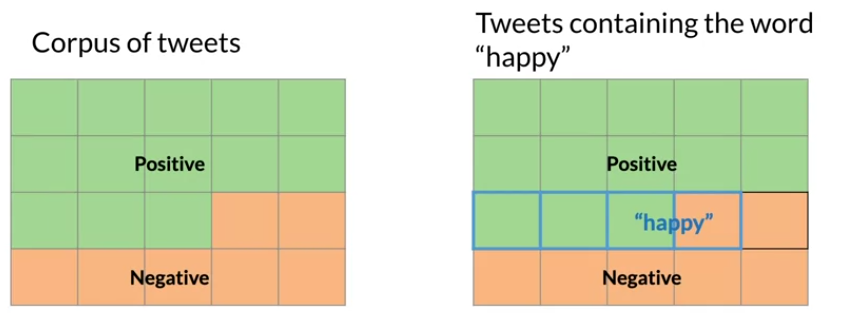
Imagine you have an extensive corpus of tweet that can be categorized

1. Positive
2. Negative

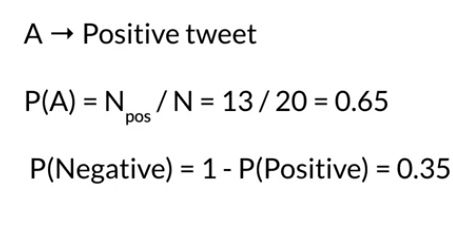
Within the croups, the word happy is something being labeled positive and something negative.



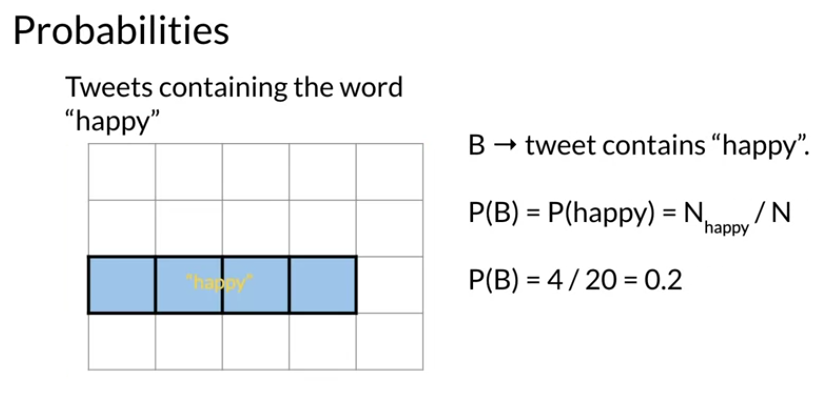
* One way to think about probabilities is by counting how frequently events occur



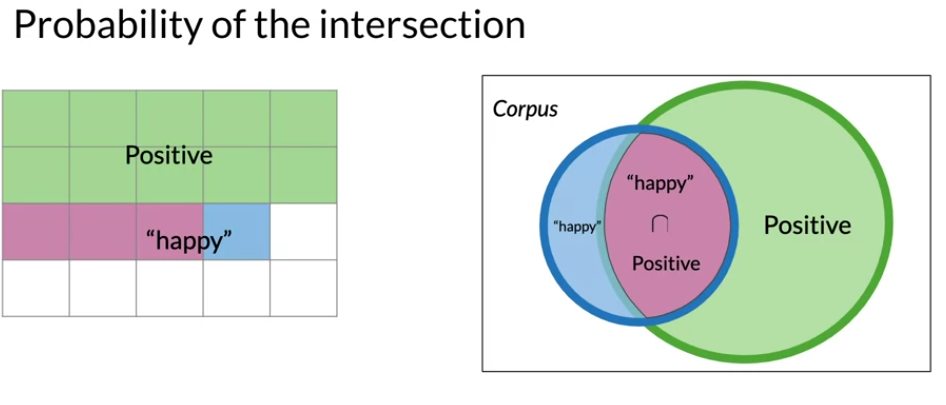
N => the corpus divided by the total number of tweets corpus.



Example



Another way

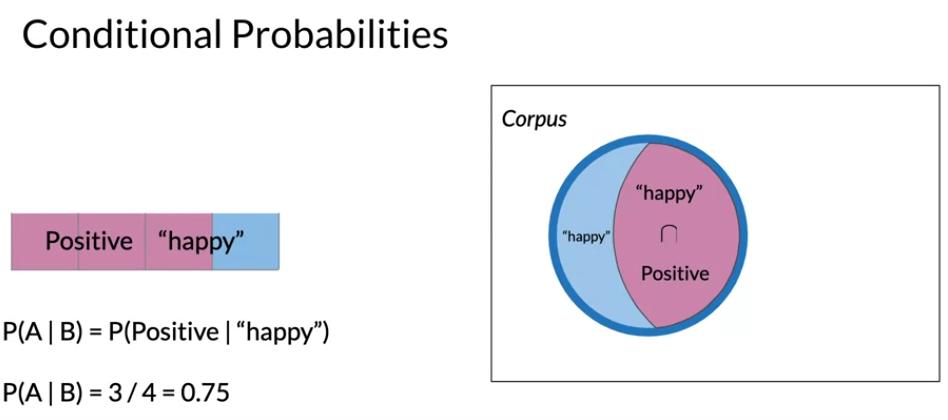


In the context of this diagram, the probability that a tweet s labeled positive and also contains the word happy is just ratio of the area of the intersection divided by the area of the entire corpus.



In the other words, if there were 20 tweets in the corpus and three of them are labeled positive and also contain the word happy, then the associated probability is 3/20 or 0.15.

Conditional probability

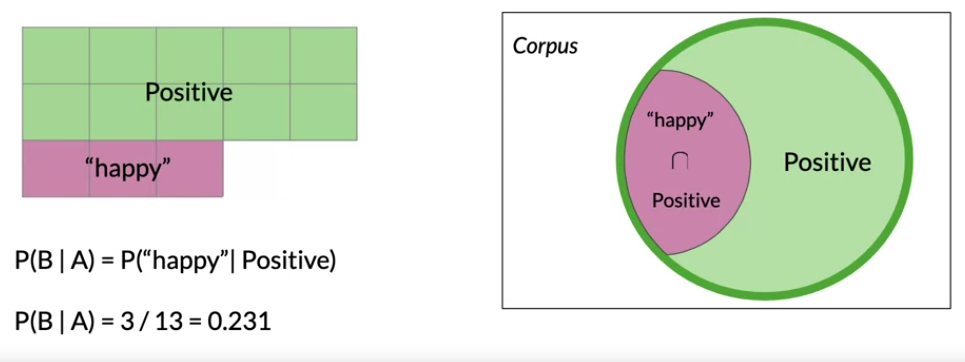


In this case, the probability that a tweet is positive given that it contains the word happy becomes the number of tweets that are positive and also contain the word happy and we divide that by the number that contain the word happy.

ملخص الكلام

لو عندك تويتات كلها بوستيف وفيها كلمة سعيد هتقسمها على تويتات ال فيها كلمة سعيد بس

You can make the same case for positive tweets.



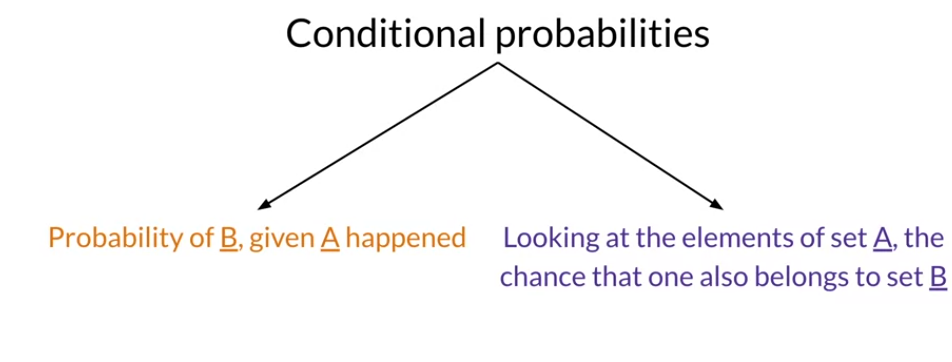
The purple area denotes the probability that a positive tweet contains the word happy.

In this case, the probability is 3 / 13.

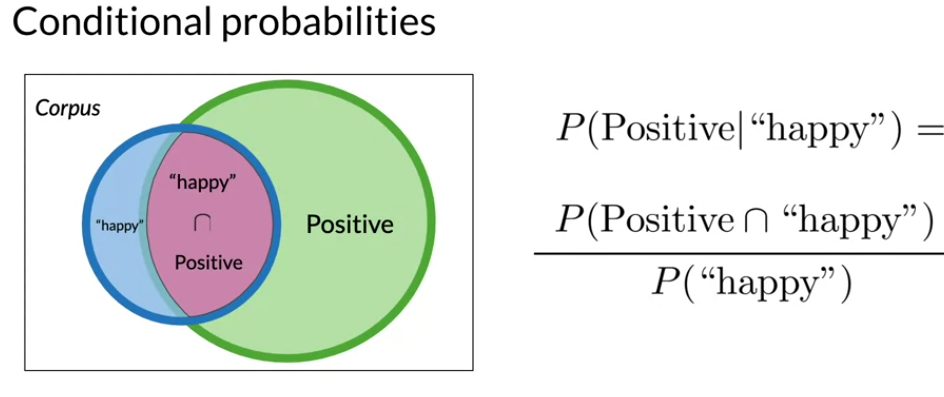
It is venn diagram

With all of this discussion of the probability of missing certain conditions,

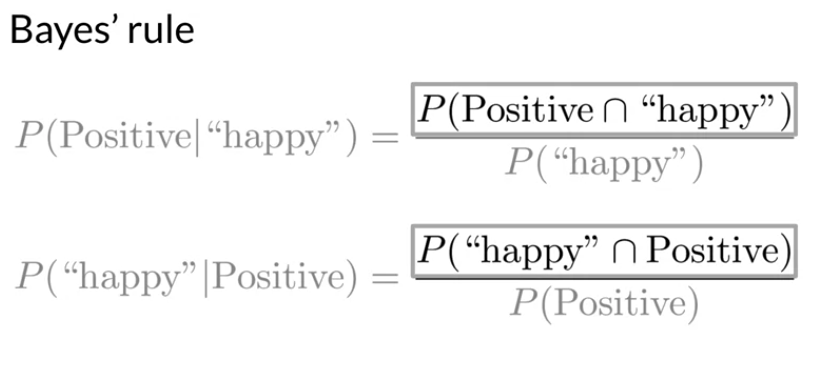
1. **Conditional probabilities could be interpreted as the probability of an outcome B knowing that event a already happened given A happened**
2. **Given that i`m looking at an element from set A , the probability that it also belongs to set B.**



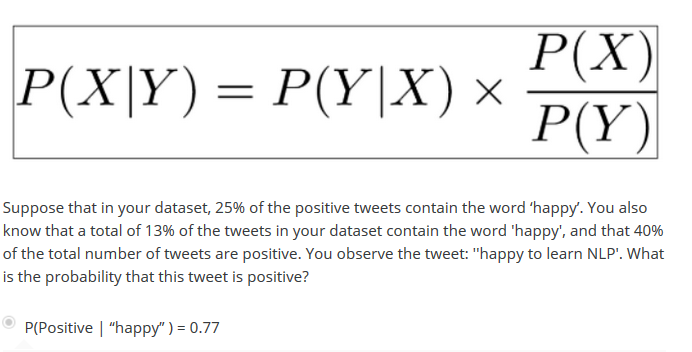
Example



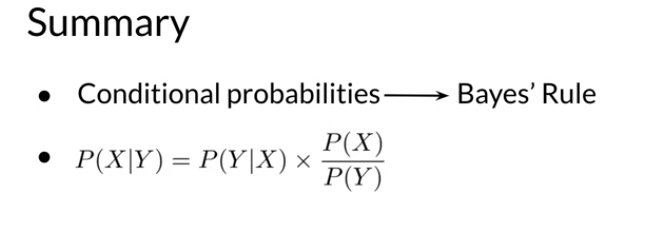
Combine this equation



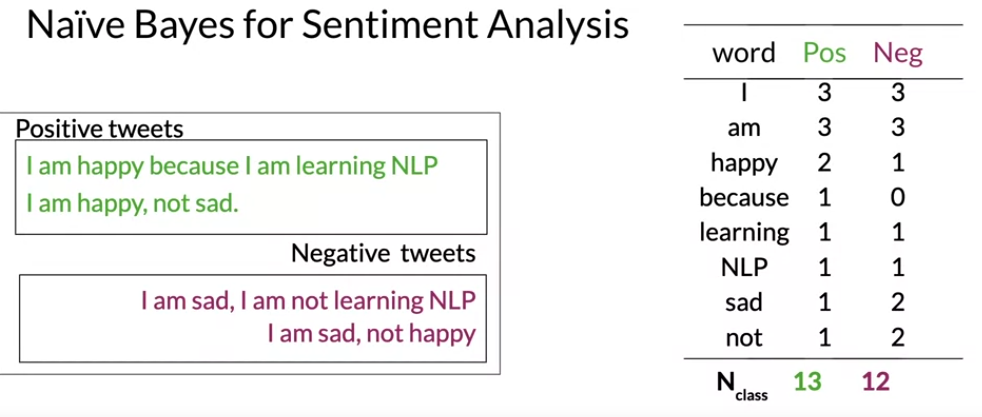
Quiz



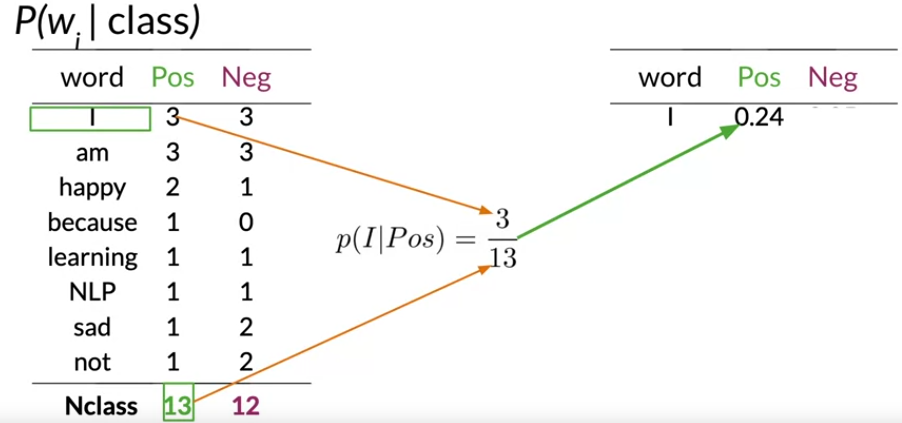
Summary



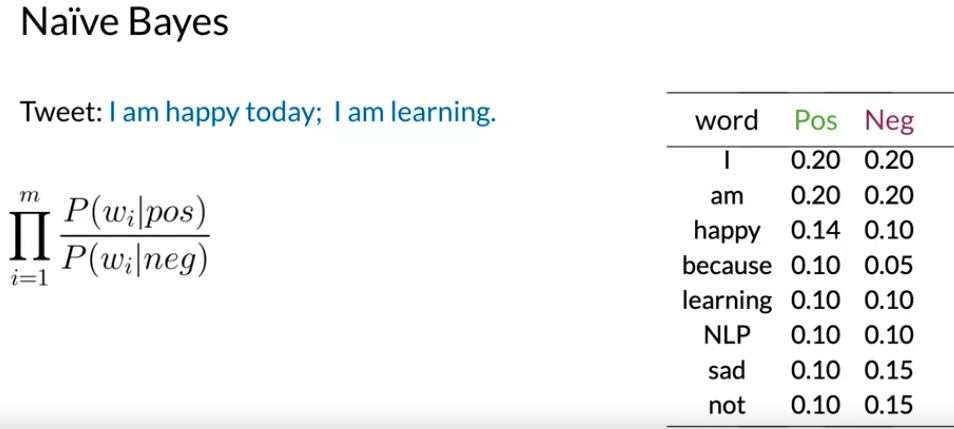
Total number of positive tweets is 13

Total number of negative tweets is 12

It allows you to compute the conditional probabilities of each word given the class as you are about to see.



First, note how many words have a nearly

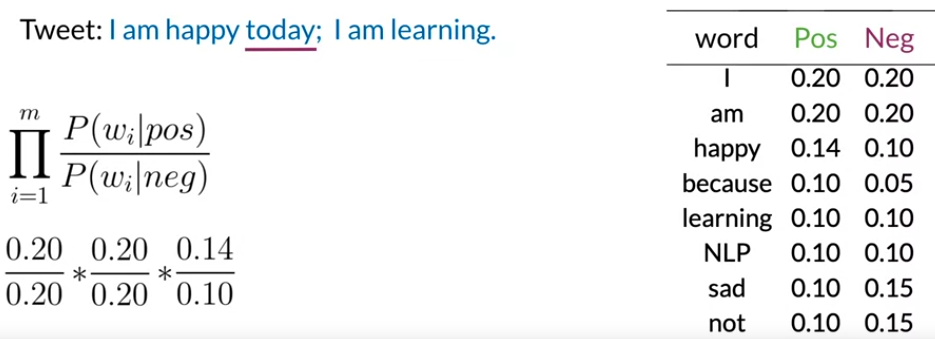


The inference condition rule for binary classification.

This expression says that you are going to take the product across of the words in your tweets of the probability for each word in the positive class divided by the probability in the negative class

I am happy today; I am learning

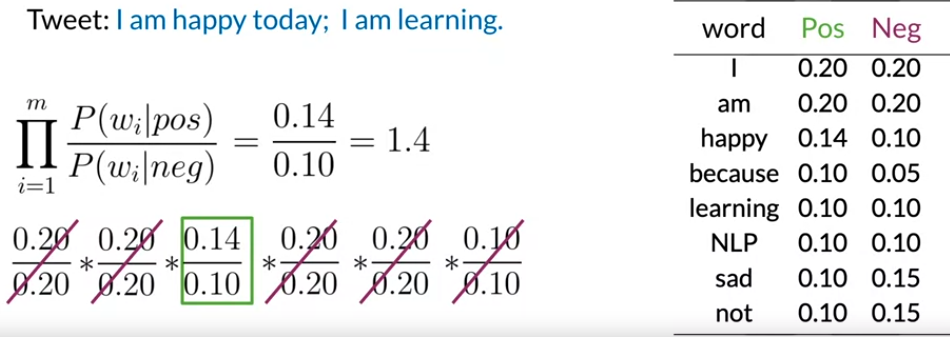
For each word, select its probabilities from the table.



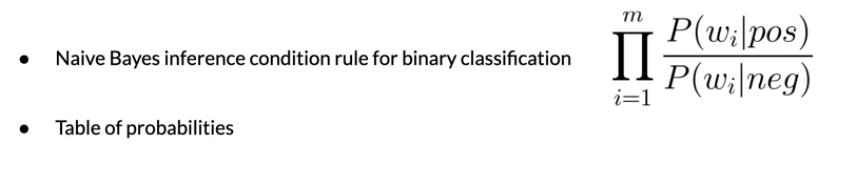
For today, you don`t find any word in table; meaning this word is not in your vocabulary.

So you won`t include any term in this score.

Now note that all the neutral words in the tweet like I and am just cancel out in the expression.



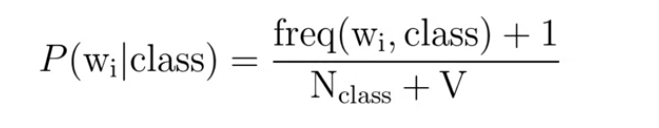
This value is higher than 1 : it mean in the tweets are more likely to correspond to positive sentiment, so you conclude that the tweet is positive.



Laplacian smoothing : it is technique to avoid probabilities being zero.

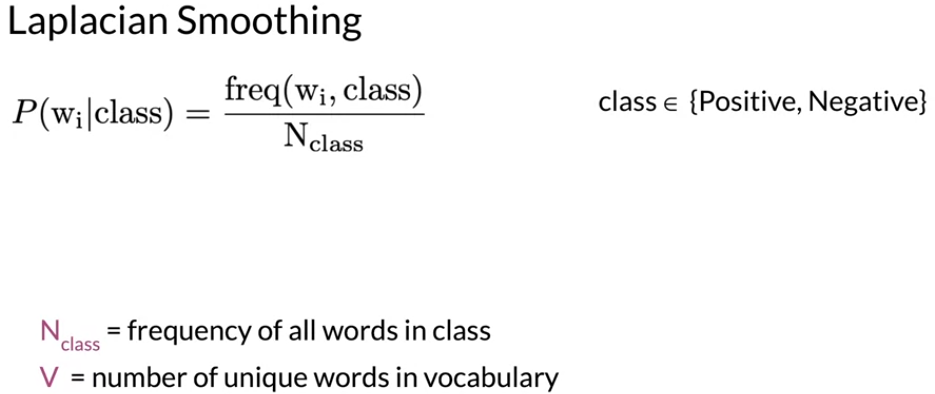
The expression used to calculate the conditional probability of a word given the class, is the frequently of the word in the corpus shown here as freq. of word I, class divided by the number of words in the corpus or N class.

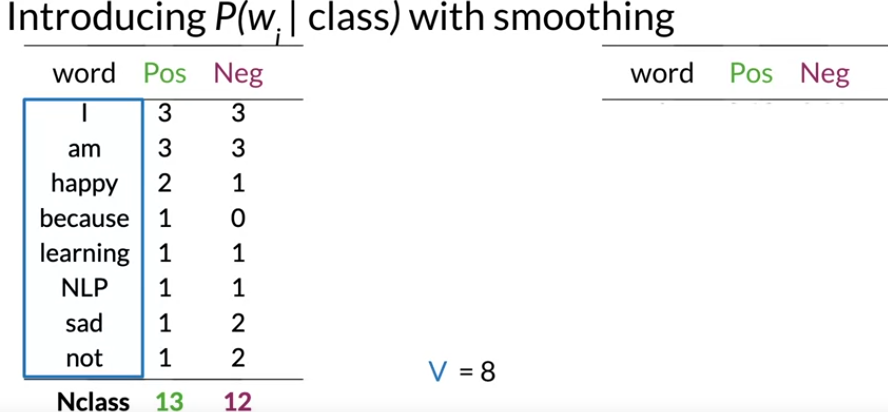
Smoothing the probability function means that you will use a slightly different formula from the original.



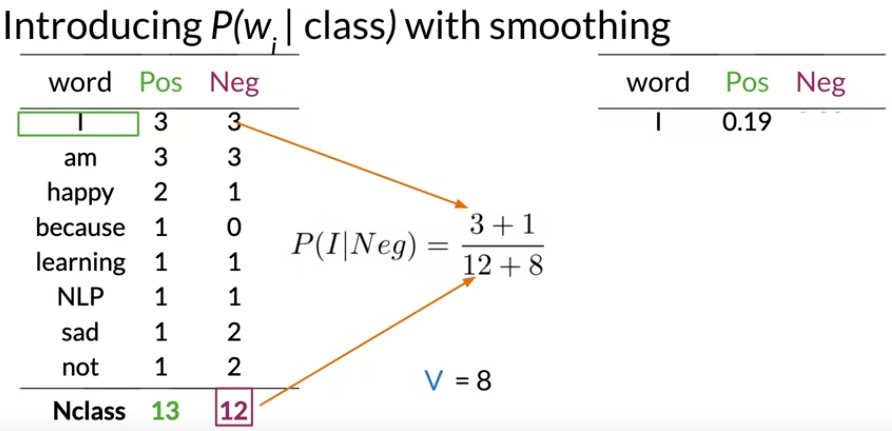
This little transformation avoids the probability being zero.

* It adds the new term to all the frequencies that is not correctly normalized by N class.
* You add new term in the denominator v class.



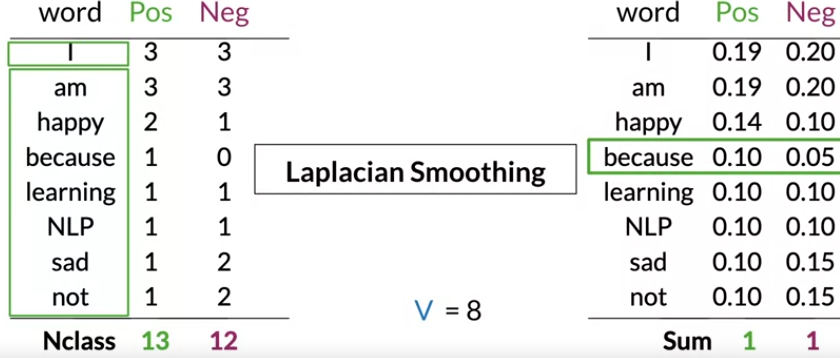


In this example, let`s calculate the probability for each word in the positive class.



Laplacian smoothing

In word because



Used Laplacian smoothing to avoid 0

Log likelihoods

Ratio of probabilities:

Words can have many shades of emotional meaning.

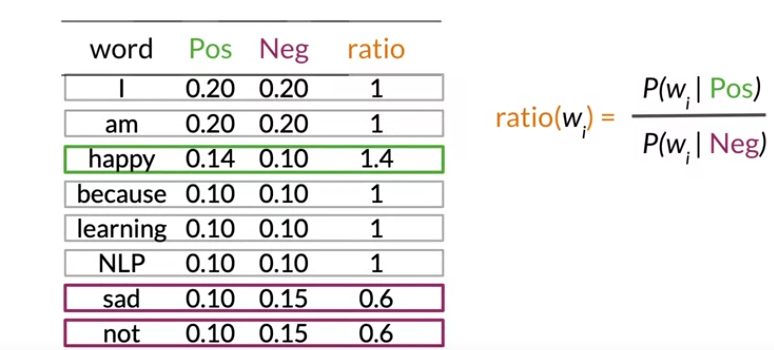
But for purpose of sentiment classification, they are simplified into three categories:

1. Neutral
2. Positive
3. Negative

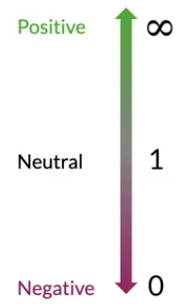
All can be identified by using their conditional probabilities.

These categories can be numerically estimated just by dividing the corresponding conditional of this table.

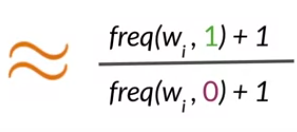
So the ratio before



Infinity => positive

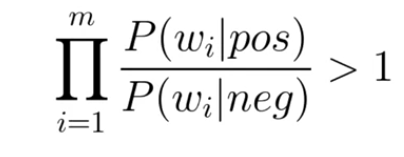


On the other hand, negative words have a ratio smaller than one.



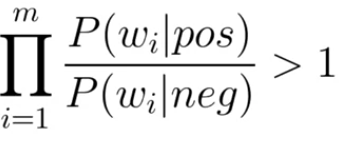
These ratio are essential in naïve Bayes’ for binary classification.

I will illustrate why using an example you have seen before



The formula to categorize a tweet as positive

* If the products of the corresponding ratios of every word appears in the tweet is bigger that one.



* We said it was negative if it was less than one.

This is called the likelihood

If your are take the ratio between the positive and negative tweets, you would have what`s called prior ratio.

In the future through, when you are building your own application, remember that this term becomes important unblamed datasets. With the addition of the prior ratio, you now have the full naïve Bayes formula for binary classification.

* A simple, fast and powerful baseline
* A probabilistic model used for classification

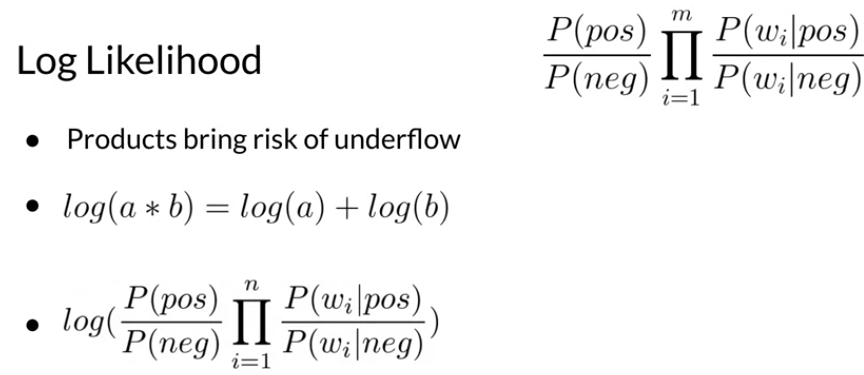


Can used to establish a baseline quickly.

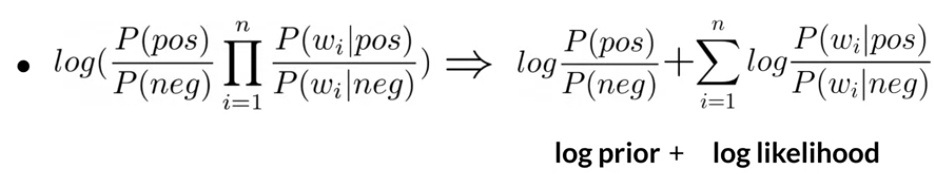
Sentiment probability calculation requires multiplication of many numbers with values between zero and one.

Carrying out such multiplications on their computer runs the risk of numerical underflow when the number returned is so small it can`t be stored on your device.

Luckily there`s a mathematical trick to solve this it involves using a property of logarithms.



The score instead of the raw score.

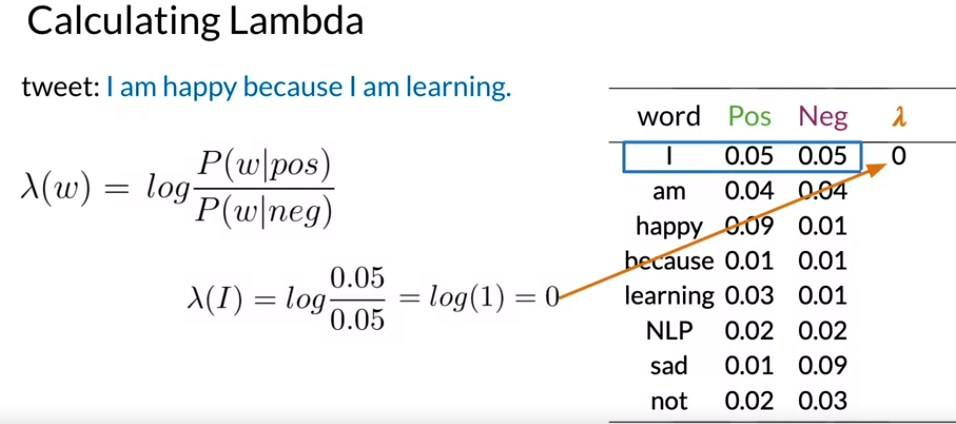


The sum of **the log prior** and the **log likelihood**, which is a sum of the logarithms of the conditional probability ratio of all unique words in your corpus.

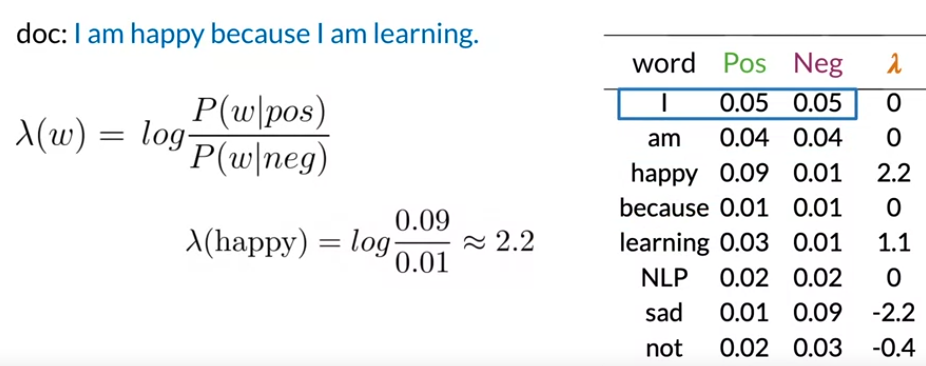
The values in the table here are unrelated to the previous dataset. It uses a bigger corpus and shows just a subset of the vocabulary. This results in lower probabilities and the sums per column not adding up to 1.

When you need to calculate the log of the score is called the lambda : this is the log of ratio of the probability that your word is positive and you divide that by the probability that the word is negative.

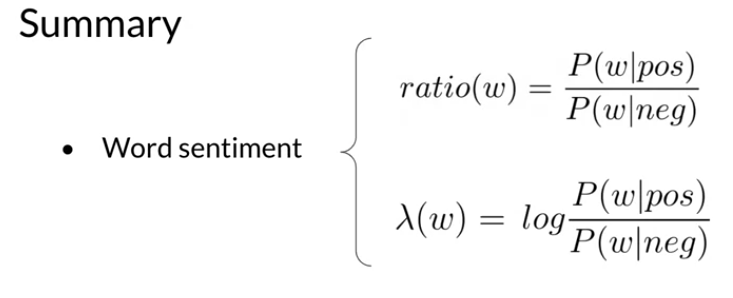
Remember, the tweet will be labeled positive if the product is larger than one.



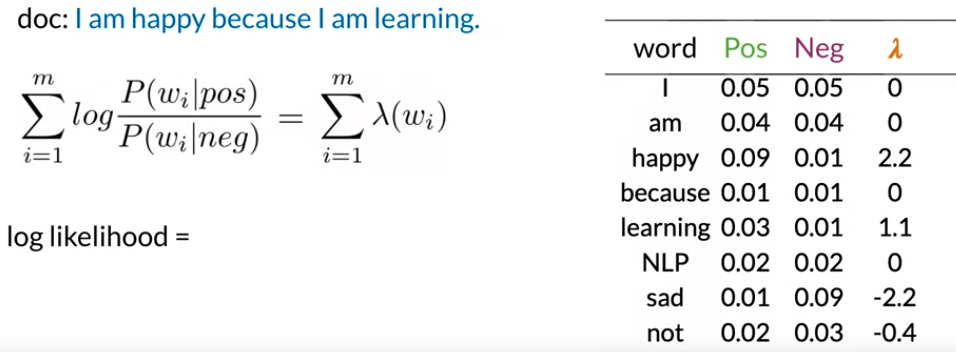
By this logic, I would classify s neutral at zero.



From here on out, you can calculate the log of score of the entire corpus just by summing out the lambdas.

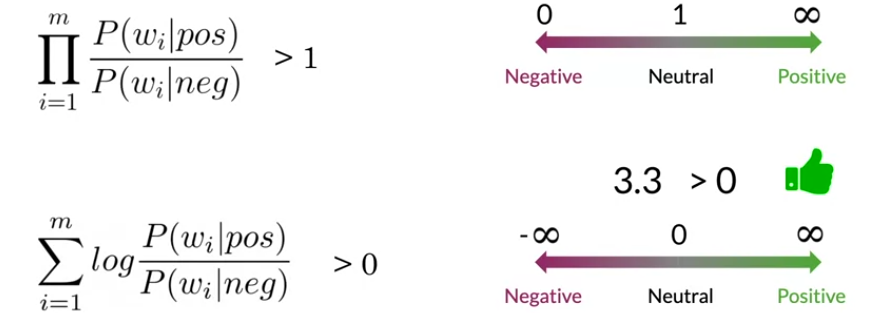


**Now you can calculate the log-likelihood of the tweets as the sum of the lambdas from each word in the tweets.**





This value is higher than 0.

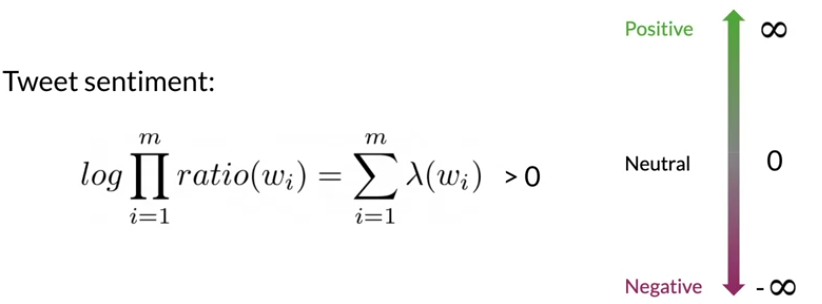


Notice: that this score is based entirely on the words happy and learning, both of which carry a positive sentiment.

All the other words are neutral and didn`t contribute to the score.

See how much influences the power words have

The score is called the log-likelihood. Sum all lambada



In naïve baye`s no gradient descent. We are just counting frequencies of words in a corpus.

انت بس هتعمل هتعد الارقام المتكرره ال فى corpus.

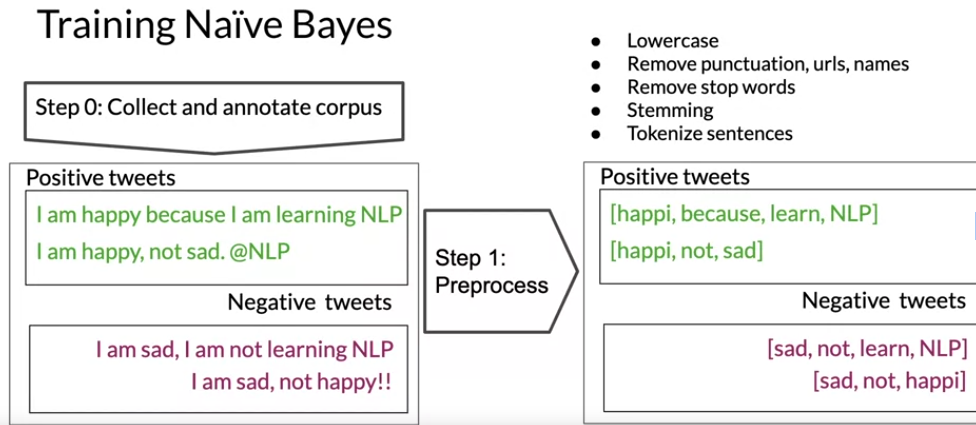
You know be creating step by step a naïve bayes model for sentiment analysis using a corpus of tweets that you are already collected.

The first step for any supervised machine learning project

1. Collect and annotate corpus. To gather the data to train and test your model.

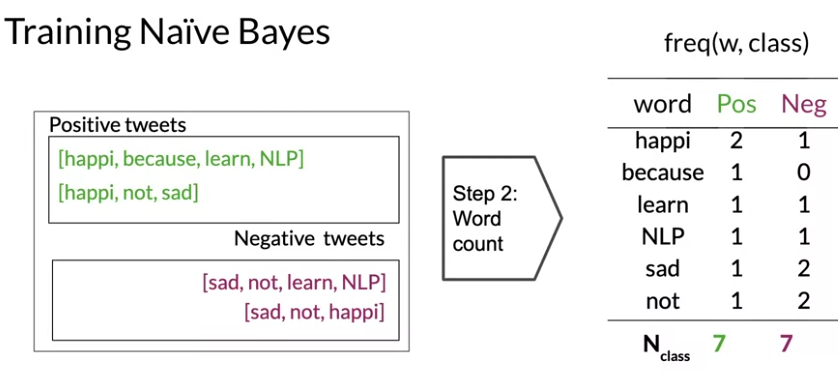
For sentiment analysis of tweets, this step involves getting a corpus of tweets and dividing it into groups, positive and negative tweets.

1. **Lowercase**
2. **Remove punctuation urls, names**
3. **Remove stop words.**
4. **Stemming**
5. **Tokenize sentence**



Second process

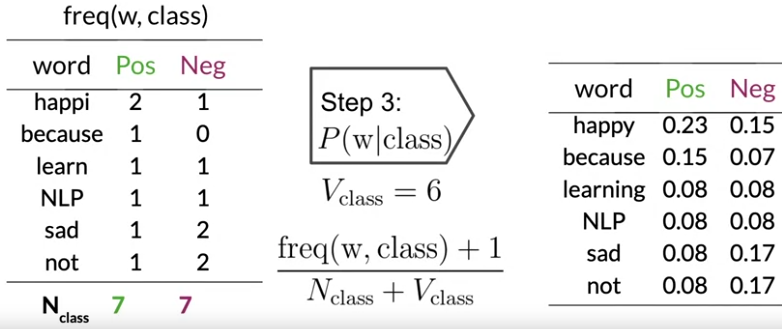
1. Computing the vocabulary for each word and class, like you did in the previous week



You can compute the sum of words and class in each corpus in this same step.

From this table of frequencies, you get the conditional probability by using Laplacian smoothing.

V is not the total number of words in the original corpus.

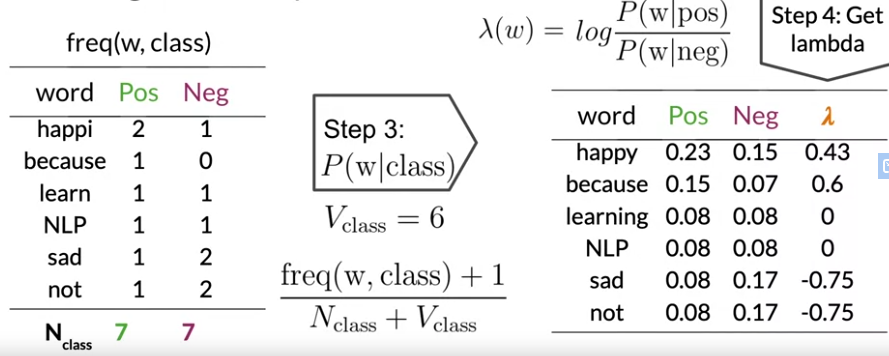


This is produce a table of conditional probabilities for each word and each class.

This table only contains value greater than 0.

The Fourth step:

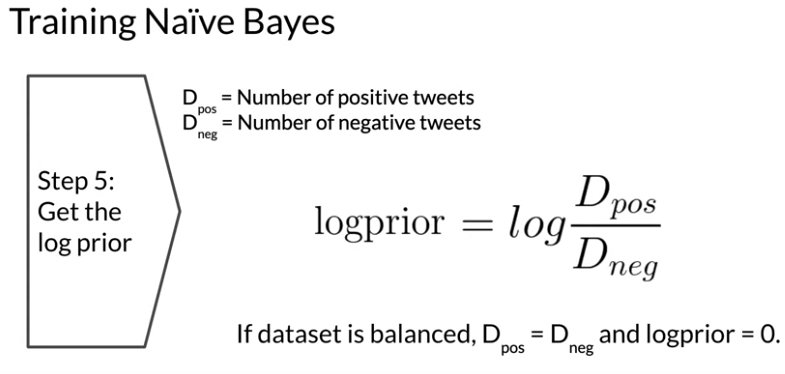
You will get the lambada square for each word, which is the log of the ratio your conditional probabilities.



The five step: is estimation of the log prior.

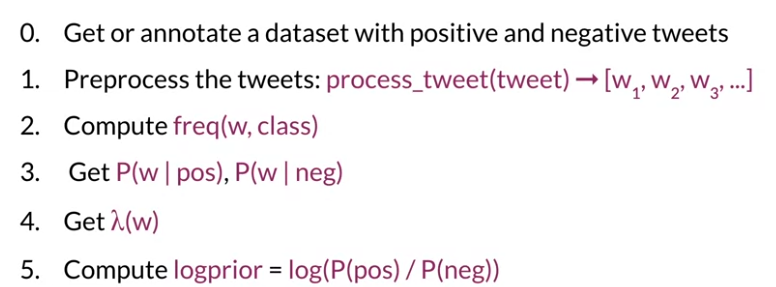
To do that: you will need to count the number of positive and negative tweets.

And then the log prior is the log of the ratio of the number of positive tweets over the number of negative tweets.



But for unbalanced data sets, this term will become important.

Summary



A confidence ellipse is a way to visualize a 2D random variable. It is a better way than plotting the points over a cartesian plane because, with big datasets, the points can overlap badly and hide the real distribution of the data. Confidence ellipses summarize the information of the dataset with only four parameters:

Center: It is the numerical mean of the attributes

\* Height and width: Related with the variance of each attribute. The user must specify the desired amount of standard deviations used to plot the ellipse.

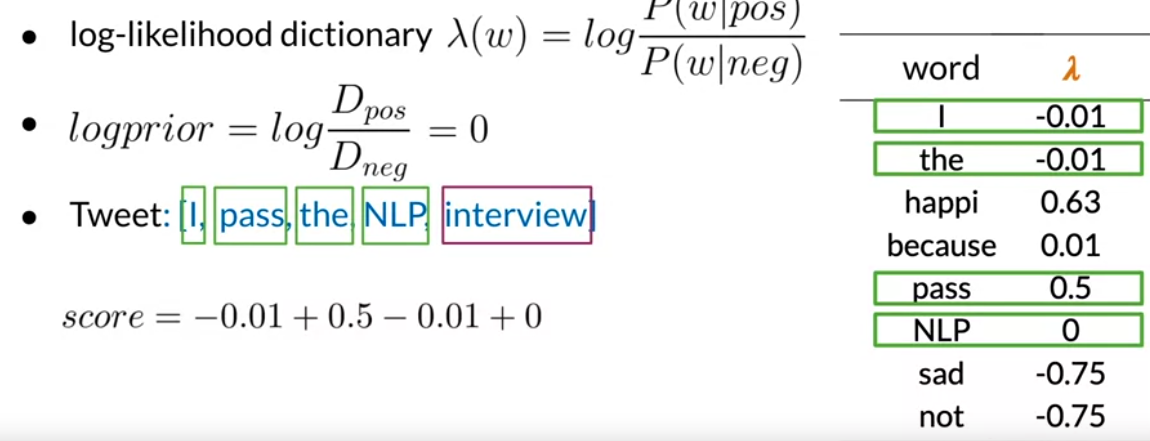
\* Angle: Related with the covariance among attributes.

The naïve Bayes classifier or real text exmpales.

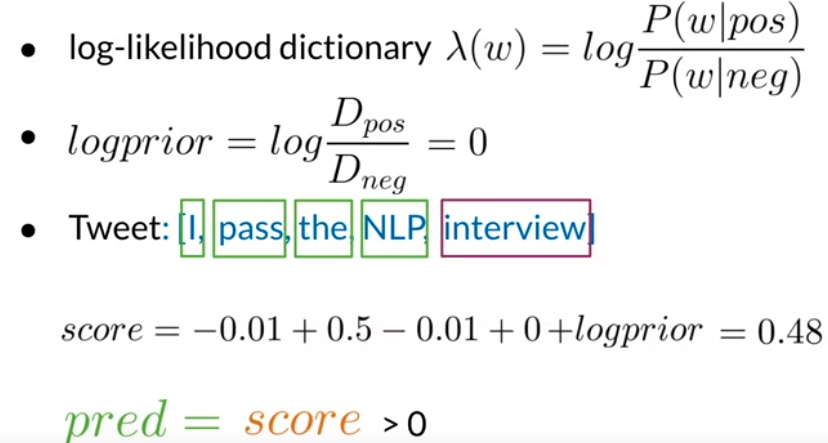
Using your validation set to compute accuracy

With the calculations you have done already, you have a table with lambda score for each unique word in your vocabulary.

With your estimation of the log prior, you can predict sentiment or a new tweet.



Now you can add the log prior to account for the balance or imbalance of classes in the dataset.



Remember is the score is bigger than zero, than this tweet has a positive sentiment.

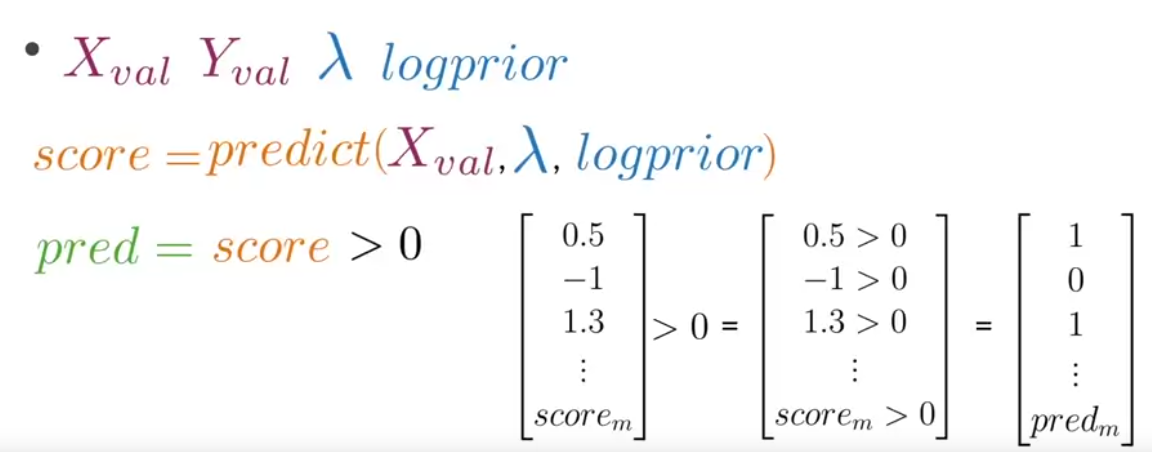
Test performance of classifier:

A different scenario in the previous module

Let`s quickly review that process as applied to naïve bayes.

This week`s assignments includes a validation set.

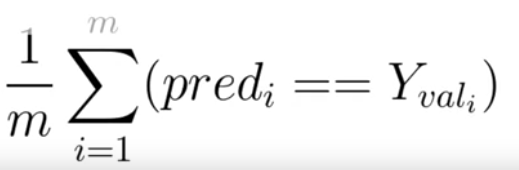
This week `s assignments includes a validation set.



Then evaluates whether each score is greater than zero.

This produces a vector populated with zeros and ones indicating if the predicted sentiment is negative or positive respectively for each tweet in the validation sets.

We can compute the accuracy if your model over the validation sets.



You will compare your predications against the true value for each observation from your validation data, y\_val.

If the values are equal and your predication is correct, you will get the value of 1 and 0 if incorrect.

To test the performance of your naïve bayes model, you use a validation set to allow you predict the sentiment score for an unseen tweet using your newly trained model.

**Application of naïve baye`s**

1. **Identify who`s an author of a text.**

if you had two large corporal, each written by different authors you could train model to recognize whether a new document was written by one or the other.

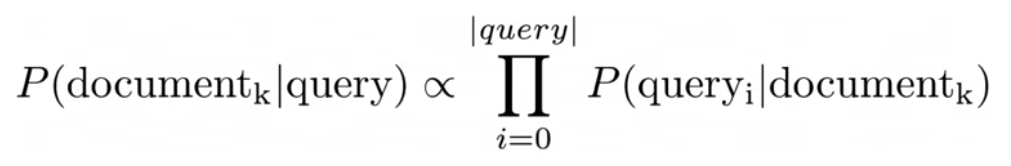
1. Spam filter

Using information taken from the sender, subject and content you could decide whether an email is spam or not.

1. Information retrieval

One of the earliest uses of naïve was filtering between relevant and irrelevant documents in a database.

* Given the sets of keywords in a query, in this case, you only need to calculate of the document given the query.



You can`t know beforehand what`s irrelevant or a relevant document looks like.

So you can compute the likelihood for each document in your dataset and then store the document based on its likelihoods.

You can choose to keep the first M result or the ones that have a likelihood larger than a certain threshold.

1. Word disambiguation

Meaning of word in two document.

Outline

The first assumption is independence between the predictors or features associated with each class.

The second: has to with your validation sets.



Naïve bayes : assume that the words in a piece of text are independent of one another, but as you can see this typically isn`t the case

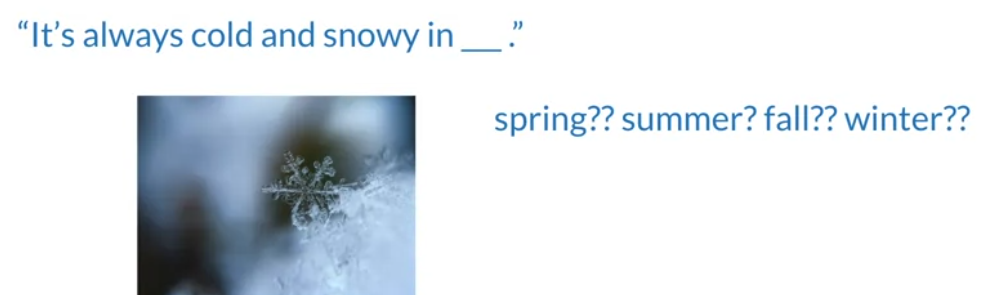
The word sunny and hot often appear together as they do in this exmpale.

Taken together as they might also related to the thing they are describing like a beach or a desert.

So the words in a sentence are not always necessarily independent of one another , but naïve bayes assumes that they are.

This could lead you to under or over estimate the conditional probability of individual words.

When using naïve bayes for example, if your task was to complete the sentence



It`s always cold and snowy in blank naïve byes might assign equal probability to the words spring summer and winter even through from the context you can see that winter should be the most likely candidate.

Another issue with naïve that it relies on the distribution of the training a good data set will contain the same proportion of positive and negative tweets as a random sample would

Most of the available annotated corpora are artificially balance just like data set

In real tweet is sent to occur more often their negative counterparts.

This is that negative tweets might contain content that is balanced by the platform muthed by the user such as inappropriate or offensive vocabulary

**Error analysis:**

1. **Removing punctuation and stops words.**
2. **Word order (how word order affects the meaning of a sentence.)**
3. **Adversarial attacks (some quirks of language come naturally to humans but confuse naïve Bayes models)**

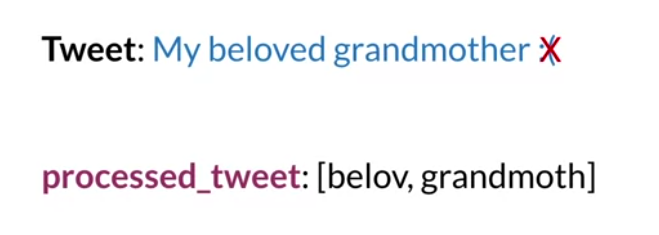
Note

One of your main consideration when analyzing errors NLP systems. Is what the processed version of the text actually looks like.

Tweet: My beloved grandmother ☹

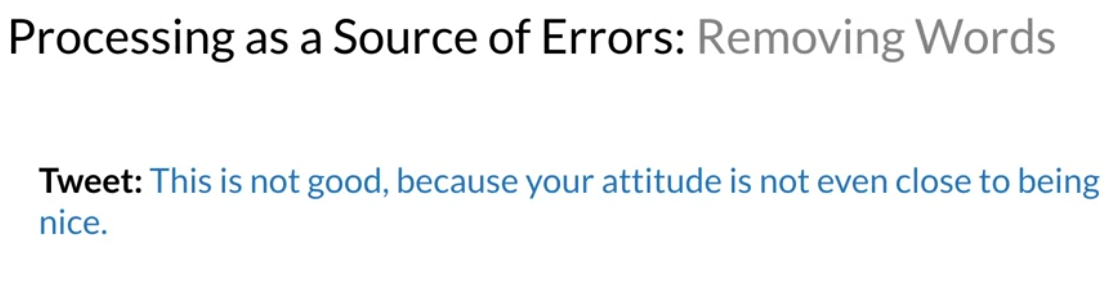
“ The sad face punctuation in this case is very important to the sentiment“

But if you are removing punctuation, then the processed tweet will leave behind only

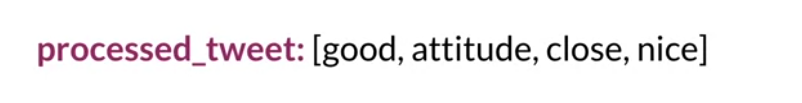


Which looks a very positive tweet.

My beloved grandmother, exclamation mark would be a very different sentiment.



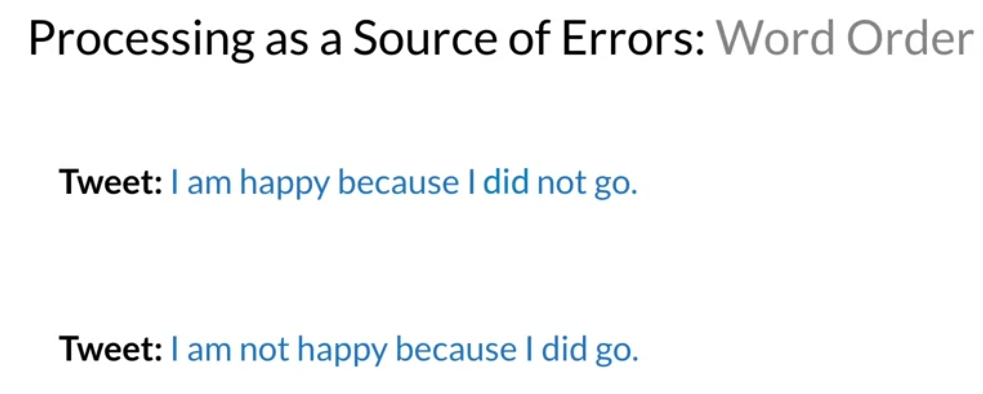
If we remove neutral words like not and this, what you are left with the following.



The any classifier will infer that this is something very positive.

I will talk later on about handling nots and word orders.

Make sure your model will be able to get an accurate read.



The inputs pipeline isn`t the only potential source of trouble

1. First : is a purely positive tweet.
2. Second: Negative sentiment

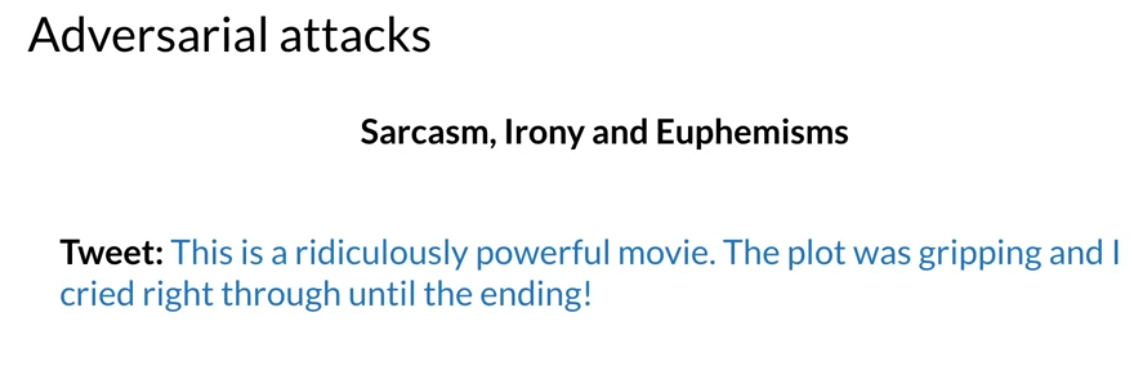
It get missed by your naïve bayes classifier.

So word order can be as important to spelling.

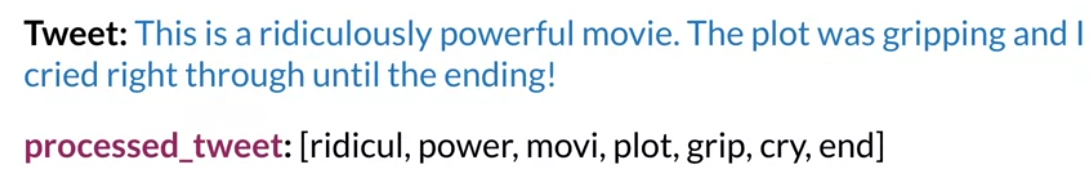
**Adversarial attacks**

Sarcasm, irony and euphemisms

The term adversarial attack describes some common language phenomenon,



A somewhat positive movie, but pre-processing might suggest otherwise.



If you pre-process this tweet, you will get a list of mostly negative words, but