Assignment C4

* Title:- Sentiment Analysis

* Objective (i) Understand idea & concept of text preprocessing.
(ii) Understand ML algorithms used for sentiment analysis.

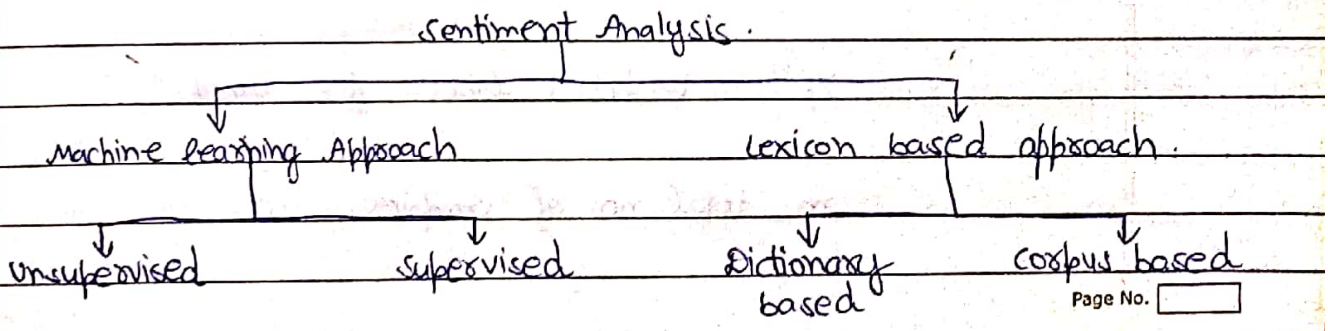
* Problem statement; Use Twitter data for sentiment analysis. Identify tweets which are hate tweets & which are not.

* Outcome:- (i) Understood Regex for text preprocessing.
(ii) Understood concept of classification algorithms.

* S/W & H/W Requirements:-
  - Jupyter notebook, GPU (preferably).
  - Python libraries.

* Theory:-

1) Sentiment Analysis:-
  - Contextual mining of text.
  - Identifier & extracts subjecting meaning of text.
  - also called opinion mining.
  - basically refers to classification of given text, here tweet, as negative or positive based on words present in the tweet.



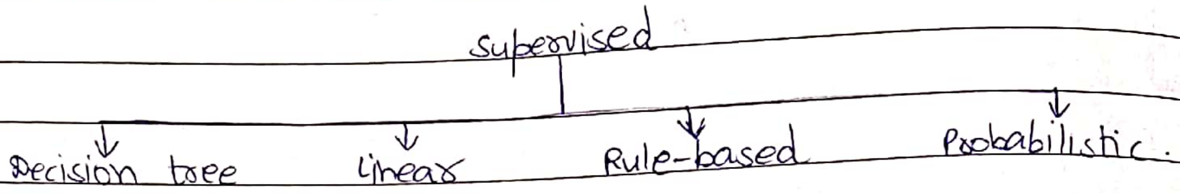


Fig:- Sentiment Analysis Technique.

- logistic Regression.
  - makes use of sigmoid fn to calculate probability of tweet belonging to a particular class.

$$P(t) = \sigma(t) = \frac{1}{1+e^{-t}}$$

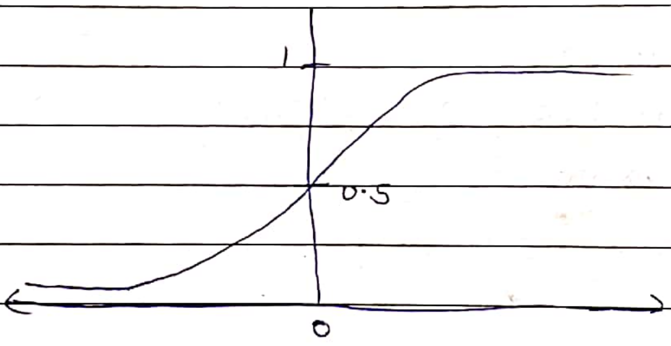


Fig:- Graph of sigmoid fn.

- idea is to minimize the cost fn

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

$y^{(i)}$: true label for i^{th} tweet.

$Y_p^{(i)}$: predicted label for ~~tweet~~ " "

m: total no. of samples.

2.) Text Preprocessing :-

- tweets contain characters & symbols which have no influence on sentiment of tweet.
- (i) Tokenization

  operation of splitting sentence into words.

  (ii) Removing stop words i.e. words which frequently occur but have no semantic value.

  (iii) Removing hastags, punctuations & other symbols.

  (iv) Stemming

  Reduction of words to root words by removal of suffixes & prefixes.

3.) Feature Extraction :-

- refers to transformation of tweets into vectors.
- (i) Bag-of-Words
  - builds vocabulary from a corpus & counts how many times a word occurs in a tweet. Poses Problem of high dimensionality.

  (ii) Term frequency - Inverse document frequency similar to BOW, but here the value of word decreases as the frequency of word increases.

* Algorithm :-

⇒ Import libraries such as NLTK, pandas, scikit-learn etc.

⇒ load twitter dataset into dataframe

⇒ preprocess tweets by

(i) removing URLs.

(ii) removing hastags.

(iii) replacing all characters other than number or alphabets with " ".

(iv) tokenize.

(v) stemming.

(vi) Removing stop words.

(vii) joining tokens to create processed tweets.

- Create corpus for positive & negative vocabulary.
- extract features by converting text to vector.
- Split dataset into train & test set.
- Train model and make predictions on test set.

* Conclusion:-

successfully implemented sentiment analysis for twitter dataset.