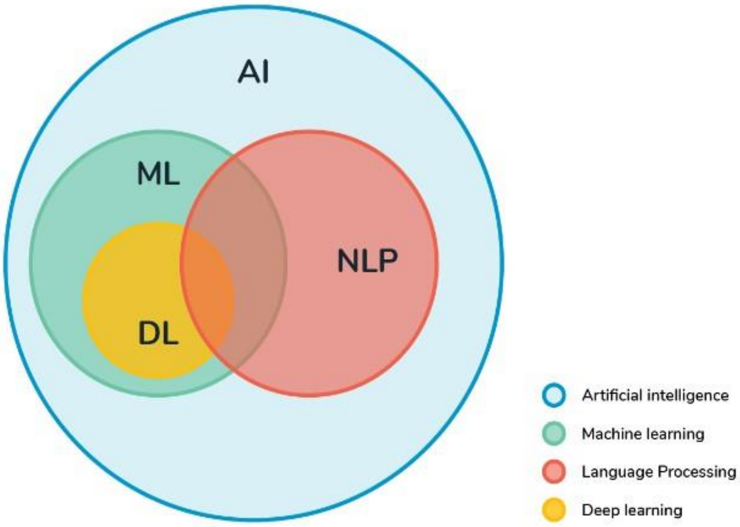
Day 6: DATA SCIENCE INTERVIEW PREPRATION QUESTIONS

Q1. What is NLP?

Natural language processing (NLP): It is the branch of artificial intelligence that helps computers understand, interpret and manipulate human language.

NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding.



Q2. What are the Libraries we used for NLP?

We usually use these libraries in NLP, which are:

NLTK (Natural language Tool kit), TextBlob, CoreNLP, Polyglot,

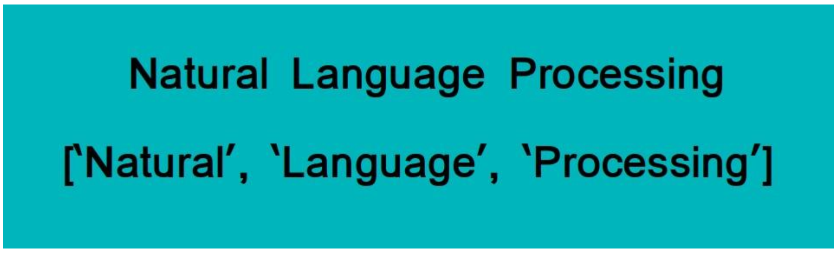
Gensim, SpaCy, Scikit-learn

And the new one is Megatron library launched recently.

Q3. What do you understand by tokenisation?

Tokenisation is the act of breaking a sequence of strings into pieces such as words, keywords, phrases, symbols, and other elements called tokens.

Tokens can be individual words, phrases, or even whole sentences. In the process of tokenisation, some characters like punctuation marks are discarded.

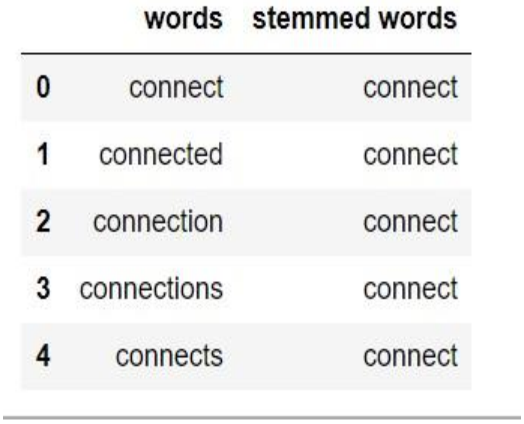


Natural Language Processing

['Natural', 'Language', 'Processing']

Q4. What do you understand by stemming?

Stemming: It is the process of reducing inflexions in words to their root forms such as mapping a group of words to the same stem even if stem itself is not a valid word in the Language.

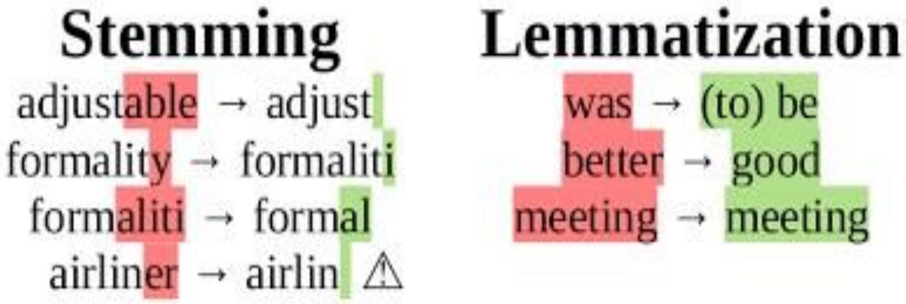


	words	stemmed words
0	connect	connect
1	connected	connect
2	connection	connect
3	connections	connect
4	connects	connect

Q5. What is lemmatisation?

Lemmatisation: It is the process of the group together the different inflected forms of the word so that they can be analysed as a single item.

It is quite similar to stemming, but it brings context to the words. So it links words with similar kind meaning to one word.



Q6. What is Bag-of-words model?

We need the way to represent text data for the machine learning algorithms, and the bag-of-words model helps us to achieve the task. This model is very understandable and to implement.

It is the way of extracting features from the text for the use in machine learning algorithms.

In this approach, we use the tokenised words for each of observation and find out the frequency of each token.

Let's do an example to understand this concept in depth.

"It is going to rain today."
"Today, I am not going outside."
"I am going to watch the season premiere."

We treat each sentence as the separate document, and we make the list of all words from all the three documents excluding the punctuation. We get,

'It', 'is', 'going', 'to', 'rain', 'today' 'I', 'am', 'not', 'outside', 'watch', 'the', 'season', 'premiere.'

The next step is the create vectors. Vectors convert text that can be used by the machine learning algorithm.

We take the first document — **"It is going to rain today",** and we check the frequency of words from the ten unique words.

"It" = 1
"is" = 1
"going" = 1
"to" = 1
"rain" = 1
"today" = 1
"I" = 0
"am" = 0
"not" = 0
"outside" = 0

Rest of the documents will be:
"It is going to rain today" = ***[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]***
"Today I am not going outside" = ***[0, 0, 1, 0, 0, 1, 1, 1, 1, 1]***
"I am going to watch the season premiere" = ***[0, 0, 1, 1, 0, 0, 1, 1, 0, 0]***

*In this approach, each word (a token) is called a **“gram”.** Creating the vocabulary of two-word pairs is called a bigram model.*

The process of converting the NLP text into numbers is called **vectorisation** in ML. There are different ways to convert text into the vectors:

- ***Counting the number of times that each word appears in the document.***
- ***I am calculating the frequency that each word appears in a document out of all the words in the document.***

Q7: What do you understand by TF-IDF?

TF-IDF: It stands for the term of frequency-inverse document frequency.

TF-IDF weight: It is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Term Frequency (TF): is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. The term frequency is often divided by the document length to normalise.

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

Inverse Document Frequency (IDF): It is a scoring of how rare the word is across the documents. It is a measure of how rare a term is, Rarer the term, and more is the IDF score.

$$IDF(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it})$$

Thus, $TF - IDFscore = TF * IDF$

Q8. What is Word2vec?

Word2Vec is a shallow, two-layer neural network which is trained to reconstruct linguistic contexts of words.

It takes as its input a large corpus of words and produces a vector space, typically of several of hundred dimensions, with each of unique word in the corpus being assigned to the corresponding vector in space.

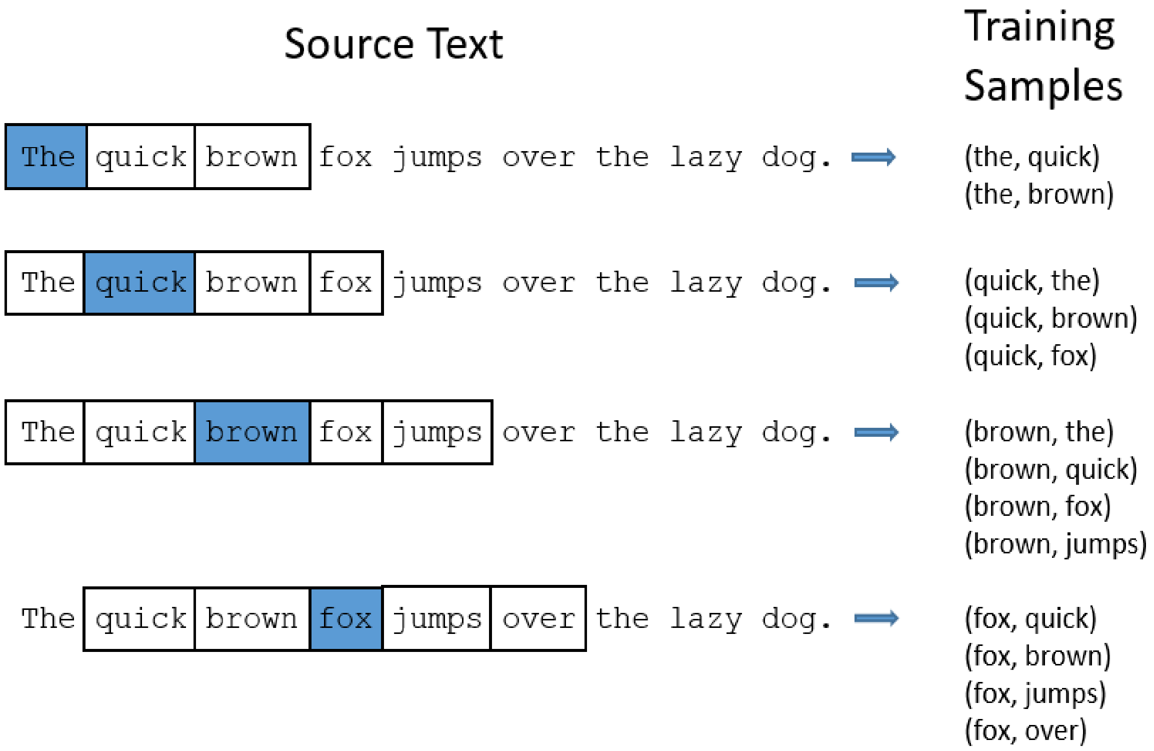
Word vectors are positioned in a vector space such that words which share common contexts in the corpus are located close to one another in the space.

Word2Vec is a particularly computationally efficient predictive model for learning word embeddings from raw text.

Word2Vec is a group of models which helps derive relations between a word and its contextual words.

Let us look at two important models inside Word2Vec: Skip-grams and CBOW.

Skip-grams



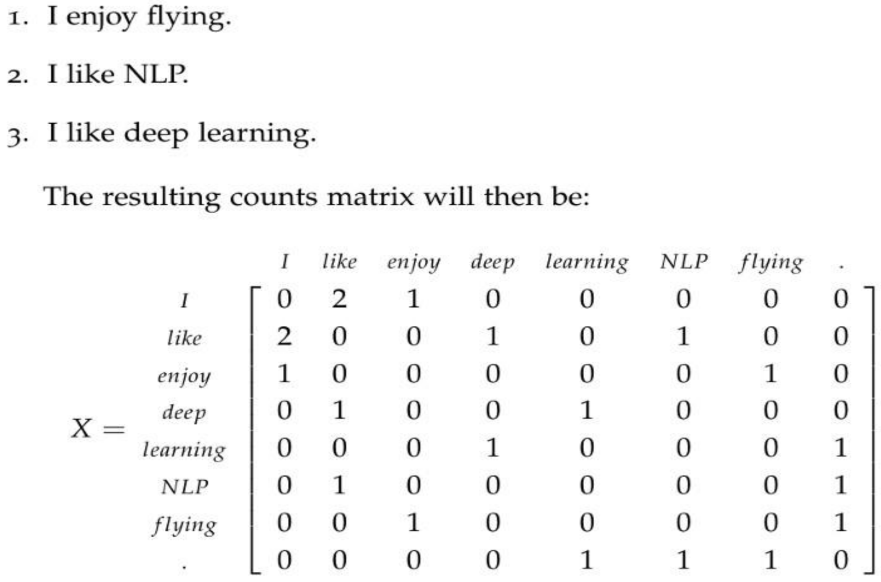
In Skip-gram model, we take a centre word and a window of context (neighbour) words, and we try to predict the context of words out to some window size for each centre word. So, our model is going to define a probability distribution, i.e. probability of a word appearing in the context given a centre word and we are going to choose our vector representations to maximise the probability.

Continuous Bag-of-Words (CBOW)

CBOW predicts target words (e.g., ‘mat’) from the surrounding context words (‘the cat sits on the’).

Statistically, it affects that CBOW smoothest over a lot of distributional information (by treating an entire context as one observation).

For the most part, this turns out to be a useful thing for smaller datasets.



1. I enjoy flying.
2. I like NLP.
3. I like deep learning.

The resulting counts matrix will then be:

$$X = \begin{array}{c|cccccccc} & I & like & enjoy & deep & learning & NLP & flying & . \\ \hline I & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ like & 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ enjoy & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ deep & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ learning & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ NLP & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ flying & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ . & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}$$

This was about converting words into vectors. But where does the “learning” happen?

Essentially, we begin with small random initialisation of word vectors. Our predictive model learns the vectors by minimising the loss function.

In Word2vec, this happens with feedforward neural networks and optimisation techniques such as Stochastic gradient descent.

There are also count-based models which make the co-occurrence count matrix of the words in our corpus; we have an exceptionally large matrix with each row for the “words” and columns for the “context”.

The number of “contexts” is, of course large, since it is very essentially combinatorial in size.

To overcome this issue, we apply SVD to a matrix. This reduces the dimensions of the matrix to retain maximum pieces of information.

Q9. What is Doc2vec?

Paragraph Vector (more popularly known as *Doc2Vec*) — Distributed Memory (*PV-DM*)

Paragraph Vector (Doc2Vec) is supposed to be an extension to Word2Vec such that *Word2Vec learns to project words into a latent d-dimensional space* whereas *Doc2Vec aims at learning how to project a document into a latent d-dimensional space.*

The basic idea behind PV-DM is inspired by Word2Vec.
In CBOW model of Word2Vec, the model learns to predict a centre word based on the contexts.
For example- given a sentence "The cat sat on the table", CBOW model would learn to predict the words "sat" given the context words — the cat, on and table.

Similarly, in PV-DM the main idea is randomly sample consecutive words from the paragraph and ***predict a centre word*** from the randomly sampled set of words by taking as the ***input — the context words and the paragraph id***.

Let us have a look at the model diagram for some more clarity.

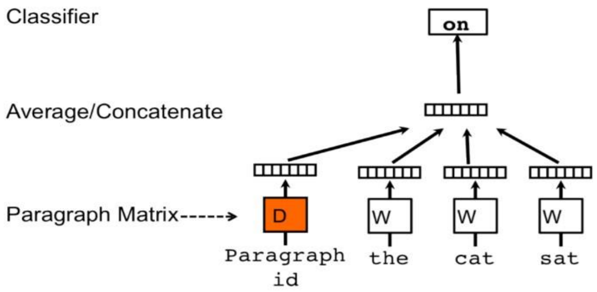
In this given model, we see Paragraph matrix, (Average/Concatenate) and classifier sections.

Paragraph matrix: It is the matrix where each column represents the vector of a paragraph.

Average/Concatenate: It means that whether the word vectors and paragraph vector are averaged or concatenated.

Classifier: In this, it takes the hidden layer vector (the one that was concatenated/averaged) as input and predicts the Centre word.

In the matrix D, It has the embeddings for "seen" paragraphs (i.e. arbitrary length documents), the same way Word2Vec models learns embeddings for words.



For unseen paragraphs, the model is again run through gradient descent (5 or so iterations) to infer a document vector.

Q9. What is Time-Series forecasting?

Time series forecasting is a technique for the prediction of events through a sequence of time.
The technique is used across many fields of study, from the geology to behaviour to economics.
The techniques predict future events by analysing the trends of the past, on the assumption that future trends will hold similar to historical trends.

Q10. What is the diff between in Time series and regression?

Time-series:

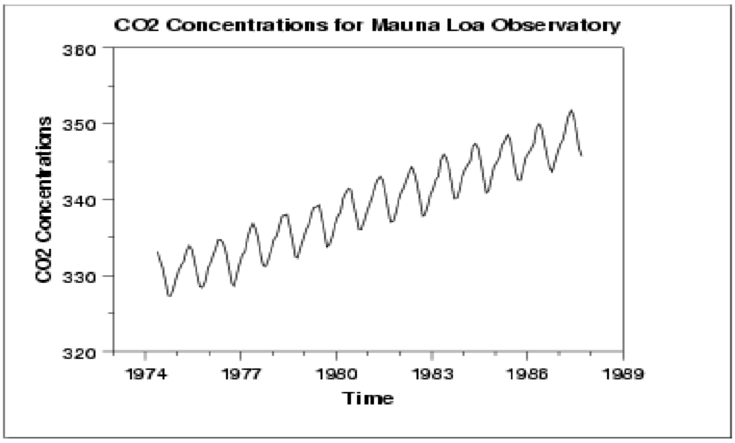
1. Whenever data is recorded at regular intervals of time.
2. Time-series forecast is Extrapolation.
3. Time-series refers to an ordered series of data.

Regression:

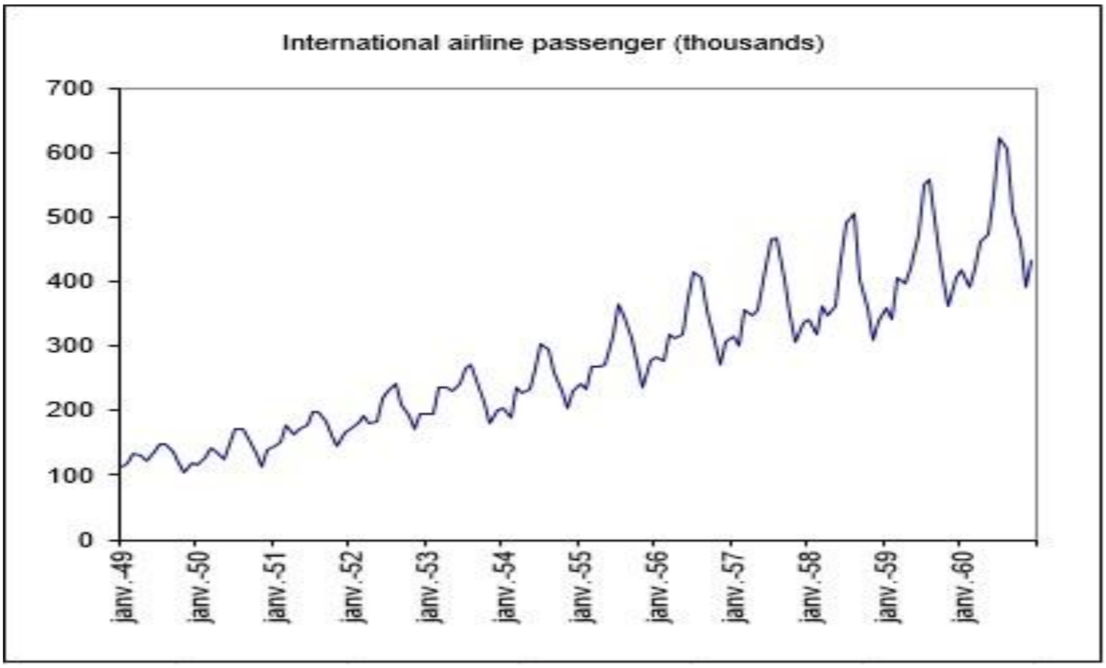
1. Whereas in regression, whether data is recorded at regular or irregular intervals of time, we can apply.
2. Regression is Interpolation.
3. Regression refers both ordered and unordered series of data.

Q11. What is the difference between stationery and non-stationary data?

Stationary: A series is said to be "STRICTLY STATIONARY” if the Mean, variance & Covariance is constant over some time or time invariant.



Non-Stationary: A series is said to be "STRICTLY STATIONARY" if the Mean, Variance & Covariance is not constant over some time or time invariant.



Q12. Why cannot you take non-stationary data to solve time series Problem?

- Most models assume stationary of data. In other words, standard techniques are invalid if data is "NON-STATIONARY".
- Autocorrelation may result due to "NON-STATIONARY".
- Non-stationary processes are a random walk with or without a drift (a slow, steady change).
- Deterministic trends (trends that are constant, positive or negative, independent of time for the whole life of the series).
