

Naive Bayes Classifier

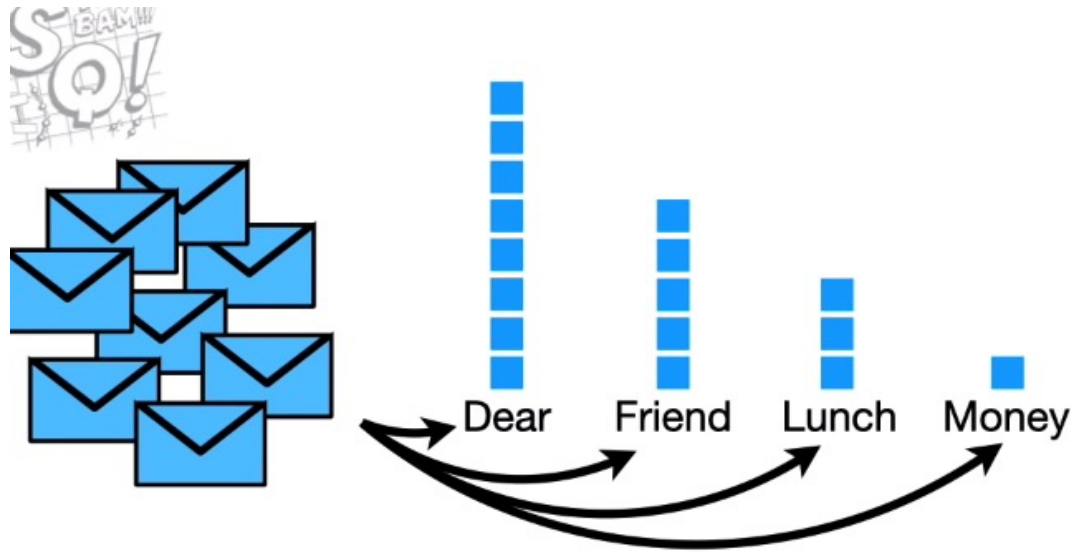


Set Up

- Imagine we get messages from friends and family
- But also unsolicited messages (advertising, scams...)
- And we don't want to sort out the messages anymore, or at least try to reduce the amount by quite a bit and create an algorithm which will automatically send most spams to a spam box.

Intuition

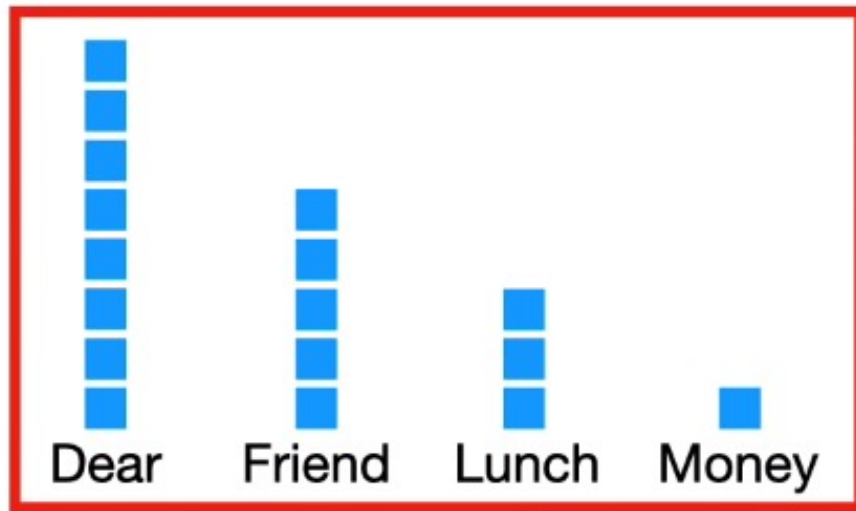
- Following pics are from <https://www.youtube.com/watch?v=O2L2Uv9pdDA&t=695s>



So, the first thing we do is make a **histogram** of all the words that occur in the **normal messages** from friends and family.

Intuition

- Can use the histogram to calculate the **probability** of seeing each word given it was in a **normal** message



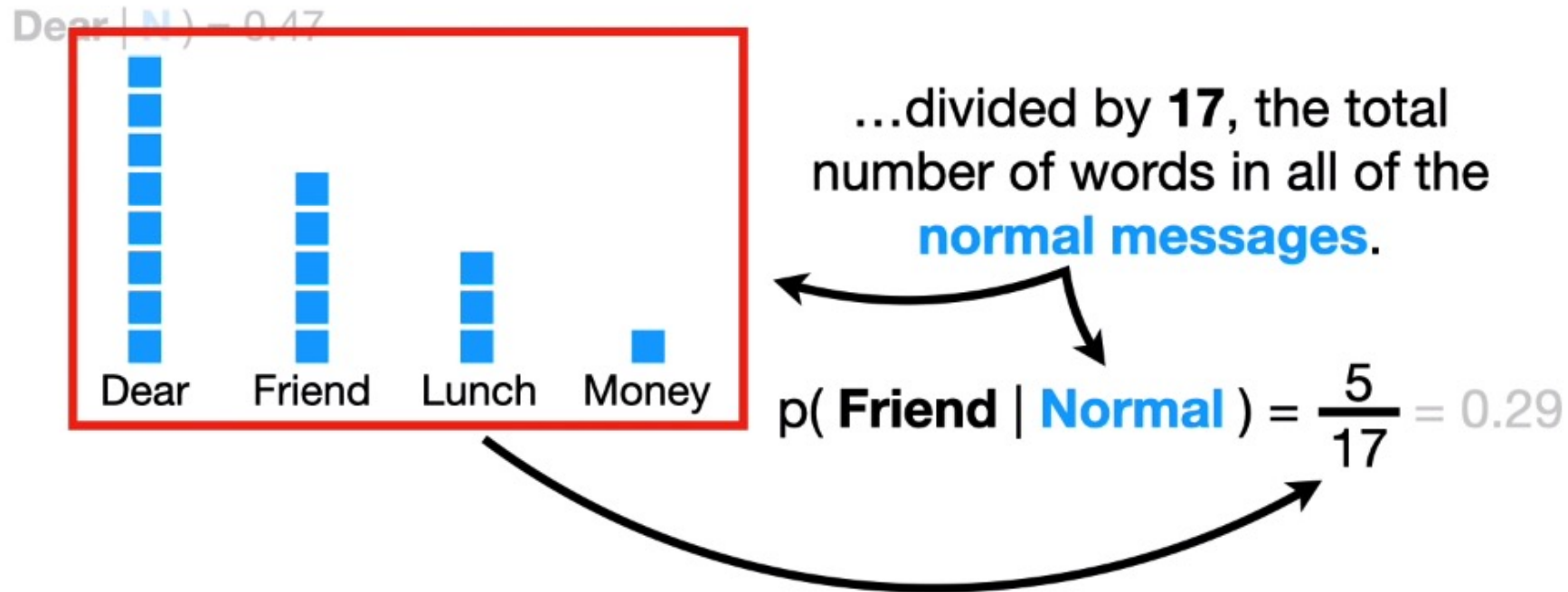
...divided by **17**, the total number of words in all of the **normal messages**.

$$p(\text{Dear} \mid \text{Normal}) = \frac{8}{17}$$

A curved arrow points from the 'Dear' bar in the histogram to the 'Dear' in the probability formula.

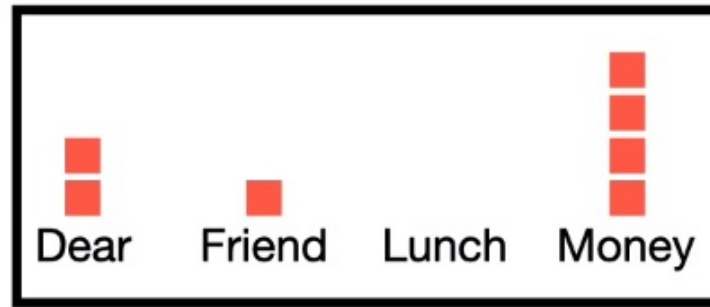
Intuition

- Can repeat the process for each word we see in our normal messages (hams).



Intuition

- Can do the exact same thing for our spams



$$p(\text{Dear} \mid \text{Spam}) = \frac{2}{7} = 0.29$$



Intuition

- We end up with a set of conditional **probabilities** or **likelihoods**.

$$\begin{aligned}p(\text{Dear} \mid \text{N}) &= 0.47 \\p(\text{Friend} \mid \text{N}) &= 0.29 \\p(\text{Lunch} \mid \text{N}) &= 0.18 \\p(\text{Money} \mid \text{N}) &= 0.06\end{aligned}$$

$$\begin{aligned}p(\text{Dear} \mid \text{S}) &= 0.29 \\p(\text{Friend} \mid \text{S}) &= 0.14 \\p(\text{Lunch} \mid \text{S}) &= 0.00 \\p(\text{Money} \mid \text{S}) &= 0.57\end{aligned}$$

Terminology Alert!!!

Because we have calculated the probabilities of discrete, individual words, and not the probability of something continuous, like weight or height, these **Probabilities** are also called **Likelihoods**.

Intuition 1st Example

- Now imagine we get a message that says:

Dear friend

- **We want to decide where it should go:**
 - Our normal inbox
 - Or our spam box

Intuition 1st Example

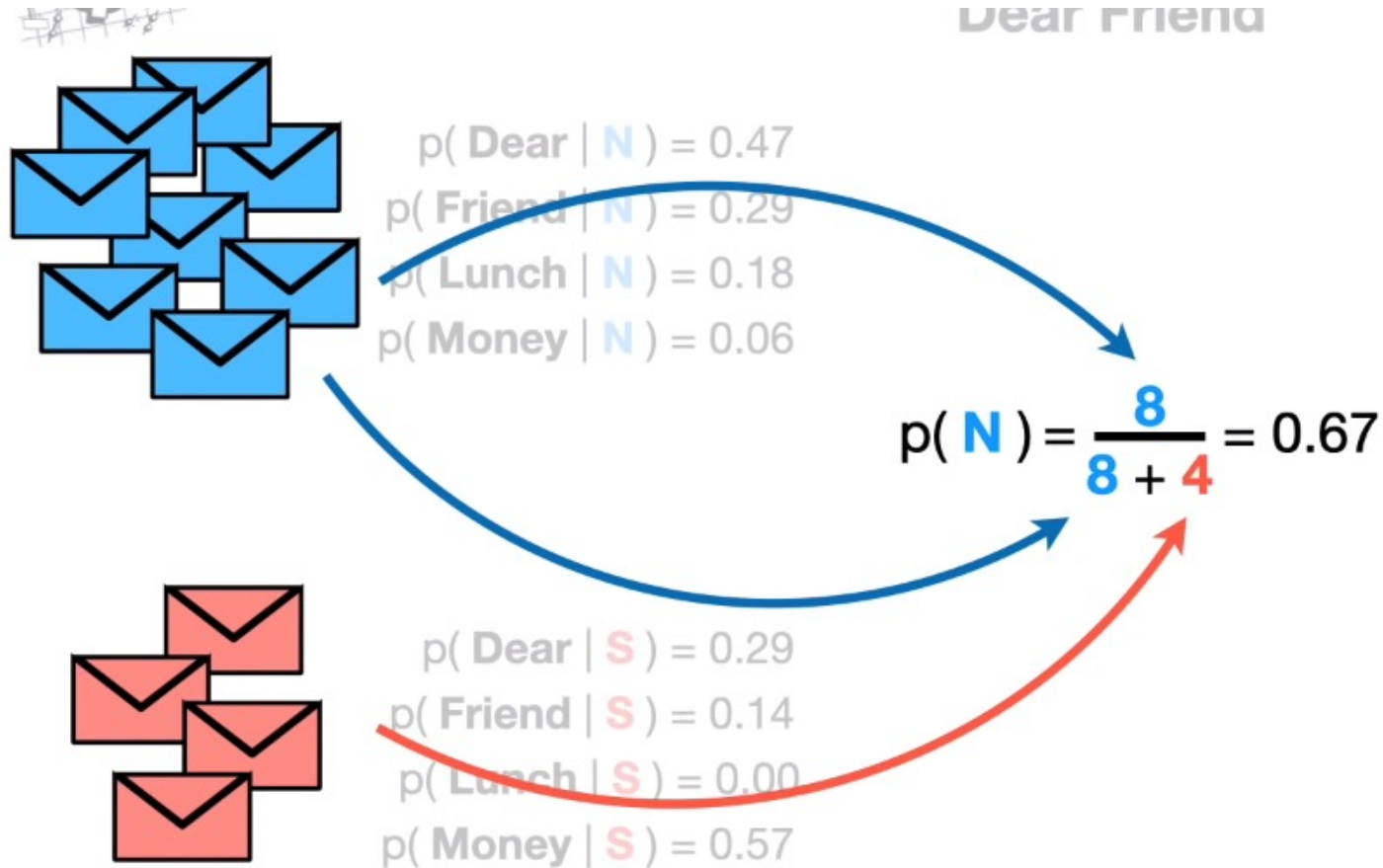
Dear Friend

We start with an initial guess about the probability that any message, regardless of what it says, is a **normal message**.

$p(\mathbf{N})$



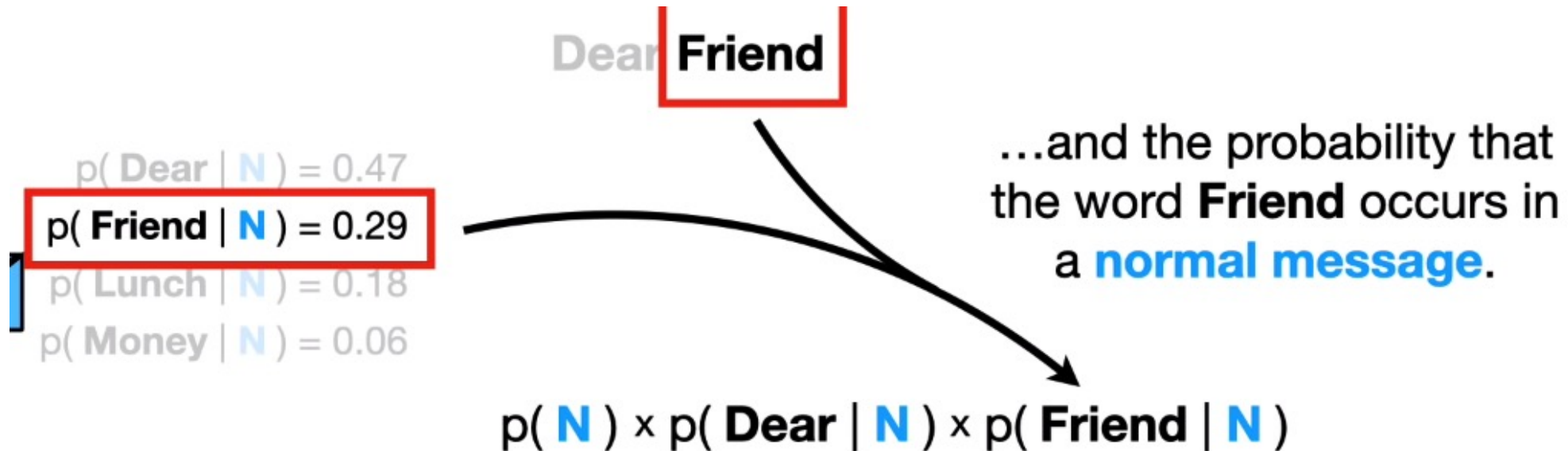
Intuition 1st Example



For example, since **8** of the **12** messages are **normal messages**, our initial guess will be **0.67**.

Intuition 1st Example

- Multiply the probabilities with each other



Intuition 1st Example

However, technically, it is
proportional to the probability
that the message is **normal**,
given that it says **Dear Friend**.


$$0.67 \times 0.47 \times 0.29 = 0.09 \propto p(\text{N} \mid \text{Dear Friend})$$

Intuition 1st Example

- Same process for spam:

Like before, we can think of **0.01** as the score that **Dear Friend** gets if it is **Spam**.

$$0.33 \times 0.29 \times 0.14 = 0.01$$

Intuition 1st Example

- Compare both results :

$$p(\text{N}) \times p(\text{Dear} \mid \text{N}) \times p(\text{Friend} \mid \text{N}) = 0.09$$

$$p(\text{S}) \times p(\text{Dear} \mid \text{S}) \times p(\text{Friend} \mid \text{S}) = 0.01$$

Intuition 2nd Example

- Now let's try and classify :

Lunch Money Money Money Money

Intuition 2nd Example

1 $p(\text{Dear} | \text{N}) = 0.47$

2 $p(\text{Friend} | \text{N}) = 0.29$

3 $p(\text{Lunch} | \text{N}) = 0.18$

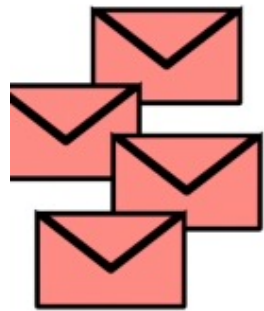
4 $p(\text{Money} | \text{N}) = 0.06$

...and the probability we see
Money four times, given that it
is in a **normal message**.


$$p(\text{N}) \times p(\text{Lunch} | \text{N}) \times p(\text{Money} | \text{N})^4$$

$$p(\text{N}) \times p(\text{Lunch} | \text{N}) \times p(\text{Money} | \text{N})^4 = 0.000002$$

Intuition 2nd Example



$p(\text{Dear} | \text{S}) = 0.29$
 $p(\text{Friend} | \text{S}) = 0.14$
 $p(\text{Lunch} | \text{S}) = 0.00$
 $p(\text{Money} | \text{S}) = 0.57$

$$p(\text{S}) \times p(\text{Lunch} | \text{S}) \times p(\text{Money} | \text{S})^4 = 0$$

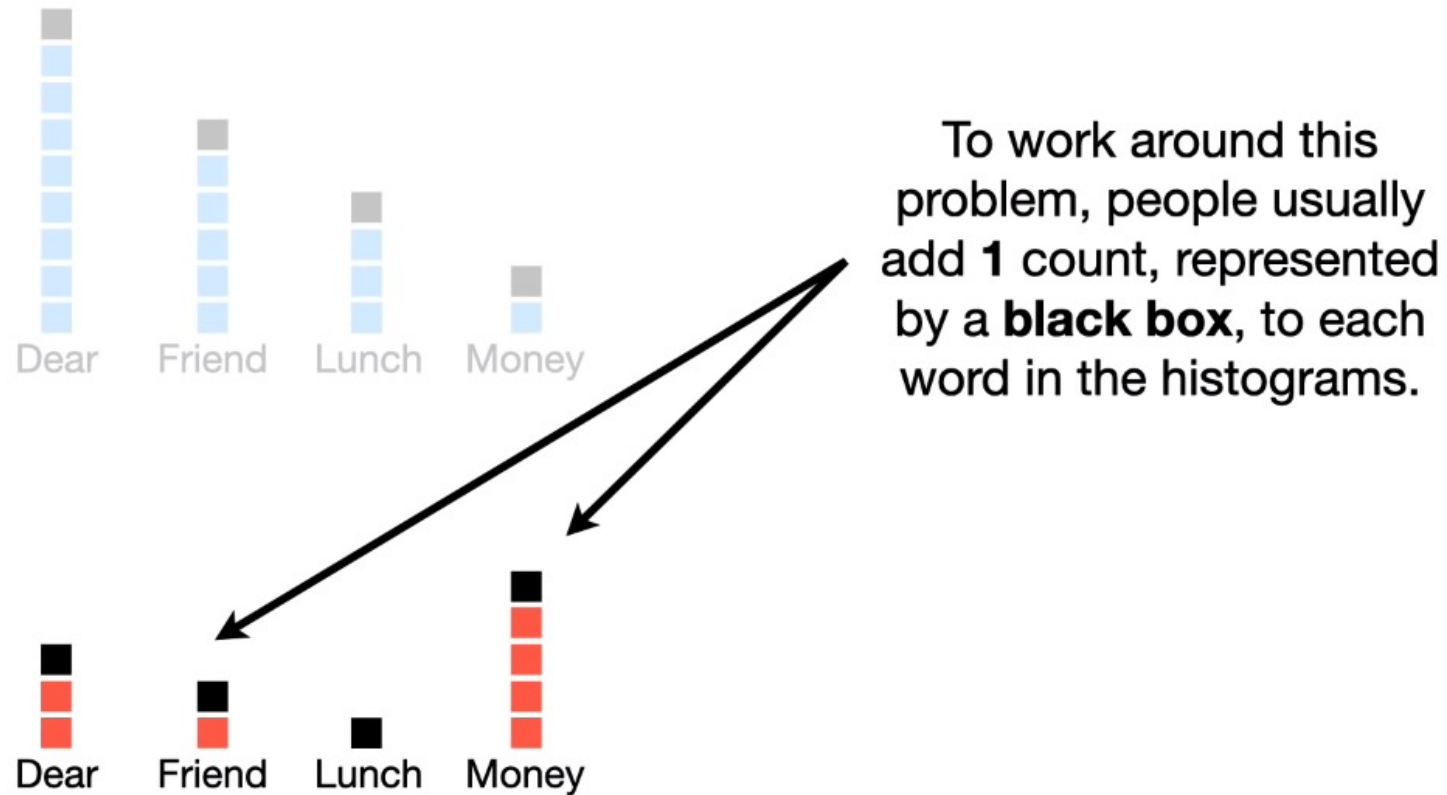
This is because the probability we see **Lunch** in **spam** is **0**, since it was not in the **Training Data**.

Intuition 2nd Example

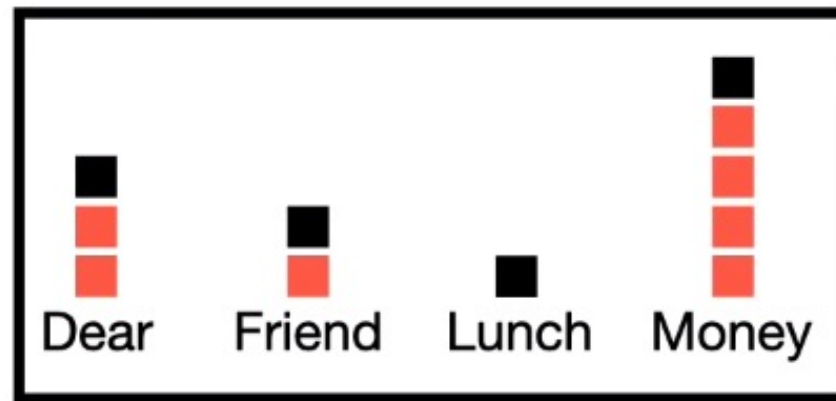
- In other words, if a message contains lunch, it will not be classified as spam...
- This is not satisfactory !
- No matter how many times we see the word **money**, or any other word which has a high probability of being in spam, the end result will be 0 ...

Pseudocounts

- The pseudocount is referred to with the Greek letter α (1 here)



- Now, when calculating the probas of observing each word, we never get 0.
- Careful : we now need to add 4 (total counts added = our vocab length) to the denominator



$$p(\text{Lunch} | \text{Spam}) = \frac{1}{7 + 4}$$

- Our values for P(Normal) and P(Spam) do not change however.
- We are 'hallucinating' data so our model will *generalize* better; meaning we're taking into account seeing certain words that our data doesn't account for so that if we *do* ever see this in the future, we have a probability estimate that we can use.

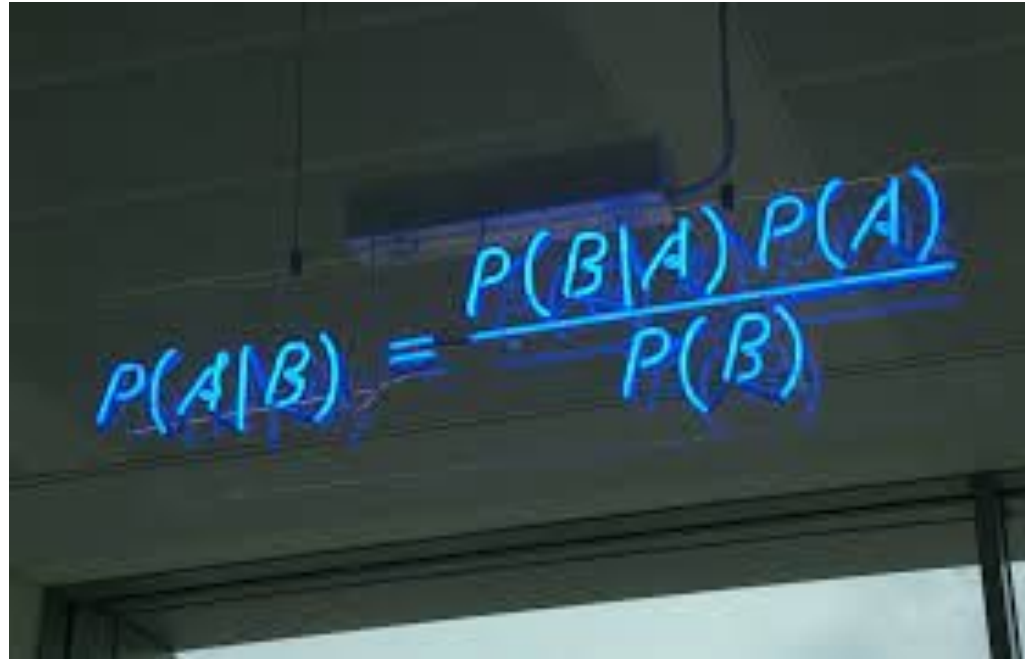
$$p(\mathbf{N}) \times p(\mathbf{Lunch} \mid \mathbf{N}) \times p(\mathbf{Money} \mid \mathbf{N})^4 = 0.00001$$

$$p(\mathbf{S}) \times p(\mathbf{Lunch} \mid \mathbf{S}) \times p(\mathbf{Money} \mid \mathbf{S})^4 = 0.00122$$

Why is this classifier called *Naive* Bayes ?

- This classifier treats all word orders the same, there is no sequence info taken into account... only single word counts.
- That said it tends to perform surprisingly well...

Putting things into perspective with the formula

A photograph of a blue neon sign mounted on a dark wall. The sign displays the formula for conditional probability: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. The text is written in a stylized, glowing blue font. The background is dark, and the sign is the primary light source in the image.
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In the case of Spam Filtering

- $P(A|B)$ is *proportional* to our estimated probabilities

$$P(\text{Spam}|w_1, w_2, \dots, w_n) \propto P(\text{Spam}) \cdot \prod_{i=1}^n P(w_i|\text{Spam})$$

- P_{spam} — the part/proportion of spam messages in our dataset
- $P_{w_i|\text{spam}}$ — the probability of a word w_i to be found in the spam messages.
- Same logic for not spam (ham) :
 - $P_{\text{not_spam}}$ — the part of non-spam messages in the dataset
 - $P_{w_i|\text{non_spam}}$ — the probability of a word to be found in the non-spam messages.

Why are we using proportionality vs equality ?

- Where did the denominator from the formula $P(B)$ go ?
- $P(B) = P(w_1, w_2, w_3, \dots)$ will be the same when calculating the probabilities of both classes, spam and ham.
- It can therefore be regarded as a constant :
 - We're not interested in the exact result, only which class has higher probability.
 - So we can save a little computation by doing so.

How do we measure the probability of a word given a class ? (same for ham)

$$P(w_i|Spam) = \frac{N_{w_i|Spam} + \alpha}{N_{Spam} + \alpha \cdot N_{Vocabulary}}$$

- $N_{vocabulary}$
 - the number of unique words in the whole dataset.
- N_{spam}
 - the total number of words in the spam messages.
- N_{wi_spam}
 - the number of times a word w_i is repeated in all spam messages.
- Alpha
 - the coefficient for the cases when a word in the message is absent in spam.

Implementation

- Now we have all we need to know to implement the algorithm 😊