NLP – SENTIMENT ANALYSIS

*(the outputs shown below are of classification of movie reviews into positive or negative) (sentdex course on youtube)

The first step in sentiment analysis supervised learning is to collect a huge amount of data to train an algorithm. The size of the data has a huge impact on the accuracy of the algorithm. So to collect we can use nltk.corpus which has which has huge categorized data set which can be used to train to model. We can also use an Api like twitter api or any other api which provides us with categorized data set.

As the data collected is very huge so to analyse this data at a quicker rate we need to use some features provided by nltk like word tokenizing, sentence tokenizing, stop words, stemming, lemmatizing…..

Word tokenizing – It is the process of splitting a large sample of text into words and is similar for sentence tokenizing.

Stop words - Stop words are words which are filtered out before or after processing of natural language data. Stop words are generally the most common words in a language. The meaning of the sentence does not change.

Example -

There is a tree near the river	**There tree near river**

We can also use lemma and antonyms to find similar or opposite words

Bow-model :- The bag of words used to simplify it for nlp by converting words to numbers.

ALGORTIHMS

1 NAÏVE BAYES CLASSIFIER

-few of the types of Naive Bayes classifier are Bernoulli, Gaussian and Multinomial classifier

- The formula for naïve bayes theorem is

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

- Example

A great movie.	positive
It was an amazing movie.	positive
The movie was horrible.	negative
The movie was not so good	negative

So if we want to calculate the probability of a word contributing to positive or negative category

$$P(\text{great}|\text{positive}) = \frac{\text{total no of times great ocurs in positive section}}{total\ \ no\ of\ words\ in\ positive\ category}$$

But this formula is not reliable as it can have an output of 0.

So we need to use Laplace Smoothing.

$$P = \frac{\text{word count} + 1}{\text{total number of words in that category} + \text{no of unique words[means the words which are not repeated]})}$$

Output accuracy

```
The accuracy of Naïve Bayes Classifier is: 91.0
```

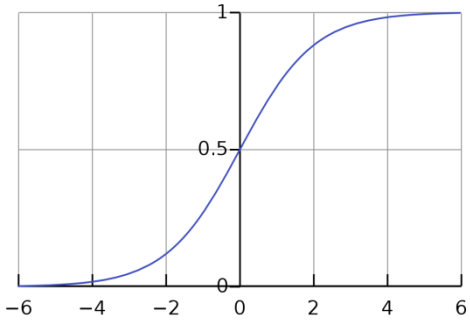
```
The accuracy of MNB Classifier is: 87.0
The accuracy of Bernoulli classifier is: 86.0
```

2. LOGISTIC REGRESSION

It is generally used for binary classification 0, 1/ positive, negative/ malignant, benign.

We cannot use linear regression for this so we use a sigmoid function

SIGMOID FUNCTION: $f(x) = \frac{1}{1+e^{-(x)}}$



so if the output is above 0.5 it is rounded off to 1 and if it is below 0.5 it is taken 0.

Cost function

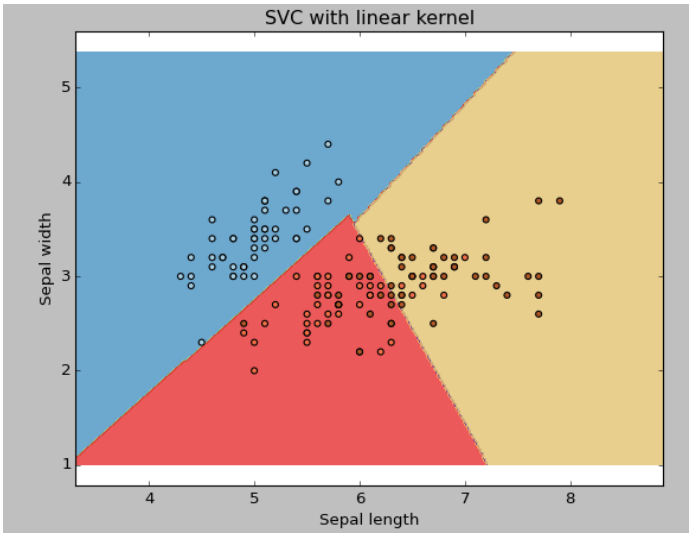
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

Output accuracy

```
The accuracy of Logistic Regression Classifier is: 78.0%
```

3. Linear Support Vector Classifier

This method divides the graph in parts using straight lines.
Y=mx + c
In case of binary classification only one line is present and so on …



This type of graph can classify it into three parts like positive negative or neutral

```
The accuracy of LinearSVC is : 75.0%
```

We can calculate the accuracy of a given classifier by:-

nltk.classify.accuracy (classifier, testing_set)*100

We can also find the best classifier for a given set of data using voted classifier:-

In this method first we take an odd no of classifiers and we find the final output of all the classifiers and the majority is the answer (positive/negative). We can also calculate the percentage. [(Positive or negative) / total]

```
classification: neg confidence % 57.14285714285714
classification: neg confidence % 100.0
classification: neg confidence % 100.0
```

In this method even number of classifier cannot be used as if we use 4 classifiers and 2 of them have positive output while the other two have negative so then we cannot get the final output.