Natural Language Processing (NLP)

Natural Language? : Human language. English, hindi, French etc.

NLP :- Processing the natural language so that we can learn useful features.

NLU : Natural Language Understanding ; syntax, grammar, construction

NLG : Natural Language Generation. ; Given an incomplete text we can complete it.

Usecases : Lots of usecases

1) Sentiment analysis : Given a review / text / article what is the sentiment? (+ve / -ve / neutral) } clarification

2) Classification of documents } clarification

3) Document cluster } clustering Problem.

Revisiting the traditional ML:

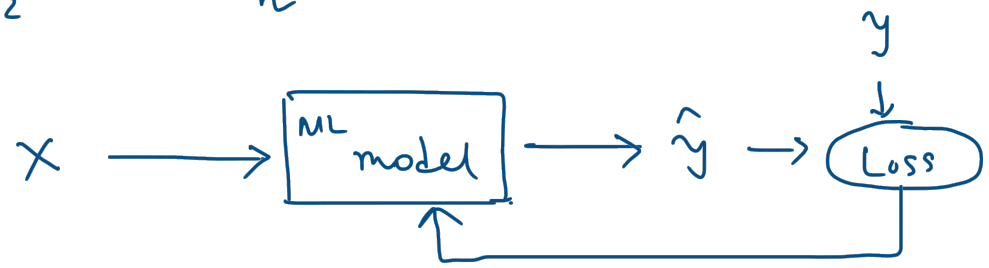
features are numerical / categorical (OHE / IE)

features

x_1 x_2 .... x_n

targets

y



(Supervised ML model)

Sentiment classification:- This can also be framed as supervised ML task.

→ I liked the product but it needs improvement. (neutral)

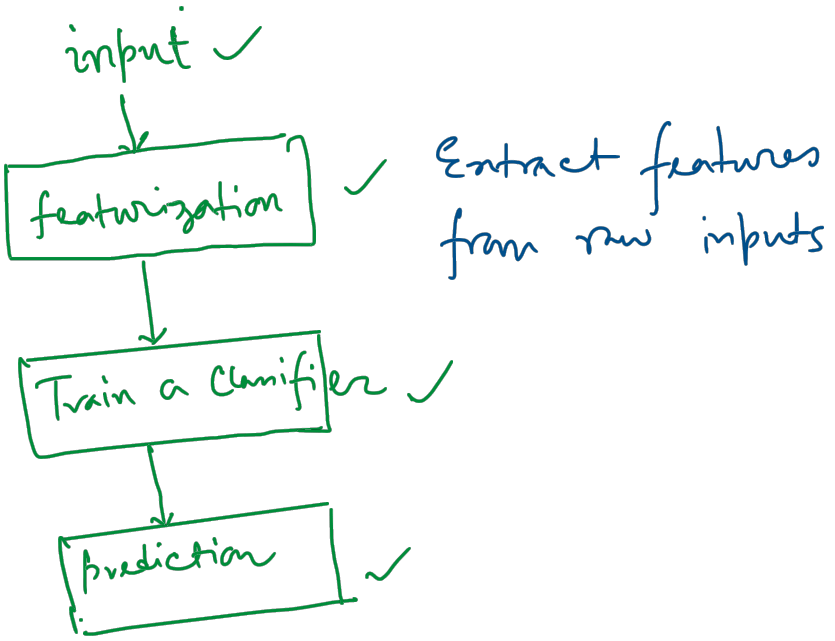
→ I am absolutely delighted by the product (+ve)

→ The product is not working after using for 2 days. (-ve)

1 - 2 * (-ve), 3 * (neutral), 4+ * (+ve)

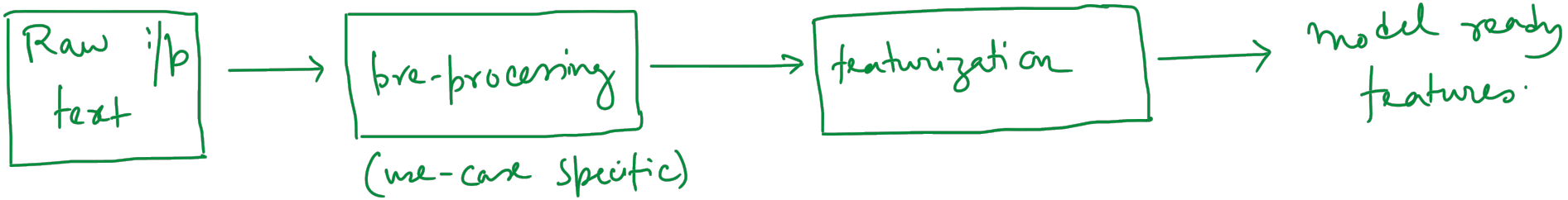
Training

input	output (sentiment)
1. Review - 1	+ve
2. Review - 2	-ve
3. Review - 3	-ve
⋮	⋮
m. Review - m	+ve



Given an unseen review can you predict the sentiment ?

The first step of any NLP problem is to do the featurization



Text Preprocessing:

@skarnakar is teaching at @LogicMojo. Please sign up: https://..../...

Preprocessing Step-1: Removing punctuations and stopwords.

punctuations	Stopwords
;	the
:	and
,	or
:/ :) :(:O	are
	is .....

} English stopwords.

Domain Specific Stopwords: Suppose you are creating a sentiment classifier for the review of air-fryers.

Total set of stopwords: Language specific $\cup$ Domain Specific.

Example: I ~~am~~ sad, because ~~the~~ movie ~~was~~ bad. [raw]

→ (sad because movie bad)

~~The~~ product ~~is~~ great. ~~I~~ ~~am~~ very happy ~~to~~ use ~~the~~ product. [raw]

→ product great very happy use product

Pre-processing step-2: Remove URLs, Handles.

Pre-processing step-3: Lowercasing

Pre-processing step-4: Stemming / Lemmatization

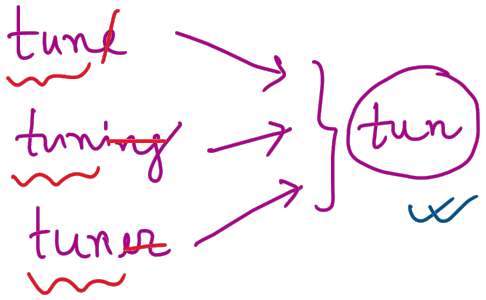
Both Stemming and Lemmatization are text normalization techniques.

Stemming: → It is a crude heuristic process that chops off word endings (prefixes / suffixes) to get the root form.

→ It doesn't care about grammer / meaning.

→ It often produces non-dictionary words (stems)

→ Porter Stemmer algorithm is the most famous stemming algo for english.



Raw	Stemmed
Caring	car
Cars	car
Studies	Studi
better	better

Lemmatization :

→ Uses vocabulary and morphological analysis of words to return the "lemma" : the base/root word. (dictionary word)

→ Takes into account the parts of speech (POS) and the context.

Raw	Lemmatized word
caring	care
studies	study
better	good

Corpus: Set of texts, intended to use for specific NLP task.

Ex:- Set of wikipedia articles.

Set of reviews of a product in Amazon.

financial times articles published between 2020 – 2025.

Corpus are used to train NLP models

One of the most common technique to collect corpus is to do web-scrapping (crawling different website and dump textual data)

Most of the sophisticated NLP models are trained on publicly available corpus (wikipedia article).

Tokenization

Corpus: { document-1. document-2. ⋮ document-n }

document: { sentence-1. Sentence-2. ⋮ }

Sentence: I liked the movie @Oppenheimer.

Tokens: Building blocks of a sentence.

In the strict sense tokens are not simple words but tokens can consist a part of word.

(liked) → lik ed

Vocabulary :

Corpus → Apply preprocessing techniques. → processed text → tokenization (breaking into words / tokens)

Create a set of unique words in the pre-processed corpus ←

Set of unique words : Vocabulary (V)

Eg:-

1) I love to read horror stories.
2) I hate my engineering stream.
3) I love my country

[love, read, horror, story]
[hate, engineer, stream]
[love, country]

V = [love, read, horror, story, hate, engineer, stream, country]