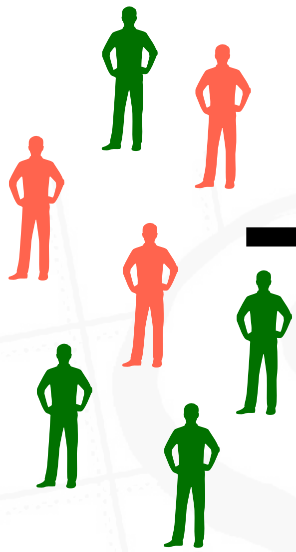
StatQuest!!!

Gaussian Naive Bayes

Study Guide!!!

The Problem

People who **Love** the movie, **Troll 2**, mixed with people who **Do Not**…

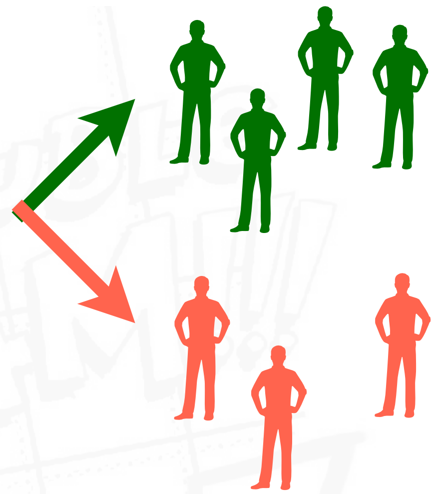


…but we have continuous data gathered from each group…

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
24.3	750.7	0.2
28.2	533.2	50.5
etc.	etc.	etc.

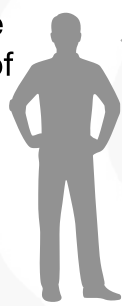
Popcorn (grams)	Soda Pop (ml)	Candy (grams)
2.1	120.5	90.7
4.8	110.9	102.3
etc.	etc.	etc.

…and we want to use it to classify the two types of people.

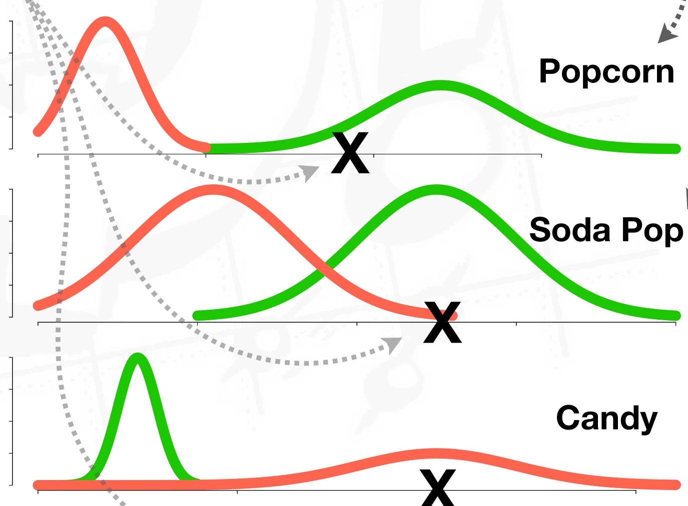


The Solution - Gaussian Naive Bayes

If someone says they consume **20** grams of **popcorn**, **500 ml** of **soda pop** and **25 grams** of **candy** each day.



The mean and standard deviation of each feature in **Training Data** is converted in to **Gaussian**, aka **Normal**, distributions.



We multiply the **Prior** probability that the person **Loves Troll 2**…

…by the **Likelihoods** given that they **Love Troll 2.**

p(**Loves Troll 2**)

× *L*(popcorn = 20 | **Loves**)

× *L*(soda pop = 500 | **Loves**)

× *L*(candy = 25 | **Loves**)

p(**No Love**)

× *L*(popcorn = 20 | **No Love**)

× *L*(soda pop = 500 | **No Love**)

× *L*(candy = 25 | **No Love**)

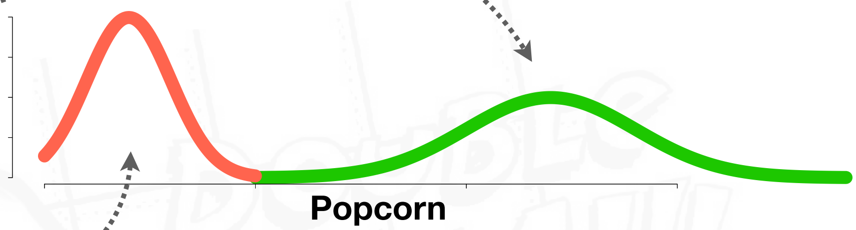
Then we do the same thing assuming **Does Not Love Troll 2.**

Whichever classification has the highest **value**, or **log(value)**, is the final classification.

Step 1) Create a Gaussian (normal) curve for each feature

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
24.3	750.7	0.2
28.2	533.2	50.5
etc.	etc.	etc.
2.1	120.5	90.7
4.8	110.9	102.3
etc.	etc.	etc.

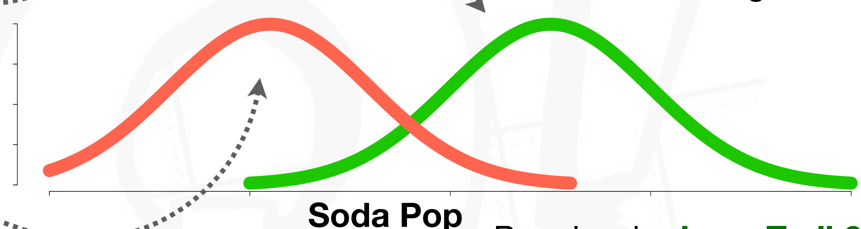
For the people who **Love Troll 2** the mean is **24** and the standard deviation is **4** for **popcorn**.



People who **Do Not Love Troll 2** eat less popcorn; the mean is **4** and the standard deviation is **2**.

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
24.3	750.7	0.2
28.2	533.2	50.5
etc.	etc.	etc.
2.1	120.5	90.7
4.8	110.9	102.3
etc.	etc.	etc.

These curves represent **soda pop** consumption for the two categories.



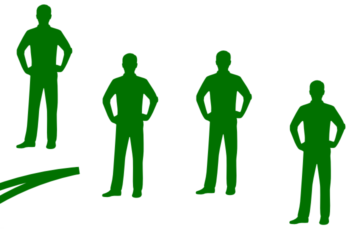
People who **Love Troll 2** drink more soda than people who **Do Not**.

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
24.3	750.7	0.2
28.2	533.2	50.5
etc.	etc.	etc.
2.1	120.5	90.7
4.8	110.9	102.3
etc.	etc.	etc.

Lastly, these curves represent **candy** consumption for the two categories.

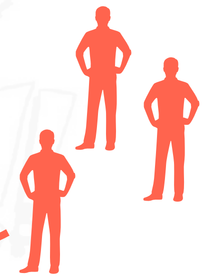
People who **Do Not Love Troll 2** eat more candy.

Step 2a) Calculate prior probability for people who Love Troll 2

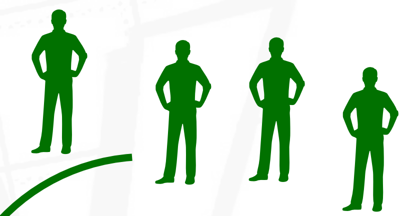


NOTE: The **Prior Probabilities** can be set to any probabilities we want, but a common guess is estimated from the training data like so:

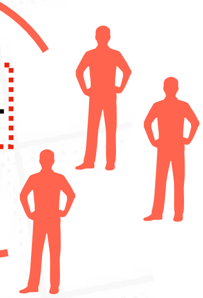
$$p(\textbf{Love}) = \frac{\textbf{\# who love Troll 2}}{\textbf{Total \# of People}} = \frac{\mathbf{4}}{\mathbf{4} + \mathbf{3}} \boxed{= 0.6}$$



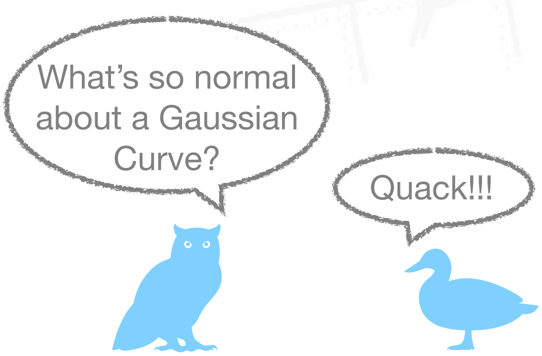
Step 2b) Calculate prior probability for people who Do Not Love Troll 2



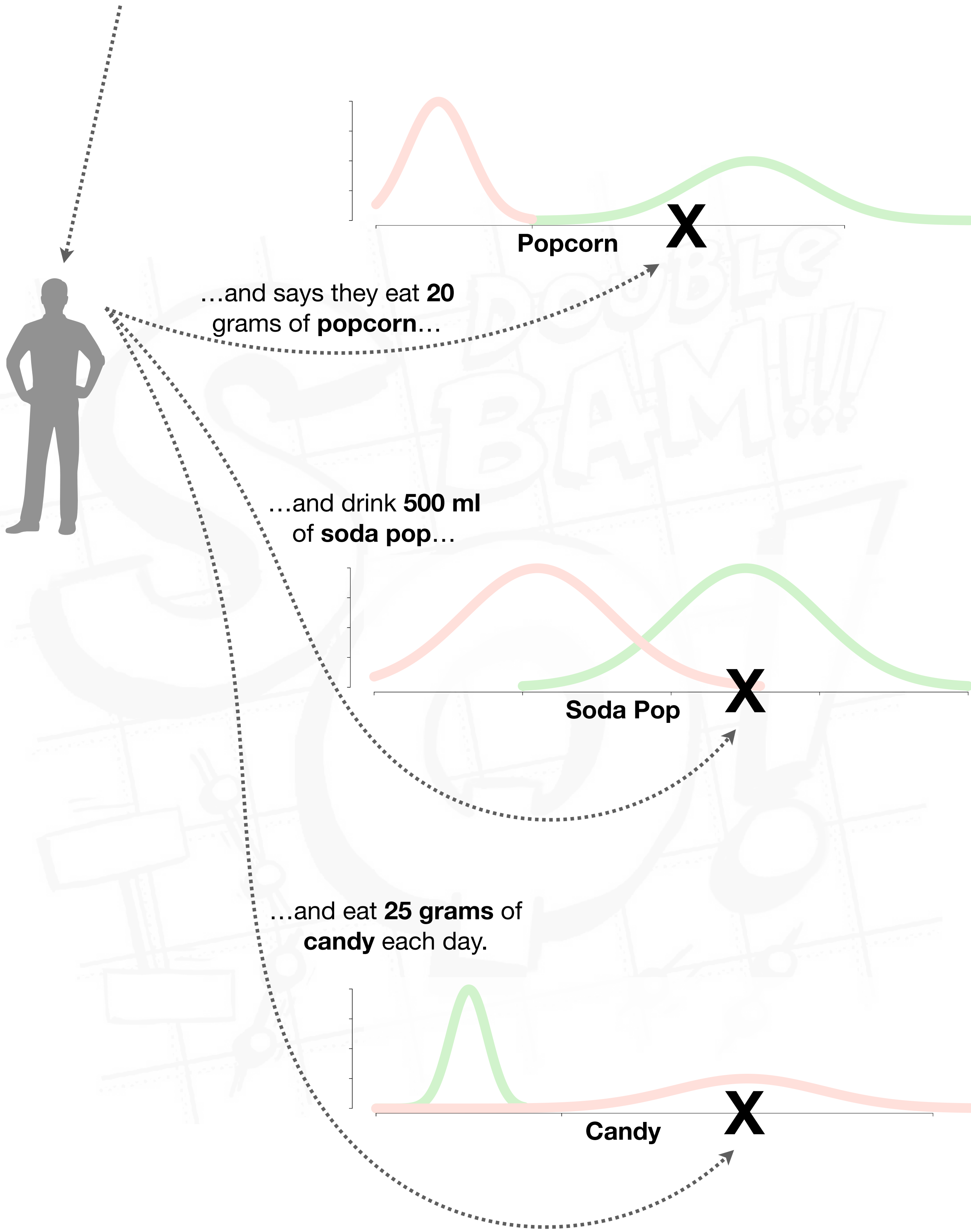
$$p(\textbf{No Love}) = \frac{\textbf{\# who don't love Troll 2}}{\textbf{Total \# of People}} = \frac{\mathbf{3}}{\mathbf{4} + \mathbf{3}} \boxed{= 0.4}$$



NOTES:



Now someone new shows up…



Step 3a) Calculate the score that the new person Loves Troll 2

The **Prior** probability that the person **Loves Troll 2**…

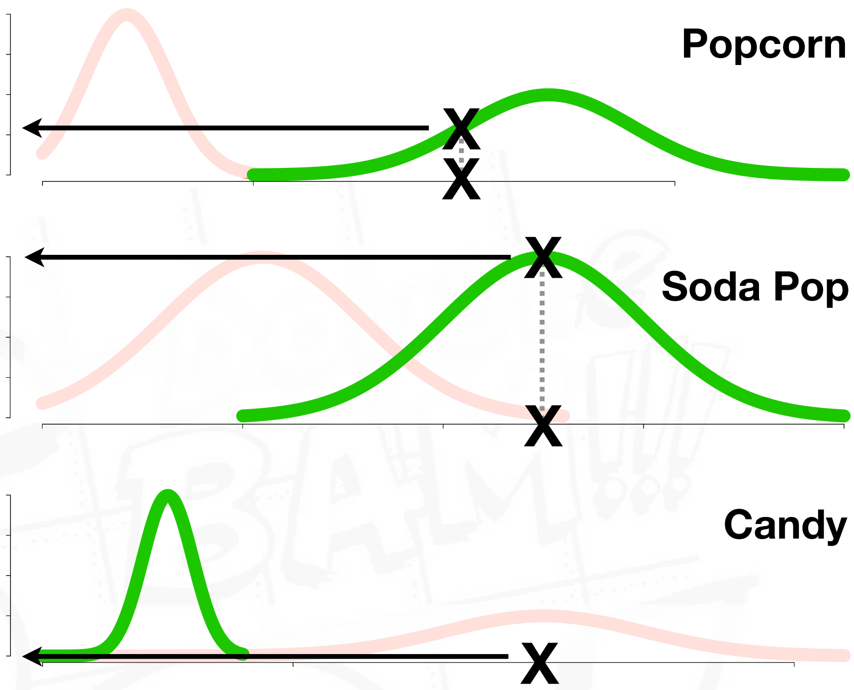
p(**Loves Troll 2**)

x *L*(popcorn = 20 | **Loves**)

x *L*(soda pop = 500 | **Loves**)

x *L*(candy = 25 | **Loves**)

…multiplied by the **likelihoods** that the new person consumes so much **popcorn**, **soda pop** and **candy** given that they **Love Troll 2**.



Plug in the numbers.

p(**Loves Troll 2**) → 0.6

x *L*(popcorn = 20 | **Loves**) → x 0.06

x *L*(soda pop = 500 | **Loves**) → x 0.004

x *L*(candy = 25 | **Loves**) → x a tiny number

Because computers have trouble with numbers very close to **0**, we take the **log.**

NOTE: In machine learning we usually use **log base *e***, aka **the natural log**…

log(0.6 x 0.06 x 0.004 x a tiny number) = **log**(0.6) + **log**(0.06) + **log**(0.004) + **log**(tiny number)

…and the **log** turns the multiplication into the sum of logs.

Doing the math gives us…

log(**Loves Troll 2 Score**) = **-124**

Step 3b) Calculate the score for Does Not Love Troll 2

The **Prior** probability that the person **Does Not Love Troll 2**…

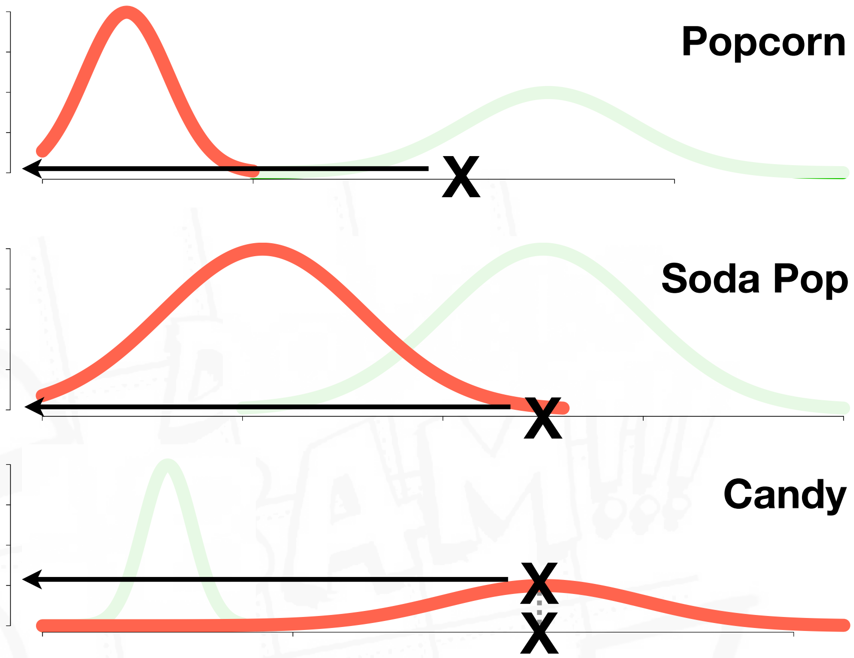
p(**No Love**)

x L(popcorn = 20 | **No Love**)

x L(soda pop = 500 | **No Love**)

x L(candy = 25 | **No Love**)

…multiplied by the **likelihoods** that the new person consumes so much **popcorn**, **soda pop** and **candy** given that they **Do Not Love Troll 2**.



Take the **log()**

log(p(**No Love**))

+ **log**(L(popcorn = 20 | **No Love**))

+ **log**(L(soda pop = 500 | **No Love**))

+ **log**(L(candy = 25 | **No Love**))

Doing the math gives us…

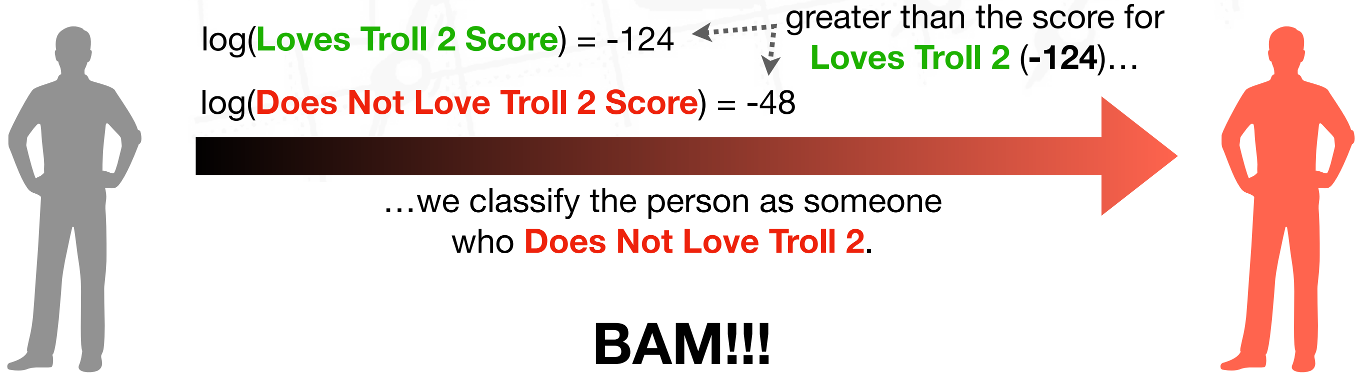
log(**Does Not Love Troll 2 Score**) = **-48**

Step 4) Classify the new person

Because the score for **Does Not Love Troll 2** (**-48**) is greater than the score for **Loves Troll 2** (**-124**)…

log(**Loves Troll 2 Score**) = -124

log(**Does Not Love Troll 2 Score**) = -48



…we classify the person as someone who **Does Not Love Troll 2**.

BAM!!!