Naive Baye's Algorithm

→ Naive Bayes is a probabilistic Machine Learning algorithm used for classification tasks. It is based on Bayes theorem and assumes that the features used in the Model are Independent of eachother.

→ Suitable for both binary and Multi classification algorithm.

In probability,

Independent Event,

Running a dice {1, 2, 3, 4, 5, 6}

$P(1) = 1/6 \qquad P(2) = 1/6.$

it is Independent event, because one outcome is not hampering or not changing the Probability of the other outcome.

conditional probability,

→ conditional probability measures the likelihood of an event occurring given that another event has already occurred.

denoted as, $P(A|B)$ → Probability of event A given event B.

$$P(A|B) = P(A \cap B) / P(B)$$

$$P(B) > 0$$

Baye's theorem,

Bayes theorem is a fundamental concept in probability theory and statistic. It describes how to update the probability of a hypothesis based on new evidence or information.

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

$P(A|B)$: posterior probability, probability of hypothesis A given evidence B.

$P(B|A)$: Likelihood, the probability of evidence B given that A is true.

$P(A)$: Initial belief about the probability of A before observing evidence.

$P(B)$: Normalization process., Ensuring that the probabilities sum to 1

In machine learning context,

Indenpendent features, x_1, x_2 and x_3
dependent feature, y (Yes | No)

The algorithm that specifically use this Bayes theorum is called as Naive Bayes algorithm.

so, basically we need to find $p(y | (x_1, x_2, x_3)) = ?$

$$p(y \mid (x_1, x_2, x_3)) = \frac{p(y) * p(x_1, x_2, x_3) \mid y}{p(x_1, x_2, x_3)}$$

$$= \frac{p(y) * p(x_1/y) * p(x_2/y) * p(x_3/y)}{p(x_1) * p(x_2) * p(x_3)}$$

dependent feature $(y) \longrightarrow$ Yes / No.

$$p(yes \mid (x_1, x_2, x_3)) = \frac{P(yes) * p(x_1/yes) * p(x_2/yes) * p(x_3/yes)}{\cancel{p(x_1) * p(x_2) * p(x_3)}}$$ constant

$$p(NO \mid (x_1, x_2, x_3)) = \frac{p(NO) * p(x_1/NO) * p(x_2/NO) * p(x_3/NO)}{\cancel{p(x_1) * p(x_2) * p(x_3)}}$$ constant

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Let's solve this problem.

using naive Bayes theorem.

for my simplicity, let's take

1. outlook (x_1)
2. Temperature (x_2)

as independent features

Outlook

	Yes	No	P(E|yes)	P(E|No)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	6/5
Rain	3	2	3/9	2/5.

Play (Y/N)

n(yes) = 9
n(no) = 5

p(yes) = 9/14

p(no) = 5/14.

Temperature

	Yes	No	P(E|yes)	P(E|No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	4/5
Cool	3	1	3/9	3/5.

P(Sunny, Hot) = ?

p(yes | (sunny, Hot)) = p(yes) * P(sunny | yes) * P(Hot | yes)

= 9/14 * 2/9 * 2/9
= 2/63
= 0.031

p(No | (sunny, hot)) = P(No) * p(sunny | No) * p(hot | No)

= 5/9 * 3/5 * 2/5
= 3/35
= 0.085

p(yes | (sunny, hot)) = $\dfrac{0.031}{(0.031 + 0.085)}$ = 0.27 => 27%

$P(No \mid (sunny, hot)) = \frac{0.085}{(0.085 + 0.031)} = 0.73 \Rightarrow 73\%$

They will NOT play tennis.

Characteristics

* Probabilistic Approach, It predicts the probability of a class given a set of feature.

* Independence Assumption, Assumes that all features are independent of one another ('Naive')

* Scalability, works well with large datasets

Variants of Naive Baye's theorem

1. Bernoulli Naive Bayes
2. Multinomial Naive Bayes
3. Gaussian Naive Bayes.

Bernoulli Naive Bayes,

whenever your features are following a Bernoulli distribution, then we need to use Bernoulli naive Bayes Algorithm.

$$P(x \mid c) = \prod_{i=1}^{n} P(x_i \mid c)^{x_i} (1 - P(x_i \mid c))^{1-x}$$

$P(x_i \mid c)$ is the probability of feature x_i

Application:

1. Text classification with binary word presence
2. Document classification

Use case: Binary features in the dataset.

Multinomial naive Bayes

Mostly used when the inputs are text form and features represent counts or frequencies, such as word occurrences in a document.

$$P(x|c) = \prod_{i=1}^{n} P(x_i|c)$$

convert the input text into numerical values.

Application:

1. Spam detection
2. Sentiment analysis.

Use case: Discrete data, especially for text classification.

Gaussian Naive Bayes

If the features are following Gaussian distribution, then we use Gaussian naive Bayes, and dataset have continuous features.

$$p(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right)$$

Application:

1. Iris data classification
2. Handwriting digit recognition.

Use case : continuous numerical features that follow a normal distribution.