# ***Natural Language Processing***

* It is a method to find meaningful information from random text,or on how to deal with text data.
* Various types of NLP :
  + Sentiment Analysis
  + Topic Modeling
  + Text Generation
  + Named Entity Recognition
  + Text Summarization
  + Text Classification
  + Keyword Extraction
* Basic Skills required for NLP

| Programming | Data | Pandas, sklearn, re |
| --- | --- | --- |
|  | NLP | nltk, TextBlob, gensim |
| Math and Stats | Clean | Corpus, document-term matrix |
|  | EDA | Word counts |
|  | NLP | Various NLP techniques (like the above) |
| Communication | Design | Scope, Visualize, Extract insights |
|  | Domain | Expertise in the topic |

* Basic Workflow of the NLP Problems

1. Start with a question
2. Get/Collect and Clean the Data
3. Perform EDA
4. Apply the Techniques
5. Share Insights

## ***Start with a Question***

* Decide on what type of problem you want to work on and what result you expect to reach at the end, i.e, create the problem statement

## ***Collect and Clean the Data***

* **Collecting Data**
  + If you are working with a predefined problem statement with its own data, there is no need to collect any new data, we can directly start working on cleaning it.
  + If we are generating a new problem statement, where we are collecting our own data we can collect the required data from various websites using Web Scraping Techniques.
* **Cleaning Data**
  + The output after the cleaning of the data will be
    - **Corpus** - A collection of text(A data frame with containing the text)
    - **Document-Term Matrix** - Word count in matrix format
  + Common Cleaning Steps
    - Make all letter lowercase
    - Remove all punctuations marks and special characters
    - Remove all numerical values
    - Remove all common nonsensical text(\n,\t,etc..)
    - Tokenize the text
    - Remove the stopwords(except words like no,not and nor)
  + Few more steps that can be taken to clean the data after tokenization
    - Stemming/Lemmatization
    - Arts of speech tagging
    - Create bi-grams, tri-grams or n-grams
    - Deal with typos
    - And many more…..

### **Vectorization**

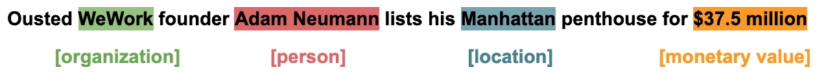
* + TF-IDF :
  + Word2Vec:

## ***Perform EDA***

* After the cleaning up of data, we try to see if the data makes sense and try to understand and find any possible connection or relation between the data, i.e., we do Exploratory Data Analysis(EDA).
* This can be done by visualizing the data in different ways.
  + For numerical data, we can find the average of the data, plot the distribution of the data, the most common values,highest and lowest values,etc.
  + Similarly for text data, we can plot the frequency of the most used words, create a CloudWord for most frequently used words, find different trends in the text data, find the size of vocabulary,etc.

## ***Apply the Techniques***

* Once all the data cleaning and EDA is done, we get a corpus and a document-term matrix to work with. Finally, we start working on the problem statement.
* We select a technique to work on based on what we want to find using NLP
  + **Text Classification** : It is the organizing of large amounts of unstructured text. It takes your text dataset then structures it for further analysis. Topic modeling, sentiment analysis, and keyword extraction are subsets of Text Classification.
  + **Sentiment Analysis** : It is the dissection of the data in order to determine whether the data is positive, negative or neutral. In this manner, sentiment analysis can transform large archives of [customer feedback](https://monkeylearn.com/customer-feedback/), reviews, or social media reactions into actionable, quantified results.
  + **Topic Modeling** : [Topic Modeling](https://monkeylearn.com/blog/introduction-to-topic-modeling/) is an unsupervised Natural Language Processing technique that utilizes artificial intelligence programs to tag and group text clusters that share common topics.
  + **Keyword Extraction** : It is the automated process of extracting the most relevant information from text using AI and machine learning algorithms.
  + **Text Summarization** : It is the breakdown of jargon, whether scientific, medical, technical or other, into its most basic terms using natural language processing in order to make it more understandable.
  + **Text Generation** : Its goal is to generate meaningful phrases and sentences in the form of human-written text from the input data.
  + **Named Entity Recognition** : It is a Natural Language Processing technique that tags ‘named identities’ within text and extracts them for further analysis.



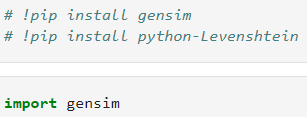
## ***Share Insights***

* Finally after the model is constructed and we get our results, we can conclude our insights throughout the process like one found during doing EDA or after the building of the model.

# ***Code Snips for Data Cleaning and Vectorization***

## ***Data Cleaning***

* The very basic of preprocessing and tokenization can be done directly by using ‘*simple\_preprocess*’ function from the ‘*gensim*’ library
  + It will convert all the words to lowercase, remove all punctuations,trimming spaces, remove some stopwords(not all) and tokenize the words.
  + For very simple cleaning data, we can use this and directly start with vectorization and modeling.





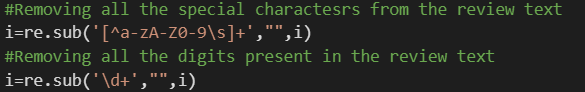
* Similarly we also have a library that helps us decontracting the words like ‘*don’t*’ to ‘*do not*’, ‘*I’m*’ to ‘*I am*’, etc.
  + We use a library called ‘*contractions*’ to do this task





* We can also use the ‘re’ library that helps us use the regular expressions functions on the text to clean the data manually as per the need from the data.

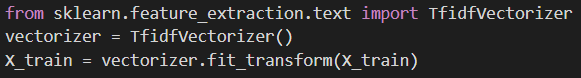


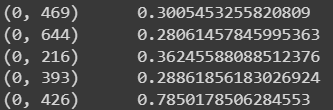




## ***Vectorization***

* To use the text data in the model we need to convert the tokenized text into numeric format using various vectorization methods like
  + TF-IDF
  + Word2Vec
  + BERT
* TF-IDF
  + So all the words in a sentence are given a numeric value

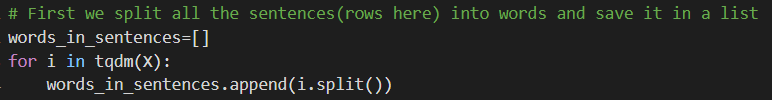




* + As we can see, sentence 1(index 0 here) has 5 words and all of which is vectorized using the formula of TF-IDF.
* Word2Vec



* + We first create a list of words as input instead of the sentences.



* + Once the list is created we will create a Word2Vec model where it will vectorize all the words in the vocabulary

