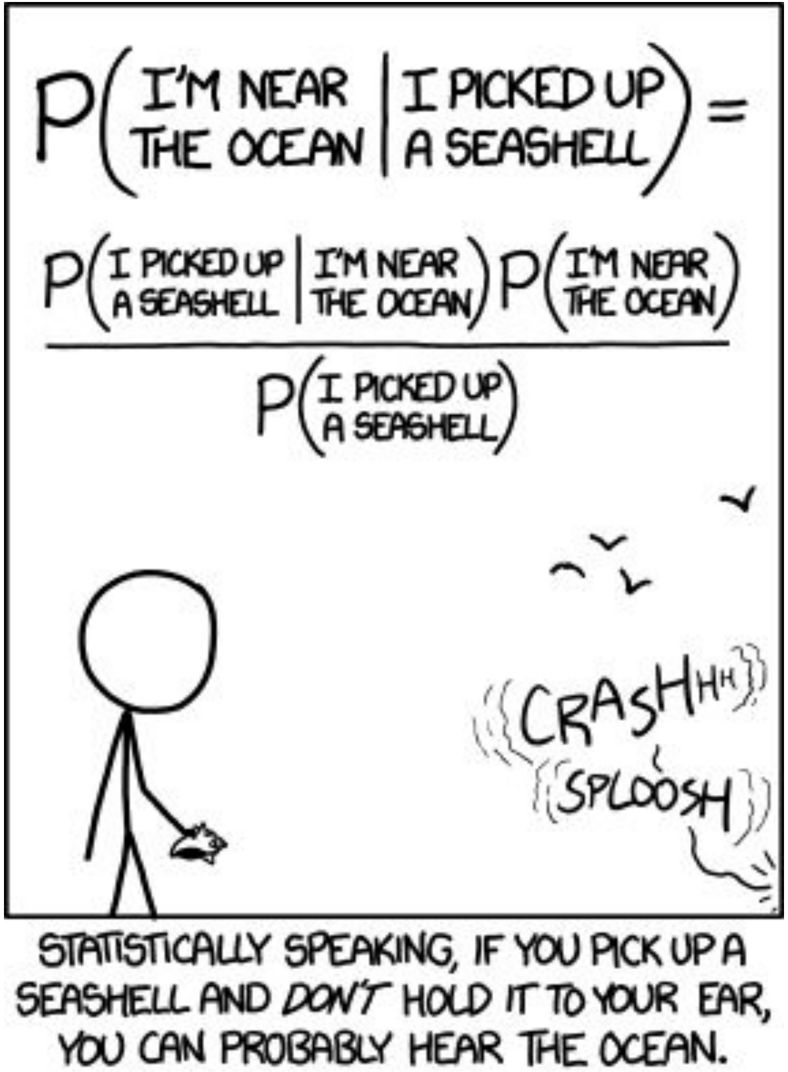


The Naive Bayes Classifier for Sentiment Analysis

LIN f313 Language and Computers
Fall 2025

Instructor: Gabriella Chronis


VEGEMITE
KRAFT
PRODUCT OF AUSTRALIA

Bridie @BBBridie · Jun 12

i just wanna eat **vegemite** toast all the time

1

Bridie @BBBridie · Jun 12

i just wanna eat **vegemite** toast all the time

1

Grant Stoddart @grants780 · Jun 10

Replying to @ArtSimone

I put tomato sauce in the fridge and **vegemite** in the bin 🗑️

1 23

Bridie @BBBridie · Jun 12

i just wanna eat **vegemite** toast all the time

1

Grant Stoddart @grants780 · Jun 10

Replying to @ArtSimone

I put tomato sauce in the fridge and **vegemite** in the bin 🗑️

1 23

GRAY @grayola · Jun 10

who tf eats **vegemite** we need to talk

4


Liam Wyatt @Wittylama · Jun 13
Lockdown has clearly been hitting the Australians in Italy hard...
"Frequently bought together" on @AmazonIT for ~$57AUD!
twin pack of 220g @Vegemite + 1L of @absolutvodka
amazon.it prime
Torna all'inizio
Spesso comprati insieme
VEGEMITE
ABSOLUT VODKA
Prezzo totale: 36,43 €
Aggiungi entrambi al carrello
Questi articoli sono spediti e venduti da venditori diversi. Mostra dettagli
12

Our Task:

Given a new tweet about Vegemite, determine the **polarity** of the tweet:

- positive (POS) or negative (NEG) (assuming we've already weeded out the neutral)

Resources:

- We have access to a dataset of 5000 tweets about Vegemite, already labeled for polarity.

IDEA: Use the **joint probabilities** we observe of different events happening together in order to make predictions.

Breakfast of Champions

Kerry Carpenter @Kezzacarpenter · Jun 23, 2020 ···
Breakfast. **Vegemite** toast!!!! I paid for THIS!!

23 16

David Feng @DavidFeng · Jun 9 ···
Bit of a cunt I am forgetting **Vegemite** ad of late. Cued for **breakfast** tomorrow morning now.

Andrew McDonald @andrewmcdonald · Mar 17, 2020 ···
A guide to ‘**Vegemite**’ and what it means in Australia

VEGEMITE: a jar of **vegemite**
LITTLE VEGEMITES: children
BIG VEGEMITES: the giant containers of **vegemite** that only big families and cafes buy
THE BIG **VEGEMITE**: A rental villa in southern NSW
BREAKFAST: **vegemite** on toast

8 28

Breakfast of Champions


Kerry Carpenter @Kezzacarpenter · Jun 23, 2020

Breakfast. **Vegemite** toast!!!! I paid for THIS!!

23 16

NEG


David Feng @DavidFeng · Jun 9

Bit of a cunt I am forgetting **Vegemite** ad of late. Cued for **breakfast** tomorrow morning now.

POS


Andrew McDonald @andrewmcdonald · Mar 17, 2020

A guide to '**Vegemite**' and what it means in Australia

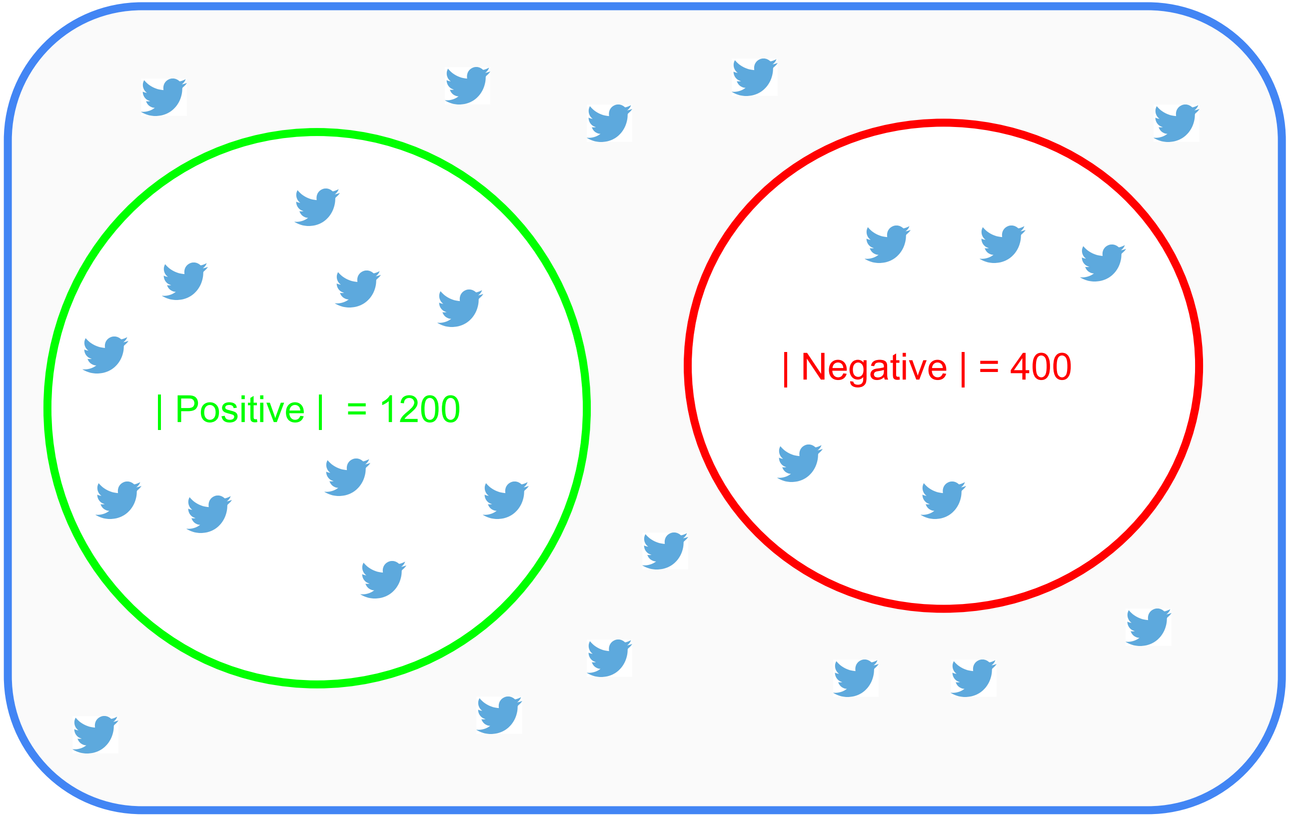
VEGEMITE: a jar of **vegemite**
LITTLE VEGEMITES: children
BIG VEGEMITES: the giant containers of **vegemite** that only big families and cafes buy
THE BIG **VEGEMITE**: A rental villa in southern NSW
BREAKFAST: **vegemite** on toast

8 28

??????

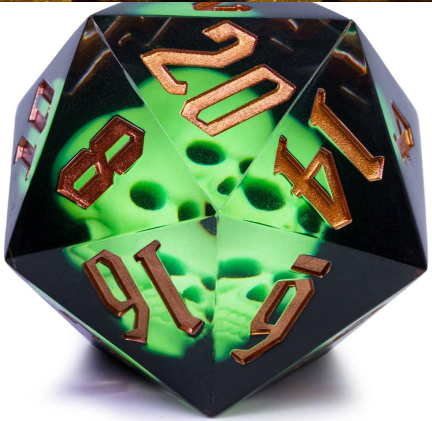
U =  tweets

| U | = 5000



Probability Review:

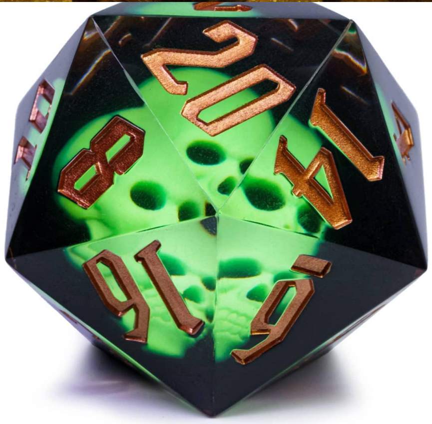
Q: What is the probability that you roll a 20?



Probability Review:

Q: What is the probability that you roll a 20?

Q: What is the probability that you roll two 20s in a row?



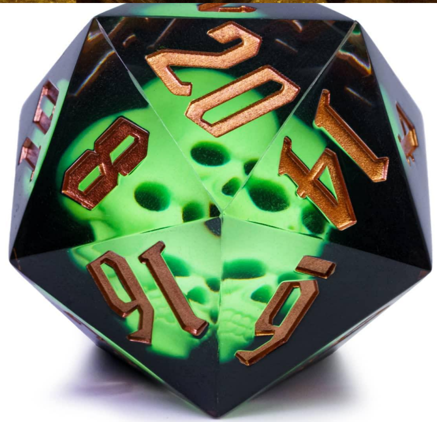
Probability Review:

Q: What is the probability that you roll a 20?

Q: What is the probability that you roll two 20s in a row?

Q: **Given** that you have already rolled a 20, what is the probability that you roll another 20?

Hint: these are two *separate* rolling events, and we only care about the second one!



Probability Review:

Q: What is the probability that you roll a 20?

A: P(rolling 20) = 1/20

Q: What is the probability that you roll two 20s in a row?

A: P (rolling two 20s) = P(rolling 20) x P(rolling 20

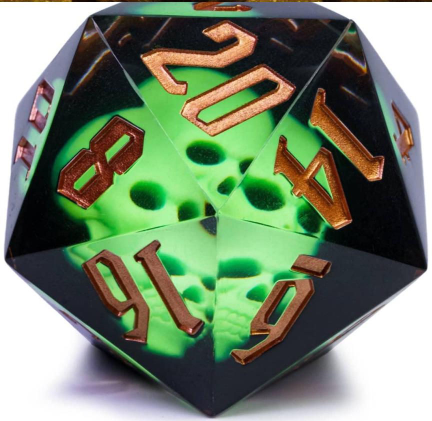
= 1/20 x 1/20

= 1/400

Q: **Given** that you have <u>already</u> rolled a 20, what is the probability that you roll another 20?

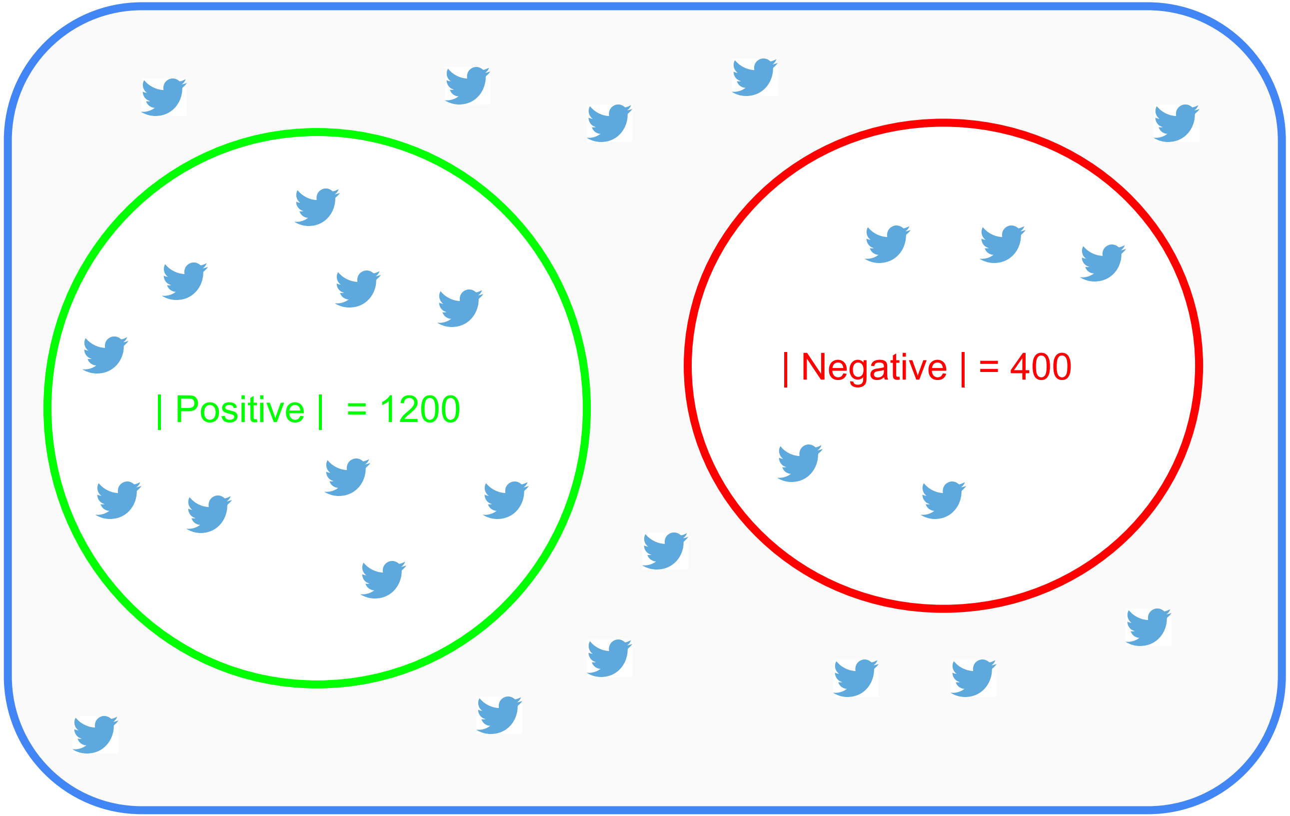
Hint: these are two *separate* rolling events, and we only care about the second one!

A: P(rolling 20 | rolled a 20) = 1 / 20



U =  tweets

| U | = 5000



Making Predictions

Question: how do we use our dataset to make *automatic* predictions about the sentiment of a tweet that we have never seen before?

Making Predictions

Question: how do we use our dataset to make *automatic* predictions about the sentiment of a tweet that we have never seen before?

Answer: First, treat our little dataset like a model of the universe. Then, calculate the probability that a tweet is positive, **given** that we've rolled these particular words

It's so . . . easy?

Just find the probability that a tweet is positive, given that it contains the words "I" and "tried" and "own" and "again" and…!

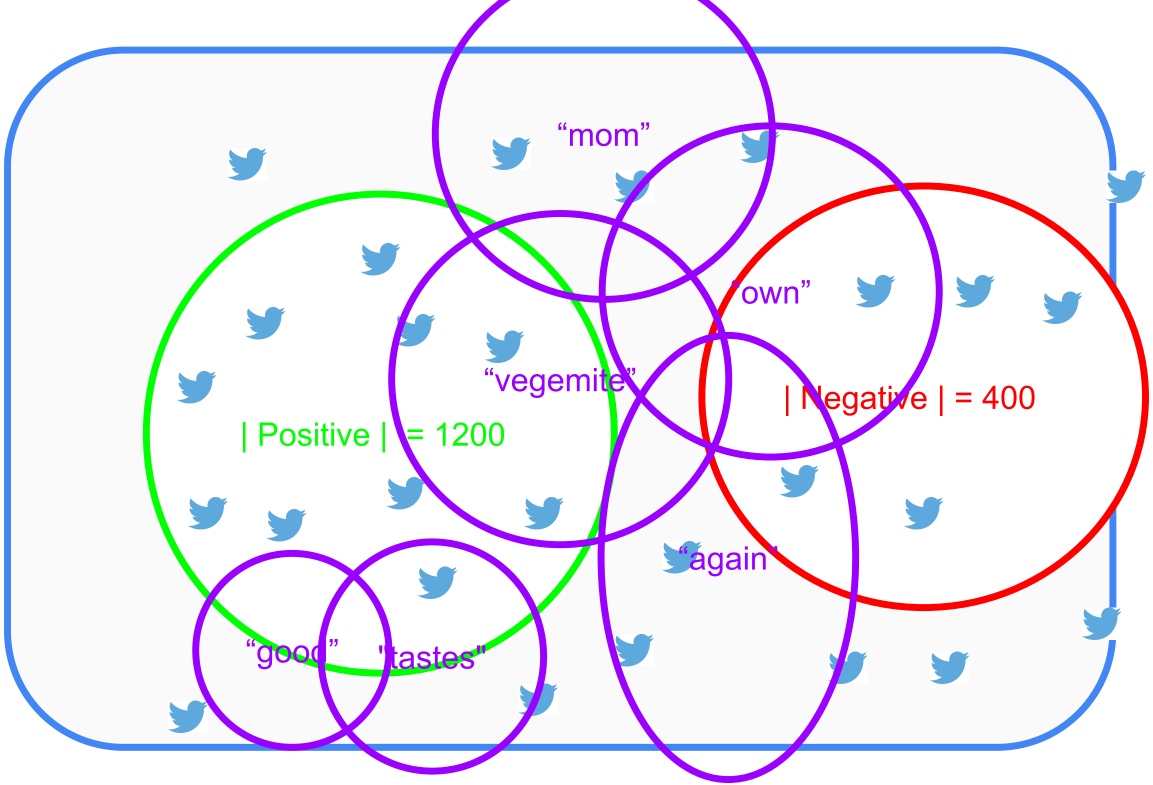
P (Pos | I AND tried AND own AND again AND tastes AND good AND mom AND scared)

= (positive tweets containing all of these words) / (tweets containing all of these words)

We just look at our giant chart and find the **intersection** of all these probabilities!

U =  **tweets**

| U | = 5000

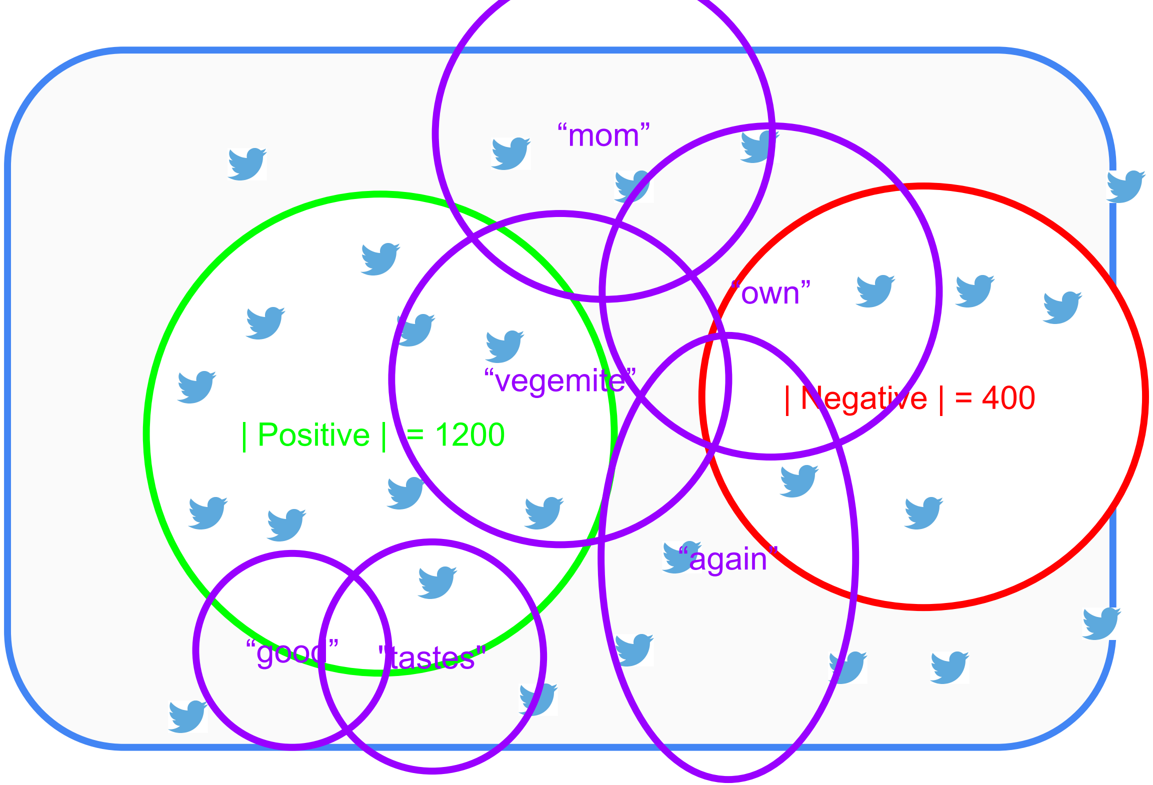


P (Pos | I AND tried AND own AND again AND tastes AND vegemite AND mom AND scared)

= (positive tweets containing all of these words) / (tweets containing all of these words)

U = tweets

| U | = 5000



PROBLEM:
Language is
way too clever
for that!

P (Pos | I AND tried AND own AND again AND tastes AND vegemite AND mom AND scared)

= (positive tweets containing all of these words) / (tweets containing all of these words)

Workaround: Independence assumption: Treating language like a 20 sided die

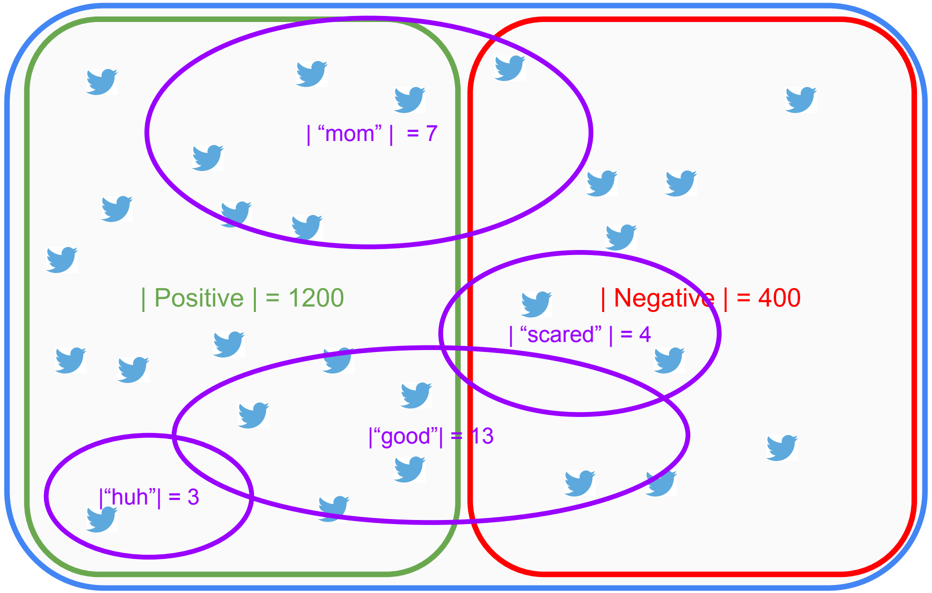
Let's say our vocabulary is 1 million words. A million-sided die

Then each tweet is like a group of dice rolls.

A10 word tweet is like rolling 10 million-sided dice!

U =  tweets

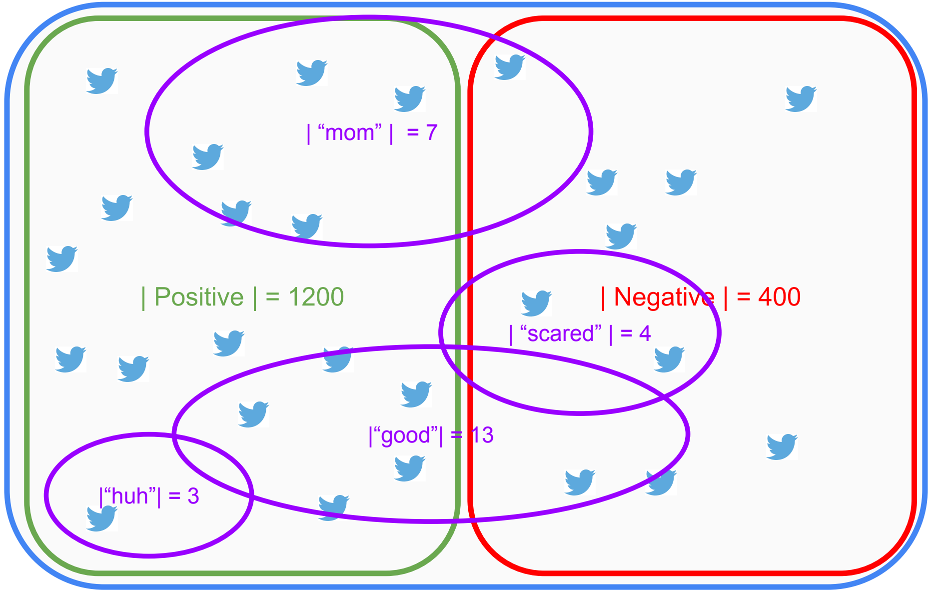
| U | = 1600 = subjective tweets only!



vocabulary: good, mom, huh, scared

U =  tweets

| U | = 1600 = subjective tweets only!



GOAL:

1. Calculate

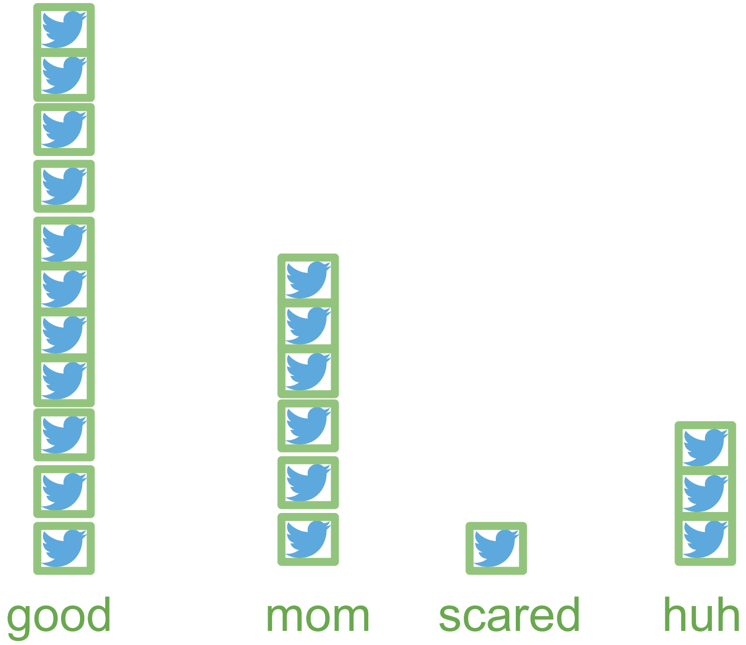
P(POS | "good", "mom")

P(NEG | "good", "mom")

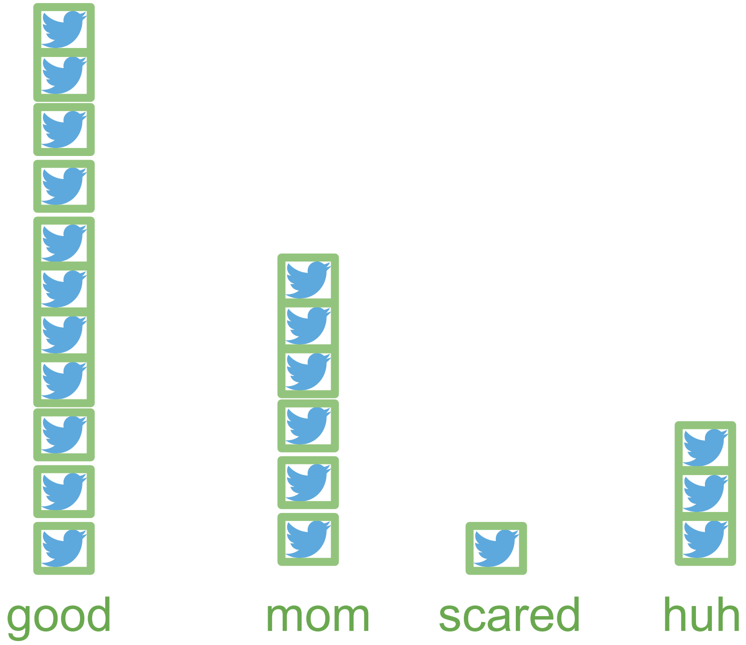
2. compare

vocabulary: good, mom, huh, scared

Histogram



Histogram



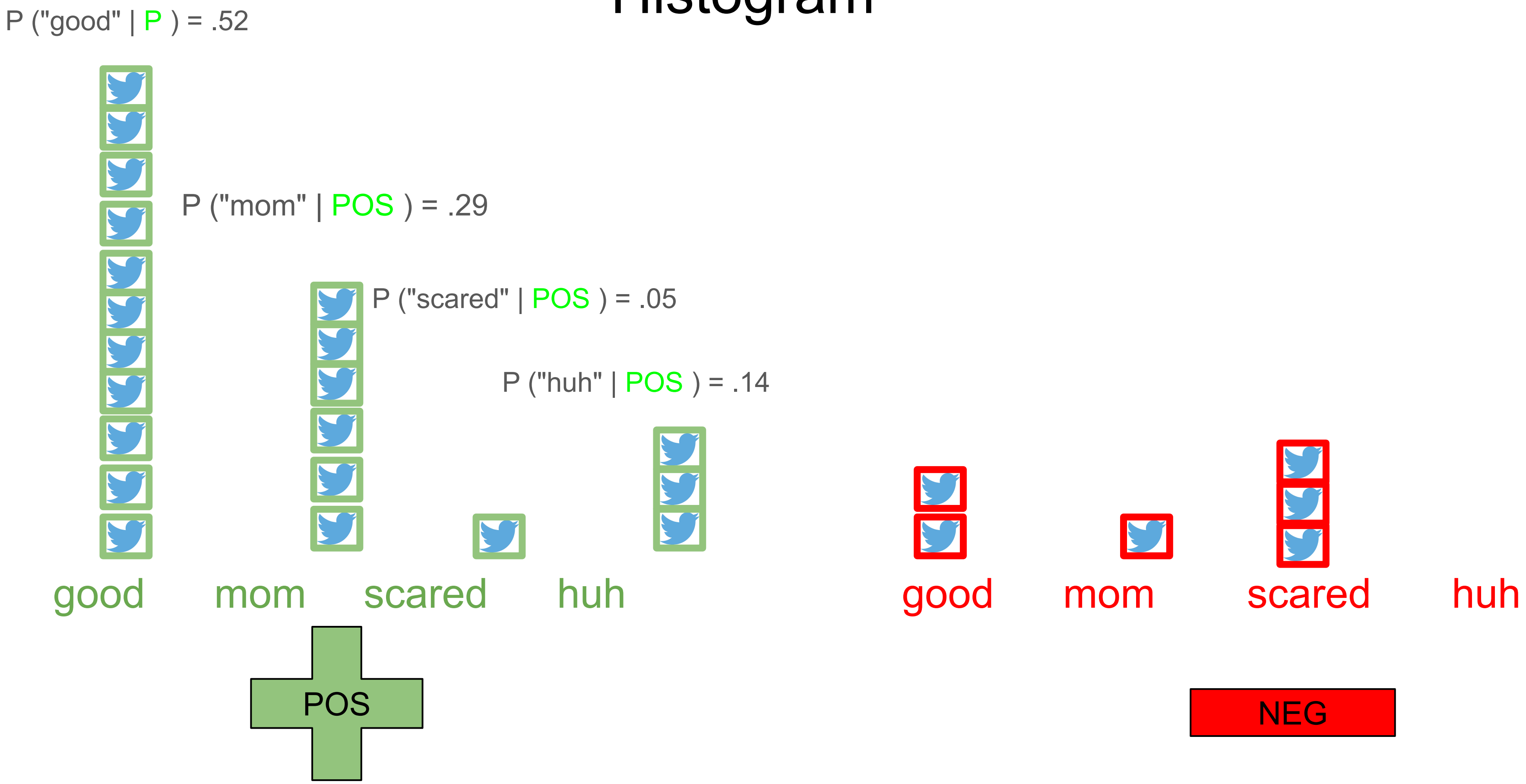
P ("good" | POS) = 11 / 21

P ("mom" | POS) = 6 / 21

P ("scared" | POS) = 1 / 21

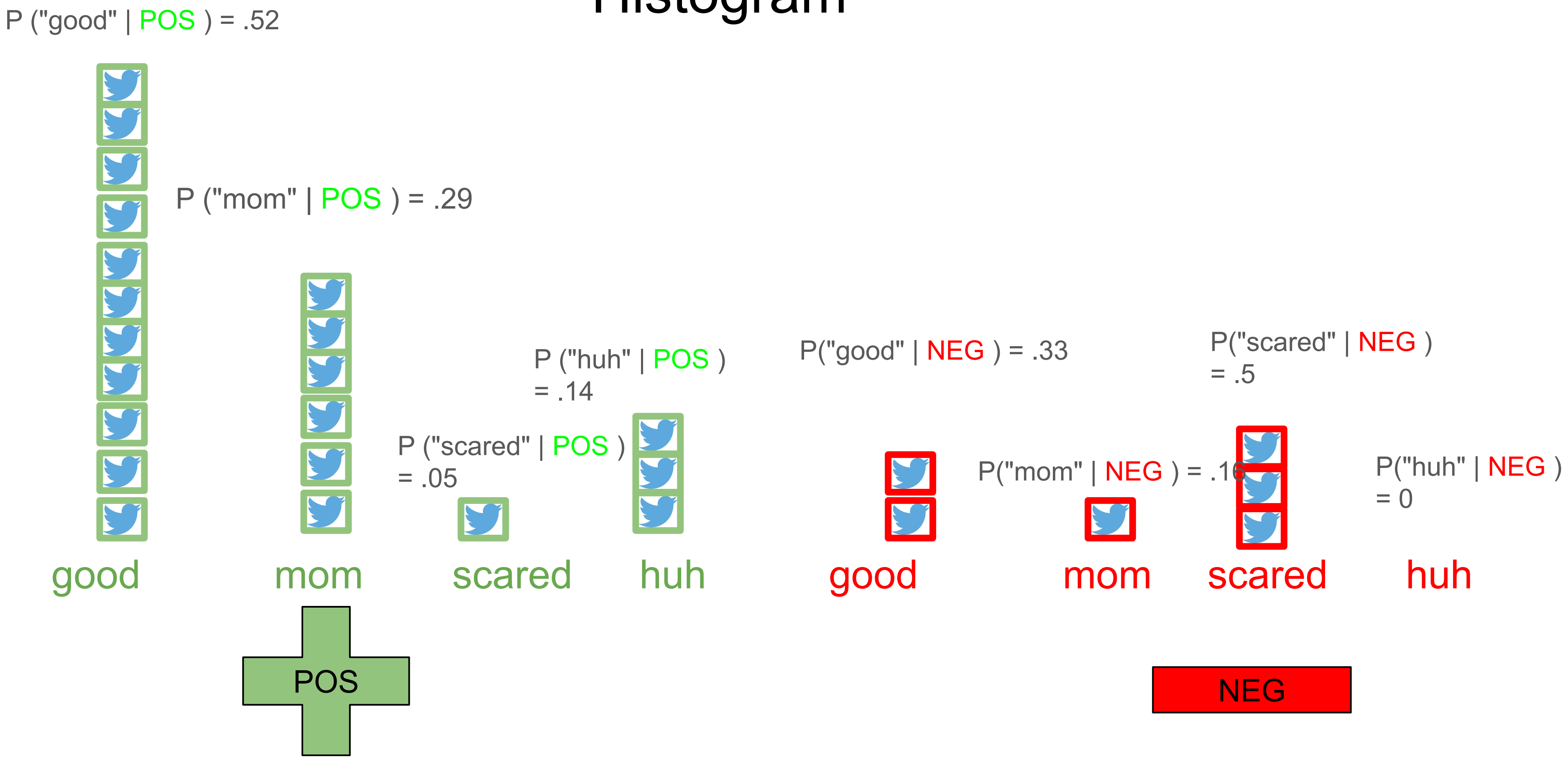
P ("huh" | POS) 3 / 21

Histogram



P ("good" | P) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14
good
mom
scared
huh
POS
good
mom
scared
huh
NEG

Histogram


P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("huh" | POS) = .14
P ("scared" | POS) = .05
P("good" | NEG) = .33
P("scared" | NEG) = .5
P("mom" | NEG) = .16
P("huh" | NEG) = 0
good
mom
scared
huh
good
mom
scared
huh
POS
NEG

Probabilities

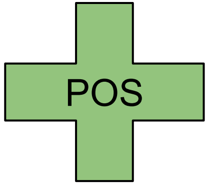
POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

NEG

Joint Probabilities



$P(\text{"good"} \mid \text{POS}) = .52$
$P(\text{"mom"} \mid \text{POS}) = .29$
$P(\text{"scared"} \mid \text{POS}) = .05$
$P(\text{"huh"} \mid \text{POS}) = .14$

$P(\text{"good"} \mid \text{NEG}) = .33$
$P(\text{"mom"} \mid \text{NEG}) = .16$
$P(\text{"scared"} \mid \text{NEG}) = .5$
$P(\text{"huh"} \mid \text{NEG}) = 0$

Q: Given we have a 2-word positive tweet, What's the probability that it contains "good" and "mom"?

Joint Probabilities

POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

NEG

Q: Given we have a 2-word positive tweet, What's the probability that it contains "good" and "mom"?

P(A,B) = P(A) x P(B)
P("good", "mom" | POS) = P("good" | POS) x P("mom"| POS)

Joint Probabilities

POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

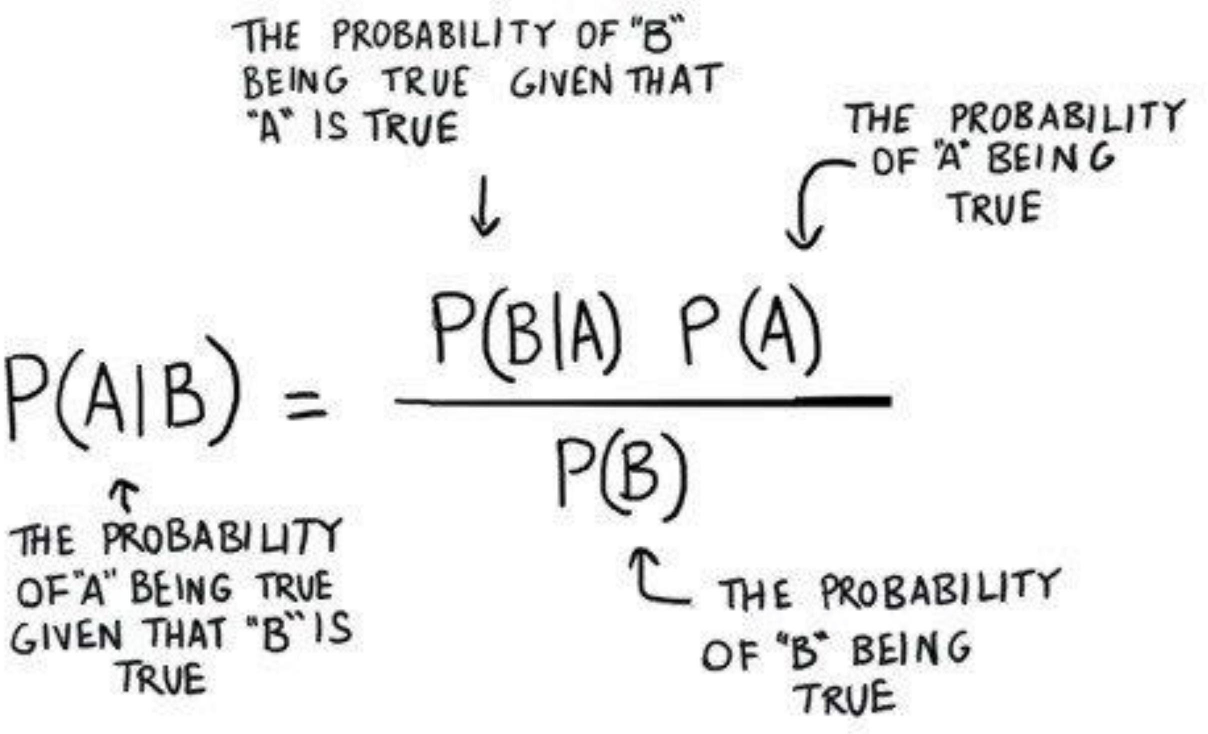
NEG

Q: Given we have a 2-word positive tweet, What's the probability that it contains "good" and "mom"?

$P(A,B) = P(A) \times P(B)$
P("good", "mom" | POS) = P("good" | POS) x P("mom"| POS)

PROBLEM: we **have** P("good", "mom" | POS)
but we **want** P(POS | "good", "mom")

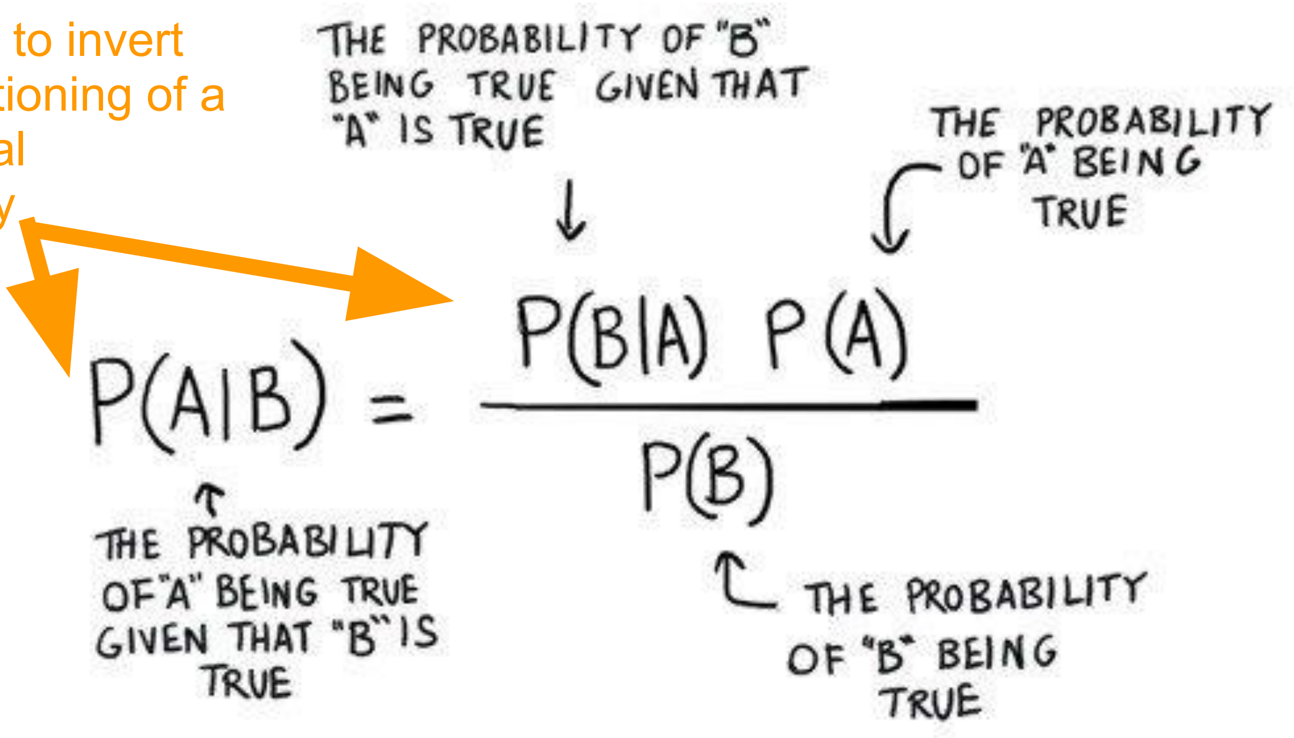
Bayes' Theorem to the Rescue



https://medium.com/@chonghankhai/bayesian-thinking-in-everyday-life-bf82fe2ab0af

Bayes’ Theorem to the Rescue

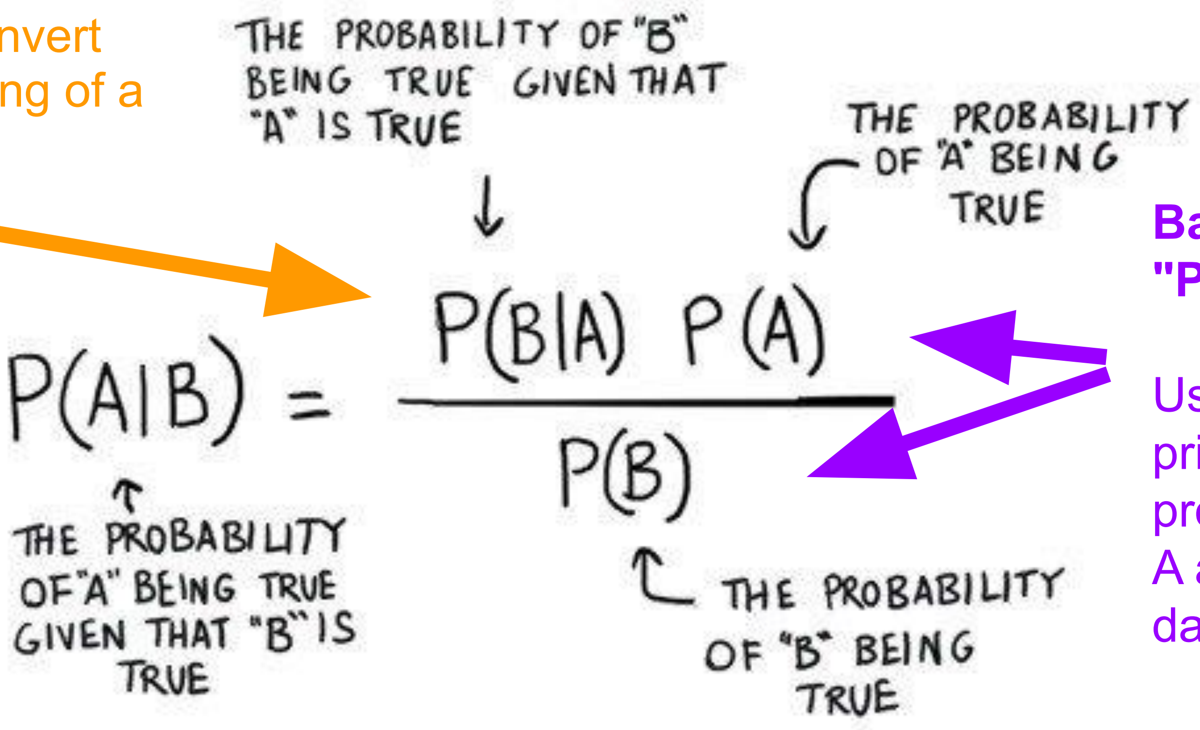
Allows us to invert the conditioning of a conditional probability



https://medium.com/@chonghankhai/bayesian-thinking-in-everyday-life-bf82fe2ab0af

Bayes’ Theorem to the Rescue

Allows us to invert the conditioning of a conditional probability



Bayesian "Priors":

Using the prior probabilities of A and B in our dataset

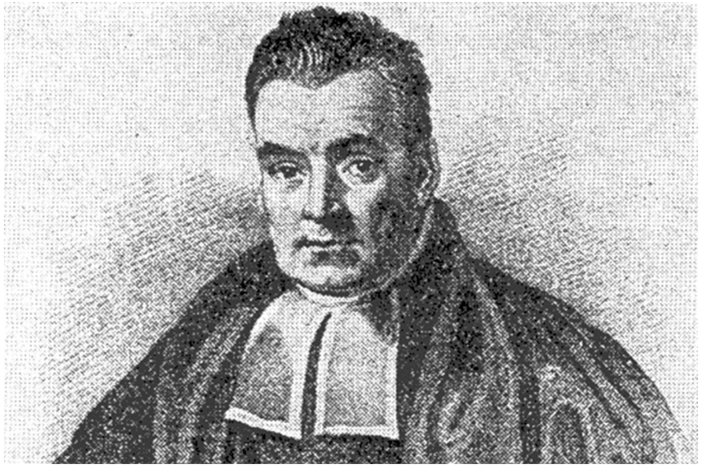
https://medium.com/@chonghankhai/bayesian-thinking-in-everyday-life-bf82fe2ab0af

Thomas Bayes

- Philosopher, Presbyterian minister, and *very* influential statistician

- *Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures (1731)*

- Later became interested in probability



Bayes to the Rescue

We have a way to get there with Bayes Rule!

$$P(\text{POS} \mid \text{"good"}, \text{"mom"})$$

$$\propto \; P(\text{"good"}, \text{"mom"} \mid \text{POS}) \times P(\text{POS})$$

Bayes to the Rescue

We have a way to get there with Bayes Rule!

P(POS | "good", "mom")

$\propto$ P("good", "mom" | POS) x P(POS)

P(NEG | "good", "mom", scared")

$\propto$ P("good", "mom" | NEG) x P(NEG)

Bayes to the Rescue

We have a way to get there with Bayes Rule!

P(POS | "good", "mom")

$\propto$ P("good", "mom" | POS) x P(POS)

P(NEG | "good", "mom", scared")

$\propto$ P("good", "mom" | NEG) x P(NEG)

Bayesian "Priors":

The the probability that any given tweet is POS or NEG

Bayes to the Rescue

We can break down the values we want by the values we have

P(POS | "good", "mom", scared", "own")

$\propto$ P("good", "mom" | POS) x P(POS)

P(NEG | "good", "mom", scared")

$\propto$ P("good", "mom" | NEG) x P(POS)

Applying Bayes Rule

POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P("good", "mom" | POS) = ?

P(POS) = ?

NEG

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

P("good", "mom" | NEG) = ?

P(NEG) = ?

Applying Bayes Rule

POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P("good", "mom" | POS)

= P("good" | POS) x P("mom"| POS)
= .52 x .29 x

= 0.15

P(POS) = 0.75

NEG

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

P("good", "mom" | NEG)

= P("good" | NEG) x P("mom"| NEG)
= .33 x .16

= 0.05

P(NEG) = 0.75

Applying Bayes Rule

POS

P ("good" | POS) = .52
P ("mom" | POS) = .29
P ("scared" | POS) = .05
P ("huh" | POS) = .14

P(POS) = .75

P("good", "mom", scared" | POS)

= P("good" | POS) x P("mom"| POS)
= .52 x .29

= 0.15

NEG

P("good" | NEG) = .33
P("mom" | NEG) = .16
P("scared" | NEG) = .5
P("huh" | NEG) = 0

P(POS) = .25

P("good", "mom", scared" | NEG)

= P("good" | NEG) x P("mom"| NEG)
= .33 x .16 x .5

= 0.05

Applying Bayes Rule

POS

P(POS) = .75

P("good", "mom", scared" | POS) = 0.15

NEG

P(POS) = .25

P("good", "mom", scared" | NEG) = .05

P(POS | "good", "mom", scared", "own")

∝ P("good", "mom", scared", "own" | POS) x P(POS)

P(NEG | "good", "mom", scared")

∝ P("good", "mom", scared" | NEG) x P(POS)

Applying Bayes Rule

POS

P(POS) = .75

P("good", "mom", scared" | POS) = 0.15

NEG

P(POS) = .25

P("good", "mom", scared" | NEG) = .0.05

P(POS | "good", "mom", scared", "own")

$\propto$ P("good", "mom", scared", "own" | POS) x P(POS)

$\propto$ 0.15 x 0.75

$\propto$ 0.1125

P(NEG | "good", "mom", scared")

$\propto$ P("good", "mom", scared" | NEG) x P(POS)

$\propto$ 0.05 x 0.25

$\propto$ 0.0125

TA-DA! The Naive Bayes Classifier

Bayesian probability model + decision rule = tweet classifier!

(what we just did)

Decision rule: a way of choosing a label based on the probability estimates. Essentially, you look at P(POS|‘good’,'mom', 'scared') and P(NEG|’good’, 'mom', scared') and choose the one that’s higher. The fancy word for choosing the label with the highest probability is the *argmax* function.

Problem?

Question: what happens when we run our Naive Bayes classifier on this tweet?

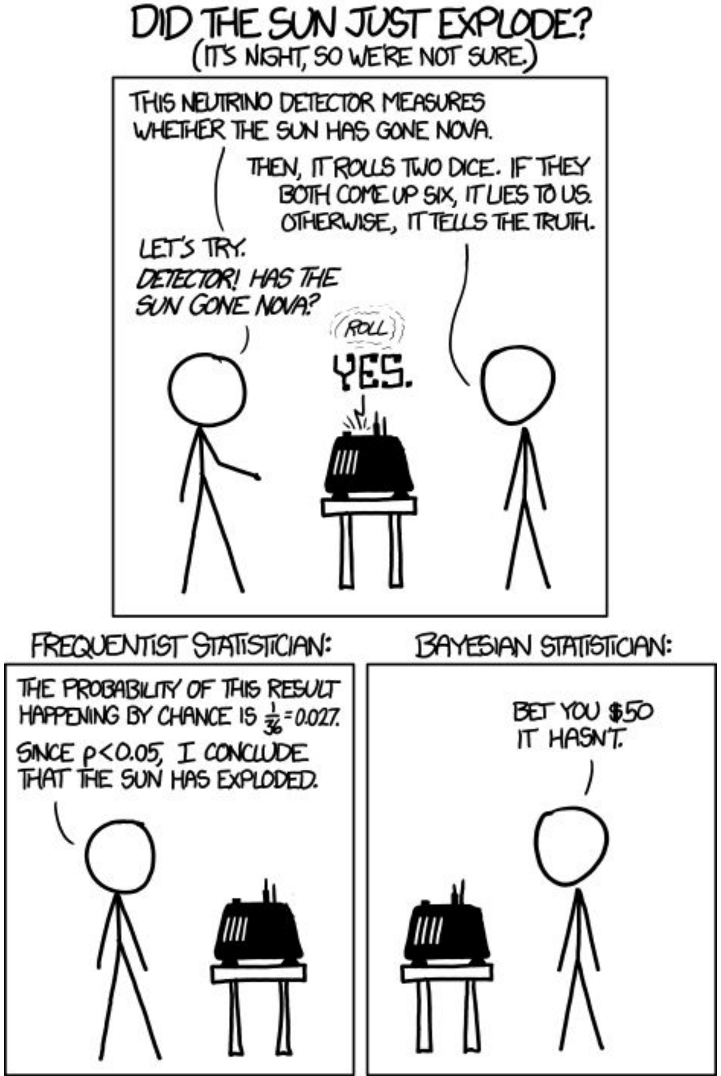
Problems?

- Independence assumption
- sarcasm / irony
- ambiguity
- ….

That all made perfect sense, right?

https://www.youtube.com/watch?v=8al5cSQNmME

(Skip to 5:45)


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
(ROLL)
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36}$=0.027.
SINCE $p<0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

Administrivia

For next class:

- read the syllabus
- Read golden gate claude blog post (3 mins)
- Watch subliminal messaging video (15 mins)