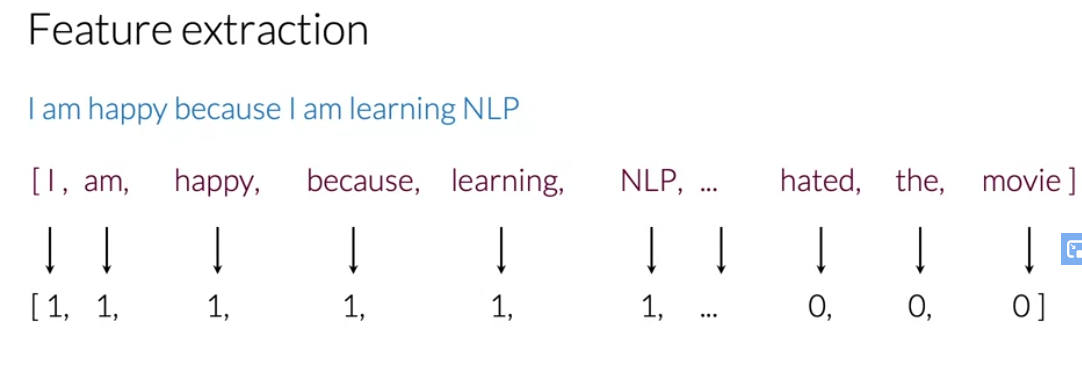
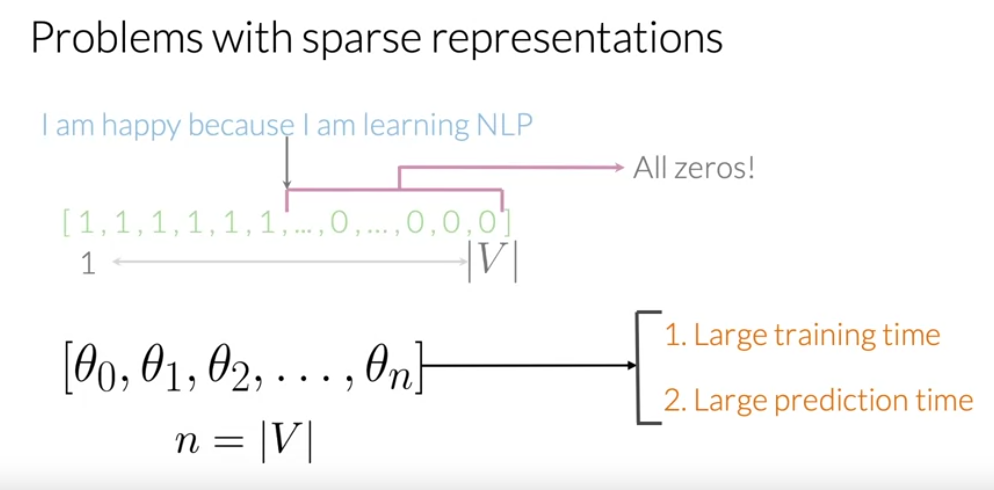
**Locality sensitive hashing: which method will help you with efficient search.**

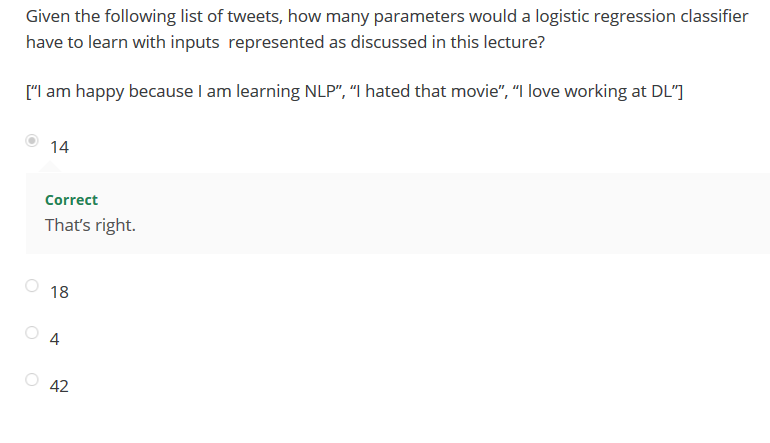
**Sparse representation**



Problem in the sparse



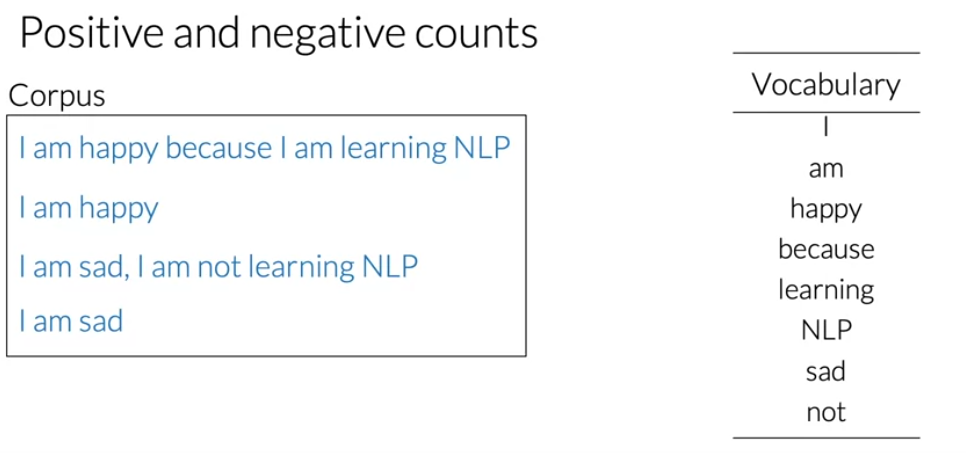
Quiz



How it is work ?

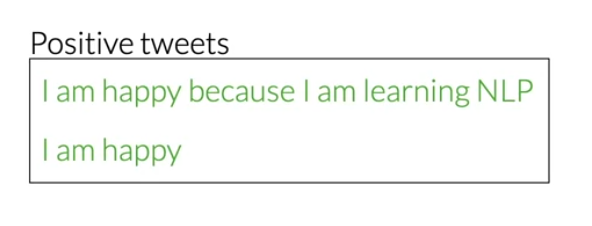
Specifically, given a word you want to keep track of the number of times, that`s where it shows up as the positive class. And given another word you want to keep the number of times that word showed up in the negative class.

Negative and positive Frequencies



You have a nine vocabulary

And we have two tweet as a positive



And negative class

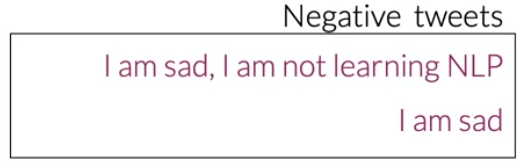


Table for tweet

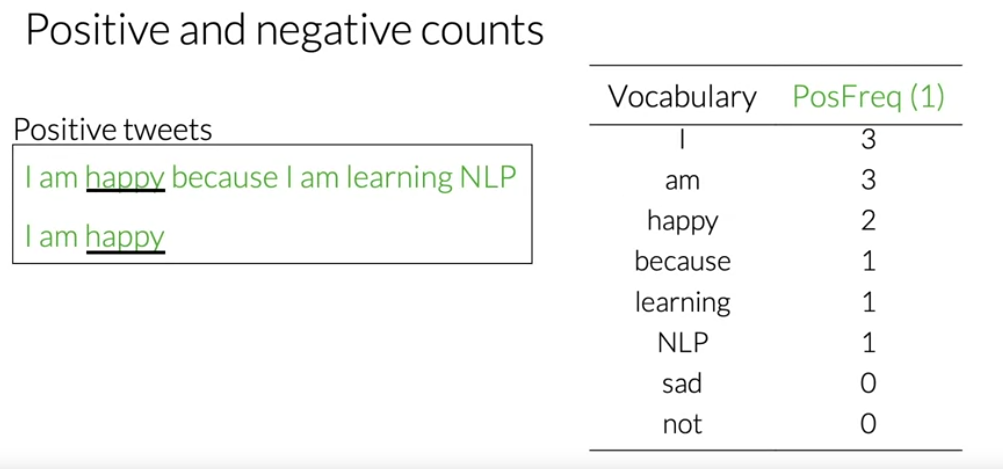
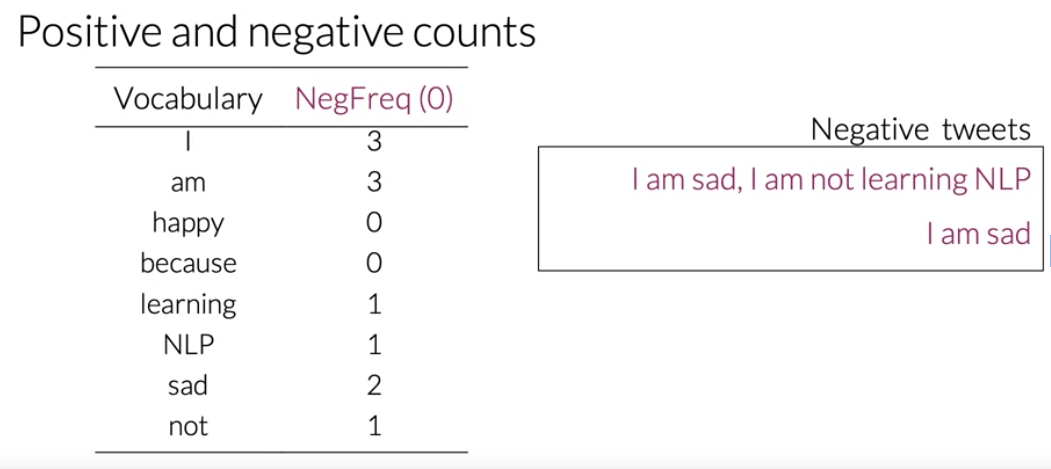
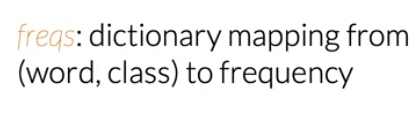


Table for tweet negative

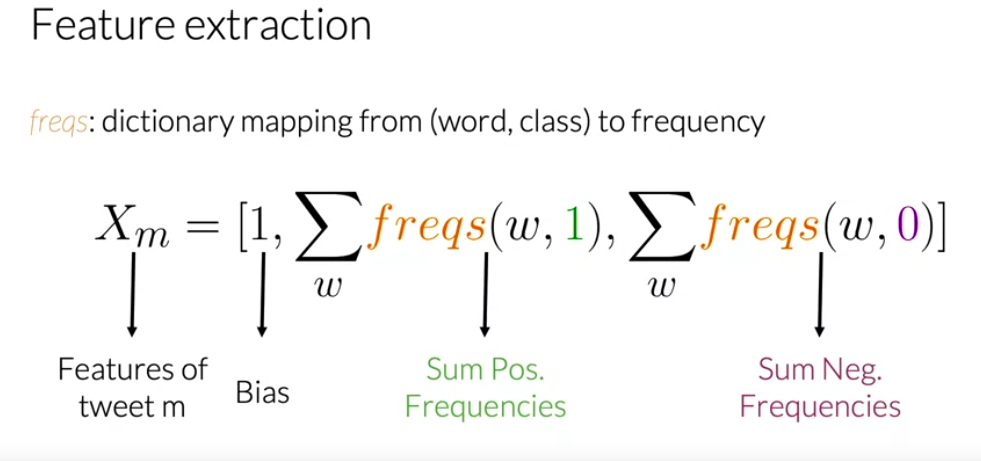


Dic

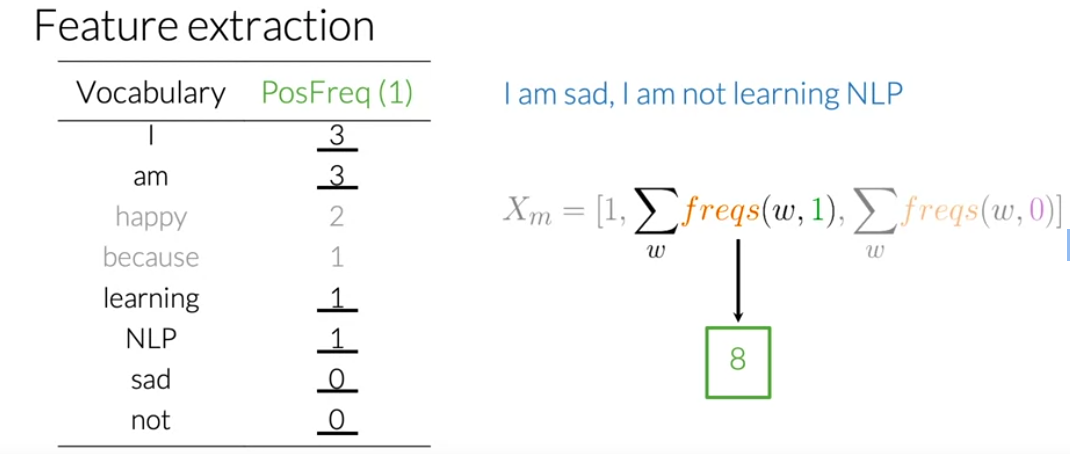


In doing so, you will have a much faster speed for your logistic regression classifier, because instead of learning V features you only to have learn three features.

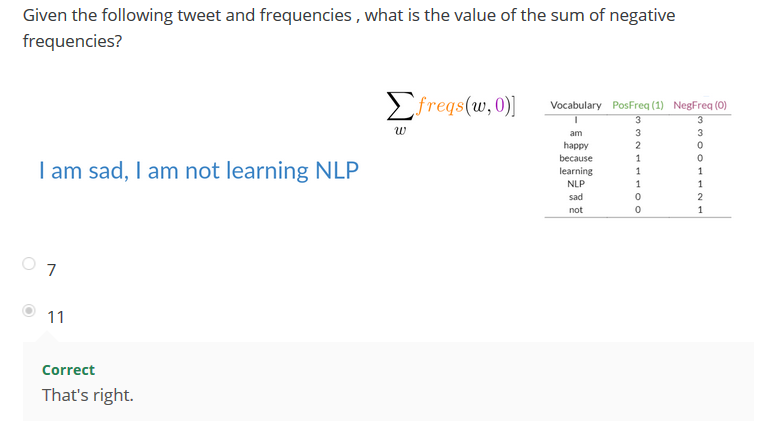
Feature extraction



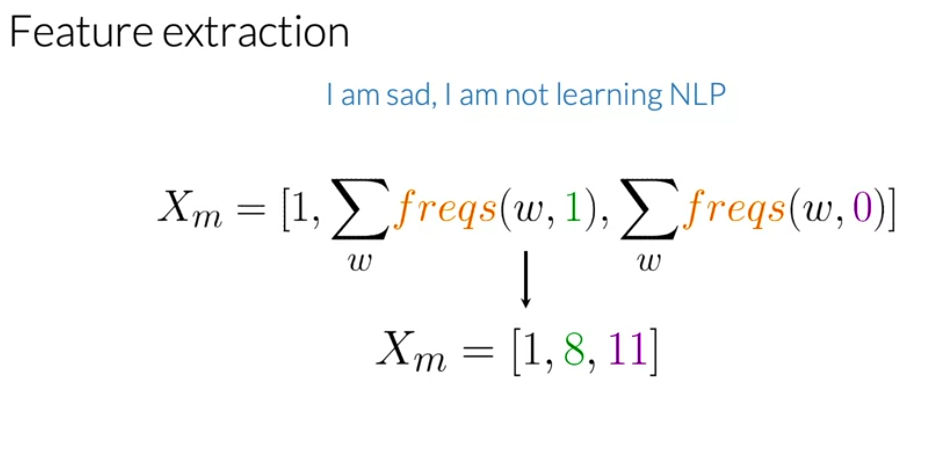
Sum the values but not sum of values not appear in tweet



For negative



Vector representation



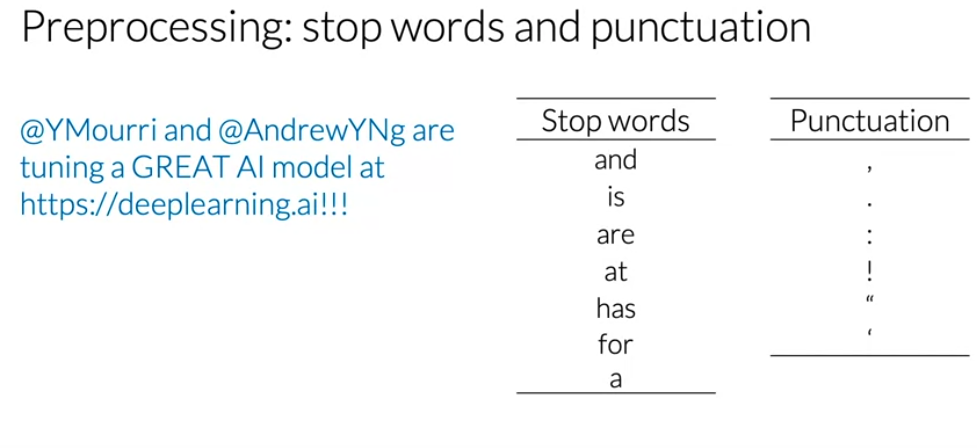
Vec (1x3) dim

Pre-processing

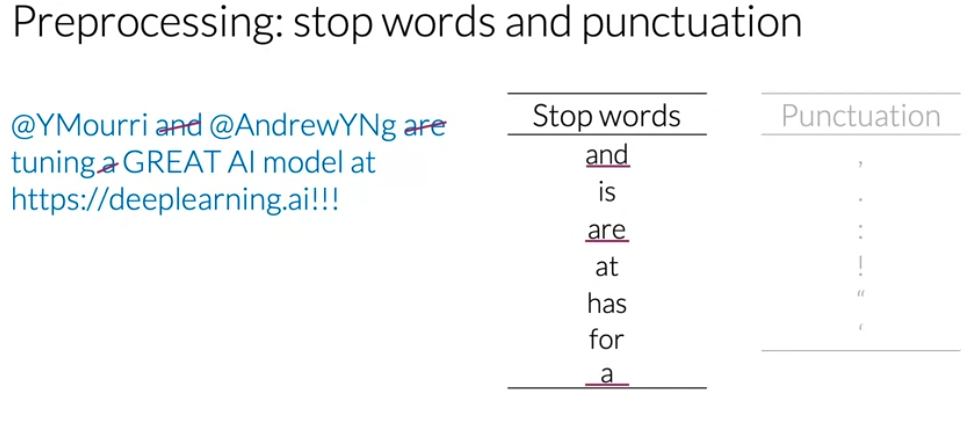
1. Stemming
2. Stop words

You would have compare your tweet against two list.

One stop words English and another with punctuation

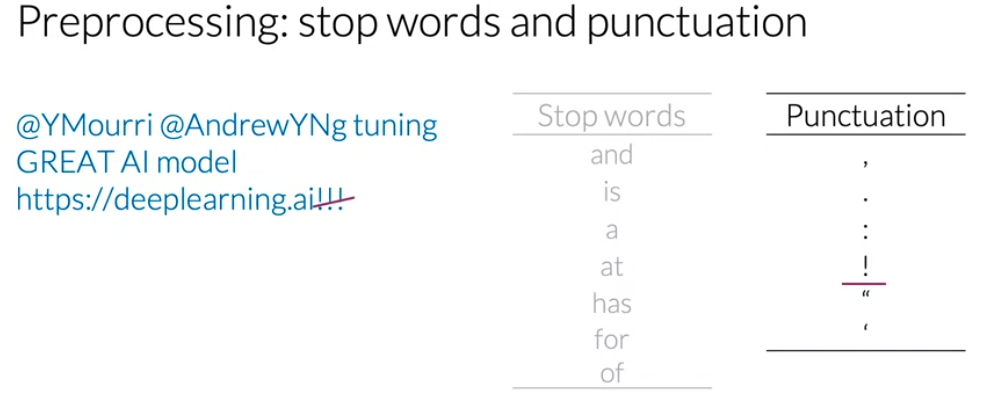


* Every word from the tweet that also appears on the list of stop words should be eliminated.



Note that the overall meaning of the sentence could inferred without any effort

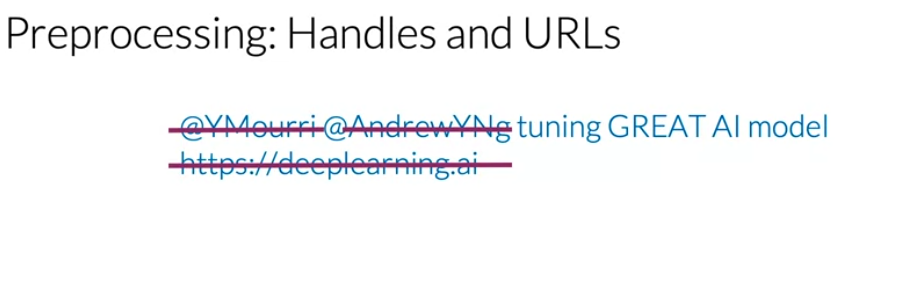
* Now, let`s eliminate every punctuation mark.



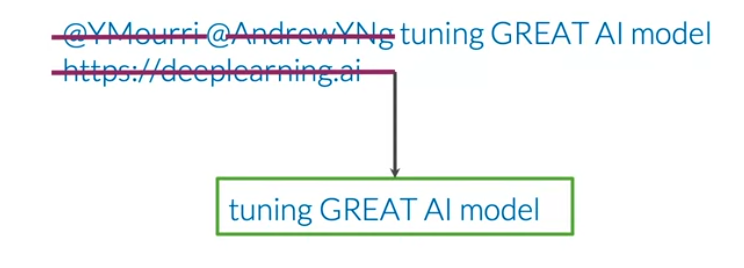
Some context you won`t have to eliminate punctuation.

You should care about important information to your specific NLP task or not.

Eliminate handle and URLs

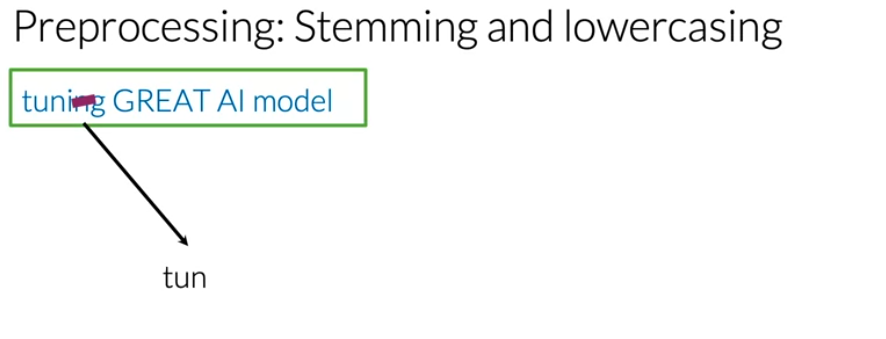


After pre-processing

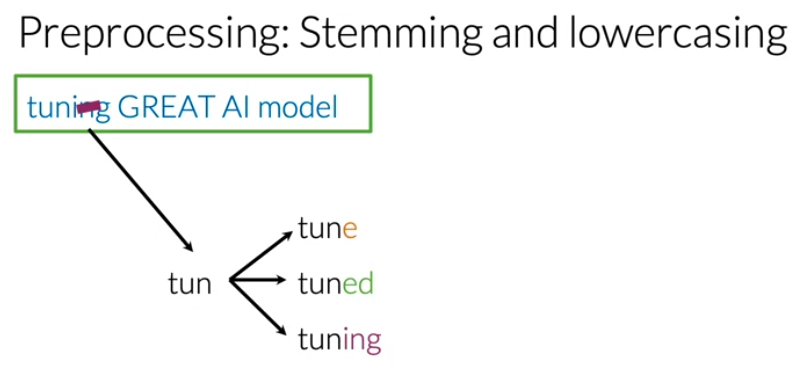


You have a necessary information,

Stemming in NlP, transforming any word to its base stem which you could define as the set of character that are used to construct the word and its derivatives.



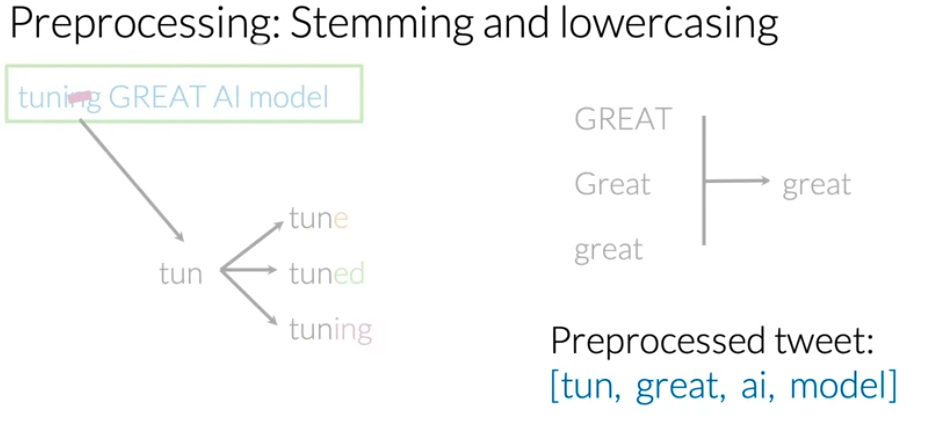
1. Add the word e
2. Adding the suffex ed
3. Adding the suffex-ing



After you perform stemming on your corpus the suffex will be reduce to the stem tun.

* To reduce your vocabulary even further without losing valuable information would have to lowercase

Final pre-processing



Preprocess raw text for Sentiment analysis

Data preprocessing is one of the critical steps in any machine learning project. It includes cleaning and formatting the data before feeding into a machine learning algorithm. For NLP, the preprocessing steps are comprised of the following tasks:

1. Tokenizing the string
2. Lowercasing
3. Removing stop words and punctuation
4. Stemming

import re # library for regular expression operations

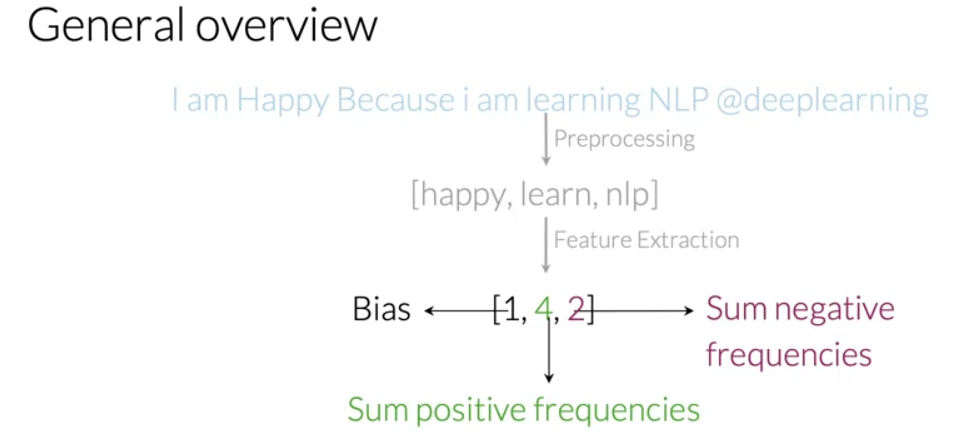
import string # for string operations

from nltk.corpus import stopwords # module for stop words that come with NLTK

from nltk.stem import PorterStemmer # module for stemming

from nltk.tokenize import TweetTokenizer # module for tokenizing strings

Stemming is the process of converting a word to its most general form, or stem. This helps in reducing the size of our vocabulary.

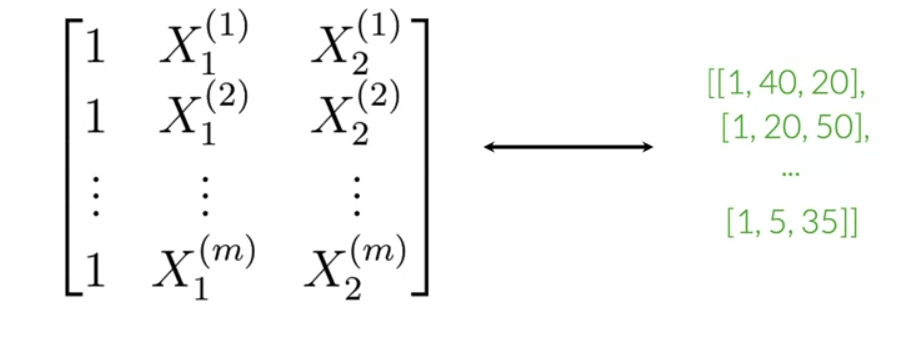


Extract feature using a frequency dictionary mapping.

And the end u have matrix

M => row

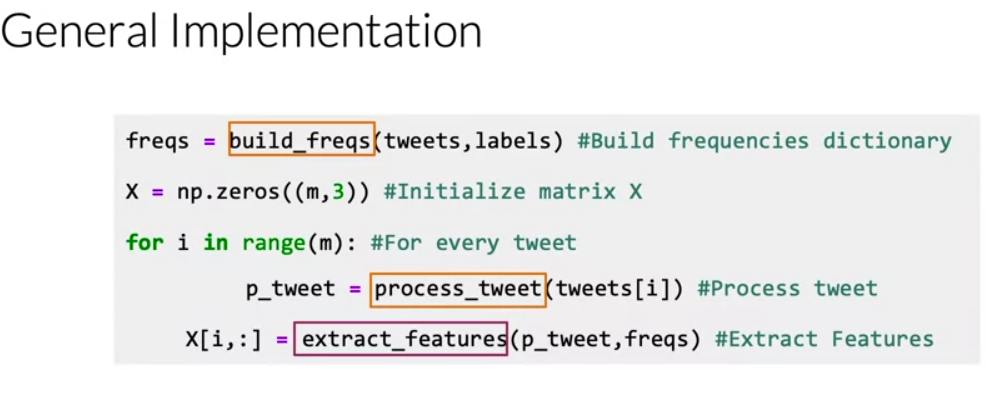
And 3 => column



Helper function :

1. build\_freqs
2. process\_tweet

you have to implement function: extract\_feature



After this feed logistic classifier

Word frequency dictionary

"""

# Convert np array to list since zip needs an iterable.

# The squeeze is necessary or the list ends up with one element.

# Also note that this is just a NOP if ys is already a list.

yslist = np.squeeze(ys).tolist()

"""

freqs = {}

for y, tweet in zip(yslist, tweets):

for word in process\_tweet(tweet):

pair = (word, y)

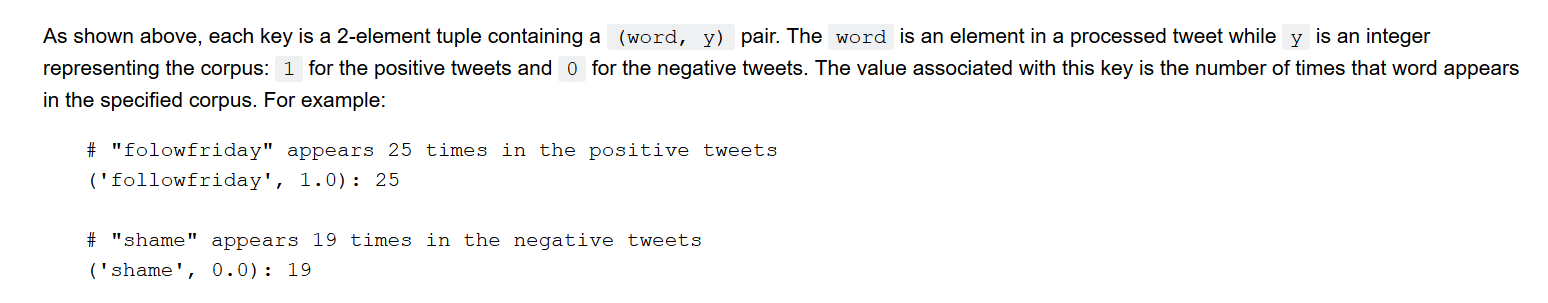
if pair in freqs:

freqs[pair] += 1

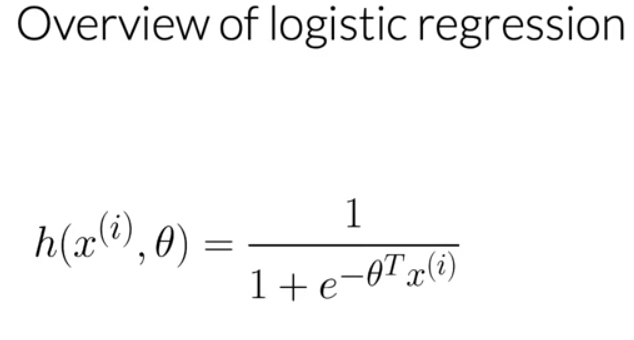
else:

freqs[pair] = 1

return freqs



Logistic regression :

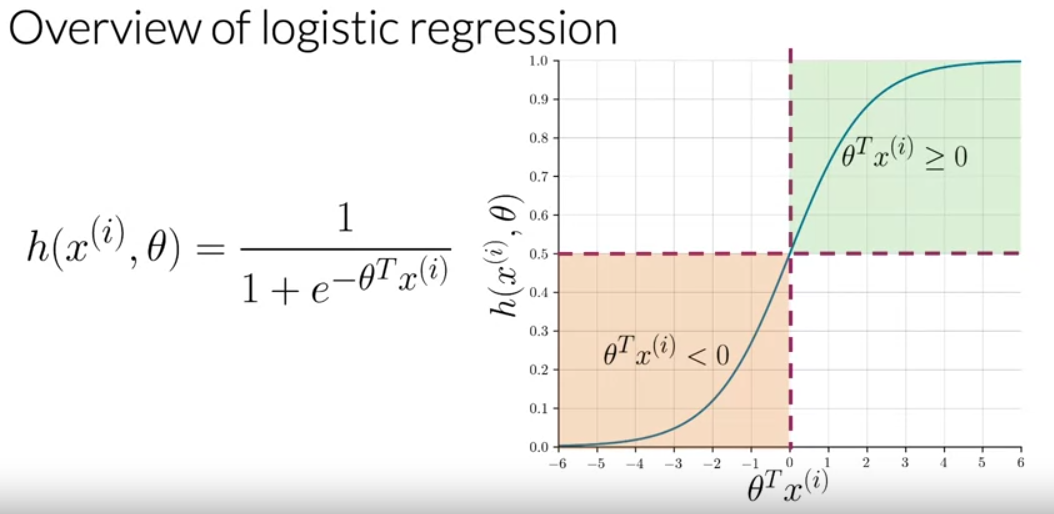


I denote the observation : data points

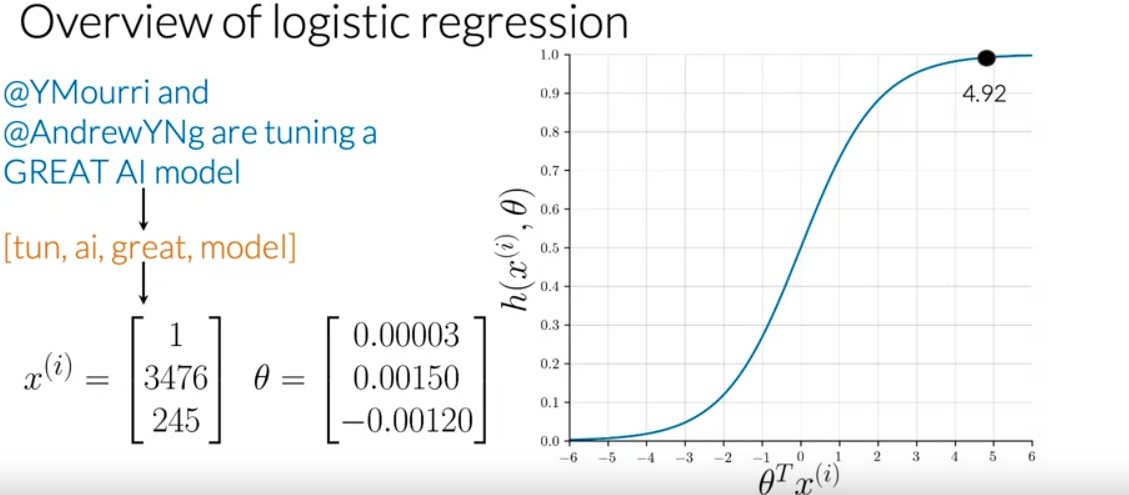
For classification threshold is needed:

Usually it is set to be 0.5 and this value corresponds to a dot product between theta transpose and x equal to zero.

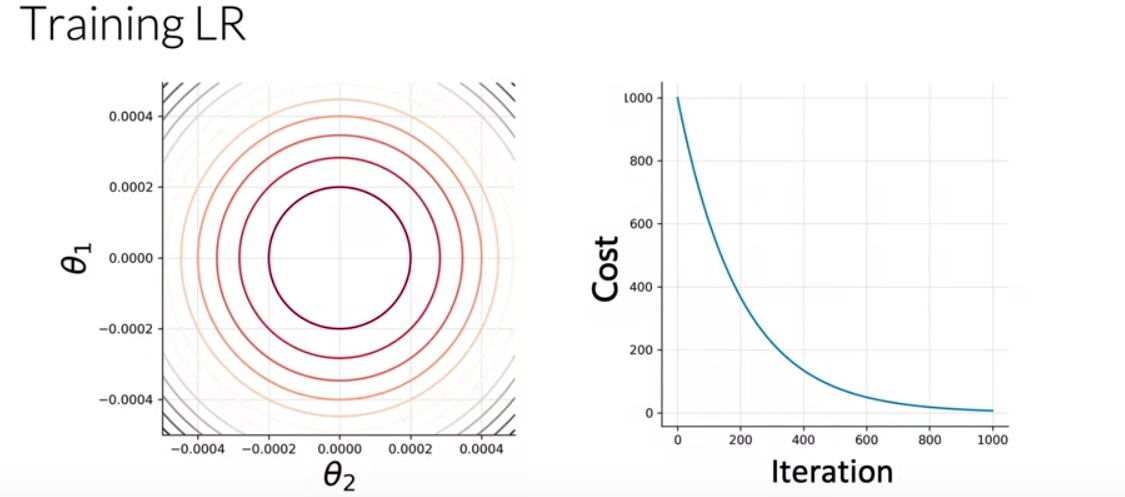
1. Dot product is greater or greater than zero, the prediction is positive



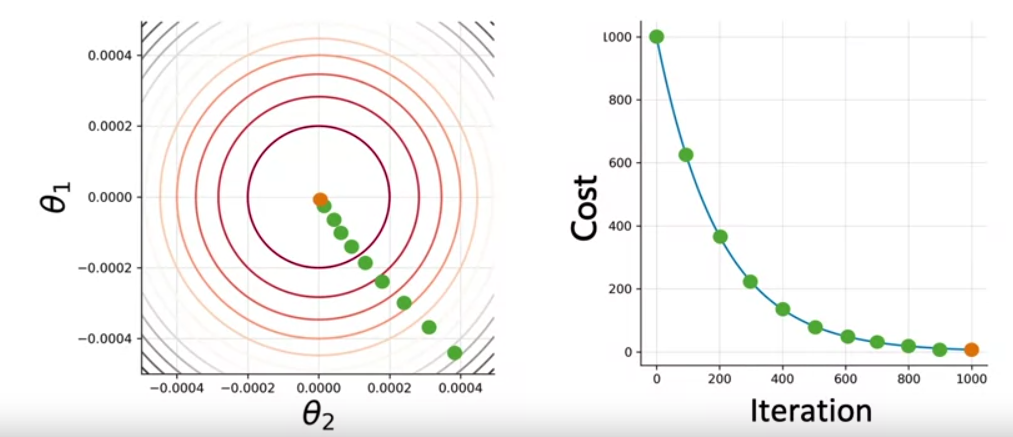
Example overall



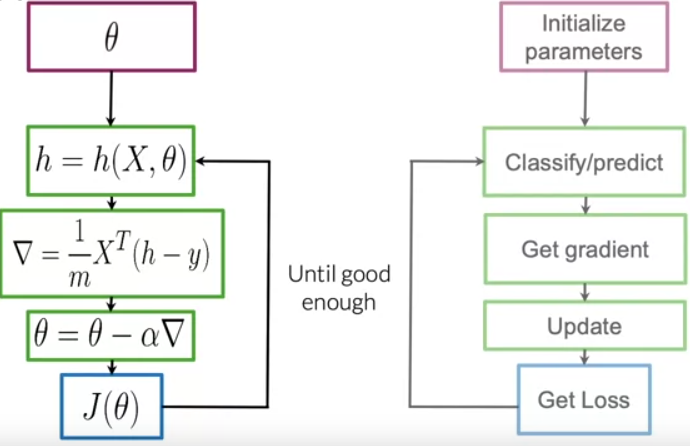
TRaing LR



After iteration and process many iteration



Gradient descent

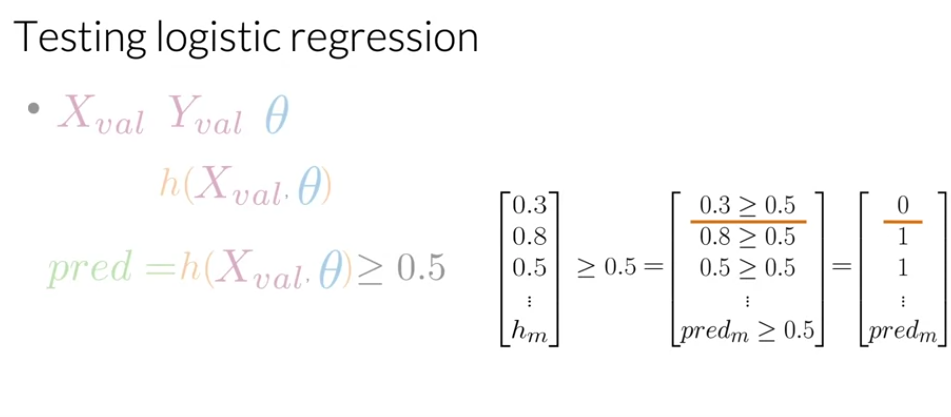


# Equation for the direction of the sentiments change

# We don't care about the magnitude of the change. We are only interested

# in the direction. So this direction is just a perpendicular function to the

# separation plane

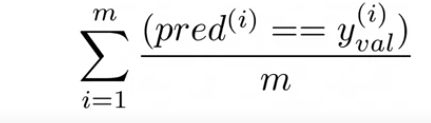


First example => No

Second example =>Yes

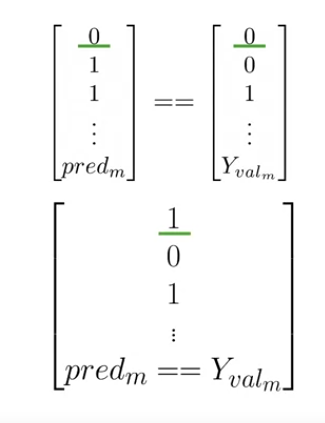
Third example =>Yes

After building the prediction vector you can compute the accuracy of your model over validation sets.



The two vector

Real and prediction



After you have compared the values of every prediction with the true labels of your validation test.

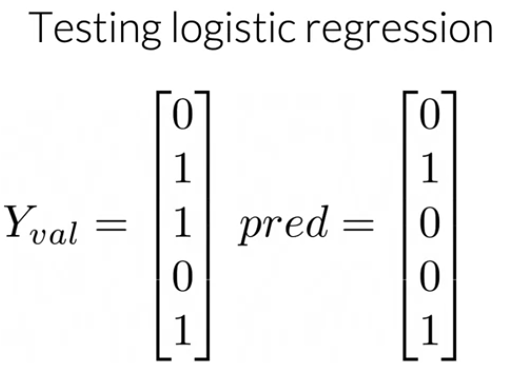
You can get the total times that your predictions were correct by summing up the vector of the comparisons.

Finally, you will divide that number m of total observations your validation sets.

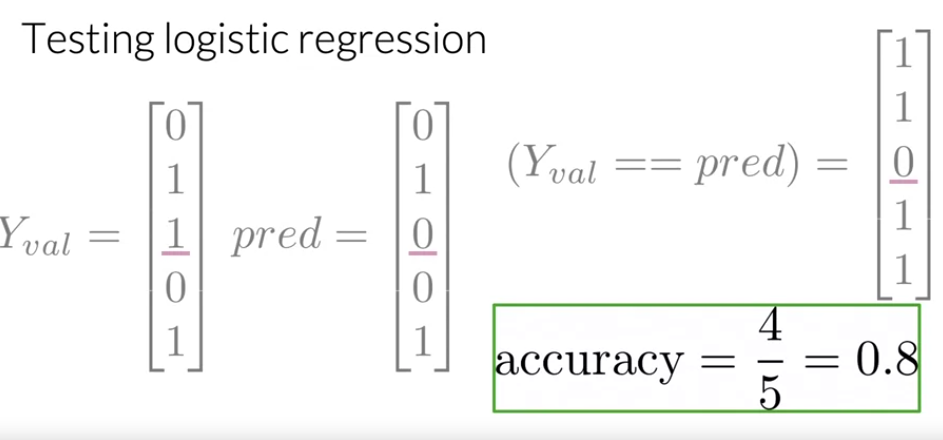
This metric gives an estimate of the times that your logistic regression will correctly work on unseen data.

If you accuracy is equal to 0.5, it means that 50 percent of the time, your model is expected to work well.

For instance if your Y\_val and prediction vectors for five observations look like this, you will compare each of their values and determine whether they match or not.



Get accuracy



You have to sum the number of times that your predictions were right and divide that number by the total number of observations in your validation sets.

Cost Function

