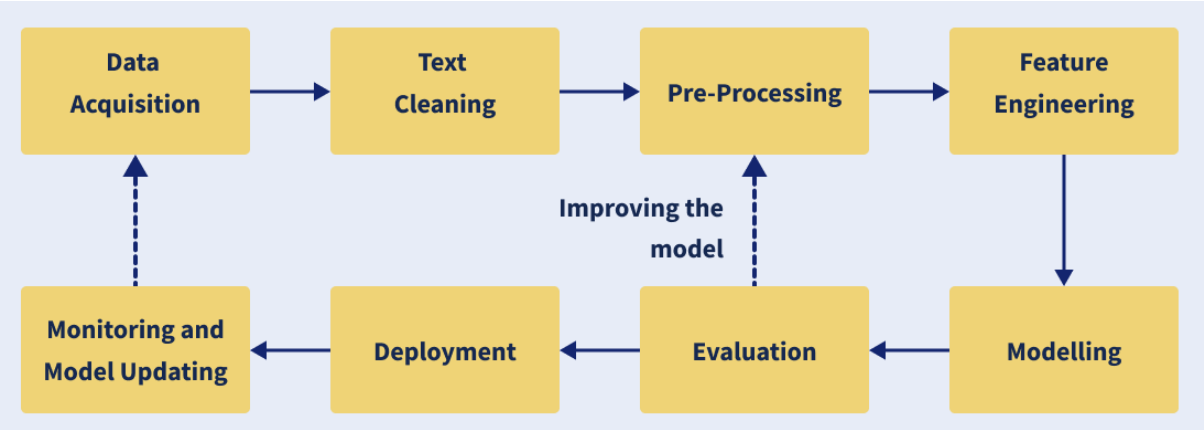
NATURAL LANGUAGE PROCESSING

Natural language processing is a subfield of machine learning that allows the computers to process the natural language and gain insight into the dataset. It helps in information retrieval that can be applied in tasks like speech recognition, text classification, language translation, chatbots, advertisement marketing, sentiment analysis etc.



NLP Life Cycle

Data Acquisition: getting a collection of relevant data for gaining correct insights.

Text Cleaning: handling missing data, correcting improper format, removing corrupt data etc.

Preprocessing: preparing unstructured data for modelling.

- Tokenization:
  - Word tokenize: creating list of individual words of sentences or paragraphs.
  - Sentence tokenize: creating a list of individual sentences of a paragraph.
- Removing stop words: common words that do not necessarily add any context to the sentence. Grammatical fillers.
- Stemming: converting words to their root form, cuts part of the word indiscriminately.

- Lemmatization: grouping together words that share the same root word. Returns a proper word.
- Spell checker: fix misspellings.
- Part of speech tagging: assigning parts of speech to the tokens.
- Named Entity Recognition: extract entities from text. Groups similar entities like person, organization etc. together.
- Co-reference Resolution: resolving reference words by assigning proper values to them.

Feature Engineering:

- Word frequency: calculates frequency of each word occurring in the text in order to find the most common term.
- Collocation: applied using bigrams, trigrams or n-grams. It helps in figuring out words that often occur together.
- Concordance: examines a word in different settings which helps in identifying the context of the word.
- TF-IDF: gives significance of term in a corpus collection.
    - Term frequency: Number of times a word occurs in a document with respect to the total number of words.
        - TF(t,d) = frequency of t in d / number of words in d
    - Document frequency: Number of documents t occurs in corpus of documents.
    - Inverse document frequency: IDF=log(Number of documents / Document frequency).
- Text Summarization: this involves compressing text without losing the key points.
    - Extractive Summarization: summarizes text using concise phrases from the text.
    - Abstractive Summarization: summarizes text using new phrases to convey the key points of the text.
- Text Classification: classifying an open-ended text into a given number of pre-defined categories.

Modelling: Depending on the purpose of NLP, relevant machine learning and deep learning models like Naïve Baye's, Logistic Regression, Feedforward Networks, and Recurrent Neural Networks etc. are used.

NLTK AND SPACY

NLTK and Spacy are both python packages used for text processing task. However, the two have their advantages and disadvantages.

Comparison:

- NLTK supports more languages than Spacy along with named entity recognition for those languages.
- NLTK provides a wider selection of algorithms while as Spacy has fixed number of best algorithms.
- NLTK does not support word vectors while as Spacy does.
- NLTK is more appropriate for research while Spacy is better option for production environment.
- NLTK is slower than Spacy and less efficient but provides more flexibility and fine-tuning.
- NLTK has extensive documentation compared to Spacy, hence better community support.
- NLTK has less effective tokenization as Spacy uses syntactic tree for tokenization.
- For smaller projects, NLTK provides more flexibility while as for larger projects Spacy provides speed and efficiency.
- For multilingual projects NLTK provides better support while as for specialized projects Spacy works optimally.