Sentiment analysis of tweets using NLTK library (NLP)

# Introduction

### **Natural language processing** (**NLP**) is a subfield of AI, which help computers to extract information and insights contained in various documents like pdfs, news feeds and social media messages.

### Sentiment analysis is one of the main applications of natural language processing. Thousands of text documents can be processed for sentiment (and other features including named entities, topics, themes, etc.) in seconds, compared to the hours it would take a team of people to manually complete the same task.

**Main Objective**

* To train a classifier to predict sentiment from thousands of Twitter tweets about various mobile phone reviews.
* This project could be practically used by any company with social media presence to automatically predict customer's sentiment (i.e. whether their customers are happy or not with the product).

**Data Description**

For training the models, a labelled tweets dataset is provided. The dataset is provided in the form of a csv file with each line storing a tweet id, its label and the tweet.

**Dataset**

* tweets.csv: 7,920 tweets
* label column : 0 for negative

1 for positive

**Approach and Implementation**

**Data Preparation**

Tweets are mostly highly unstructured data with many hashtags, punctuation marks, misspelled words, html entities etc. So, they have to be cleaned first.

Tweets file has been pre-processed using a standardized process as given below:

* Removing any URLs, twitter user handles, punctuation marks, numbers and whitespaces from the tweets using **regex library**.
* Converting text to lower case, removing stop words, and lemmetizing further to normalize the text datasets to feed into different vector models

**Visualization**

The most common 25 words in the tweets were extracted and visualized using Matplotlib library.

**Model Building**

Frequency of every word in vocabulary is calculated and used to build the feature space using CountVectorizer class in sklearn.

**Train-test split:**

The train data is split into train and validation set for training and validation in the ration 0.7:0.3

**Model evaluation:**

**LogisticRegression** achieved an f1 score of 76.55 on the validation dataset

**Naïve Bayes classifier** achieved an f1 score of 81.46 on the validation dataset

**Note: All the code is written in python version 3.7**