

# NATURAL LANGUAGE PROCESSING MINI PROJECT

## SENTIMENT ANALYSIS

What is Sentiment Analysis?

Sentiment Analysis (or sentiment classification) is a category of text classification where a given phrase or sentence is categorized into negative, positive or neutral attributes. Generally, a classifier does this by labelling the phrases into two attributes- negative and positive- to keep the classification binary.

Since recent times, attributes like, slightly positive, slightly negative are also into considerations.

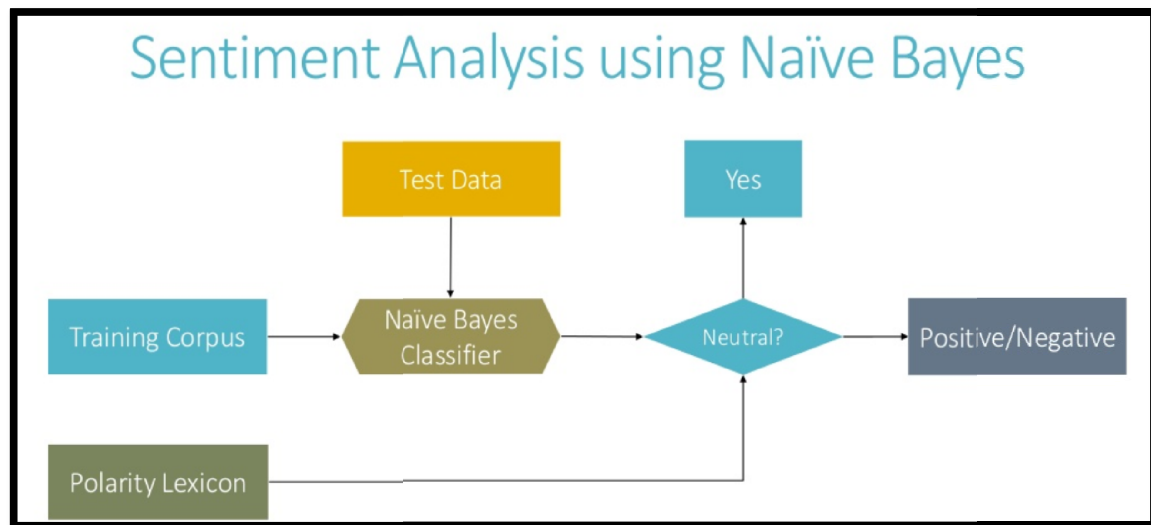


Fig. 01

**Problem Statement:** To train a model to recognize a review as positive or negative using Naive Bayes Classifier. We will be using a binary classification- positive and negative.

### **Flowchart:**

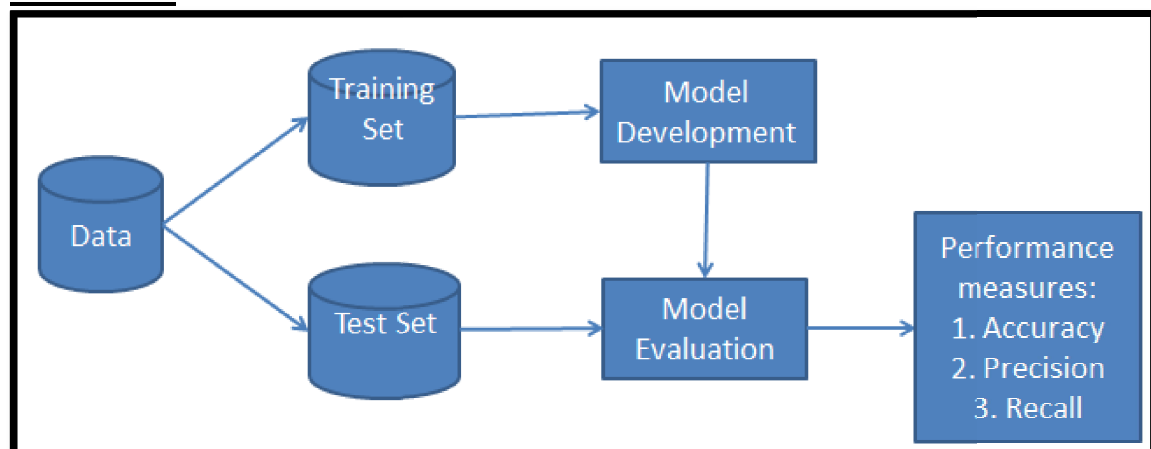


Fig.02

## **Method:**

1. Reviews are divided into- 10% testing set & 90% training set.
2. Training set – builds positive words and negative words dictionary.
3. Calculating important probability values,

$P(\text{word}|\text{negative})$  and  $P(\text{word})$

$$P(\text{word}|\text{positive})=N_{\text{word\_pos}}/N_{\text{all\_pos}}$$

$$P(\text{word}|\text{negative})=N_{\text{word\_neg}}/N_{\text{all\_neg}}$$

$$P(\text{word})=N_{\text{word}}/N_{\text{all\_word}}$$

Where,

$N_{\text{word\_pos}}$  - equals the number of times a word appears in the positive dictionary.

$N_{\text{all\_pos}}$  - equals the number of all the positive sentiment words from the training set (words are counted if it appears repetitively.)

$N_{\text{word\_neg}}$  - equals the number of times a word appears in the positive dictionary.

$N_{\text{all\_neg}}$  - equals the number of all the negative sentiment words from the training set (words are counted if it appears repetitively.)

$N_{\text{word}}$  - equals the number of times a word appears in the positive dictionary.

-equals 1, when there is no showing up of the word in the dictionary, which is a smoothing method.

$N_{\text{all\_word}}$  - equals the number of all words.

4. The most useful words (including bigrams) are used for deciding sentiment by looking at the proportion of

$\text{Power}(\text{word})=P(\text{word} \text{positive})/(p(\text{word} \text{positive})+p(\text{word} \text{negative}))$
--

Where the bigger ***Power(word)*** is the more useful a word is.

5. There is a fnction developed to include trigrams. It's worth noticing that trigramList consists of unigrams, bigrams and trigrams.

## **Conclusion:**

Thus, we achieved the model to classify negative and positive sentiments using Naive Bayes Classifier. So we implemented Sentiment Analysis.

## OUTPUT:

```
object-12192 SentimentAnalysis x + v
localhost:8888/notebooks/Desktop/NLP%20Project-121922016/SentimentAnalysis.ipynb
jupyter SentimentAnalysis Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code

['enigma', 'is', 'well', 'made', 'but', "it's", 'just', 'too', 'dry', 'and', 'too', 'placid', 'enigma_is', 'is_well', 'well_made', 'made_but', 'but_it's', 'it's_just', 'just_too', 'too_dry', 'dry_and', 'and_too', 'too_placid', 'enigma_is_well', 'is_well_made', 'well_made_but', 'made_but_it's', 'but_it's_just', 'it's_just_too', 'just_too_dry', 'too_dry_and', 'dry_and_too', 'and_too_placid']

Naive Bayes

Films (Train Data, Naive Bayes) Accuracy (All)=0.82 (7844/9595)
Films (Train Data, Naive Bayes) Precision (Pos)=0.93 (3300/3564)
Films (Train Data, Naive Bayes) Recall (Pos)=0.69 (3300/4787)
Films (Train Data, Naive Bayes) F-measure (Pos)=0.79
Films (Train Data, Naive Bayes) Precision (Neg)=0.75 (4544/6031)
Films (Train Data, Naive Bayes) Recall (Neg)=0.95 (4544/4808)
Films (Train Data, Naive Bayes) F-measure (Neg)=0.84
Films (Test Data, Naive Bayes) Accuracy (All)=0.80 (851/1068)
Films (Test Data, Naive Bayes) Precision (Pos)=0.80 (436/545)
Films (Test Data, Naive Bayes) Recall (Pos)=0.80 (436/544)
Films (Test Data, Naive Bayes) F-measure (Pos)=0.80
Films (Test Data, Naive Bayes) Precision (Neg)=0.79 (415/523)
Films (Test Data, Naive Bayes) Recall (Neg)=0.79 (415/524)
Films (Test Data, Naive Bayes) F-measure (Neg)=0.79

NEGATIVE:
['generic', 'badly', 'waste', 'mediocre', 'unfunny', 'routine', 'the_problem', 'ends_up', 'poorly', 'supposed_to', 'mindless', 'stale', 'boring', 'feels_like_a', 'shoot', 'of_the_characters', '90_minutes', 'pointless', 'wasn't', 'nowhere', 'unless', 'waste_of', 'not_enough', 'sit_through', 'only_thing', 'the_only_thing', 'far_too', 'stupid', 'mess', 'disaster', 'wants_to_be', 'meandering', 'should_have_been', 'never_really', 'bore', 'disguise', 'annoying', 'flat', 'a_script', 'inept', 'it_wasn't', 'apparently', 'save', 'offensive', 'tiresome', 'plodding', 'lousy', 'feature_length', 'lifeless', 'pinocchio']

POSITIVE:
['a_thoughtful', 'playful', 'iranian', 'resonant', 'but_still', 'tour', 'captivating', 'spare', 'grown', 'best_films', 'russpott', 'a_look', 'makes_up', 'makes_up_for', 'heartwarming', 'charming_and', 'captures', 'tender', 'a_terrific', 'polished', 'strength', 'wry', 'is_at', 'a_sweet', 'study_of', 'gem', 'lively', 'vividly', 'chilling', 'portrait_of_a', 'extraordinary', 'powerful', 'an_engaging', 'a_smart', 'mesmerizing', 'wonderful', 'wonderfully', 'love_and', 'a_powerful', 'vivid', 'in_years', 'refreshingly', 'refreshing', 'realistic', 'performances_from', 'riveting', 'inventive', 'what_makes', 'engrossing', 'of_the_rest']
```

Fig.03