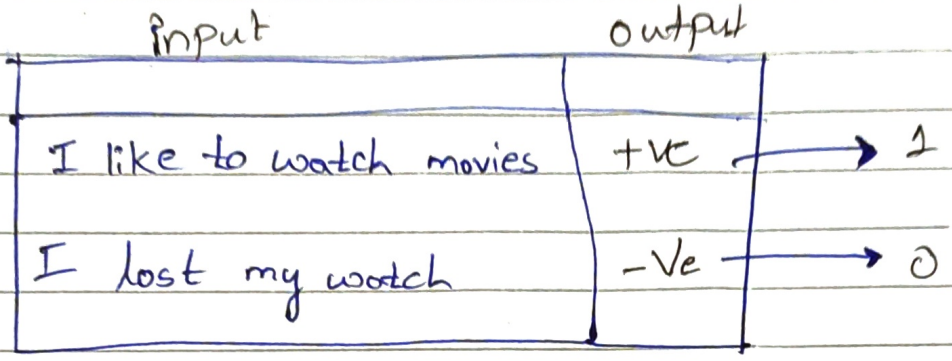
Natural Language Processing (NLP)



→ Machine can't understand the data, till now we deal with the models which are having numerical no text & no images.

→ So, we need to convert this need to numerical data.

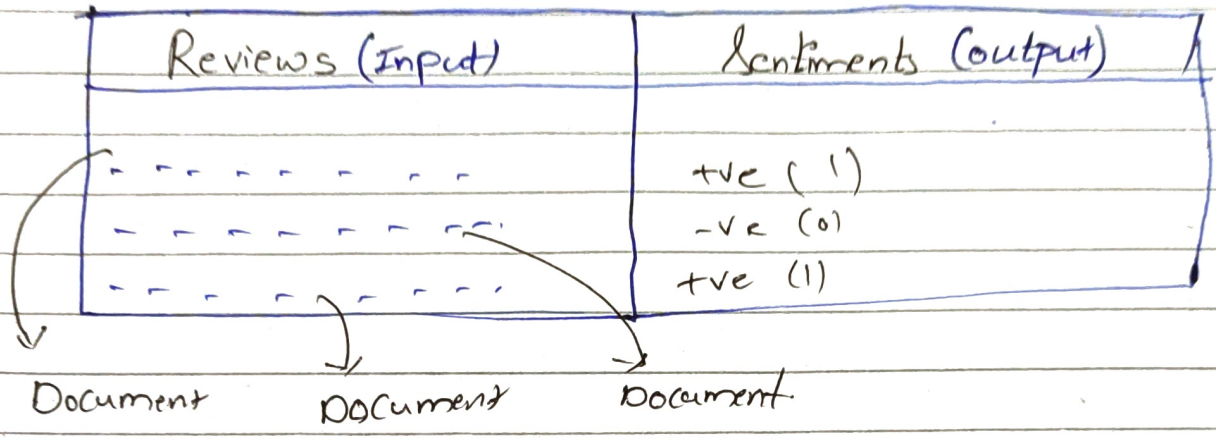
then we can predict the output (0 (or) 1) → (Binary classification)

→ Converting Text to numerical algorithms are

1. BoW → Bag of Words.
2. TF-IDF
3. Word 2 Vectorizer
4. BERT

Spacy & NLTK are the Libraries for Text data
SKlearn library for ML Models.

My dataset is having Reviews & Sentiments



→ Each Row is a "document"

→ Collection of Documents is also known as Corpus.

1st: Collect Data:-

Ex:

Input

it was the best of times it
It was the worst of times
It was the age of wisdom
It was the age of Foolishness.

each row is a document.

→ (1)

2nd step:→ Learn Vocabulary

→ First we need to understand the Vocabulary

↳ It's nothing but learning the unique words in the Corpus.

it	It	was	the	best	of	Times	worst	age	Wisdom

→ Vocabulary.

→ Now update the Input data with "Count" with the Vocabulary.

(Lowercase, Uppercase (different))

it	It	Was	the	best	of	times	worst	age	wisdom	Foolishness
2										
1	0	1	1	1	1	1	0	0	0	0
0	1	1	1	0	1	1	1	0	0	0
0	1	1	1	0	1	0	0	1	1	0
0	1	1	1	0	1	0	0	1	0	1

→ Document Vector

↳ Now, Here we are showing the "Count" of the each word.

↳ And we Converted it into Numerical form.

↳ Each row is called Document Vector

(Converted Document to Document Vector)

↓ Text ↓ Numerical

↳ Collection of Document Vector is called Document Term Matrix.

→ In the above example, we have 1st two Columns i.e It, it

↳ Both are same, Unneccessarily increased the Dimensionality (more no. of Columns)

(It, it)

↳ We Can Convert it into a Standard Case then (upper (or) Lower)

then the no. of Columns are Reduced.

Stop Words:

→ If we look into the Dataset (each Document)

↳ we have unnecessary words which are used for Sentence Formation. This words are called Stop words.

↓

This words won't make any impact on the the Dataset (~~corpus~~ Document term Matrix)

↳ best, times, Worst, age, freedom, Wisdom, Foolishness

↓

all these are not Stopwords.

↓

This words will define my input

If, way, on, of, at, ...

↳ all these are Stopwords

↳ It won't have any impact.

→ We can Remove this words. otherwise if we keep this Stopwords then My dimensionality will increase.

→ ~~And~~ L

Stemming:→

↳ Converting the Word into Root word

→ 1. Iam the happiest person right now

2. Iam happy person right now

happy & Happiest.

↳ Happ (trimmed)

Both are of same words.

↳ If we trim the Words then it will be good

↳ In a single word (happ)

So, my dimensionality will reduce.

Lemmitization & Stemming → both are same

↳ But the Words are Converted into Meaningful form in Lemitization

↳ In above Example Lemitization → is → Happ.

Remove special characters.

↳ Sometimes in the corpus we have •, @, #, ···, ;, ?, !

↳ all these are special words, it won't make any impact to the Corpus.

↳ This special characters are for our purpose like tagging some particular location, we use @hyderabad

↳ we remove @ & mentioned hyderabad is also same.

→ So, we can remove special characters.

step-1 :- 1) Remove special characters

2) Convert to lower case

3) Remove the stop words.

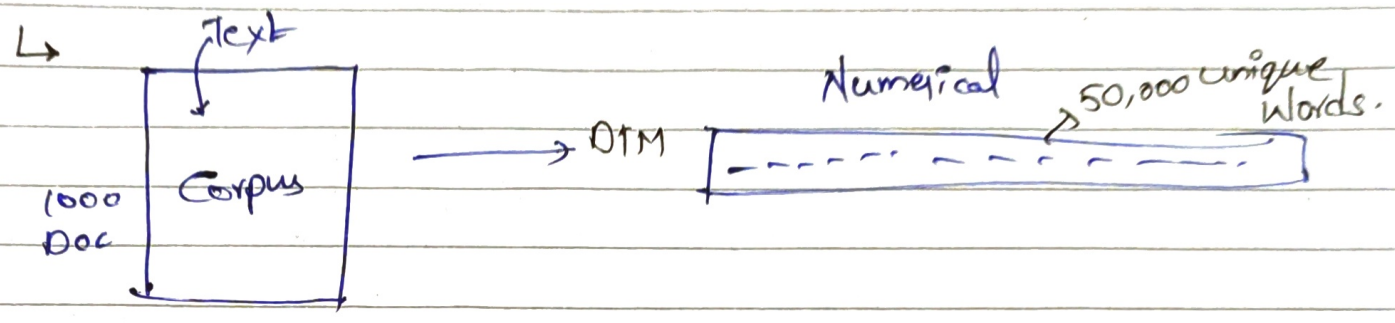
4) Stemming / Lemitization

} Preprocessing steps for Text preprocessing (Cleaning the Data)

step-2:- Learn the Vocabulary

step-3:→ Convert Corpus to Document Term Matrix

↳ Suppose if my Corpus is having 1000 Doc. and it has 50,000 unique Words.



↳ 50,000 Columns is Very Very Very high & we cant Run ML Model on this

↳ our accuracy will gonna Reduce a lot.

↳ That's why we need to do Text preprocessing.

(Note:- But the issue is this Bow is not preserving the Order)

Ex:- 1) I do have a house?
2) Do I have a house?

I	do	have	a	house
1	1	1	1	1
1	1	1	1	1

Here both the statements are different, but my Document Vector is same.

→ So, we need preseve the sequential order

→ In this case we uses 1-gram (1word)