* Text Representation *

1. What is Feature extraction from text?
→ Converting Text into numbers
(Text Vectorization) (Text Representation)

2. Why do we need it?
→ Good the features algorithm will work great and good.

→ It is difficult to create features from Text (Vectors)

3. What is core idea?
→ Numbers (vector) will Convey Semantic meaning of Text.
e.g OHE, BOW, ngrams, TF-IDF, embeddings, Custom features. → Common Techniques for Text Representat

* Common Terms.

1. Corpus → All words in Datasets [Combination of all words]
2. Vocabulary → Unique words in Corpus
3. Document → Individual Sentence in Corpus or Dataset
4. Word → Words in a Sentence.

* One Hot Encoding

D_1 People watch Campux
D_2 Campux watch Campusx
D_3 people write Comment
D_5 Campusx write Comment.

Corpus
people watch Campusx Campux
watch Campusx people write
Comment Campusx write Comment

Vocabulary
people watch Campux write Comment

V=5

Word → V dimension

People	Watch	Campusx	Write	Comment
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0

$D_1 = [[1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0]]$

Pros → Very Intuitive | easy to implement.

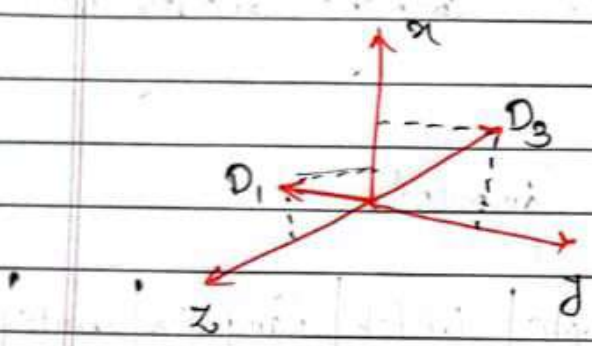
Cons / flaws:
1. Sparsity
2. Size of Document ~~Not~~ fixed Size input.
3. Out of Vocabulary words
4. No Capturing of Semantic meaning.

* Bag of Words (BOW):

→ Order of Words doesn't matter.
→ Contextual meaning doesn't matter.

→ Can handle Out of Vocabulary words.

		people	watch	Campusx	Write	Comment
D_1	People watch Campusx	1	1	1	0	0
D_2	Campusx watch Campusx	0	1	2	0	0
D_3	People write Comment	1	0	0	1	1
D_4	Campusx write Comment	0	0	1	1	1



binary = True
↳ Used for Sentiment Analysis
[Use To denote presence of word]

Pro's → 1) Simple and Intuitive
2) Can handle different input size
3) Resolve Out of Vocabulary words [By Ignoring]
4) Can capture Semantic meaning in certain quantity better than ONE.

Con's → 1) Sparsity on big data [problem of Overfitting]
2) Cannot Consider Out of Vocabulary words.
3) Cannot Consider Ordering of words.
4) Can Consider Sentence with so opposite meaning as neighbors and consider as a same sentence.

e.g A very good name
A not very good name.

* N-grams: [Bag of N-grams].

bi-gram → Two Words Tri-gram → Three words.

Bag of Bi-gram

D_1 People Watch Campusx
D_2 Campusx watch Campusx
D_3 People Write Comment
D_4 Campusx write Comment

	People watch	Watch Campusx	Campusx Watch	People Write	Write Coompent
D_1	1	1	0	0	0
D_2	0	1	1	0	0
D_3	0	0	0	1	1
D_4	0	0	0	0	1

Bag of Tri-gram

	S_1	S_2	S_3	S_4
D_1	1	0	0	0
D_2	0	1	0	0
D_3	0	0	1	0
D_4	0	0	0	1

Benefits: → i) Able to Capture Semantic of the Sentence
ii) Easy to implement.

Disadvantage: → i) Dimensionality of Vocabulary increases
ii) Computational power and time increases.
iii) Time Complexity increases as slow downs algo.
iv) No Solution for Out of Vocabulary Words.

DATE / /

* TF - IDF :

→ Assigns a value to words
→ Word which occurs most in document and less occurance in Corpus gives most importance or max weight (by assigning value)

TF → Term Frequency

$TF(t,d) = \frac{\text{Number of occurrences of term t in document d}}{\text{Total number of terms in document d}}$

D_1 = People watches Campus X.

$TF(people, D1) = \frac{1}{3}$

TF → less → rare occurance
TF → more → most occurance in document.

$0 < TF < 1$ → Probablity.

IDF → Inverse Document Frequency

$IDF(t) = \log_e \frac{\text{(Total no. of Document in the Corpus)}}{\text{(Number of document with term t in them)}}$

	IDF
people	log (4/2)
Watch	log (4/2)
Campus x	log (4/3)
Write	log (4/2)
Comment	log (4/2)

* Word with very high frequency have low IDF.
* Word with low frequency have high IDF.

D_1 = People watch Campus X

people	TF	IDF	TF-IDF
people	1/3	log (4/2)	0.3 x 0.125
watch	1/3	log (4/2)	0.125 x 0.3
CampusX	1/3	log (4/3)	0.125 x 0.4

* Advantage: i) Use in information Retrieval

* Disadvantage: 1) Sparsity
2) Higher Dimension Arrays
3) Out of Vocabulary words.
4) No Semantic meaning Capturing.

* Custom Features:

i) No. of positive words
ii) No of negative words.
iii) Word Count.
iv) Character Count
v) Ratio of positive and negative words.