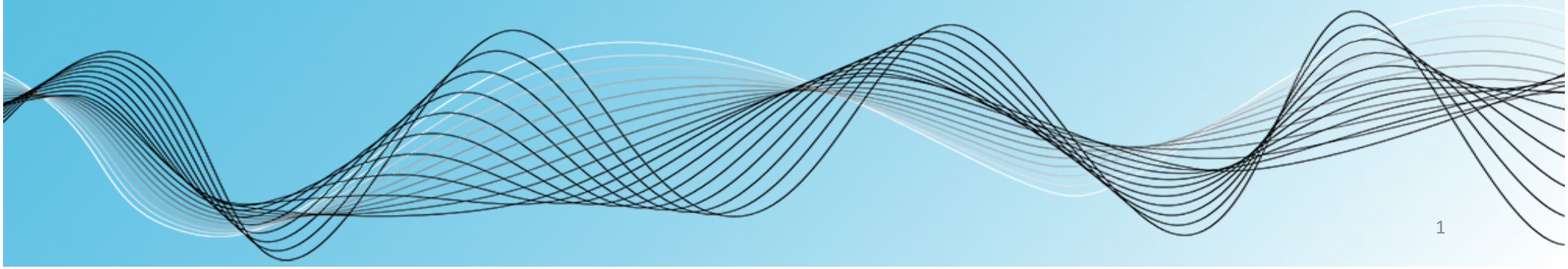
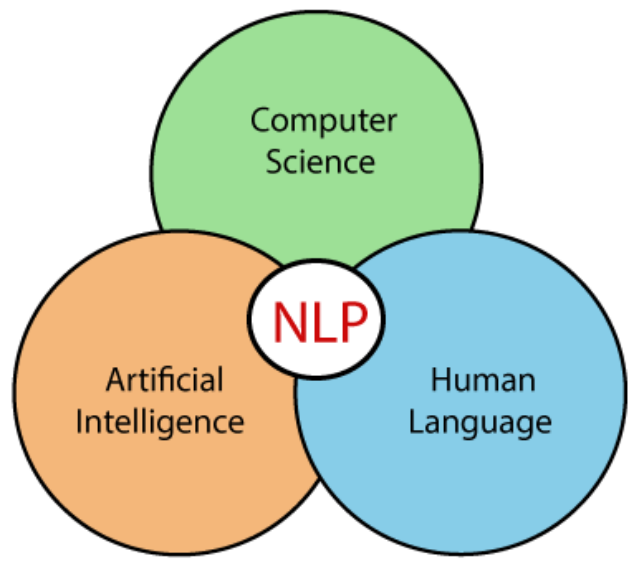
Text, Tokens



Basic NLP

- NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence.
- It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages.
- It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.

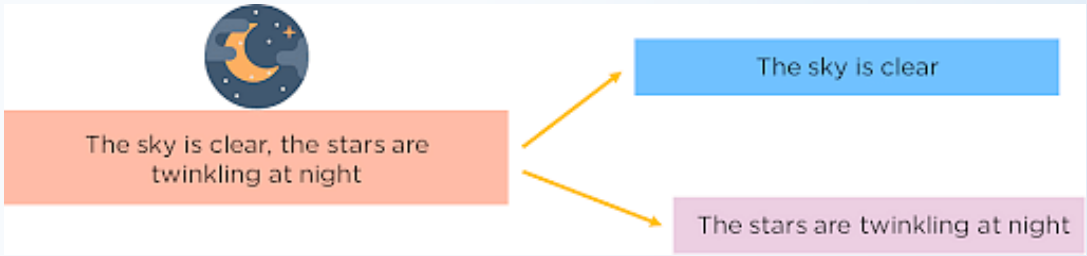


Basic NLP

- NLP combines the field of linguistics and computer science to decipher language structure and guidelines and to make models which can comprehend, break down and separate significant details from text and speech.
- In order to produce significant and actionable insights from text data, it is important to get acquainted with the basics of Natural Language Processing (NLP).
- What Can Natural Language Processing Do?
  - Intelligent Assistants
  - Search Results
  - Intuitive Typing
  - Translation
  - Data Analysis
  - Spam Detection
  - Sentiment Analysis
  - And many more .....

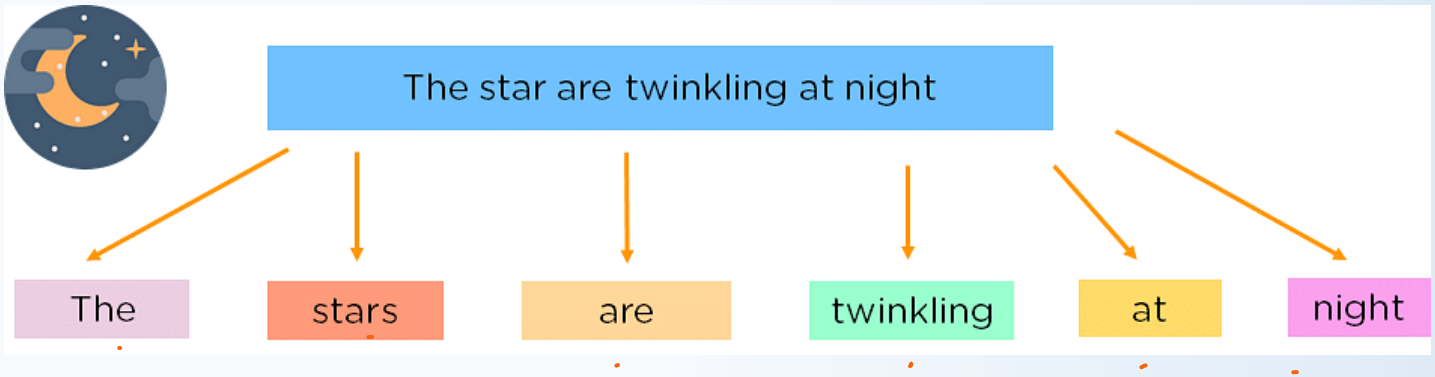
NLP : Text , Tokens

- **Segmentation:**
- Sentence Segment is the first step for building the NLP pipeline. It breaks the paragraph into separate sentences.
- Example: Consider the following paragraph -
  - Independence Day is one of the important festivals for every Indian citizen. It is celebrated on the 15th of August each year ever since India got independence from the British rule. The day celebrates independence in the true sense.
- Sentence Segment produces the following result:
  - "Independence Day is one of the important festivals for every Indian citizen."
  - "It is celebrated on the 15th of August each year ever since India got independence from the British rule."
  - "This day celebrates independence in the true sense."



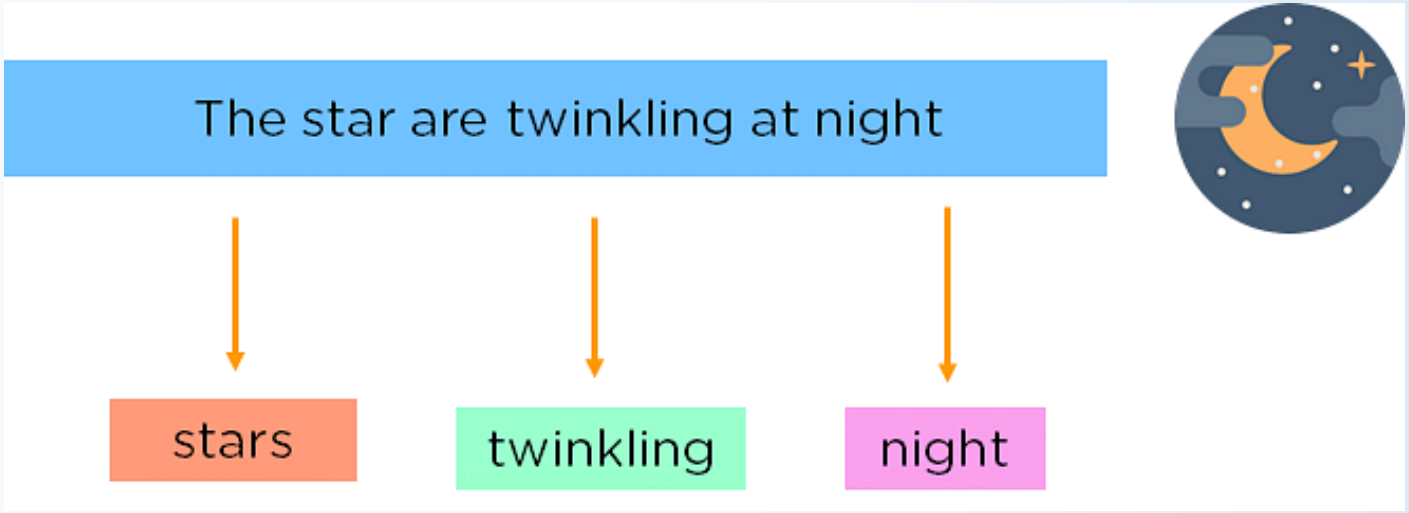
NLP : Text , Tokens

- **Tokenization:**
- We can see that unlike all the machine learning datasets we have worked with previously, the data isn't boolean, numeric, categorical etc. Usually a text is composed of paragraphs, paragraphs are composed of sentences, and sentences are composed of words. You could also go deeper into letters, but the letters have no meaning. It's only when they are combined into words, that the text starts to make sense. Hence, it is better to work at the word level.
- Tokenization is the process of splitting the text into smaller parts called tokens. Tokens are the basic units of a particular dataset. The choice of tokens could be based on the application we are working on.

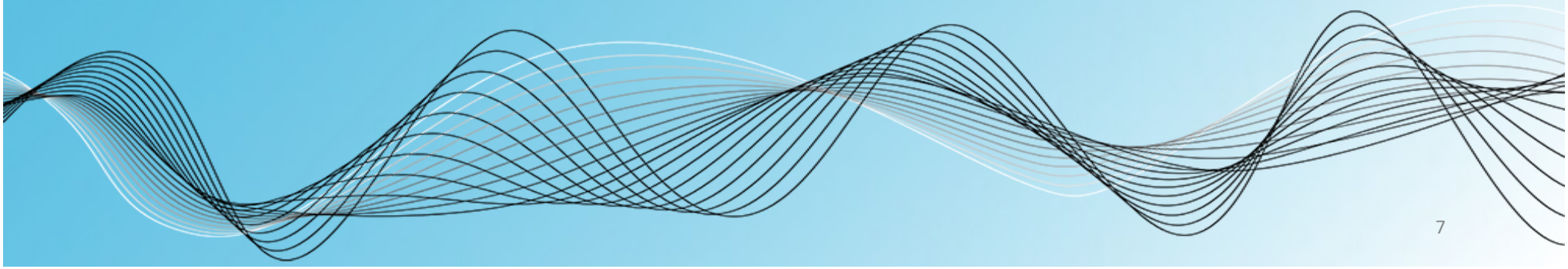


NLP : Stop words

- You can make the learning process faster by getting rid of non-essential words, which add little meaning to our statement and are just there to make our statement sound more cohesive.
- Words such as was, in, is, and, the, are called stop words and can be removed.



Regex



Regex

- A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.
- The Python module **re** provides full support regular expressions in Python.
- The re module offers a set of functions that allows us to search a string for a match:

Function	Description
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string
split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

Regex

- **Metacharacters**
- Metacharacters are characters with a special meaning:

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"planet$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
|	Either or	"falls|stays"

Regex

- **Special Sequences**
- A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

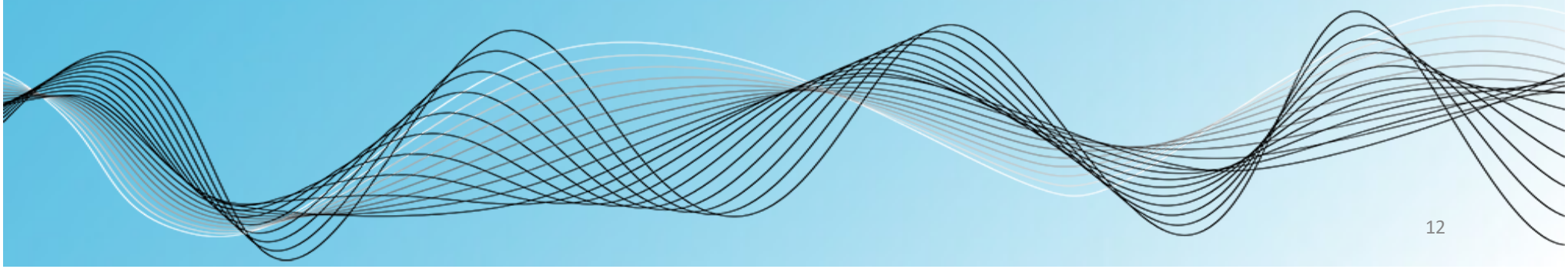
Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

Regex

- **Sets**
- A set is a set of characters inside a pair of square brackets [] with a special meaning:

Set	Description
[arn]	Returns a match where one of the specified characters (a, r, or n) is present
[a-n]	Returns a match for any lower case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a, r, and n
[0123]	Returns a match where any of the specified digits (0, 1, 2, or 3) are present
[0-9]	Returns a match for any digit between 0 and 9
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59
[a-zA-Z]	Returns a match for any character alphabetically between a and z, lower case OR upper case
[+]	In sets, +, *, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string

Stemming & Lemmatization

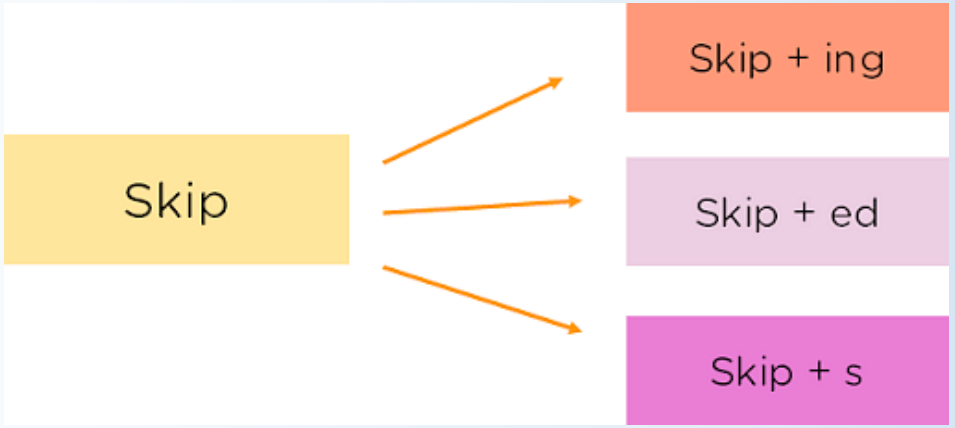


NLP : Stemming and lemmatization

- Owing to grammatical reasons, documents are going to use different forms of a word, such as discuss, discusses and discussing. Along with there are families of derivationally related words with similar meanings, such as liberal, liberty, and liberalization.
- Stemming and Lemmatization are Text Normalization (or sometimes called Word Normalization) techniques in the field of NLP that are used to prepare text, words, and documents for further processing.
- The goal of both the methods(stemming and lemmatization) is to reduce inflectional forms and derivationally related forms of a word to a common base form.
- For eg:
- am, are, is ⇒ be
- lion, lions, lion's, lions' ⇒ lion

NLP : Stemming

- **Stemming** is the process of converting the words of a sentence to its non-changing portions. So stemming a word or sentence may result in words that are not actual words. Stems are created by removing the suffixes or prefixes used with a word.
- For eg: Likes, liked, likely, unlike ⇒ like
- Lot of different algorithms have been defined for the process, each with their own set of rules. The popular ones include:
  - Porter Stemmer(Implemented in almost all languages)
  - Paice Stemmer
  - Lovins Stemmer



NLP : lemmatization

- This method is a more refined way of breaking words through the use of a vocabulary and morphological analysis of words. The aim is to always return the base form of a word known as lemma.
- Consider the following words:
  - 'Studied', 'Studious' ,'Studying'
  - Stemming of them will result in Studi
  - Lemmatisation of them will result in Study
- As it can be seen Lemmatization is more complex than stemming because it requires words to be categorized by a part-of-speech as well as by inflected form.