$

$$P(\text{class } 1 \mid X_1, X_2, \ldots X_i) = \frac{P(X_1 \mid \text{class } 1) \cdot P(X_2 \mid \text{class } 1) \cdot P(X_i \mid \text{class } 1) \cdot P(\text{class } 1)}{P(X_1, X_2, X_3)}$$

Sol^n: Given, X_1 = Sunny, X_2 = Cool, X_3 = High, X_4 = True

y = class 0 or class 1?

class 0 = Yes
class 1 = No

P(class 0 | Sunny, Cool, High, True)

$$= \frac{P(\text{Sunny} \mid \text{class } 0) \cdot P(\text{cool} \mid \text{class } 0) \cdot P(\text{High} \mid 0) \cdot P(\text{True} \mid 0) \cdot P(0)}{\text{denom } P(\text{Sunny}, \text{cool}, \text{High}, \text{True})}$$

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{P(\text{Sunny}, \text{cool}, \text{High}, \text{True})}$$

$$= \frac{0.0052}{\text{denom}}$$

P(class 1 | Sunny, cool, High, True)

$$= \frac{P(\text{Sunny} \mid \text{class } 1) \cdot P(\text{cool} \mid \text{class } 1) \cdot P(\text{High} \mid 1) \cdot P(\text{True} \mid 1) \cdot P(1)}{\text{denom}}$$

$$= \frac{\frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14}}{\text{denom}} = \frac{0.0205}{\text{denom}}$$

P(class 1 | Sunny, Cool, High, True) > P(class 0 | Sunny, ...

So, Prediction = class 1 (NO).

Note:

$P(\text{class } 1 \mid X_1, X_2, \ldots X_n) \neq 1 - P(\text{class } 0 \mid X_1, X_2, \ldots)$

But,

$P(\text{class } 1 \mid X_i) = 1 - P(\text{class } 0 \mid X_i)$

Pros

→ Very fast and efficient

→ Handles missing data without throwing valueError.

→ Naturally fits for multi-class problems.

→ Less prone to overfitting.

Cons

→ Strong independence assumption is unrealistic.

→ Zero-frequency problem (solvable via laplace smoothing)

→ Not ideal for regression.

zero-frequency problem

Outlook	Play Golf
Sunny	No
Sunny	No
Overcast	Yes
Raining	Yes

$p(\text{Sunny} \mid \text{Yes}) = 0$

$p(\text{Sunny} \mid \text{No}) = \frac{2}{2} = 1$

$p(\text{Overcast} \mid \text{Yes}) = \frac{1}{2} = 0.5$

$p(\text{overcast} \mid \text{NO}) = 0$

(Laplace Smoothing)

$p(\text{Sunny} \mid \text{Yes}) = \frac{0+1}{2+3} = \frac{1}{5}$

Formula

$p(X_i \mid \text{class}) = \frac{\text{count}(X_i, \text{class}) + \alpha}{\text{count}(\text{class}) + \alpha \cdot N_i}$

α = small constant

N_i = possible values of X_i

CRITICAL FAILURE CASE: What if outlook = "foggy"?

→ Usual ML models will return an error, but Naive Bayes will make a biased decision, as all probabilities become 0.