07_Naive_Bayes_1

15 May 2025 20:41

The next two techniques nowadays are not used in the industry - Naïve Bayes & SVM

Then why to study them?

1. Because both of them are extremely important from interview perspective.
2. Some concepts that they use are foundation of some modern machine learning techniques such as **Conditional Independence**.

Flow:

1. Introduction to Naïve Bayes
2. Business case introduction
3. Mathematical Intuition
4. Assumptions - Naïve
5. Training
6. Testing

Try (as a human) to identify spam & non-spam texts from amongst these two messages:

1. **I, Nigerian prince need help, send money.**
2. **Scheduling a meeting at 5pm, kindly revert.**

Clearly the first text is spam, but how?
Some ways which can be used to determine this are:

1. Keywords - "money"
2. Group of words
3. Context:
   a. Consider this statement: … and I started looking for the bin everywhere like crazy.
   What could be the meaning of this statement? What do you visualize?
   I must have some garbage in my hands & I am running here & there searching for the dustbin.
   Let me complete the sentence now:
   I accidently deleted an important document and I started looking for the bin everywhere like crazy.

4. Semantics:
   a. Meaning of these two sentences are different even though they have the same words:
   You can cage a swallow.
   You can swallow a cage.
   Or a better example can be:
      i. She fed her dog a shark. Vs.
      ii. She fed her dog to a shark.

5. And many more things are there

But we will learn all these later in the NLP, right now we are going to focus only on the first two.

Pre-processing the data:

Step - 1: Converting everything into lower case

i, nigerian prince need help, send money.

Step - 2: Tokenizing the sentence

i, nigerian prince need help, send money.

Step - 3: Removing Punctuation for example: , . ? ! : ;

i nigerian prince need help send money

Step - 4: Removing "stop words" e.g., 'am', articles such as 'a', 'an', 'the', 'I', 'me', 'you', 'of', 'and'

~~i~~ nigerian prince ~~need~~ help ~~send~~ money

Final sentence after pre-processing:

nigerian prince help money

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$w_1 \quad w_2 \quad w_3 \quad w_4$$

Although we have considered only one sentence, an actual email would have many sentences in it. So to figure out if a mail is spam or ham (non-spam), we also check the frequency of each word so that if some keywords are repeated quite often then we can decide accordingly. But here, each word is coming just once:

$w_1 \to 1$, $w_2 \to 1$, $w_3 \to 1$, $w_4 \to 1$

Let's say the labels are: spam $\to 1$ $(=y)$ non-spam $\to 0$ $(=y)$

Text-1 $\to$ $P(y=0 \mid \text{Text-1}) = 0.3$

Text-1 $\to$ $P(y=1 \mid \text{Text-1}) = 0.6$

$\downarrow$

Conditional Probability

Conclusion: $y=1$ hence Spam

Example Training Data:

Text	Label
I, Nigerian prince, need help send money	1
Sceduling a meeting at 5pm, kindly revert	0

What can we learn from this data?

Ans: We can learn that what text comes in a spam $(y=1)$ mail & what (type of) text occurs in a non-spam $(y=0)$ email.

As the **labels are given** to us while training, actually we are learning **P(text | label)** rather than learning **P(label | text).** For example, for the first sentence we learn **P(text-1 | y=1)** and for the second one we learn **P(text-2 | y=0)**. But while deployed, our model has to predict **P(y=0 or 1 | text)** so how will we find it?
Taking this to a simpler mathematical level:

If we know P(B | A) & we want to find P(A | B), what do we use?
Ans: **Bayes theorem**!

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

Since this algorithm uses Bayes theorem & it also has some very Naïve (childish) assumptions it is called **Naïve Bayes Algorithm**.

Using Baye's theorem, the probabilities of a mail being spam or ham will be as follows:

$$P(y=0 \mid Text_1) = \frac{P(Text_1 \mid y=0) \cdot P(y=0)}{P(Text_1)}$$

$$P(y=1 \mid Text_1) = \frac{P(Text_1 \mid y=1) \cdot P(y=1)}{P(Text_1)}$$

Looking carefully we can see that both have the same denominator & we are not interested in absolute values of their probabilities but just to compare them. Hence we can just calculate their numerators & make the comparison so that we can classify them.
But how can we find P(y = 0) & P(y = 1)?

$$P(y=0) = \frac{\text{no. of datapoints with class 0}}{\text{total no. of datapoints}} = \frac{n_0}{n}$$

$$P(y=1) = \frac{\text{no. of datapoints with class 1}}{\text{total no. of datapoints}} = \frac{n_1}{n}$$

And how will we find P(Text1 | y = 0) & P(Text1 | y = 1)?

$$P(Text_1 \mid y=0) = P(w_1 \cap w_2 \cap \dots \cap w_n \mid y=0)$$

$$P(Text_1 \mid y=1) = \underbrace{P(w_1 \cap w_2 \cap \dots \cap w_n \mid y=1)}$$

Now, how will we find ↑ this?

$$P(w_1 \cap w_2 \cap w_3 \dots \cap w_n \mid y=1) = P(w_1 \mid y=1) \cdot P(w_2 \mid y=1) \cdot \dots \cdot P(w_n \mid y=1)$$

But we can write this only if all these events are independent.

That means, we are assuming that w1 given that the mail is spam, w2 given that the mail is spam, ... , wn given that the mail is spam are all independent from each other. This is the assumption Naïve Bayes takes which is "childish" that's why it is called "Naïve".

$$\therefore P(w_1 \cap w_2 \cap \dots \cap w_n \mid y=1) = \prod_{i=1}^{n} P(w_i \mid y=1)$$

$$\therefore P(y=0 \mid Text_1) = \frac{P(y=0) \cdot \prod_{i=1}^{n} P(w_i \mid y=0)}{k}$$

$$\therefore P(y=1 \mid Text_1) = \frac{P(y=1) \cdot \prod_{i=1}^{n} P(w_i \mid y=1)}{k}$$

where $k = P(Text_1)$

Example:

$Text_1 = w_1 \cap w_2 \cap w_3$ OR

$x_q = [w_1, w_2, w_3]$

n	w_1	w_2	w_3	0	n_0
	w_2	w_3	w_4	0	
	w_5	w_6	w_1	0	
	w_1	w_2	w_4	0	
	w_5	w_6	w_2	1	n_1
	w_1	w_2	w_5	1	

$$\therefore P(y=0 \mid w_1 \cap w_2 \cap w_3)$$

$$= P(y=0) \cdot P(w_1 \cap w_2 \cap w_3 \mid y=0)$$

$$= P(y=0) \cdot \prod_{i=1}^{3} P(w_i \mid y=0) \quad \text{——} \boxed{I}$$

$x_q = [w_1, w_2, w_3]$

n	w_1	w_2	w_3	0	n_0
	w_2	w_3	w_4	0	
	w_5	w_6	w_1	0	
	w_1	w_2	w_4	0	
	w_5	w_6	w_2	1	n_1
	w_1	w_2	w_5	1	
	w_3	w_4	w_1	1	

$n=7 \quad n_0=4 \quad n_1=3$

$$\therefore P(y=0 \mid w_1 \cap w_2 \cap w_3)$$

$$= P(y=0) \cdot P(w_1 \cap w_2 \cap w_3 \mid y=0)$$

$$= P(y=0) \cdot \prod_{i=1}^{3} P(w_i \mid y=0) \quad \text{———} \; \text{(I)}$$

$$\therefore P(y=1 \mid w_1 \cap w_2 \cap w_3)$$

$$= P(y=1) \cdot P(w_1 \cap w_2 \cap w_3 \mid y=1)$$

$$= P(y=1) \cdot \prod_{i=1}^{3} P(w_i \mid y=1) \quad \text{———} \; \text{(II)}$$

$$\boxed{P(y=0) = \frac{4}{7}} \qquad \boxed{P(y=1) = \frac{3}{7}}$$

$$P(w_1 \mid y=0) = \frac{P(w_1 \cap y=0)}{P(y=0)} = \frac{3}{4} \qquad P(w_1 \mid y=1) = \frac{P(w_1 \cap y=1)}{P(y=1)} = \frac{2}{3}$$

$$P(w_2 \mid y=0) = \frac{3}{4} \qquad P(w_2 \mid y=1) = \frac{2}{3}$$

$$P(w_3 \mid y=0) = \frac{2}{4} \qquad P(w_3 \mid y=1) = \frac{1}{3}$$

$$P(w_4 \mid y=0) = \frac{2}{4} \qquad P(w_4 \mid y=1) = \frac{1}{3}$$

$$P(w_5 \mid y=0) = \frac{1}{4} \qquad P(w_5 \mid y=1) = \frac{2}{3}$$

$$P(w_6 \mid y=0) = \frac{1}{4} \qquad P(w_6 \mid y=1) = \frac{1}{3}$$

$$P(y=0 \mid Text_1) = P(y=0) \cdot \prod_{i=1}^{3} P(w_i \mid y=0)$$

$$= 4 \quad 3 \quad 3 \quad \cancel{9} \quad = \quad 9 \quad = 0.1607$$

$$= \frac{\cancel{4}}{7} \times \frac{3}{\cancel{4}} \times \frac{3}{4} \times \frac{\cancel{2}}{\underset{2}{\cancel{4}}} = \frac{9}{56} = 0.1607$$

$$P(y=1 \mid Text_1) = P(y=1) \cdot \prod_{i=1}^{3} P(w_i \mid y=1)$$

$$= \frac{\cancel{3}}{7} \times \frac{2}{\cancel{3}} \times \frac{2}{3} \times \frac{1}{3} = \frac{4}{63} = 0.0635$$

Conclusion: The mail is "ham" as P(y = 0 | Text) > P(y = 1 | Text)

Why Naïve Bayes?

1. Simple
2. We will use "Transformer" in NLP which is far too complex
3. Superfast
4. Works well for basic tasks
5. Interpretable

When does Naïve Bayes fail?

1. What if a new word w_n comes that was not there in the training data?

   $P(w_n|y=0) = 0$ and also
   $P(w_n|y=1) = 0$

   How to solve this?
   Ans: **Laplace's Smoothing** - Adding some constant to both numerator & denominator

$$P(w_i \mid y=1) = \frac{n(w_1) + \alpha}{n_1 + \alpha \cdot C}$$

for example, let's take $\alpha = 1$ & $C = 2$

$$P(w_i \mid y=1) = \frac{n(w_i) + 1}{n_1 + 2}$$

For a new word w_n let's say $n_1 = 100$

$$p(w_n|y=1) = \frac{0+1}{100+2} = \frac{1}{102} \approx 0.009$$

What will happen if alpha is too high? Let's say 10000

$$P(\omega_i|y=1) = \frac{n(\omega_i)+10,000}{n_1+10,000\times2}$$

If suppose ω_i comes 10 times in spam emails and $n_1=100$

$$= \frac{10+10,000}{100+20,000} = \frac{10,010}{20,100} \approx 0.5$$

Now suppose ω_i is occurring in 90 spam mails

$$= \frac{90+10,000}{20,100} = \frac{10,090}{20,100} \approx 0.5$$

$\therefore$ Underfit

And what if value of alpha is too small? Let's say 0.01.

$$P(\omega_i|y=1) = \frac{n(\omega_i)+\alpha}{n_1+\alpha\cdot c} \approx \frac{n(\omega_i)}{n_1}$$

<u>case-1</u> $$= \frac{10+0.01}{100+0.02} = \frac{10.01}{100.02} = 0.1$$

<u>case-2</u> $$P(\omega_i|y=1) = \frac{90+0.01}{100+0.02} = \frac{90.01}{100.02} \approx 0.$$