Naive Bayes clearly explained!

Text classification using naïve bayes

Conditional Probability

1. In probability theory, conditional probability is a measure of the probability of an event given that another event has already occurred.

2. If the event of interest is A and the event B is assumed to have occurred, "the conditional probability of A given B", or "the probability of A under the condition B", is usually written as P(A|B), or sometimes PB(A).

Example

Chances of cough

The probability that any given person has a cough on any given day maybe only 5%. But if we know or assume that the person has a cold, then they are much more likely to be coughing. The conditional probability of coughing given that person have a cold might be a much higher 75%.

Marbles in a bag

2 blue and 3 red marbles are in a bag.

What are the chances of getting a blue marble?

???

Marbles in a bag

2 blue and 3 red marbles are in a bag.

What are the chances of getting a blue marble?

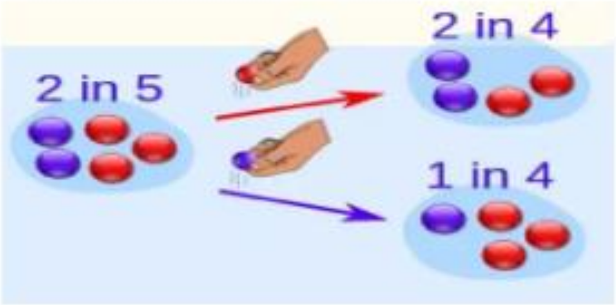
Answer: - The chance is 2 in 5

Marbles in a bag

But after taking one out of these chances,

situation may change!

So the next time:



1. if we got a red marble before, then the chance of a blue marble next is 2 in 4
2. if we got a blue marble before, then the chance of a blue marble next is 1 in 4

Probabiliy given pre-condition

Likewise:-

Drawing a second ace from a deck given we got the first ace

Finding the probability of having a disease given you were tested positive

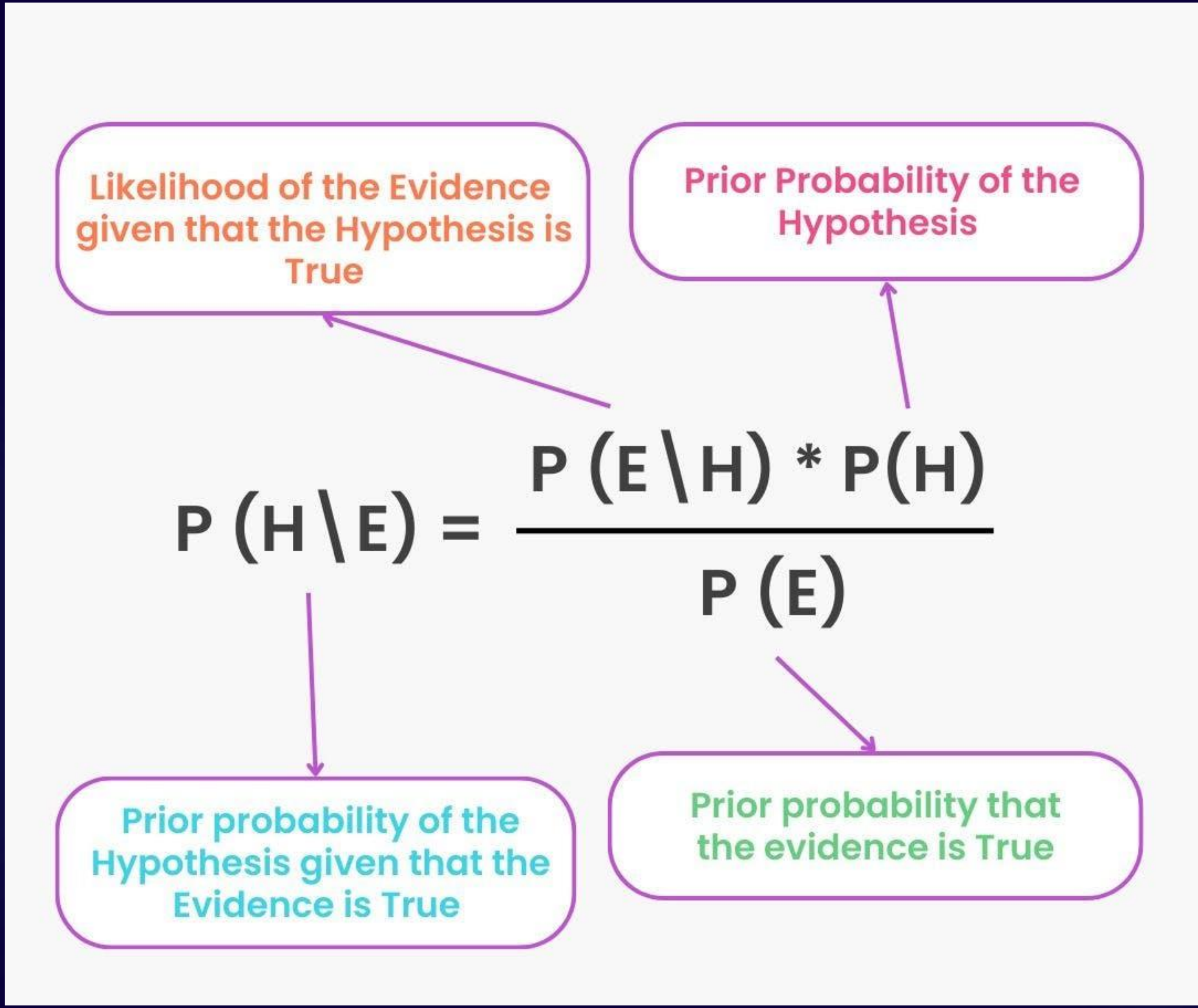
Finding the probability of liking Harry Potter given we know the person likes fiction.

Bayes Theorem

1. In probability theory and statistics, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

2. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately to assess the probability that they have cancer, compared to the assessment of the probability of cancer made without knowledge of the person's age.

Bayes Theorem



Bayes Theorem

The Formula for Bayes' theorem

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

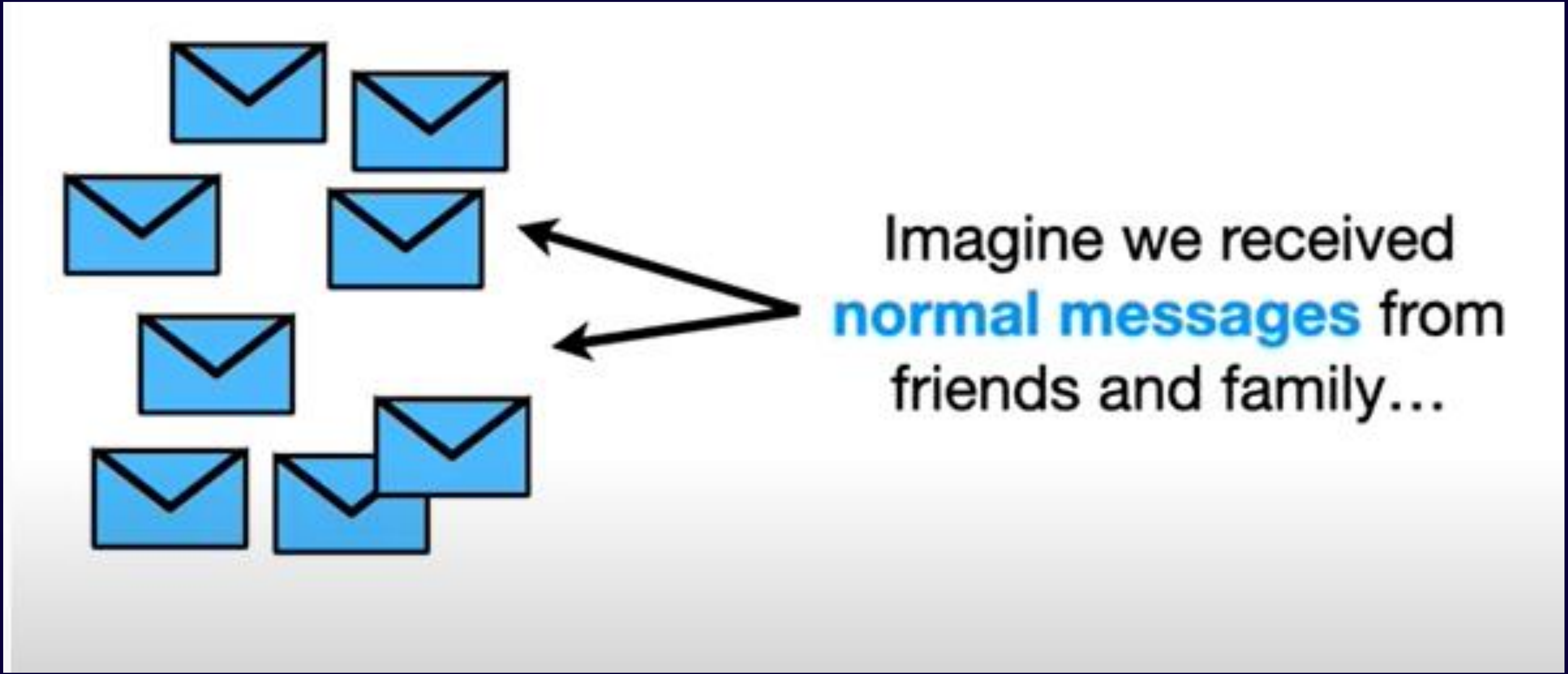
where

1. P(H) is the probability of hypothesis H being true. This is known as the prior probability.
2. P(E) is the probability of the evidence(regardless of the hypothesis).
3. P(E|H) is the probability of the evidence given that hypothesis is true.
4. P(H|E) is the probability of the hypothesis given that the evidence is there.

Naïve Bayes

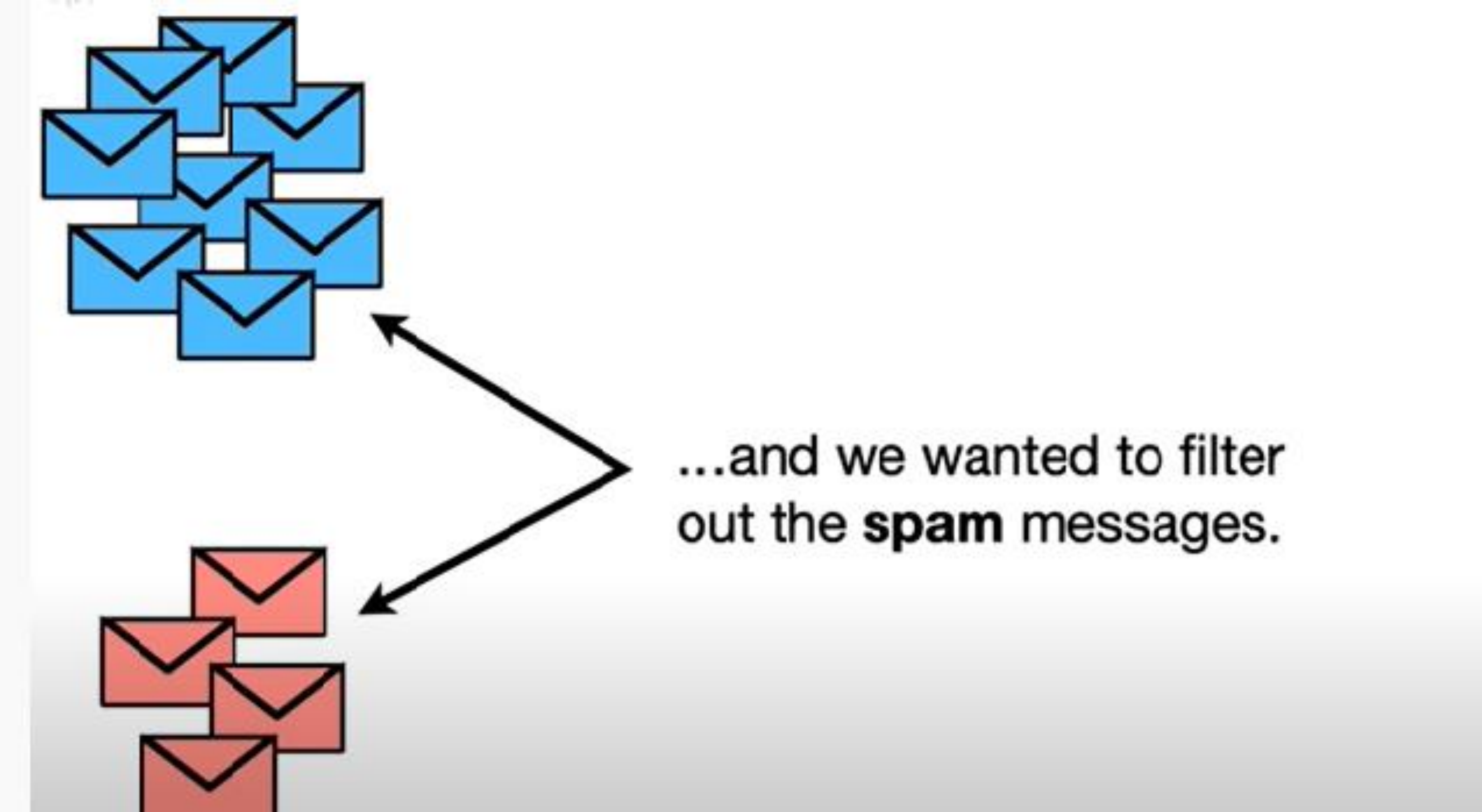
- 3 types
- Gaussian Naïve Bayes
- Binomial Naïve Bayes
- Multinomial Naïve Bayes

Email Classification

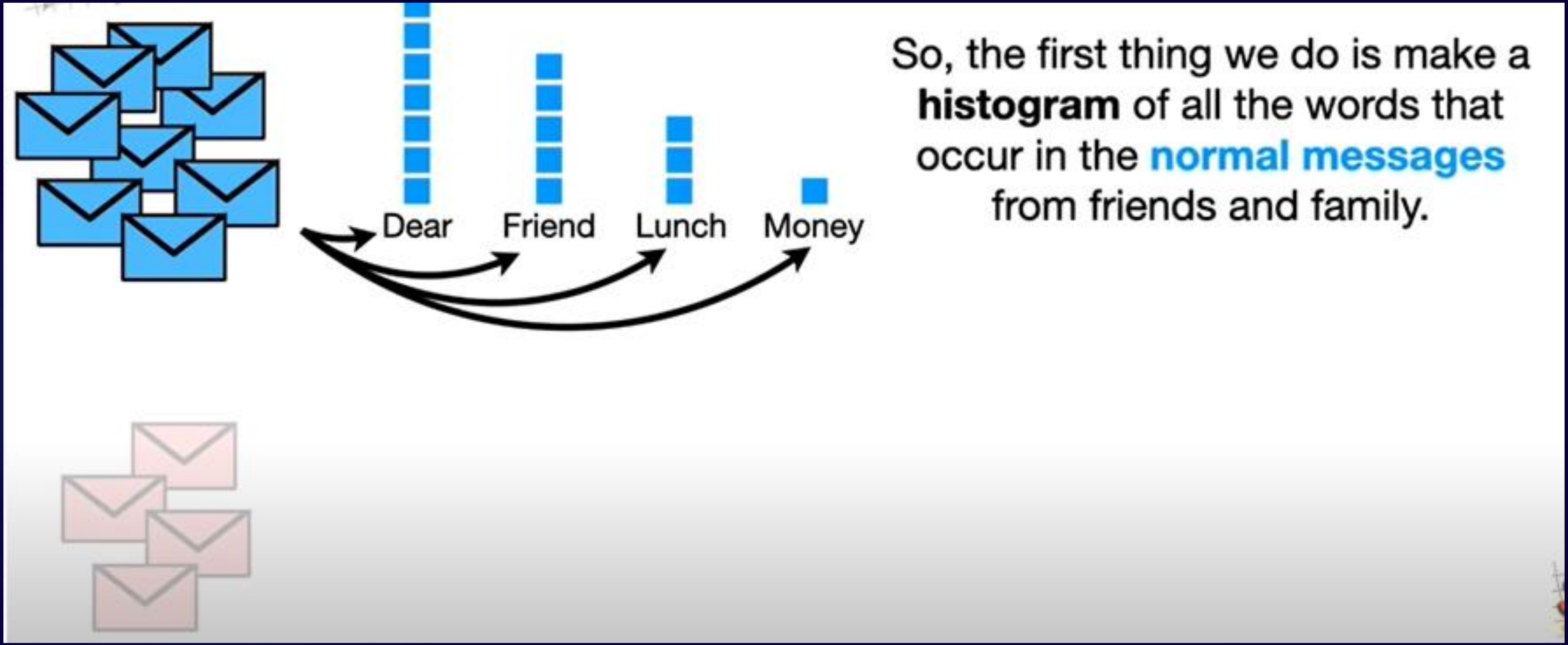


Email Classification

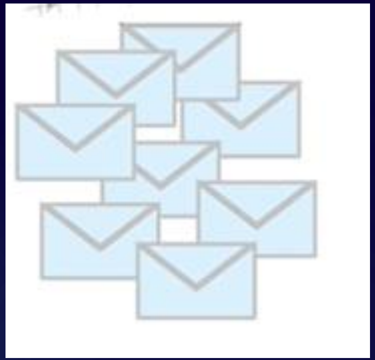
Email Classification

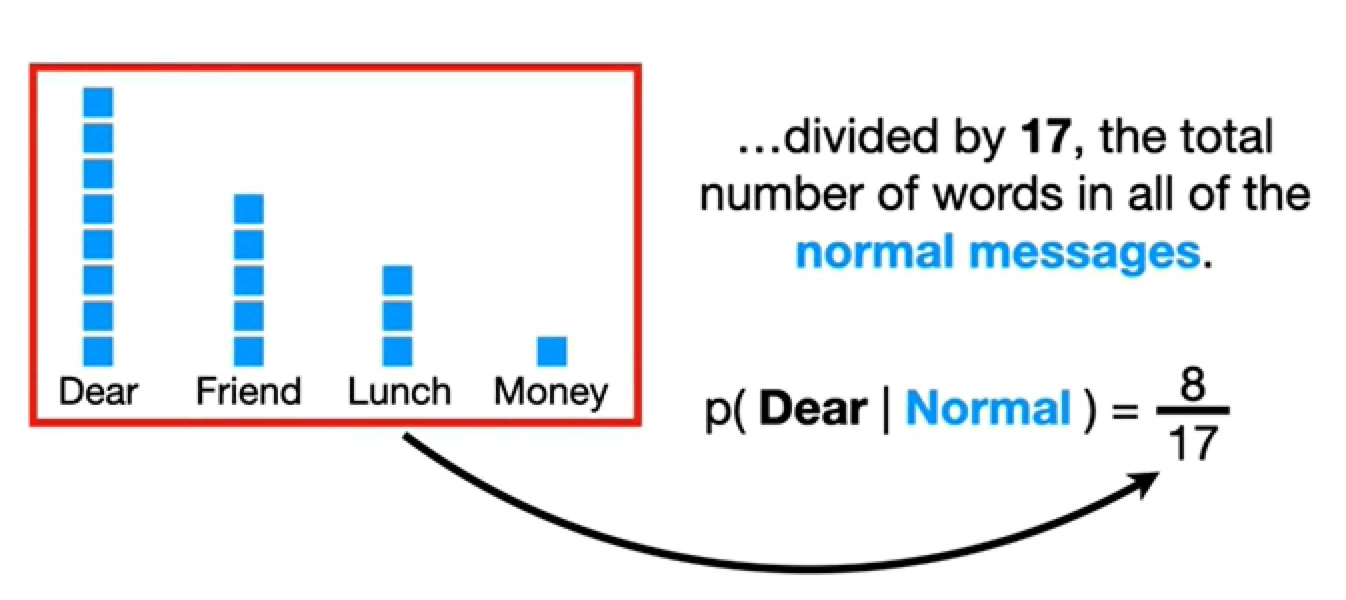


Email Classification

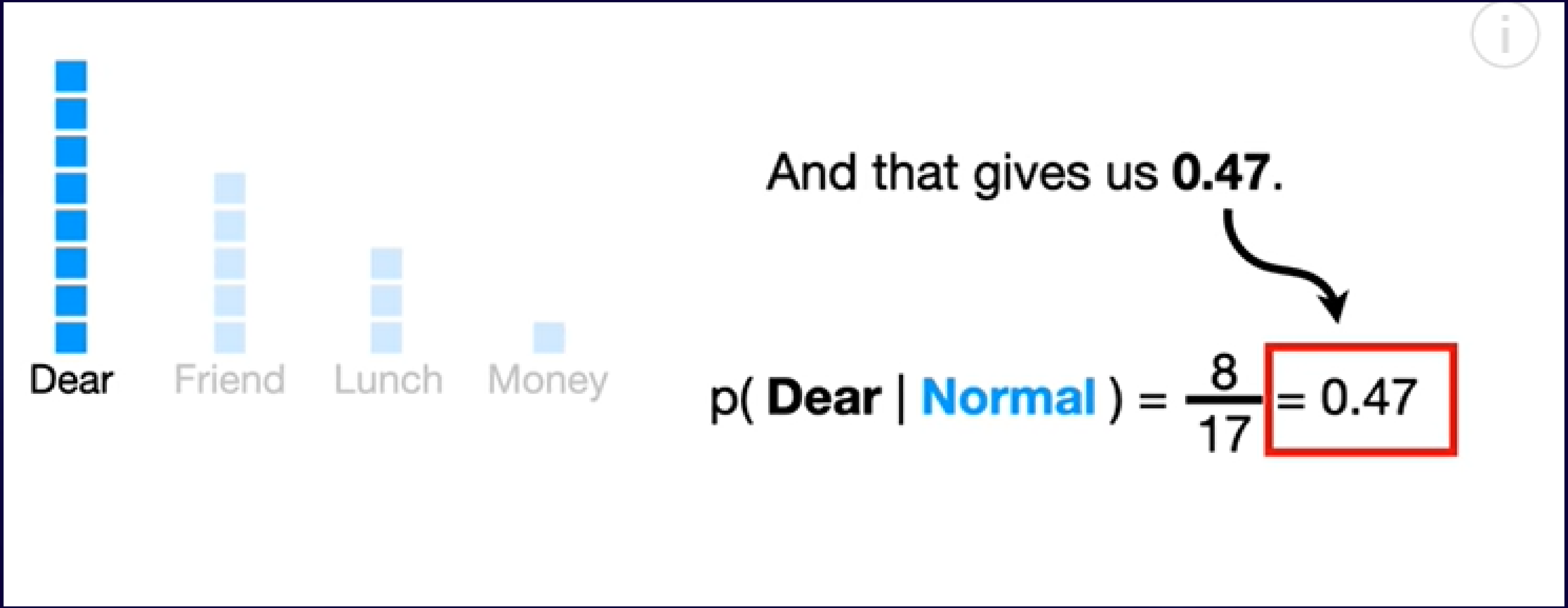


Email Classification

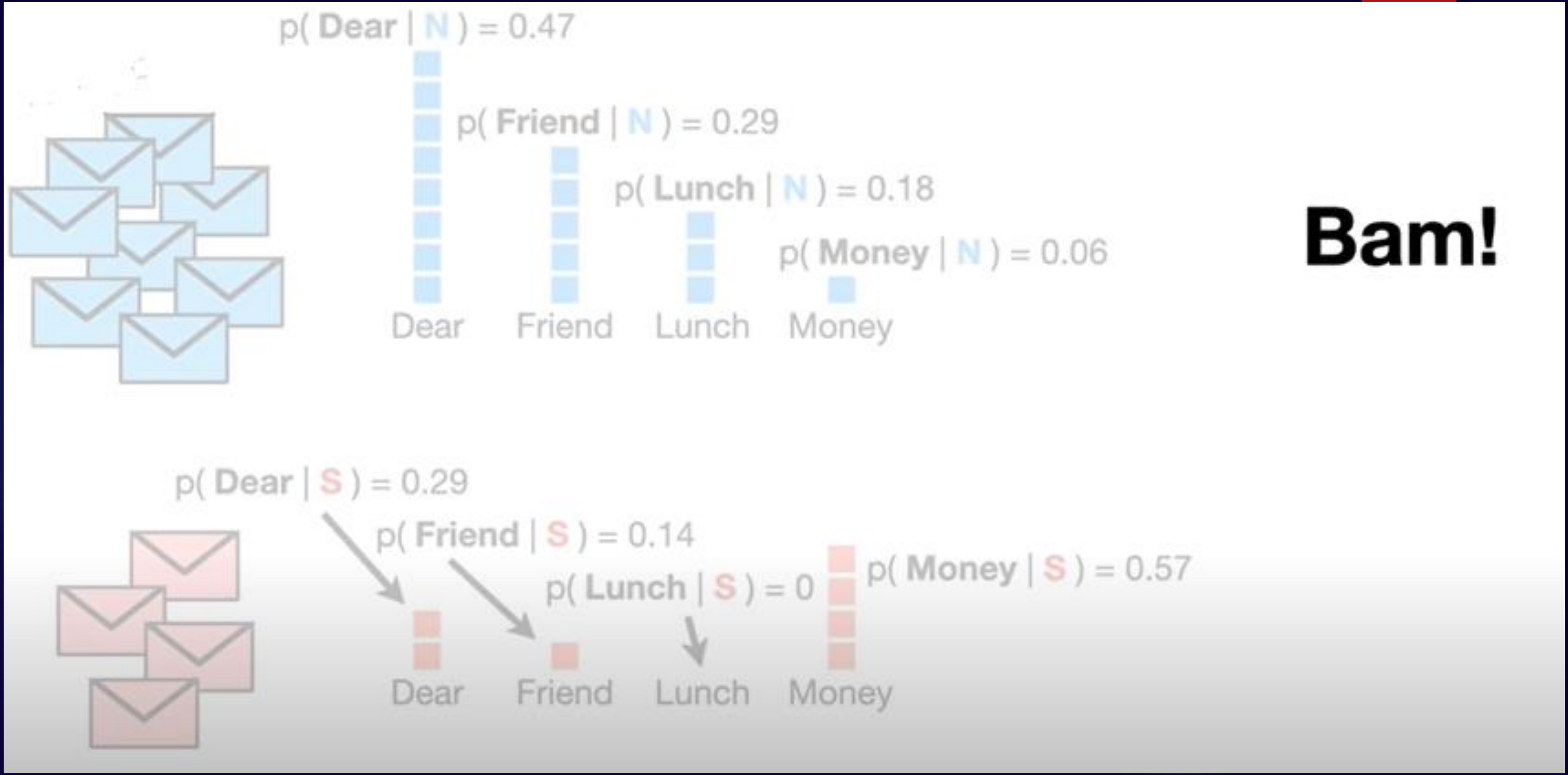




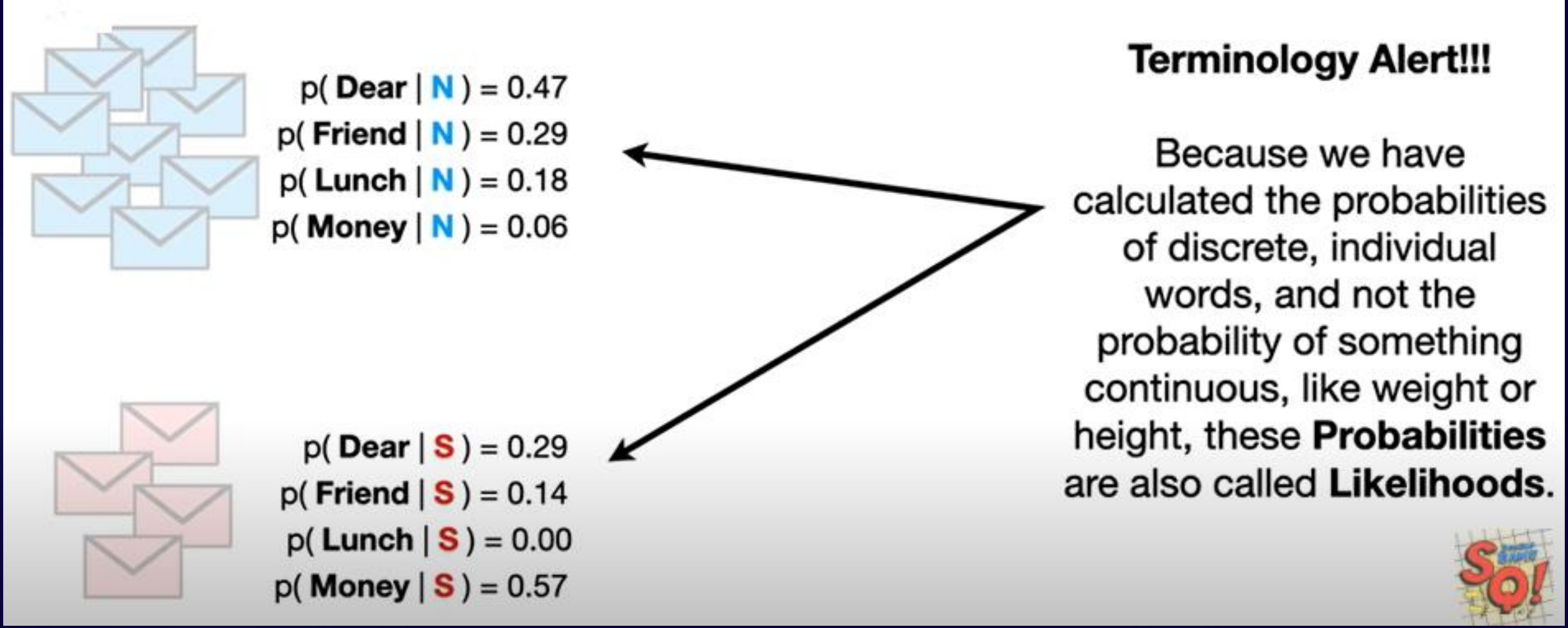
Email Classification



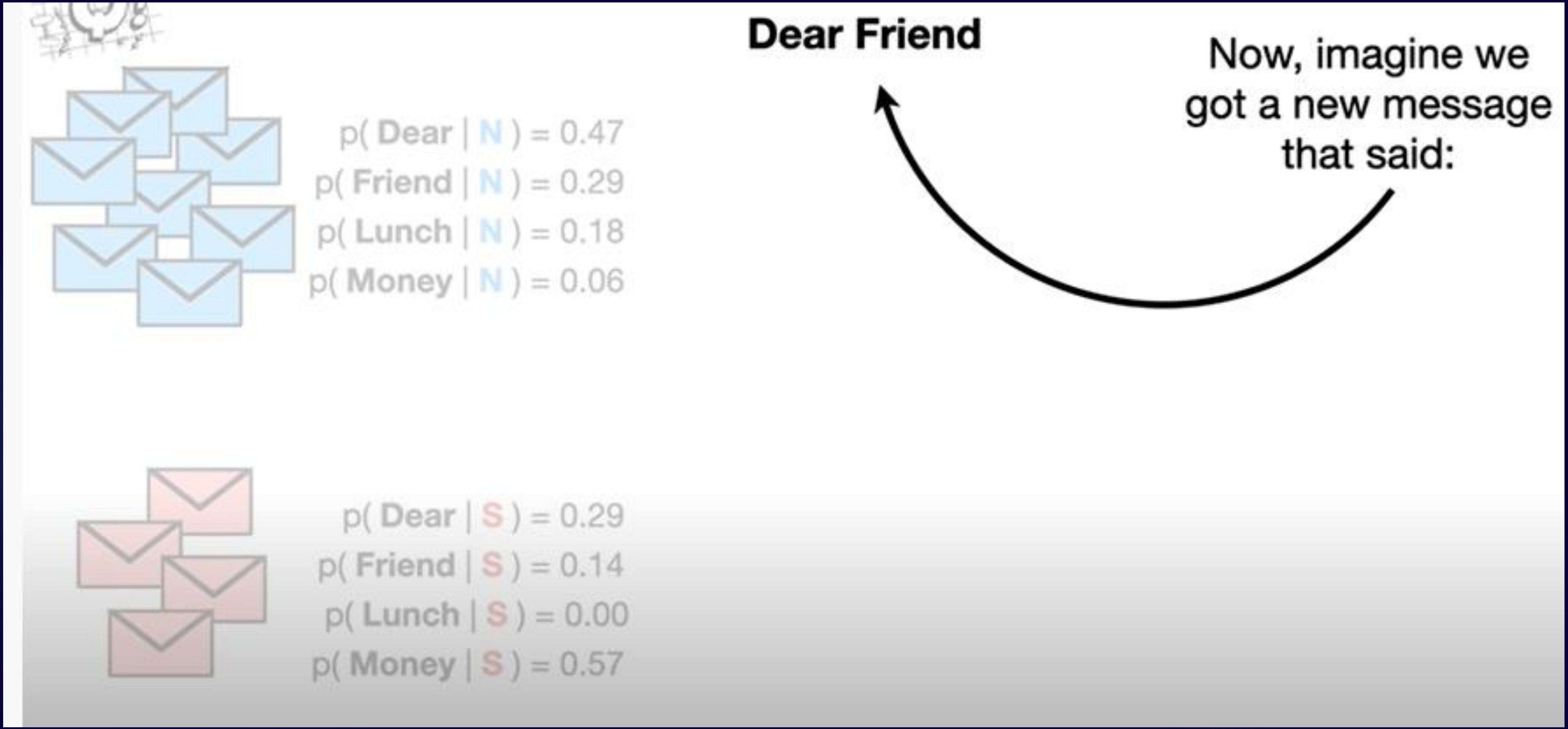
Email Classification



Email Classification

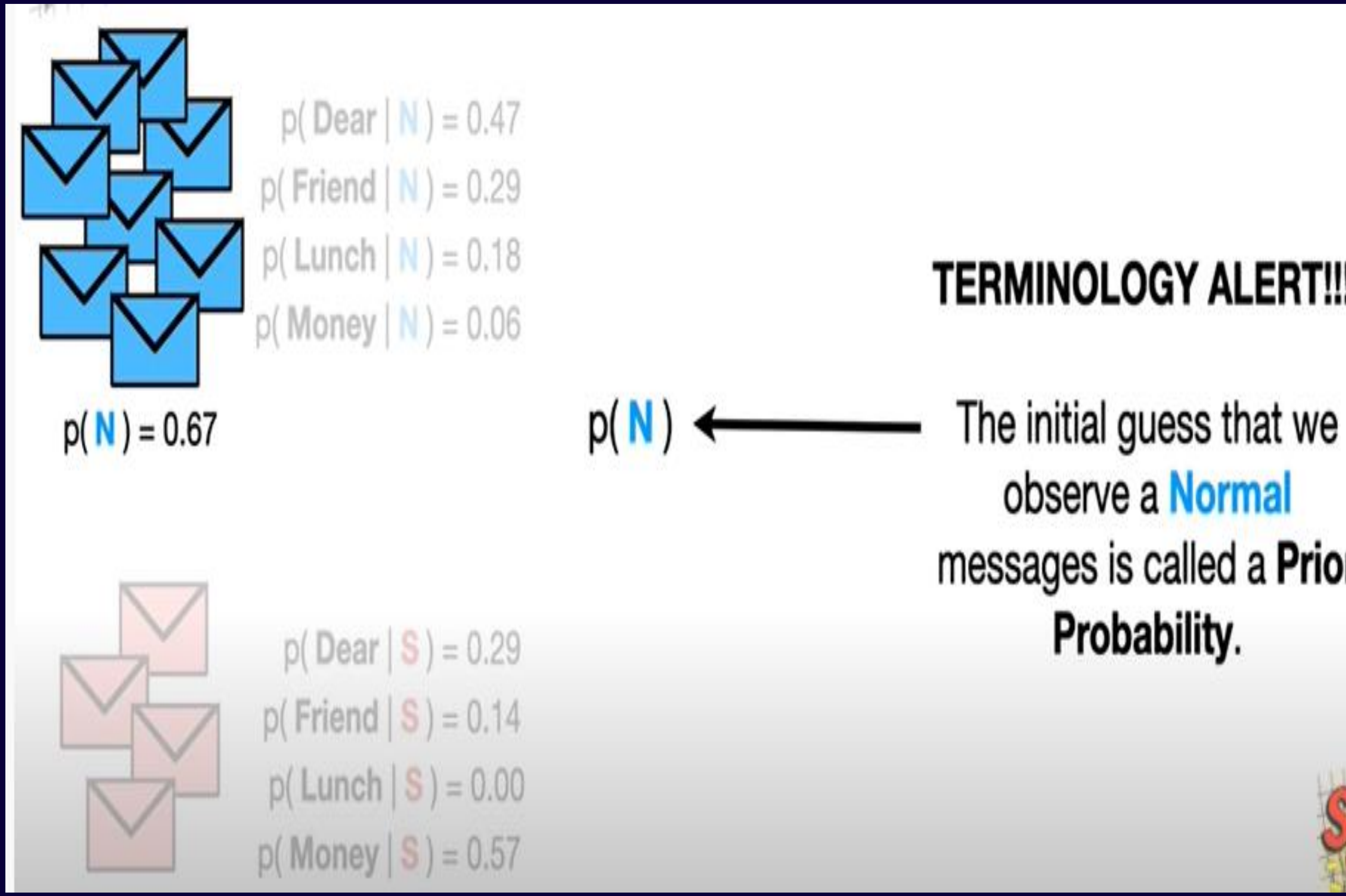


Email Classification

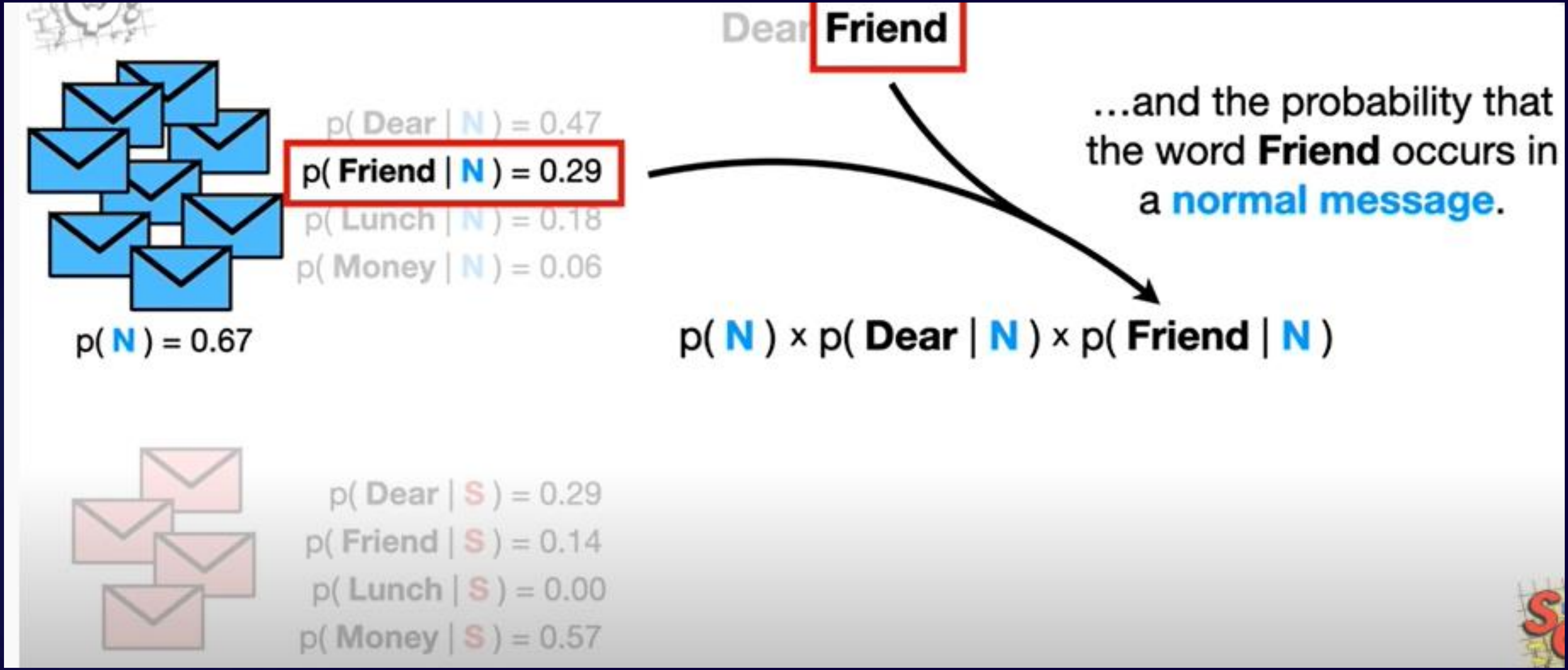


Email Classification

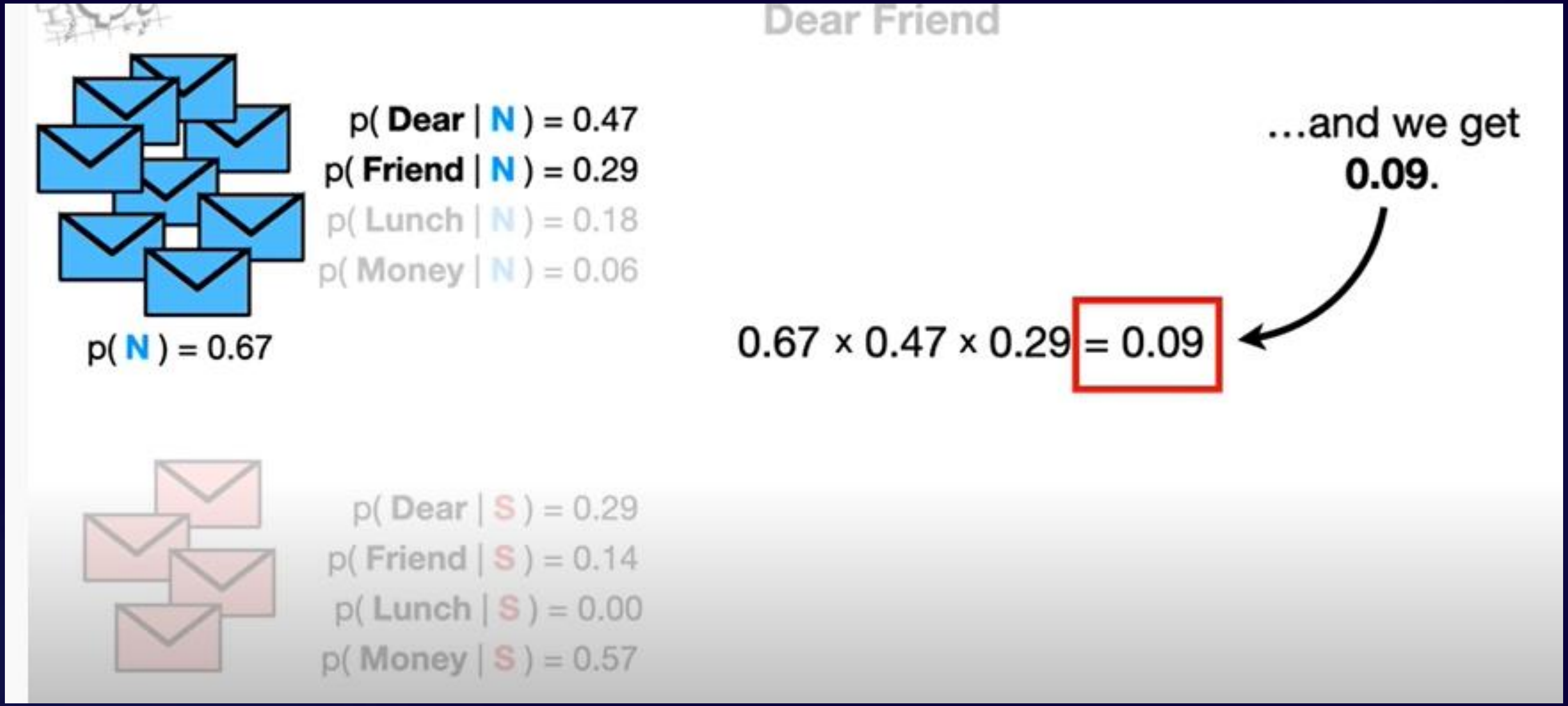
- Start with an initial guess of the probability it is a normal message
- Let 8 normal messages
- 12 total messages
- This P(N) = 8/12 =0.67



Email Classification

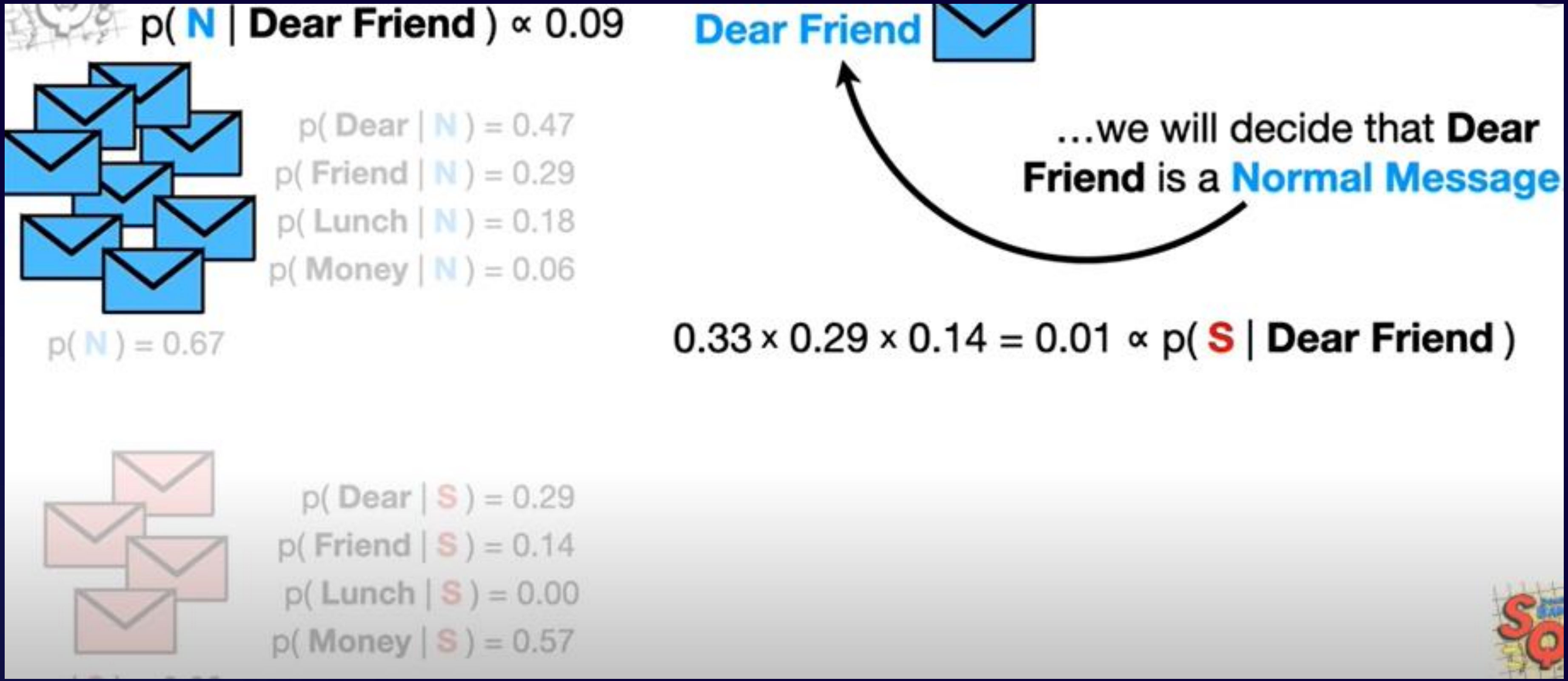


Email Classification

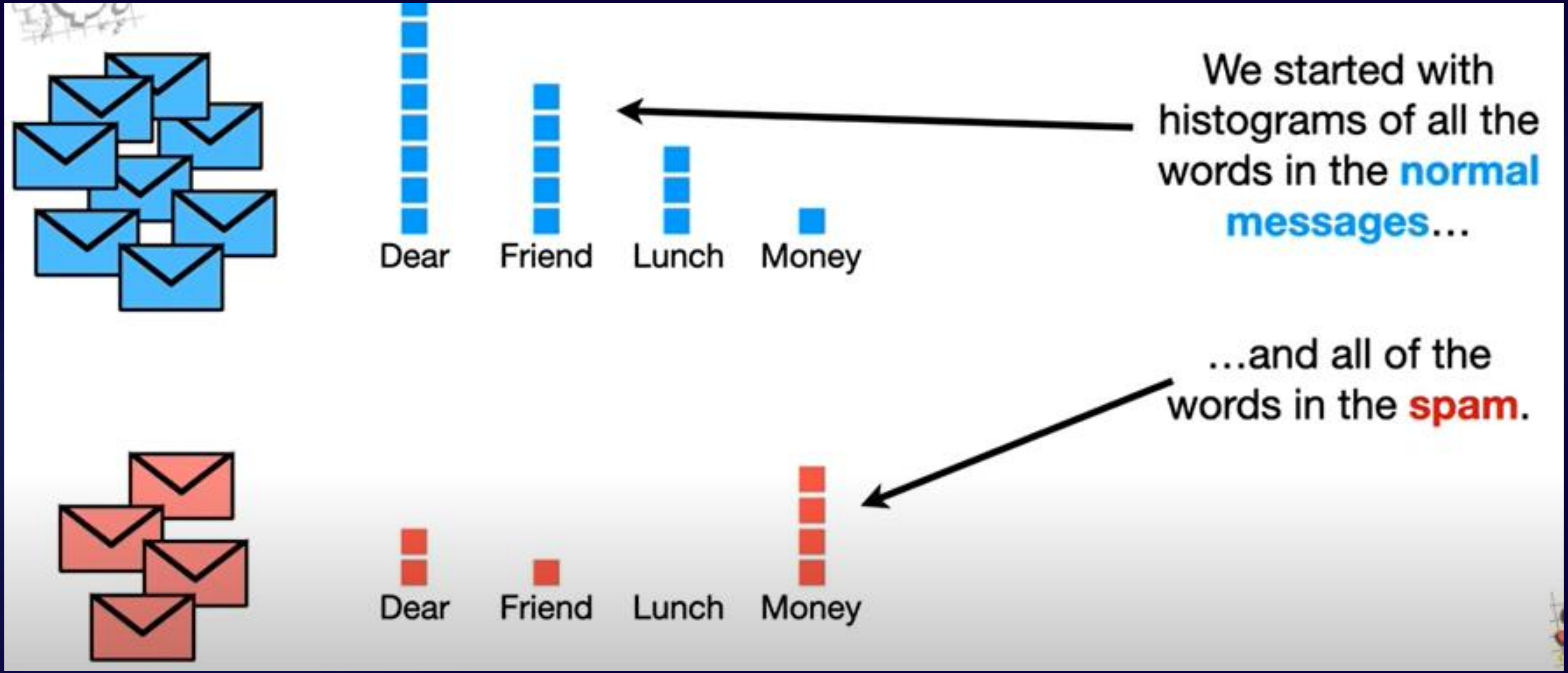


P(S) = no. of spam messages / total messages = 4/12 = 0.33
Think of .01 as the score that Dear Friend gets if it is Spam
P(S | Dear Friend) = P(S) * P(Dear | S) * P(Friend | S)
=0.33 * 0.29 * 0.14 = 0.01

Email Classification



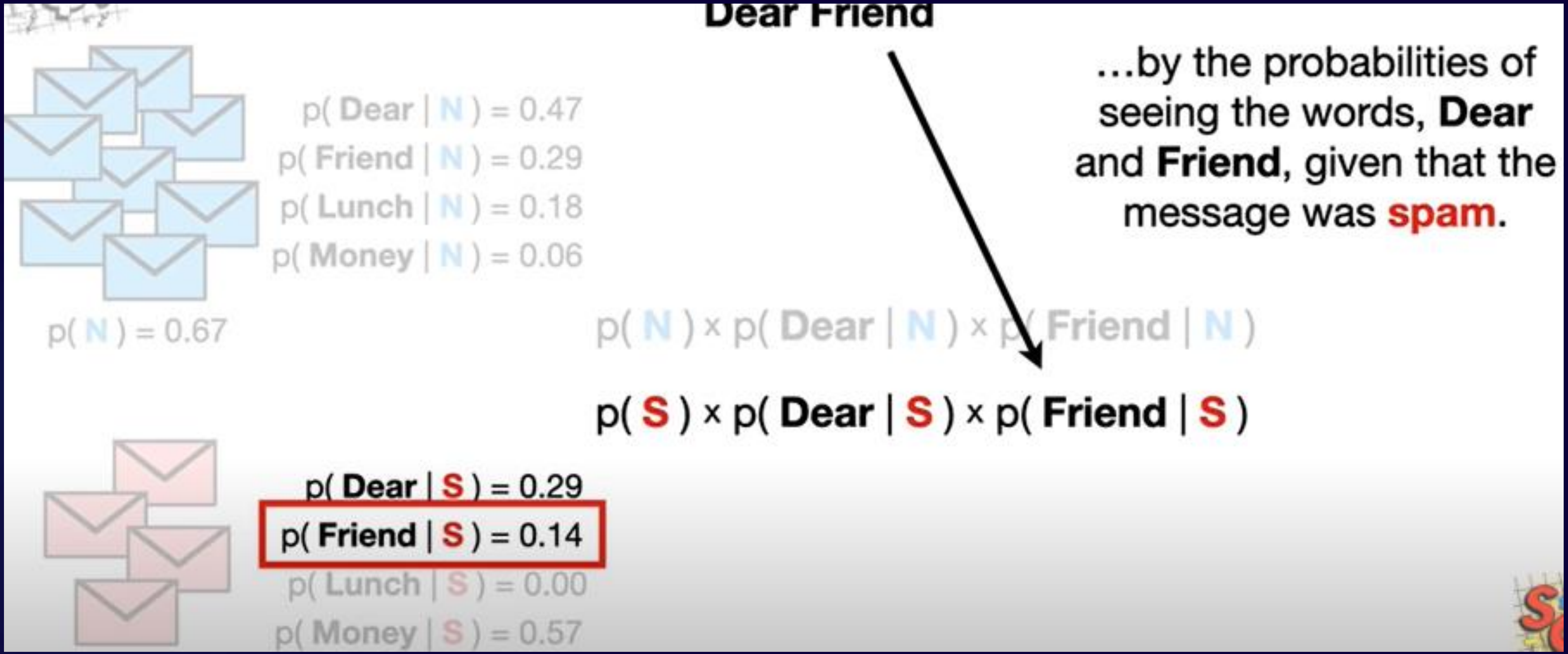
Recap


Dear
Friend
Lunch
Money
We started with histograms of all the words in the normal messages…
…and all of the words in the spam.
Dear
Friend
Lunch
Money

Recap

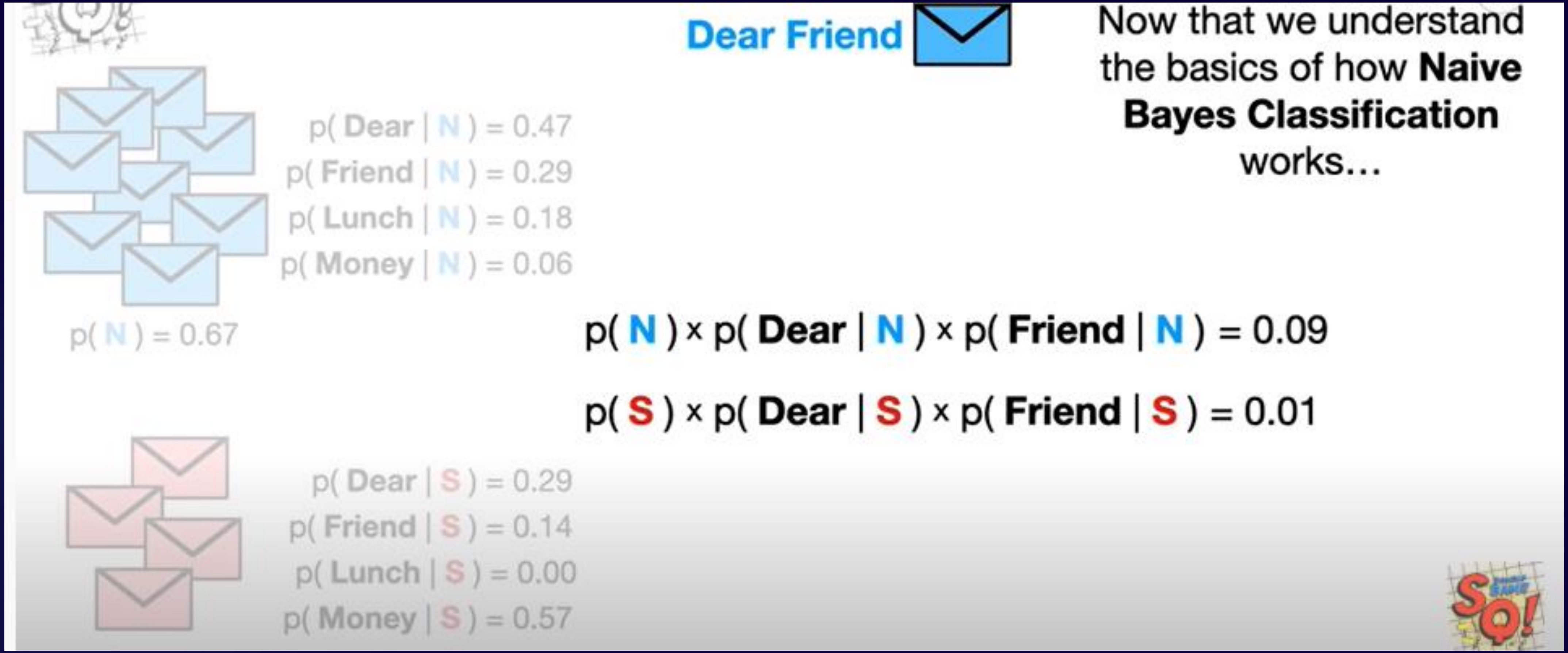
- We started with histogram of words in normal message and spam
- Calculate probability of each word in normal message or spam
- Calculate probability of seeing a normal message
- This is based on training dataset
- Calculate probability of seeing a spam

Recap

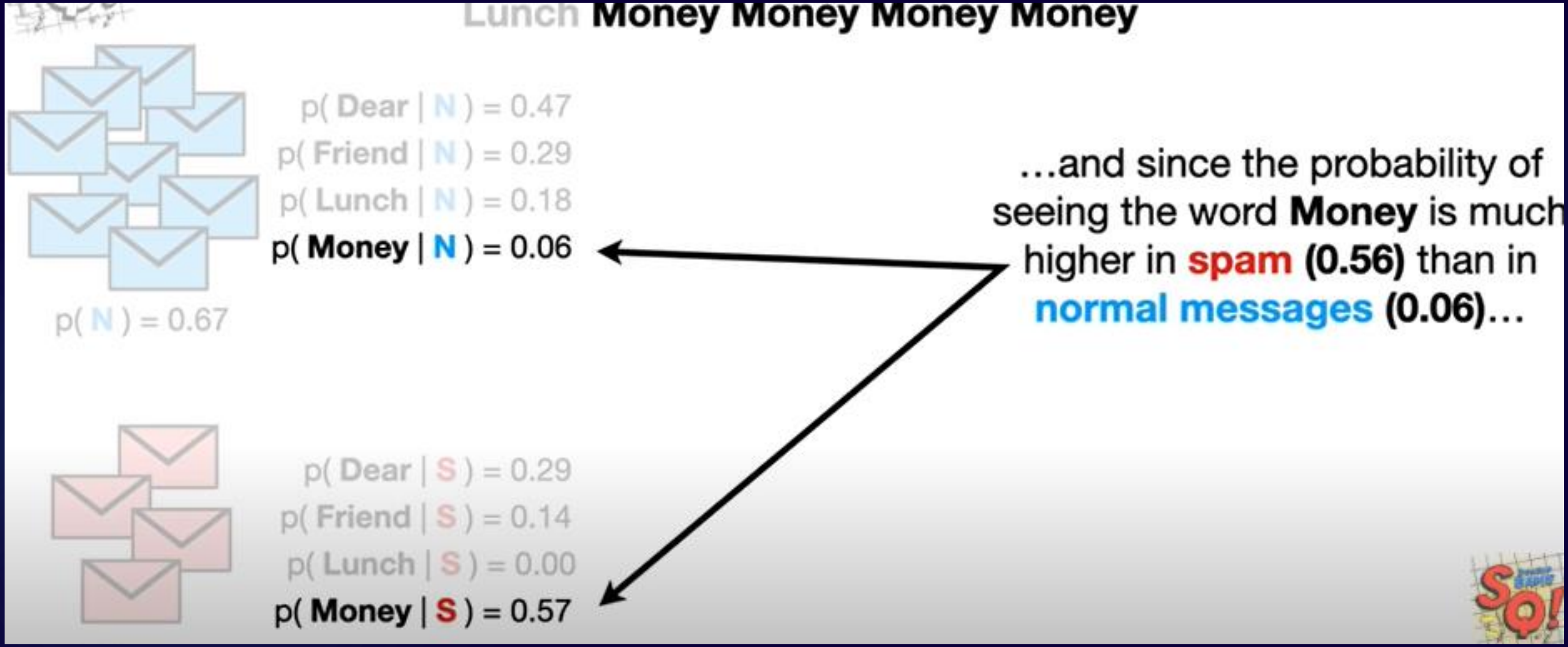


Dear Friend
...by the probabilities of seeing the words, Dear and Friend, given that the message was spam.
p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N) = 0.06
p(N) = 0.67
p(N) x p(Dear | N) x p(Friend | N)
p(S) x p(Dear | S) x p(Friend | S)
p(Dear | S) = 0.29
p(Friend | S) = 0.14
p(Lunch | S) = 0.00
p(Money | S) = 0.57

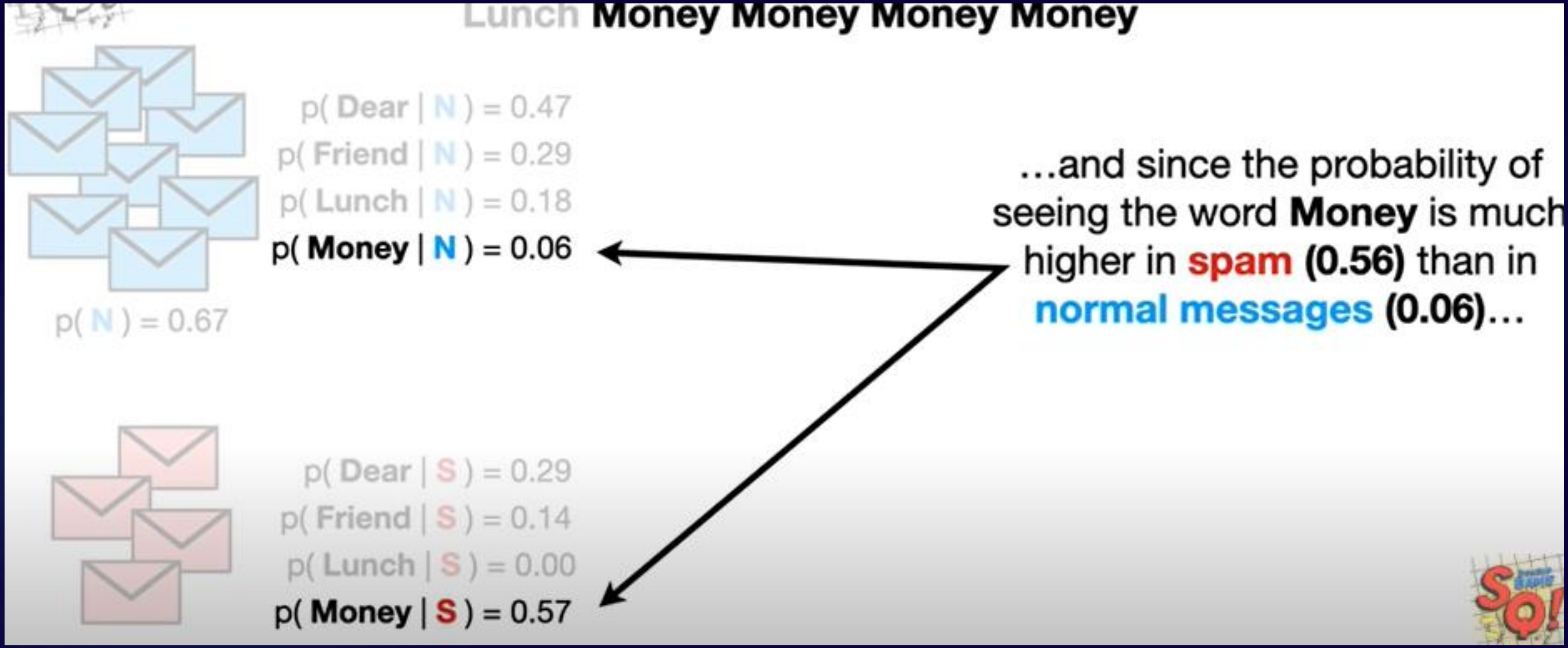
Recap


Dear Friend
Now that we understand the basics of how Naive Bayes Classification works...
p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N) = 0.06
p(N) = 0.67
p(N) x p(Dear | N) x p(Friend | N) = 0.09
p(S) x p(Dear | S) x p(Friend | S) = 0.01
p(Dear | S) = 0.29
p(Friend | S) = 0.14
p(Lunch | S) = 0.00
p(Money | S) = 0.57

Question

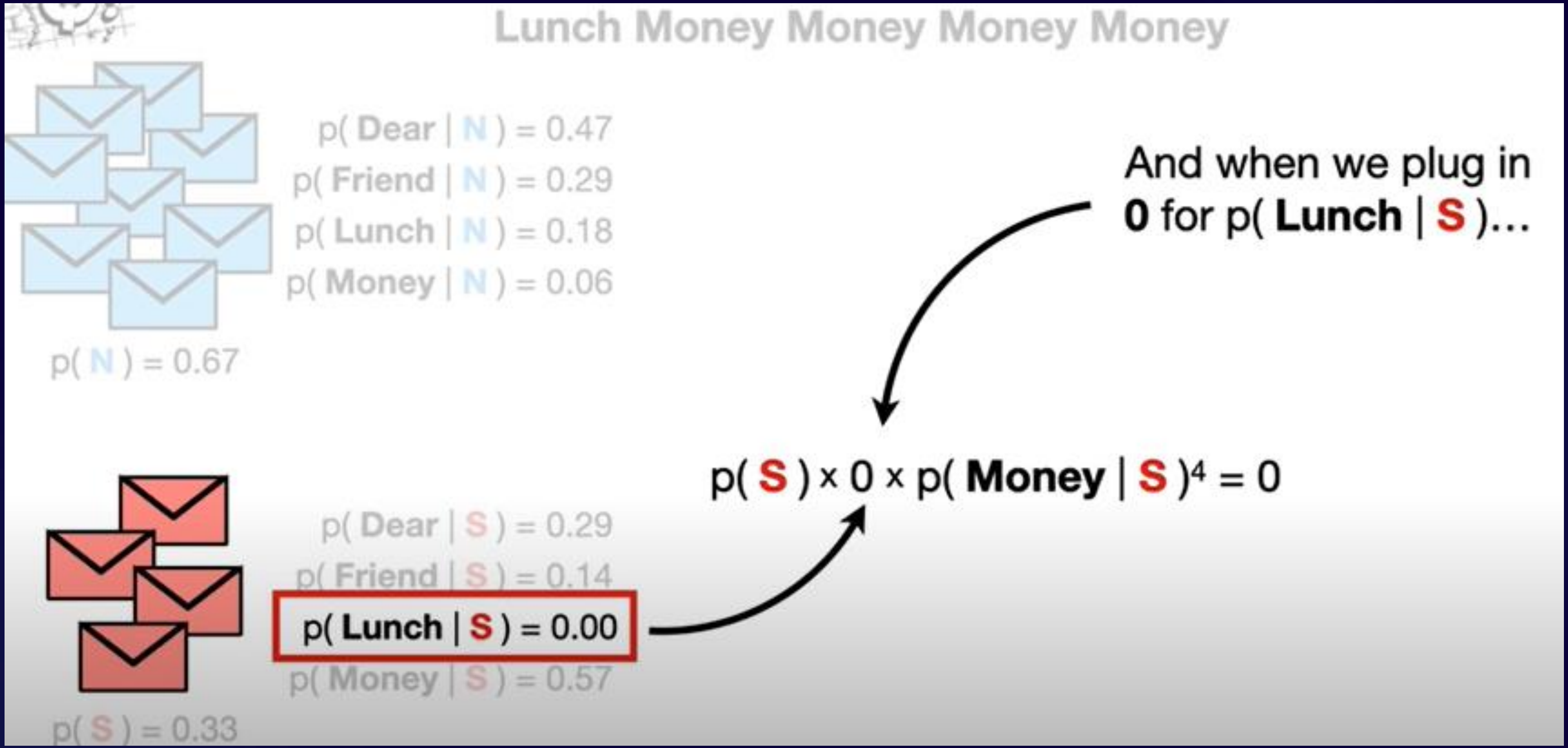

Lunch Money Money Money Money
p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N) = 0.06
p(N) = 0.67
...and since the probability of seeing the word Money is much higher in spam (0.56) than in normal messages (0.06)...
p(Dear | S) = 0.29
p(Friend | S) = 0.14
p(Lunch | S) = 0.00
p(Money | S) = 0.57

Question



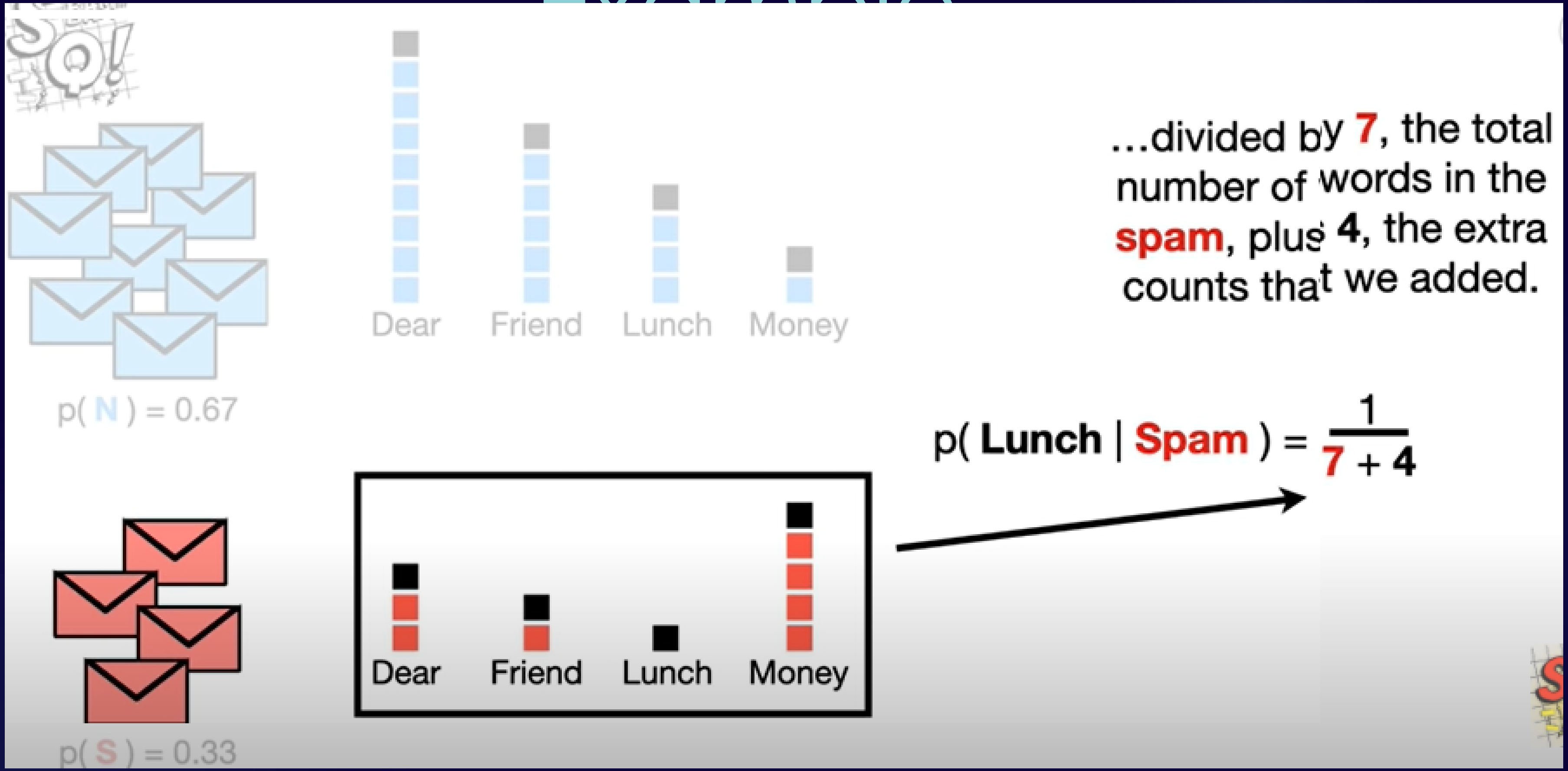
Lunch Money Money Money Money
p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N) = 0.06
p(N) = 0.67
...and since the probability of seeing the word Money is much higher in spam (0.56) than in normal messages (0.06)...
p(Dear | S) = 0.29
p(Friend | S) = 0.14
p(Lunch | S) = 0.00
p(Money | S) = 0.57

Example



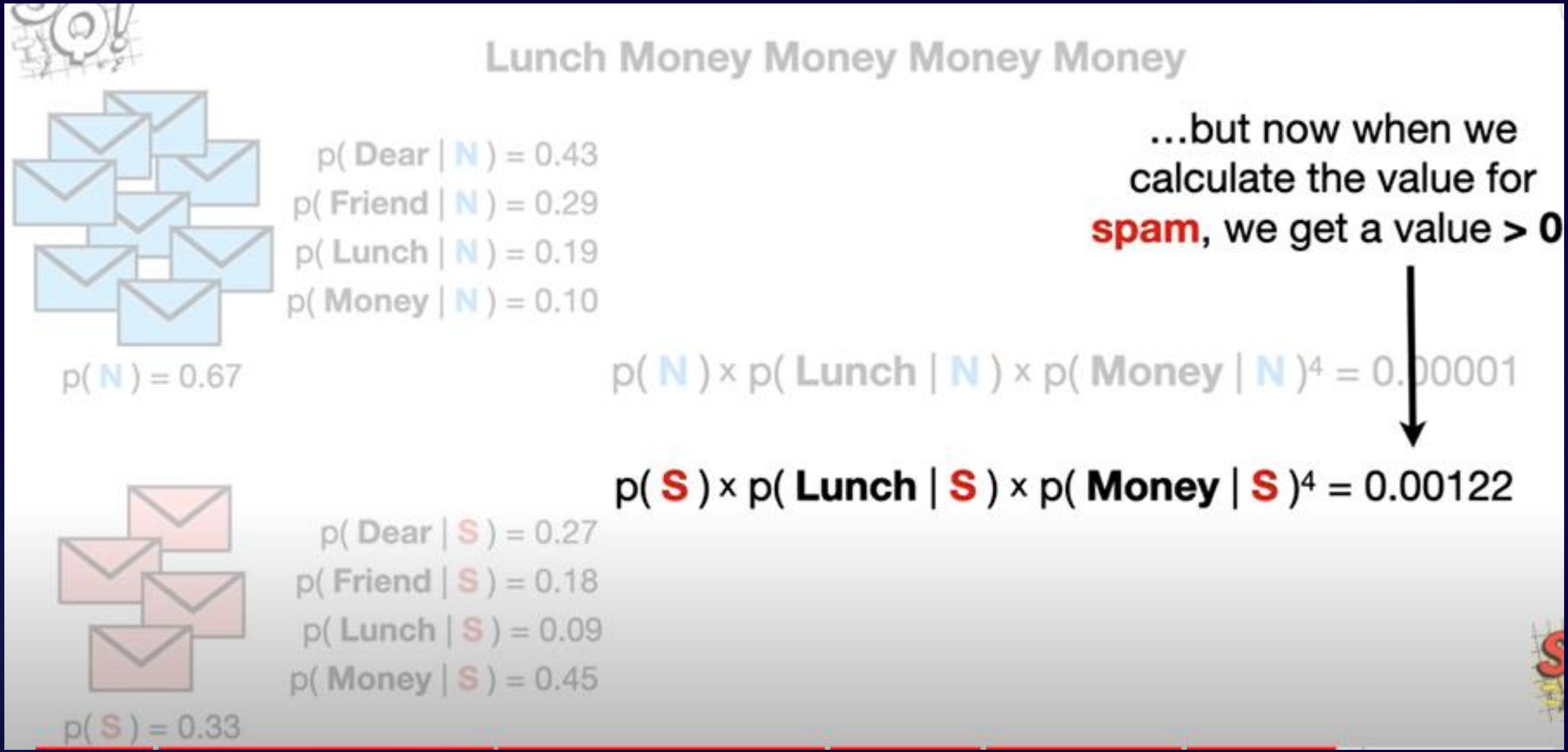
P(S) = 0 here since we have not yet seen the word lunch yet in spam

Example



We add 1/any number to all words
Now P(Lunch | spam) becomes non zero

Example

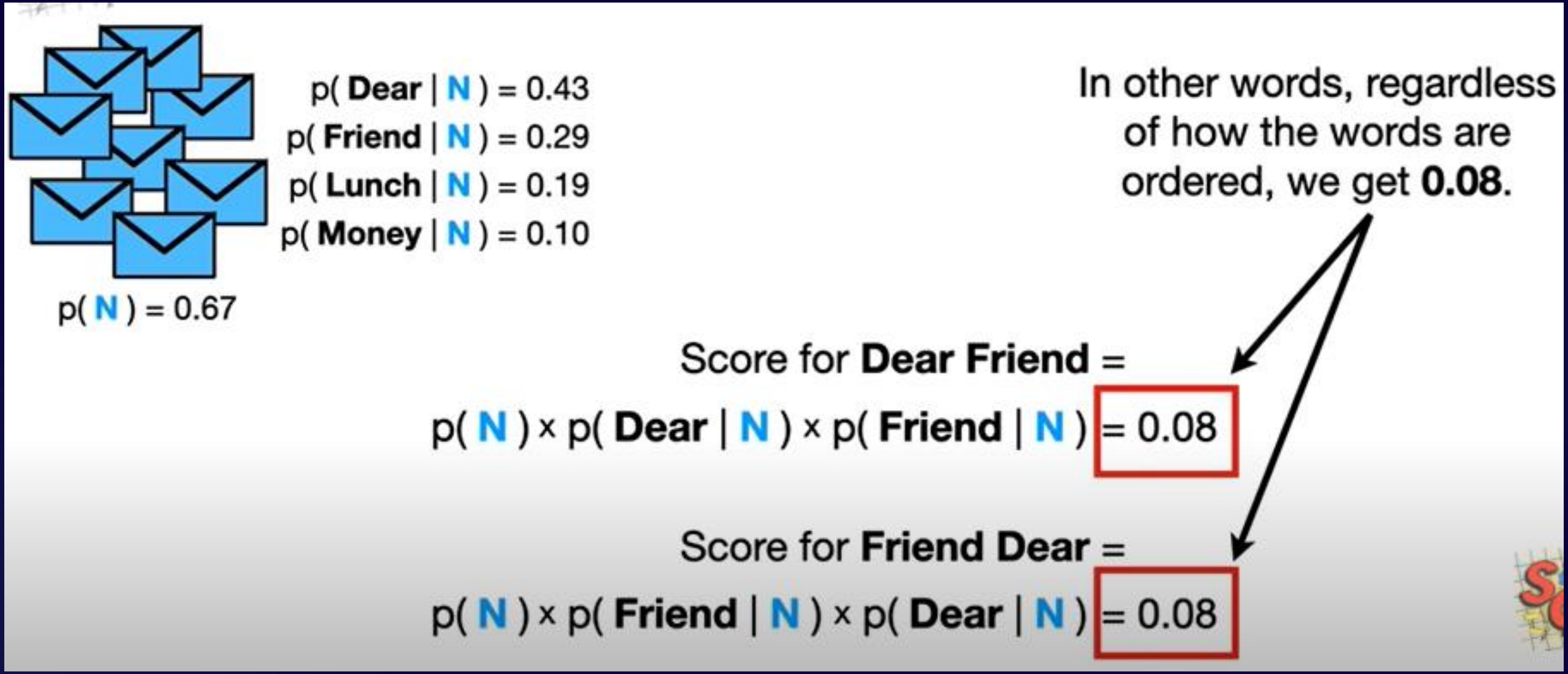


Now we calculate the probability of message being spam/normal
We get P(S) > P(N)
So we say that message Lunch Money…. Is SPAM

SPAM!!!

Why Naïve Bayes is naive

Why Naïve Baiyes is Naive



- Treats all word orders the same
- Regardless of order of word dear friend/ friend dear
- Probability is 0.08

Why Naïve Baiyes is Naive

- Treating all word orders equal is very different from how we communicate
- Every language has grammar rules
- And common phrases but Naïve Bayes ignores all that
- Naïve Bayes treats language like a bag full of words
- We can say that Naïve Bayes has a high bias
- Has low variance

Theory Behind Naïve Bayes

Theory Behind Naïve Baiyes

Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

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

Theory Behind Naïve Baiyes

Naive assumption

Now, its time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into the independent parts.

Now, if any two events A and B are independent, then,

```
P(A,B) = P(A)P(B)
```

Theory Behind Naïve Baiyes

Hence, we reach to the result:

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

which can be expressed as:

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, as the denominator remains constant for a given input, we can remove that term:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

Question

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Now which tag does the sentence A very close game belong to?

In our case, we have P (Sports | a very close game), so using this theorem we can reverse the conditional probability:

$$P(sports|a\,very\,close\,game) = \frac{P(a\,very\,close\,game|sports) \times P(sports)}{P(a\,very\,close\,game)}$$

Since for our classifier we're just trying to find out which tag has a bigger probability, we can discard the divisor —which is the same for both tags— and just compare

$$P(a\,very\,close\,game|Sports) \times P(Sports)$$

with

$$P(a\,very\,close\,game|Not\,Sports) \times P(Not\,Sports)$$

Question

$$P(a\,very\,close\,game) = P(a) \times P(very) \times P(close) \times P(game)$$

This assumption is very strong but super useful. It's what makes this model work well with little data or data that may be mislabeled. The next step is just applying this to what we had before:

$$P(a\,very\,close\,game|Sports) = P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports)$$

Calculating Probabilities

- P (Sports) is ⅗. Then, P (Not Sports) is ⅖
- Then, calculating P (game | Sports)
- Count of word "game" appears in Sports texts (2) divided by the total number of words in sports (11) Therefore,
- P(game|sports) =2/11
- "close" doesn't appear in any *Sports* text!
- P (close | Sports) = 0

$$P(a|Sports) \times P(very|Sports) \times 0 \times P(game|Sports)$$

Calculating Probabilities

- To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1

possible words are `['a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match']`.

Calculating Probabilities

$$P(game|sports) = \frac{2+1}{11+14}$$

The full results are:

Word	P (word | Sports)	P (word | Not Sports)
a	(2 + 1) ÷ (11 + 14)	(1 + 1) ÷ (9 + 14)
very	(1 + 1) ÷ (11 + 14)	(0 + 1) ÷ (9 + 14)
close	(0 + 1) ÷ (11 + 14)	(1 + 1) ÷ (9 + 14)
game	(2 + 1) ÷ (11 + 14)	(0 + 1) ÷ (9 + 14)

Calculating Probabilities

Now we just multiply all the probabilities, and see who is bigger:

$$P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports) \times P(Sports)$$
$$= 2.76 \times 10^{-5}$$
$$= 0.0000276$$

$$\mathrm{P}(\mathrm{a}\ |Not\ Sports) \times P(very|Not\ Sports) \times P(close|Not\ Sports) \times P(game|Not\ Sports) \times P(Not\ Sports)$$
$$= 0.572 \times 10^{-5}$$
$$= 0.00000572$$

Excellent! Our classifier gives “A very close game” the **Sports** tag.

Part of Speech Tagging

Part Of Speech Tagging

- Is the task of assigning POS tags to words
- Selecting among more than one tags that apply
- Can be used for further NLP tasks
- Information extraction, Question Answering etc.

Part Of Speech Tagging

- Since the greeks 8 basic POS have been distinguished
  - Noun, verb, pronoun, preposition, adverb,
  - conjunction, adjective, article
- Modern works use extended list of POS:
  - 45 in Penn Treebank corpus
  - 87 in Brown corpus
- Used for syntactic processing
  - Speech recognition – Pronounciation may change

Part Of Speech Categories

__Closed class.__ Function words: prepositions, pronouns, determiners,conjunctions, numerals, auxiliary verbs and particles (preposition or adverbs in phrasal verbs)

__Open class__:

Nouns: people, place and things proper nouns, common nouns,count nouns and mass nouns

Verbs: actions and processes. Main verbs, not auxiliaries

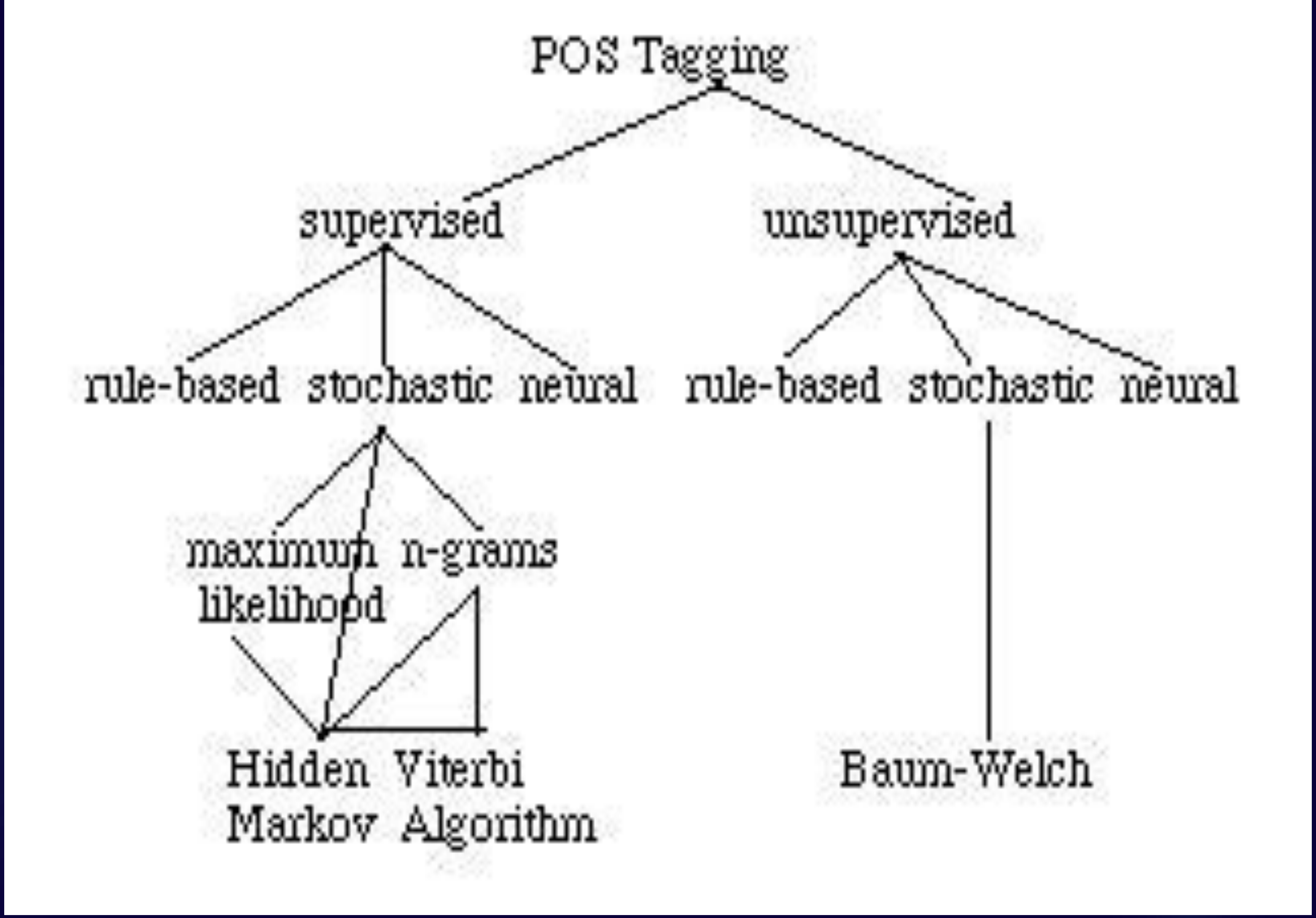
Adjectives: Properties

Adverbs

Existing Taggers

- Rule based tagging
  - Brill tagger
- Stochastic tagging
  - CLAWS tagger
  - Tree tagger

Existing Taggers



Rule-Based Tagger

- The Linguistic Complaint
  - Where is the linguistic knowledge of a tagger?
  - Just a massive table of numbers
  - Aren't there any linguistic insights that could emerge from the data?
  - Could thus use handcrafted sets of rules to tag input sentences, for example, if input follows a determiner tag it as a noun.

Modified from Diane Litman's version of Steve Bird's notes

The Brill tagger

- An example of Transformation-Based Learning
  - Basic idea: do a quick job first (using frequency), then revise it using contextual rules.
  - Painting metaphor from the readings
- Very popular (freely available, works fairly well)
- A supervised method: requires a tagged corpus

Slide modified from Massimo Poesio's

Brill Tagging: In more detail

- Start with simple (less accurate) rules…learn better ones from tagged corpus
  - Tag each word initially with most likely POS
  - Examine set of transformations to see which improves tagging decisions compared to tagged corpus
  - Re-tag corpus using best transformation
  - Repeat until, e.g., performance doesn't improve
  - Result: tagging procedure (ordered list of transformations) which can be applied to new, untagged text

An example

- Examples:
  - They are expected to race tomorrow.
  - The race for outer space.
- Tagging algorithm:
  1. Tag all uses of "race" as NN (most likely tag in the Brown corpus)
     - They are expected to race/NN tomorrow
     - the race/NN for outer space
  2. Use a transformation rule to replace the tag NN with VB for all uses of "race" preceded by the tag TO:
     - They are expected to race/VB tomorrow
     - the race/NN for outer space

Slide modified from Massimo Poesio's

Example Rule Transformations

```
             B       |
   S    F    r    O  |         Score = Fixed - Broken
   c    i    o    t  |   R     Fixed = num tags changed incorrect -> correct
   o    x    k    h  |   u     Broken = num tags changed correct -> incorrect
   r    e    e    e  |   l     Other = num tags changed incorrect -> incorrect
   e    d    n    r  |   e
-------------------+----------------------------------------------------------
  10   18    8   14  | NN -> NNP if the tag of words i+1...i+2 is 'NNP'
   9   10    1    2  | NN -> VB if the tag of the preceding word is 'TO'
   8    9    1   18  | NN -> VBD if the tag of the following word is 'DT'
   7    7    0    9  | NN -> VBD if the tag of the preceding word is 'NNS'
   6   13    7    8  | NN -> JJ if the tag of the preceding word is 'DT', and th
e tag of the following word is 'NN'
   7    9    2    2  | NN -> NNP if the tag of the preceding word is 'NN', and t
he tag of the following word is ','
   8   16    8   12  | NN -> NNP if the tag of words i+1...i+2 is 'NNP'
   4    6    2   11  | NN -> IN if the tag of the preceding word is '.'
   3    4    1    2  | NNP -> NN if the tag of words i-3...i-1 is 'JJ'
   3    3    0    2  | NN -> JJ if the tag of the following word is 'JJ'
   3    3    0    4  | NN -> VBP if the tag of the preceding word is 'PRP'
   3    3    0    0  | WDT -> IN if the tag of the following word is 'DT'
```

Sample Final Rules

```
Rules:
NN -> NNP if the tag of words i+1...i+2 is 'NNP'
NN -> VB if the tag of the preceding word is 'TO'
NN -> VBD if the tag of the following word is 'DT'
NN -> VBD if the tag of the preceding word is 'NNS'
NN -> JJ if the tag of the preceding word is 'DT', and the tag of the followi
ng word is 'NN'
NN -> NNP if the tag of the preceding word is 'NN', and the tag of the follow
ing word is ','
NN -> NNP if the tag of words i+1...i+2 is 'NNP'
NN -> IN if the tag of the preceding word is '.'
NNP -> NN if the tag of words i-3...i-1 is 'JJ'
NN -> JJ if the tag of the following word is 'JJ'
NN -> VBP if the tag of the preceding word is 'PRP'
WDT -> IN if the tag of the following word is 'DT'
NN -> JJ if the tag of the preceding word is 'IN', and the tag of the followi
ng word is 'NN'
NN -> VBN if the tag of the preceding word is 'VBP'
VBD -> VB if the tag of the preceding word is 'MD'
NN -> JJ if the tag of the preceding word is 'CC', and the tag of the followi
ng word is 'NN'
```