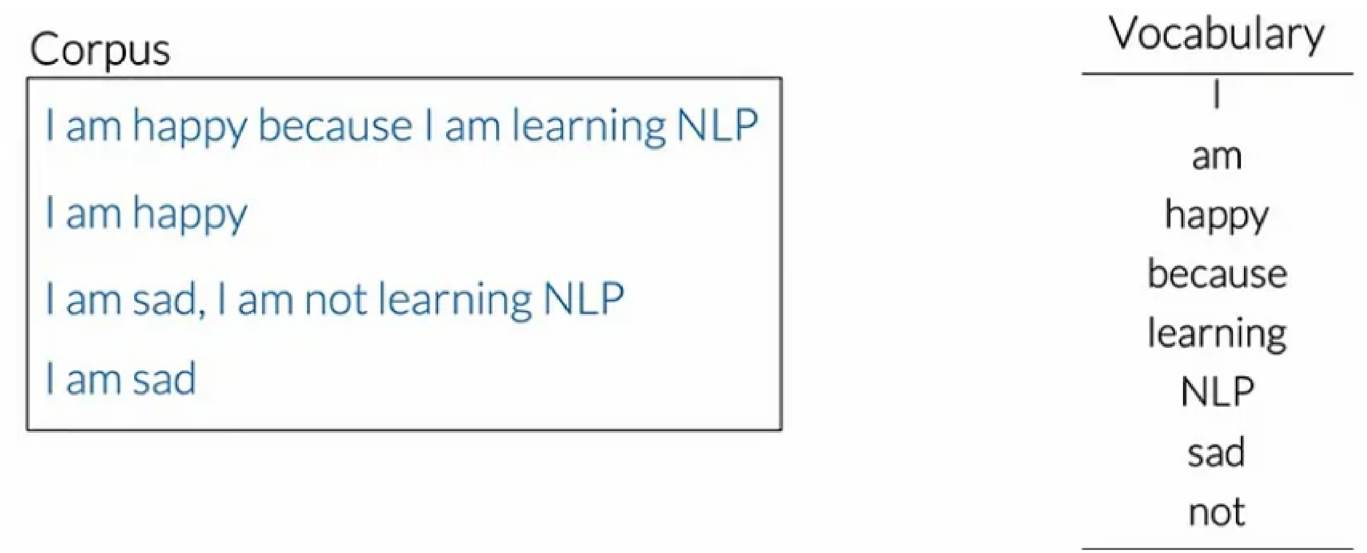
Day 1



Corpus

I am happy because I am learning NLP

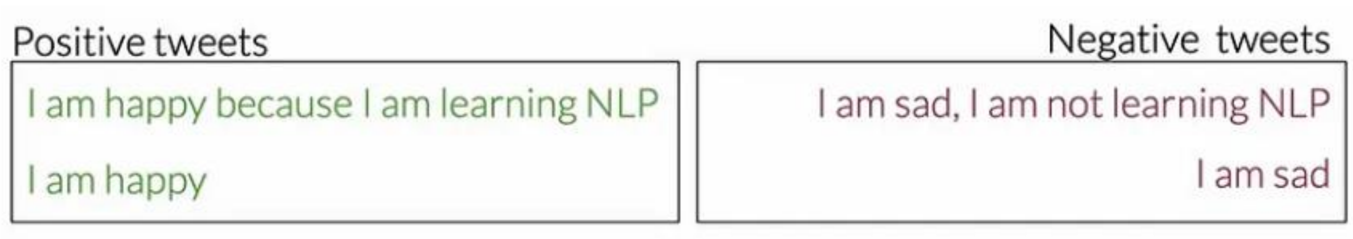
I am happy

I am sad, I am not learning NLP

I am sad

Vocabulary

I
am
happy
because
learning
NLP
sad
not



Positive tweets

I am happy because I am learning NLP

I am happy

Negative tweets

I am sad, I am not learning NLP

I am sad

Positive tweets

I am happy because I am learning NLP

I am happy

Vocabulary	PosFreq (1)
I	3
am	3
happy	2
because	1
learning	1
NLP	1
sad	0
not	0

Vocabulary	NegFreq (0)
I	3
am	3
happy	0
because	0
learning	1
NLP	1
sad	2
not	1

Negative tweets

I am sad, I am not learning NLP

I am sad

How does Bag of Words work ?

Suppose we have three sentences :

Sentence 1 : the cat sat
Sentence 2 : the cat sat in the hat
Sentence 3 : the cat with the hat

After removing stop words we are left with :

Sentence 1 : cat sat cat
Sentence 2 : cat sat hat
Sentence 3 : cat hat

When we convert frequency table to vectors it is know as **BOW**

The f1,f2,f3 features will be independent features for the model building.

Words	Frequency
cat	4
sat	2
hat	2

	f1	f2	f3
	cat	sat	hat
Sent 1	2	1	0
Sent 2	1	1	1
Sent 3	1	0	1

Disadvantage of Bag of Words

In this technique, the words are given equal importance and does not have any semantic difference .
Like in our case , in **Sentence 1** cat and sat are represented by 1 .

We can not classify which word is important than the other in this technique so to tackle this problem we have another feature engineering technique which is
TF-IDF

TF - IDF

TF-IDF stands for
Term Frequency – Inverse Document Frequency .

This is a technique to quantify a word in documents, we generally compute a weight to each word which signifies the importance of the word in the document .

This technique coverts text into numerical values , giving more importance to some words by multiplying TF and IDF i.e final result is got by TF*IDF

How does TF- IDF work ?

Lets First understand TF i.e. Term Frequency .

TF = No. of repetition of words in the sentence / No. of words in the sentence

So our , TF table will look like this for our example :

Sentence 1 : cat sat
Sentence 2 : cat sat hat
Sentence 3 : cat hat

	f1	f2	f3
	cat	sat	hat
Sent 1	1/2	1/2	0
Sent 2	1/3	1/3	1/3
Sent 3	1/2	0	1/2

How does TF- IDF work ?

Now understand IDF i.e. Inverse Document Frequency .

IDF = log(No. of reviews/ No. of reviews containing words)

So our , IDF table will look like this for our example :

Sentence 1 : cat sat
Sentence 2 : cat sat hat
Sentence 3 : cat hat

Here , no. of documents means no.of sentences.

Words	IDF
cat	Log(3/3) = 0
sat	Log(3/2)
hat	Log(3/2)

How does TF- IDF work ?

Finally ,

TF * IDF table in our case is :

Here we can see that the word sat is given the most importance in Sentence 1 out of other words .

Similarly in Sentence 2 and 3 as well .

	f1	f2	f3
	cat	sat	hat
Sent 1	½*0 = 0	½*log(3/2)	0*log(3/2)=0
Sent 2	1/3*0 = 0	1/3*log(3/2)	1/3*log(3/2)
Sent 3	1/2*0 = 0	0*log(3/2)=0	1/2*log(3/2)