Generative vs Discriminative models

A way to categorize learning algorithms

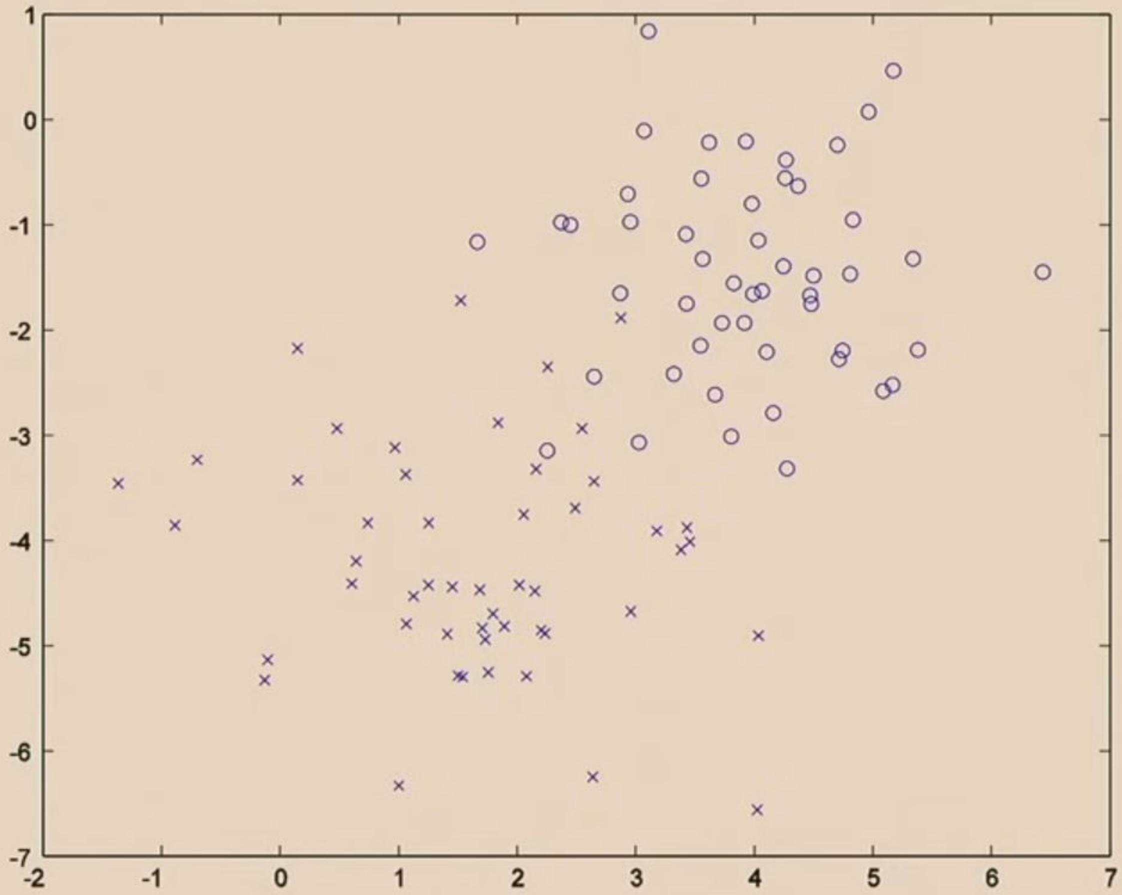
- Both can be used for classification
- Both are supervised methods

Discriminative Models

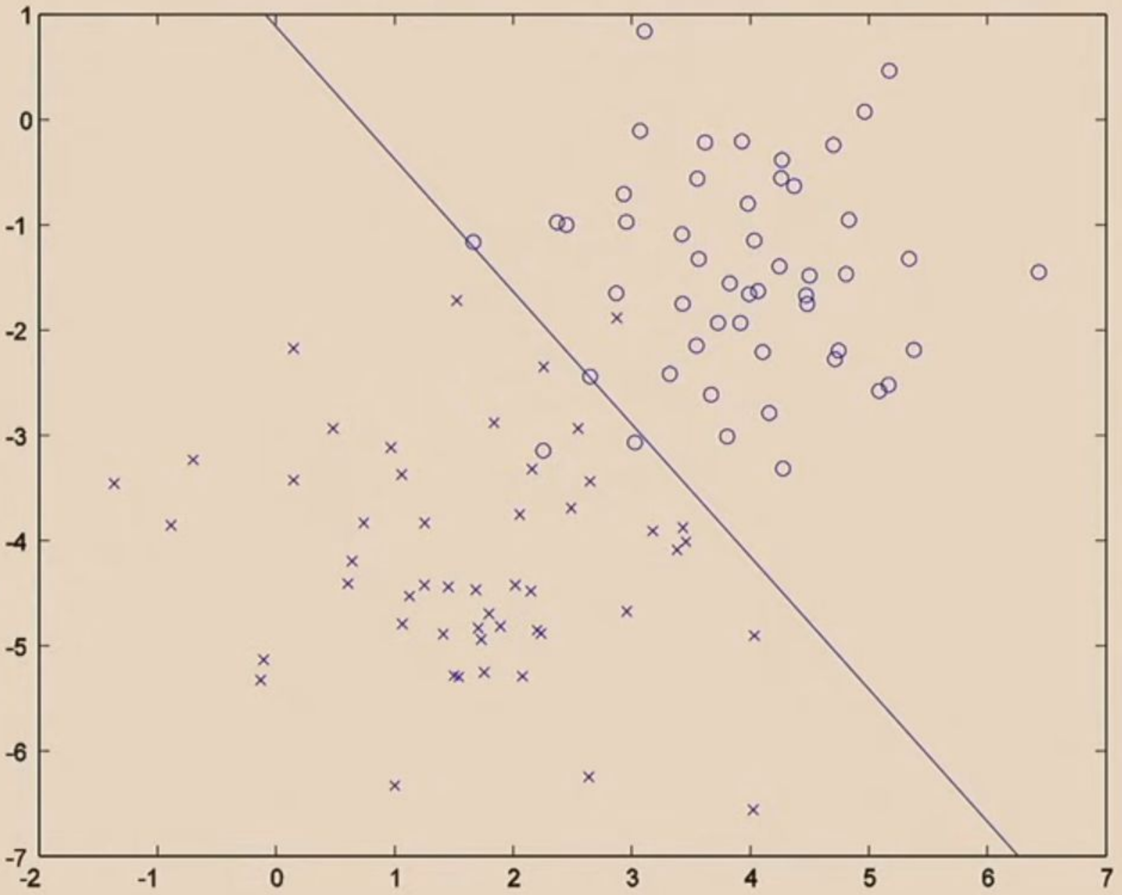
- Focus on **modeling the decision boundary** between classes
- More powerful if we have a lot of data
- Cannot be used for unsupervised tasks

Generative Models

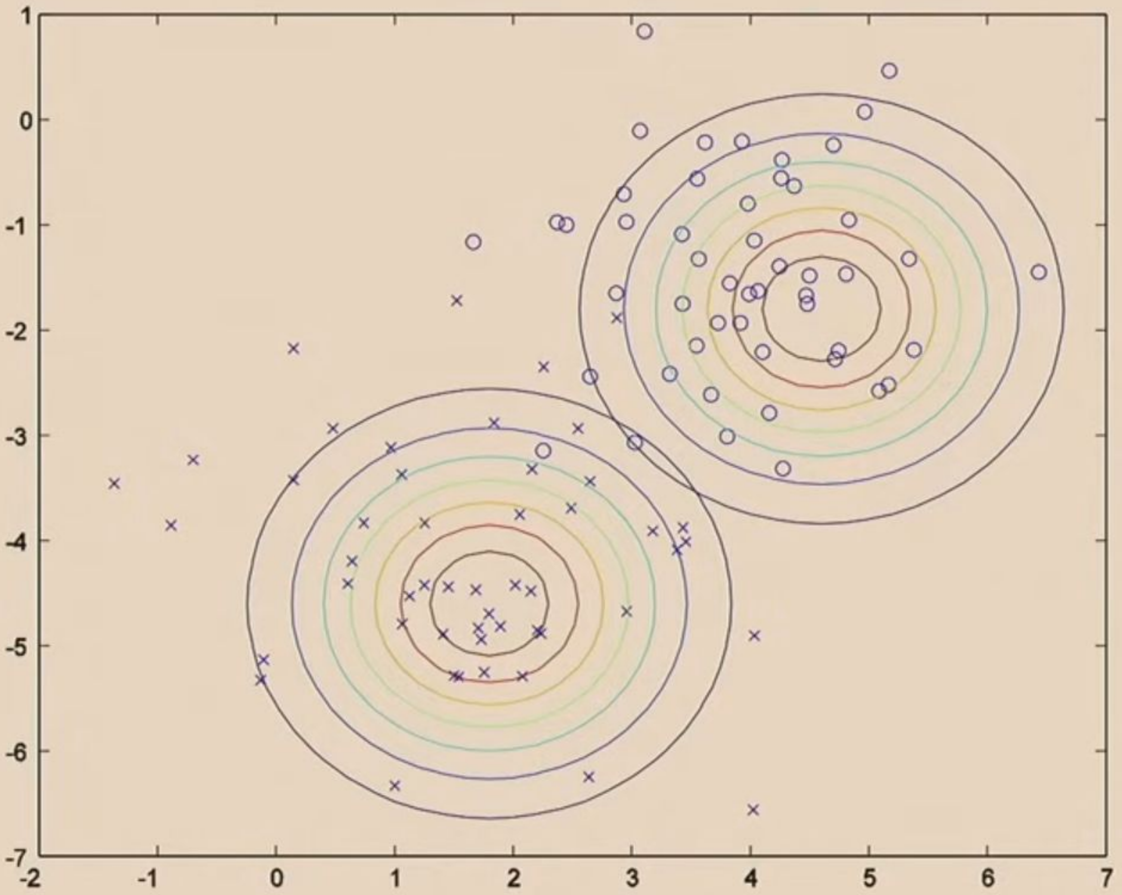
- A generative models learns the distribution for each class
- A generative model builds a model for what each of the classes look like.
- At test time, it evaluates any new example against each distribution and checks it belongs to which one more
- Where does the decision boundary comes from?
  - Where one model becomes more likely than the other
- Best if we know the underlying distribution or if we have some estimate!
- However if we don’t know then using discriminative would be best


1
0
-1
-2
-3
-4
-5
-6
-7
-2
-1
0
1
2
3
4
5
6
7

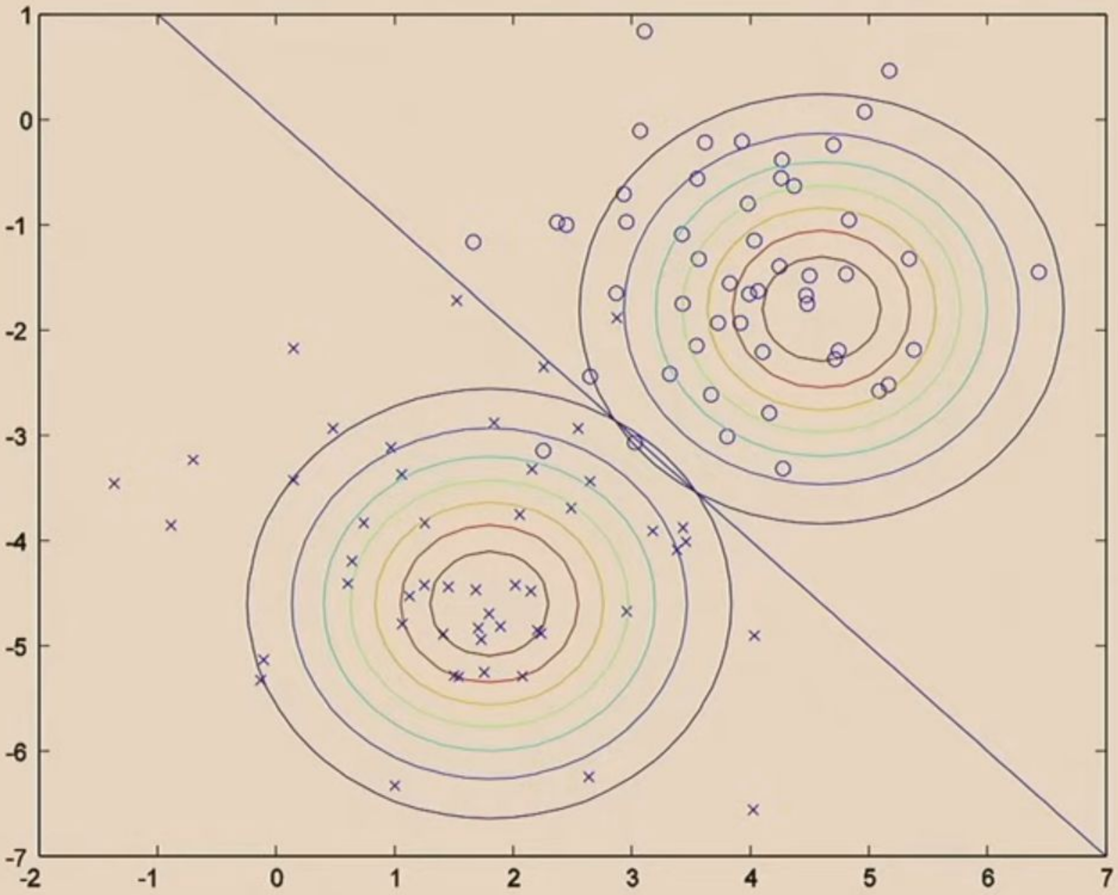
Discriminative learning algorithm: Logistic regression



Generative learning algorithm: GDA



Generative learning algorithm: GDA

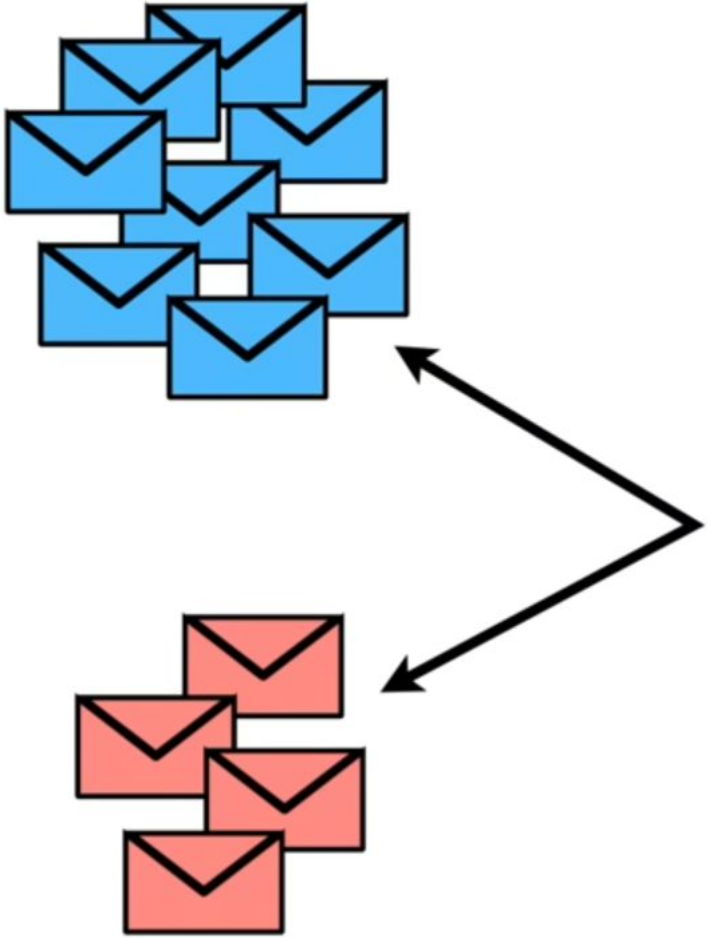


Naive Bayes / Multinomial Naive Bayes

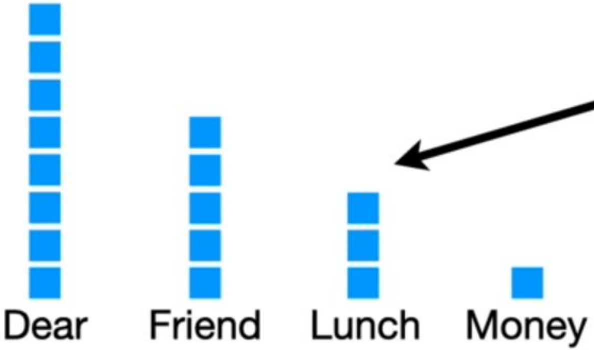
Multinomial: It assumes **that features are counts or frequencies** (like how many times a word appears in a document).

Multinomial distribution: which gives probabilities of observing counts of different outcomes.

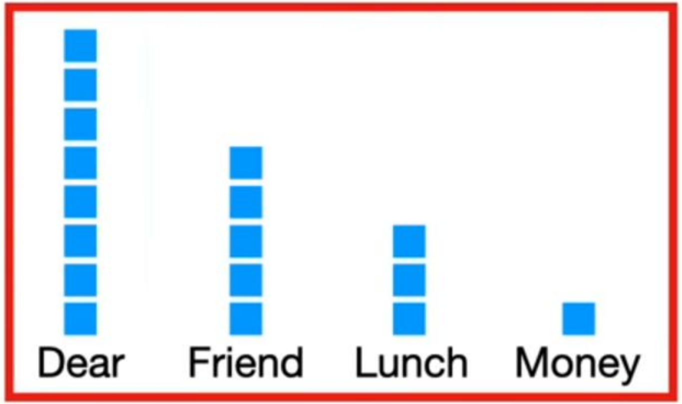
Why is it called Bayes? And Why is it naive?



...and we wanted to filter out the **spam** messages.

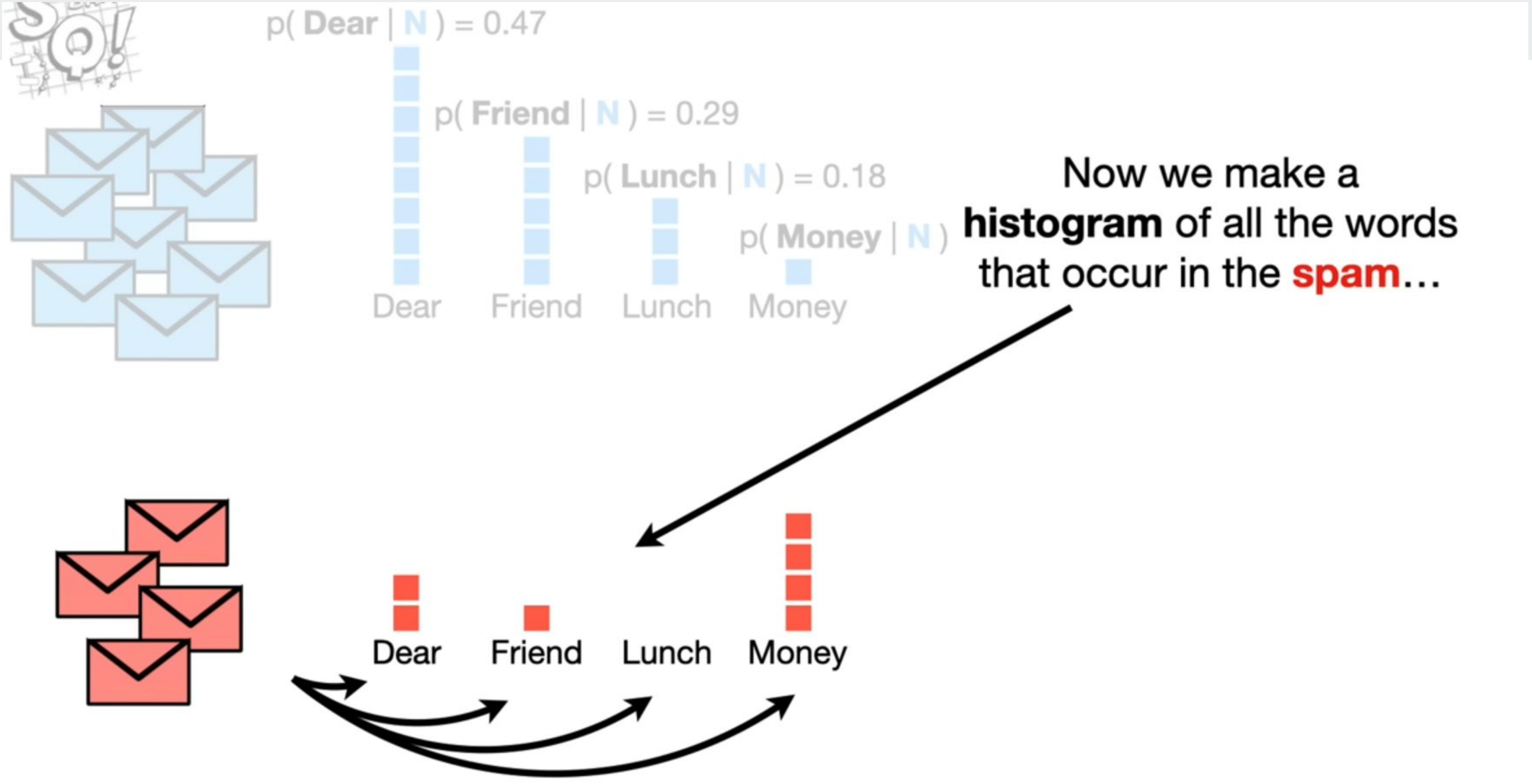


We can use the histogram to calculate the probabilities of seeing each word, given that it was in a **normal message**.



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

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


p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N)
Dear
Friend
Lunch
Money
Now we make a **histogram** of all the words that occur in the **spam**…
Dear
Friend
Lunch
Money

These probabilities are also called likelihoods

$$p(\textbf{Dear} \mid N) = 0.47$$
$$p(\textbf{Friend} \mid N) = 0.29$$
$$p(\textbf{Lunch} \mid N) = 0.18$$
$$p(\textbf{Money} \mid N) = 0.06$$

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

Naive Bayes: Why is it called Bayes?

When inference / Testing a new example:

We essentially want to get **P(class A | this example)**

This is where Bayes comes in picture:

Bayes Rule:

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

Naive Bayes: Why is it called Bayes?

Bayes Rule for Classification:

Bayes' Rule for classification:

$$P(c|d) \propto P(c) \cdot P(d|c)$$

Why not just use the original bayes rule?

- Because we don't need P(B) or P(features)

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

Naive Bayes: Why is it Naive?

- **General Probability Rule**

For two features A, B given a class C:

$$P(A, B|C) = P(A|B, C) \cdot P(B|C)$$

This is **always true** (chain rule of probability).

For three features A, B, D:

$$P(A, B, D|C) = P(A|B, D, C) \cdot P(B|D, C) \cdot P(D|C)$$

Conditional independence

- Two variables A,B are *independent* if

$$P(A \wedge B) = P(A) * P(B)$$
$$\forall a,b : P(A = a \wedge B = b) = P(A = a) * P(B = b)$$

- Two variables A,B are *conditionally independent given* C if

$$P(A,B \mid C) = P(A \mid C) * P(B \mid C)$$
$$\forall a,bc : P(A = a \wedge B = b \mid C = c) = P(A = a \mid C = c) * P(B = b \mid C = c)$$

Conditional Independence

Definition: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Which we often write

$$P(X|Y, Z) = P(X|Z)$$

E.g. $P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$

Naïve Bayes uses assumption that the X_i are conditionally independent, given Y

Given this assumption, then:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$

$$= P(X_1|Y)P(X_2|Y)$$

Chain rule

Conditional Independence

in general: $P(X_1...X_n|Y) = \prod_i P(X_i|Y)$

$(2^n-1)x2$

2n

Naïve Bayes Algorithm – discrete X_i

- Train Naïve Bayes (given data for X and Y)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

Training Naïve Bayes Classifier Using MLE

- From the data D, estimate *class priors*.
  - For each possible value of Y, estimate $Pr(Y=y_1)$, $Pr(Y=y_2)$,.... $Pr(Y=y_k)$
  - An MLE estimate: $\widehat{\pi}_k = \widehat{P}(Y = y_k) = \dfrac{\#D\{Y = y_k\}}{|D|}$
- From the data, estimate the conditional probabilities
  - If every X_i has values $x_{i1},\ldots,x_{ik}$
    - for each y_i and each X_i estimate $q(i,j,k)=Pr(X_i=x_{ij} | Y=y_i)$

$$\widehat{\theta}_{ijk} = \widehat{P}(X_i = x_{ij}|Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

Number of items in dataset D for which $Y=y_k$

Naïve Bayes Algorithm – discrete X_i

- Train Naïve Bayes (given data for X and Y)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \theta_{ijk}$$

* probabilities must sum to 1, so need estimate only n-1 of these...

Naive Bayes: Why is it Naive?

The naive assumption allows us to distribute the joint probability into multiplication of conditional probabilities **HOW?**

By assuming independence between features!

- **With Naive Bayes assumption**

We simplify drastically:

$$P(A, B|C) \approx P(A|C) \cdot P(B|C)$$

Naive Bayes: Pitfalls and Issues

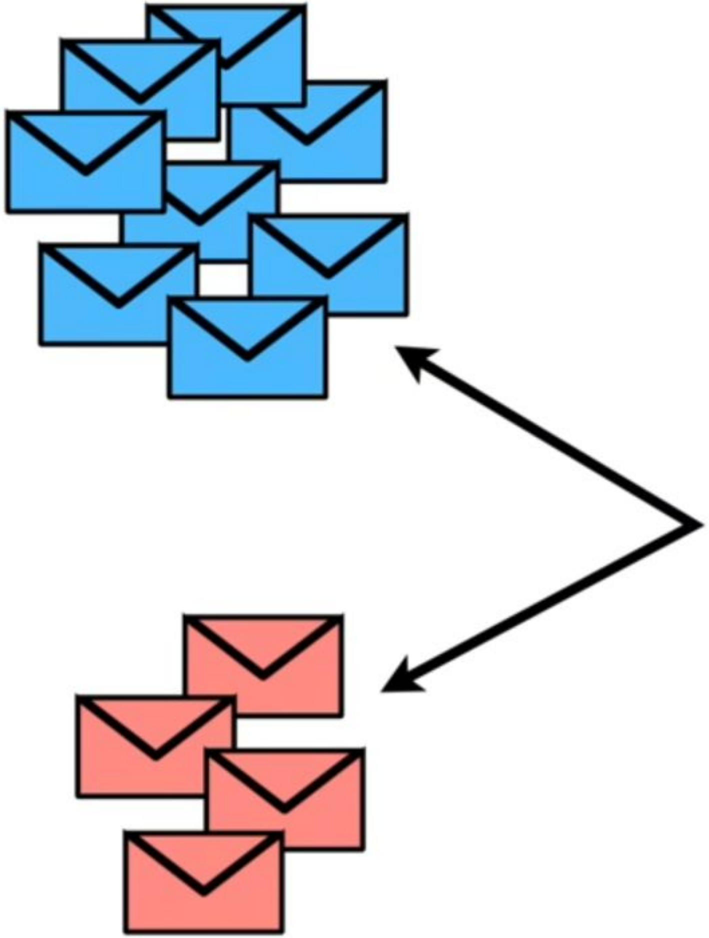
1. Zero probabilities!
    a. 1 feature not found in one of our dataset for 1 class, leads to zero probability which can be misleading.
    b. **Solution:** add 1 count to all initially (alpha)
2. The independence assumption!

Naive Bayes / Multinomial Naive Bayes

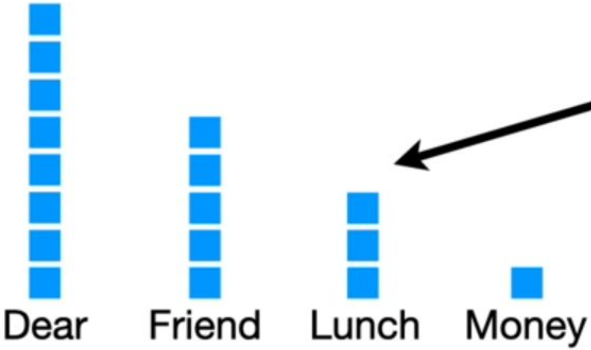
Multinomial: It assumes **that features are counts or frequencies** (like how many times a word appears in a document).

Multinomial distribution: which gives probabilities of observing counts of different outcomes.

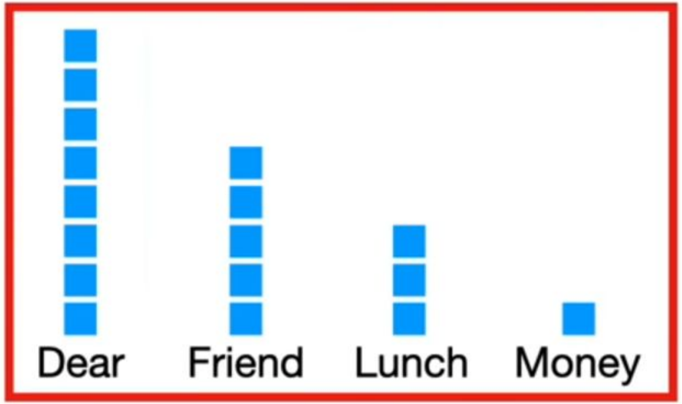
Why is it called Bayes? And Why is it naive?



...and we wanted to filter out the **spam** messages.

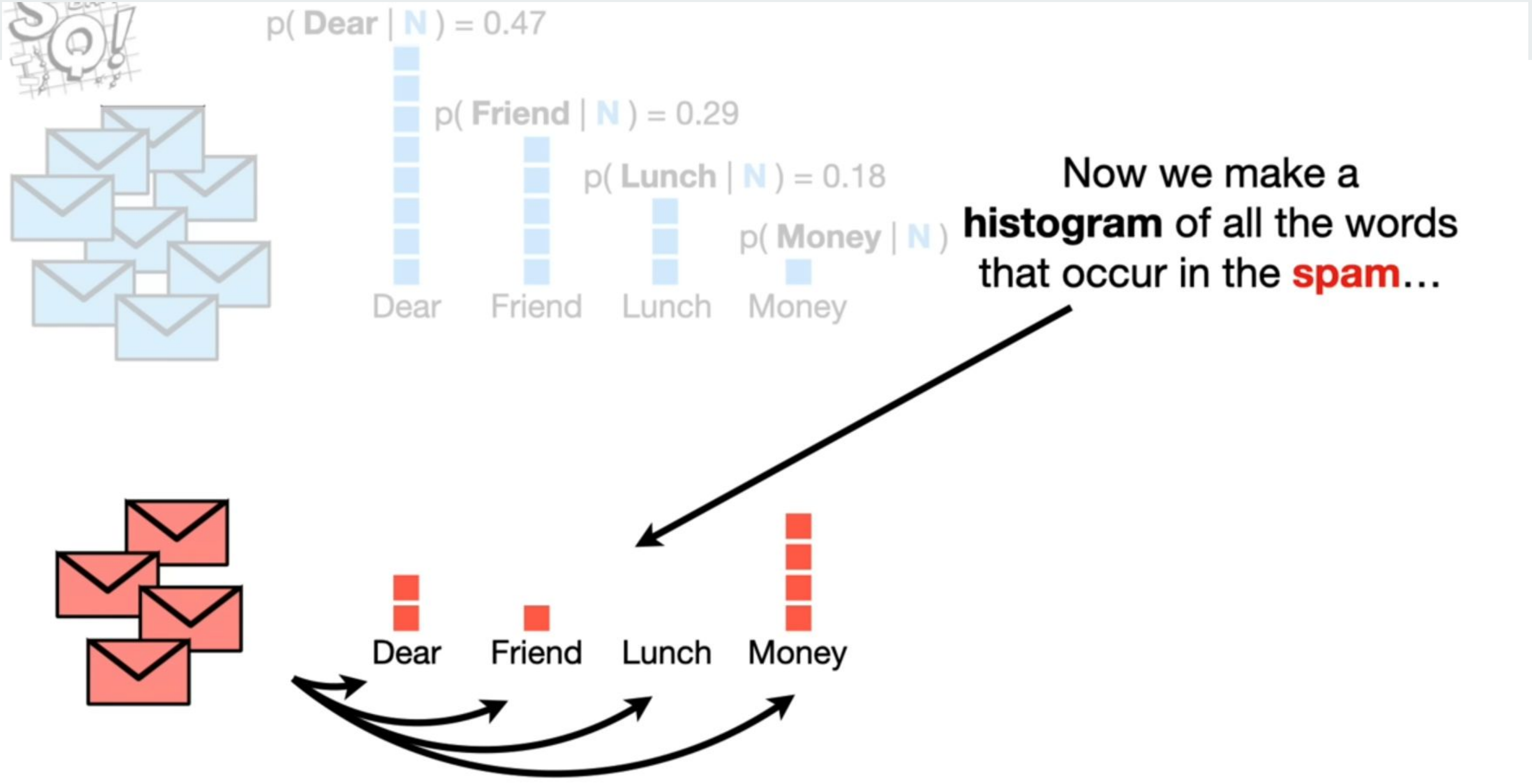


We can use the histogram to calculate the probabilities of seeing each word, given that it was in a **normal message**.



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

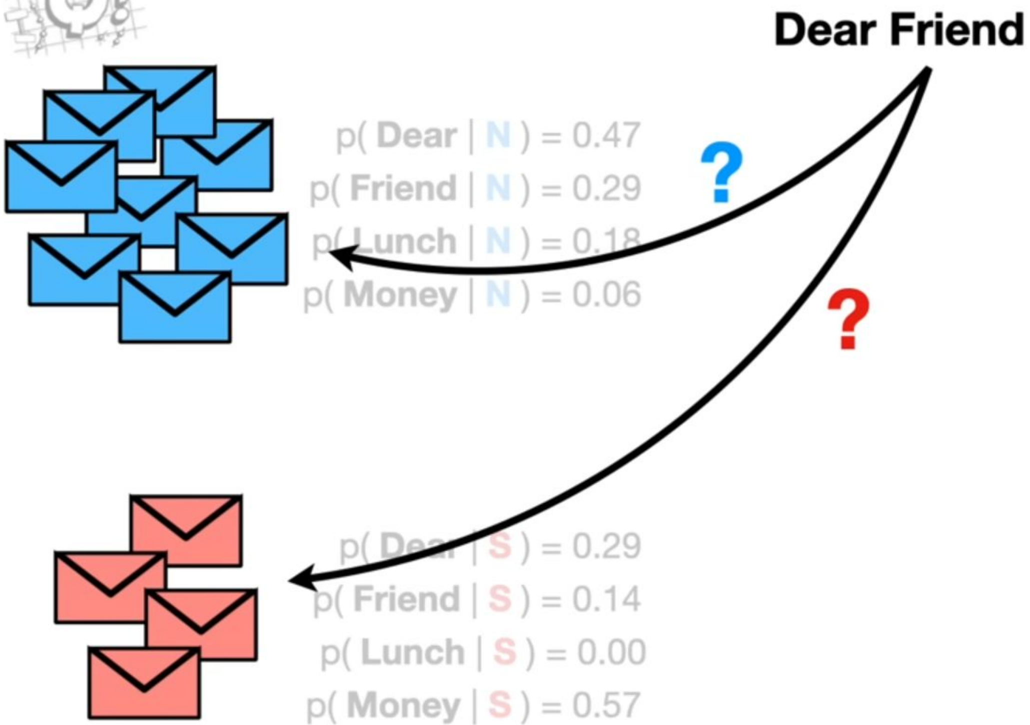
$$p(\textbf{Dear} \mid \textbf{Normal}) = \frac{8}{17}$$


p(Dear | N) = 0.47
p(Friend | N) = 0.29
p(Lunch | N) = 0.18
p(Money | N)
Dear
Friend
Lunch
Money
Now we make a **histogram** of all the words that occur in the **spam**…
Dear
Friend
Lunch
Money

$$p(\textbf{Dear} \mid N) = 0.47$$
$$p(\textbf{Friend} \mid N) = 0.29$$
$$p(\textbf{Lunch} \mid N) = 0.18$$
$$p(\textbf{Money} \mid N) = 0.06$$

These probabilities are also called likelihoods

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



And we want to decide if is a **normal message** or **spam**.

Dear Friend

p(**Dear** | N) = 0.47

p(**Friend** | N) = 0.29

p(**Lunch** | N) = 0.18

p(**Money** | N) = 0.06

$\boxed{p(N) = 0.67}$

Now we just plug in the values that we worked out earlier and do the math…

p(N) x p(**Dear** | N) x p(**Friend** | N)

p(**Dear** | S) = 0.29

p(**Friend** | S) = 0.14

p(**Lunch** | S) = 0.00

p(**Money** | S) = 0.57

How does the classification work?

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(**Dear** | **S**) = 0.29
p(**Friend** | **S**) = 0.14
p(**Lunch** | **S**) = 0.00
p(**Money** | **S**) = 0.57

p(**S**) = 0.33

Dear Friend 

Then we did the math and decided that **Dear Friend** was a **normal message** because **0.09** > **0.01**.

p(**N**) x p(**Dear** | **N**) x p(**Friend** | **N**) $\boxed{= 0.09}$

p(**S**) x p(**Dear** | **S**) x p(**Friend** | **S**) = 0.01

Resources:

Main links:

- Check this kaggle Notebook for practice
- Explanation videos drive link

Other links:

- StatQuest: Naive Bayes video
- EXTRA: StatQuest: Gaussian Naive Bayes video
- Bayes theorem, the geometry of changing beliefs