**Sentiment Analysis**

Sentiment Analysis is a branch of computer science and overlaps heavily with Machine Learning, and Computational Linguistics Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative, or neutral.

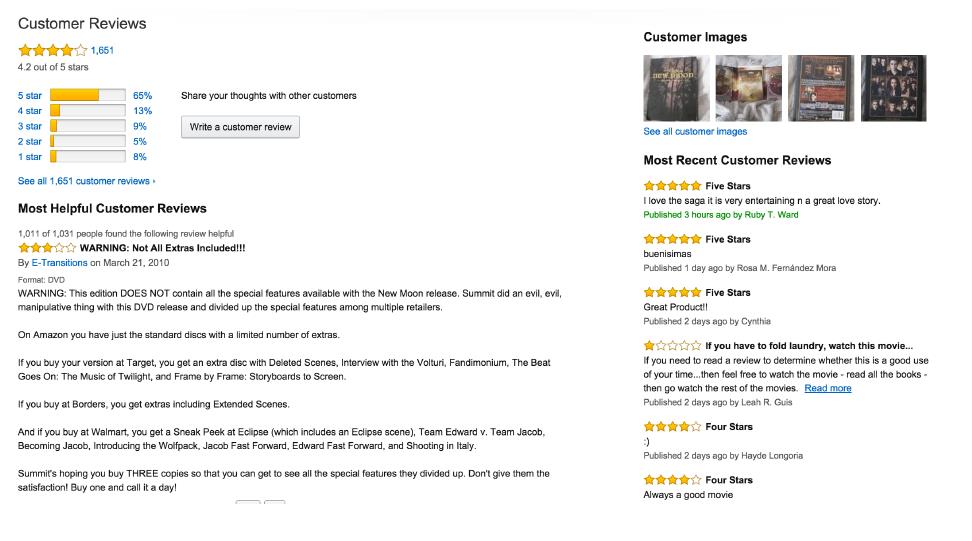
It is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer’s attitude towards a particular topic, product, etc. is positive, negative, or neutral.

Sentiment Analysis is a concept of Natural Language Processing and sometimes referred to as opinion mining, although the emphasis, in this case, is on extraction.



**Examples of the sentimental analysis are as follows:**

* Is this product review positive or negative?
* Is this customer email satisfied or dissatisfied?
* Based on a sample of tweets, how are people responding to this ad campaign/product release/news item?
* How have bloggers’ attitudes about the president changed since the election?
* The purpose of this sentiment analysis is to automatically classify a tweet as a positive or negative Tweet Sentiment wise
* Given a movie review or a tweet, it can be automatically classified in categories. These categories can be user-defined (positive, negative) or whichever classes you want.
* Sentiment Analysis for Brand Monitoring
* Sentiment Analysis for Customer Service
* Sentiment Analysis for Market Research and Analysis



**Sample Positive Tweets**

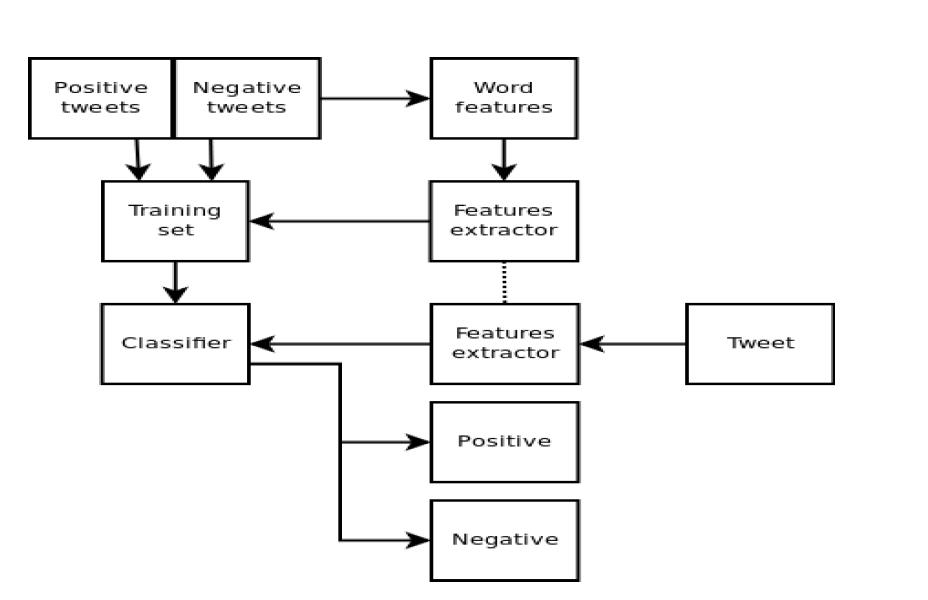
* I love this car
* This view is amazing
* I feel great this morning
* I am so excited about the concert
* He is my best friend

**Sample Negative Tweets**

* I do not like this car
* This view is horrible
* I feel tired this morning
* I am not looking forward to the concert
* He is my enemy

**Sentimental Analysis Process**

* The list of word features needs to be extracted from the tweets
* It is a list with every distinct word ordered by frequency of appearance
* The use of Feature Extractor to decide which features are more relevant
* The one we are going to use returns a dictionary indicating that words are contained in the input passed



**Naive Bayes Classifier**

* It uses the prior probability of each label, which is the frequency of each label in the training set and the contribution from each feature.
* In our case, the frequency of each label is the same for ‘positive’ and ‘negative’.
* The word ‘amazing’ appears in 1 of 5 of the positive tweets and none of the negative tweets.
* This means that the likelihood of the ‘positive’ label will be multiplied by 0.2 when this word is seen as part of the input

#### **Sentiment Analysis Example 1:**

**Training Data**

This is a good book! Positive

1. This is an awesome book! Positive
2. This is a bad book! Negative
3. This is a terrible book! Negative

**Testing Data**

* This is a good article
* This is a bad article

[NLTK](https://www.nltk.org/)has a built-in method that computes the accuracy rate of our model:

>>> from nltk.classify.util import accuracy

**Sentiment Analysis Example 2:**

Gender Identification: know that male and female names have some distinctive characteristics. Generally, Names ending in a, e, and i are likely to be female, while names ending in k, o, r, s, and t are likely to be male.

.