**Sentiment Analysis**

**Sentiment Analysis:**

* Sentiments are feelings, opinions, emotions, likes/dislikes, good/bad.
* Sentiment Analysis is a Natural Language Processing & information Extraction task aims to obtain writers feelings, expressed in positive or negative.
* Sentiment Analysis is a study of human behaviour in which we extract user opinions and emotions from plain text.
* Sentiment Analysis is also known as Opinion Mining.

Q – why we do it?

A- to check from a business POV. We check whether a customer’s likes a product or not.

3 polarity can be derived from Sentiment Analysis : Positive, Negative, & Complex

**For example:**

Sameer: Apple iPhone is great phone. It is better than any other phone I have bought.

Great = Positive

Better = Positive

Total Positives =2

Total Negatives =0

Net score = 2-0 = 2

Hence, Review is Positive

**Before we conduct Sentiment Analysis, we do Pre- Processing:**

~Tokenization

* ﻿﻿Unigram: considers only one token e.g. It is a good movie. (It, is, a, good, movie?
* ﻿﻿Bigram: considers two consecutive tokens e.g. It is not bad movie. fIt is, is not, not bad, bad movie;

~ Case Conversion

Convert the words either into uppercase or lower case. Most of the time lower case.

~Removal of punctuation (filtration)

Eg. @, #, !, ?

**Implementation**

Data > Tokenization > Sentence Splitter > Adjective Extraction > Senti WordNet Interpretation > Aggregating Scores

Data > Tokenization > Sentence Splitter ( done by Pre processing )

Adjective Extraction > Senti WordNet Interpretation > Aggregating Scores ( Done by Pre defined Algorithms)

**STEPS in TEXT PROCESSING:**

* Tokenization

Tokenization is the process by which a large quantity of text is divided into smaller parts called tokens. These tokens are very useful

And lemmatization. Tokenization also helps to substitute sensitive data elements with non-sensitive data elements. Natural language processing is used for building applications such as Text classifications, intelligent chatbot, sentimental analysis language translation, etc. It becomes vital to understand the pattern in the text to achieve the above-stated purpose.

Natural language toolkit has very important module NLTK tokenize sentence which further comprises of sub modules.

• word tokenizes

• sentence tokenizes

* Stemming

Stemming is a method of normalization of words in Natural Language Processing. It is a technique in which a set of words in sentence are converted into a sequence to shorten its lookup, in this method, the words having the same meaning but have same variations. According to the context or sentence are normalized.

* Lemmatization

Lemmatization in NLTK is the algorithmic process of finding the lemma of a word depending on its meaning and context. Lemmatization usually refers to the morphological analysis of words, which aims to remove inflectional endings. It helps in returning the base or dictionary form of a word known as the lemma.

The NLTK Lemmatization method is based on WorldNet's built-in

Morph function. Text Pre-processing includes bother stemming as well as lemmatization. Many People find the two terms confusing. Some treat these as the same, but there is a difference between stemming vs lemmatization. Lemmatization is preferred over the former because of the below reason.

* While using Stemmer it will work for few words like wait, waiting, waited, waits but not for studying, studies, study, cries, cry.
* Lemmatization will fix these issues and works with all words and can tokenize all the words in meaningful format.
* Vectorization

Machines cannot understand characters and words. So when dealing with text data we need to represent it in numbers to be understood by the machine. Countvectorizer is a method to convert text to numerical data.

To show you how it works let's take an example:

Text = [‘Hello my name is James, this is my python notebook’]

The text is transformed to a sparse matrix as shown below.

hello is James my name notebook python this

1 2 1 2 1 1 1 1

**Code Explanation: NLTK**

* word tokenize module is imported from the NLTK library.
* A variable "text" is initialized with two sentences.
* Text variable is passed in word \_tokenize module and printed the result. This module breaks each word with punctuation which you can see in the

**Ways to Perform Sentiment Analysis in Python:**

Python is one of the most powerful tools when it comes to performing data science tasks-it offers a multitude of ways to perform sentiment analysis. The most popular ones are enlisted here:

* ﻿﻿Using Text Blob
* ﻿﻿Using Vader
* ﻿﻿Using Bag of Words Vectorization-based Models
* ﻿﻿Using RNN-based Models
* Using Transformer-based Models