CSCI 544: Applied Natural Language Processing

HW1

Submission by: Sairaj Pokale (USC ID: 8392909073)

Objective: Performing Sentiment Analysis on Amazon Product Reviews Dataset, implementing the Naïve Bayes Classifier, Logistic Regression Classifier, Perceptron Classifier, SVM Classifier.

Dataset: amazon_reviews_us_Office_Products_v1_00

Average length of reviews before preprocessing: 313.16912

Average length of reviews after preprocessing: 301.82492

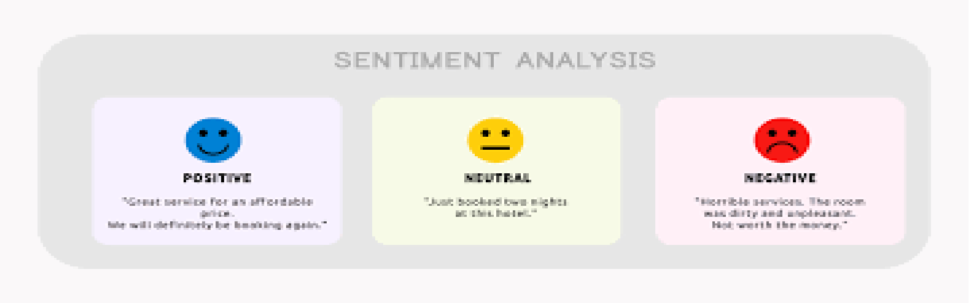
Average length of reviews after stop and lemma: 192.1008

- WHAT IS TEXT CLASSIFICATION?

  Text classification refers to the ability of a machine learning model's to predict a document's or sentence's category. The documents/sentences are classified on the basis of various features example TFIDF, since machines cannot understand the English language, this step converts words into numerical values and hence forms the basis of performing this task. Few applications of this task include Spam Email Detection, Sentiment Analysis (Objective of this Homework), Medical Document Classification, etc.

- SENTIMENT ANALYSIS

It is crucial for businesses to adapt to the rapidly evolving consumer demands, and a quick way to achieve that is by identifying the customers feedback, mostly expressed in the forms of reviews, but analyzing thousands of review is not only time consuming but also expensive in terms of man power. Here's where sentiment analysis comes in, using text classification one can determine a user's feedback with respect to the product, thus optimizing business operations.


SENTIMENT ANALYSIS
POSITIVE
"Great service for an affordable price. We will definitely be booking again."
NEUTRAL
"Just booked two nights at this hotel."
NEGATIVE
"Horrible services. The room was dirty and unpleasant. Not worth the money."

- FEATURE EXTRACTION

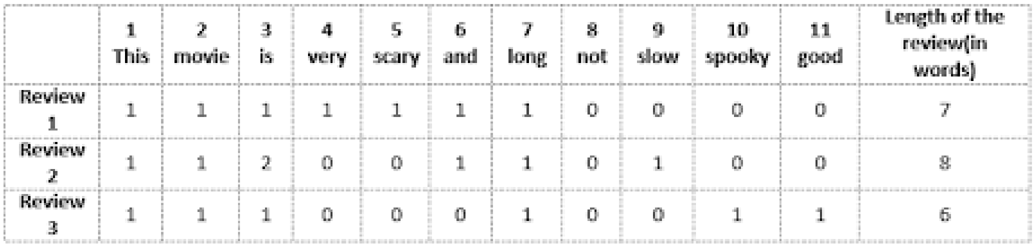
❖ TF-IDF: Term Frequency - Inverse Document Frequency is an algorithm which determines the relevancy of words to a document on the basis of their frequency.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

$$TF\text{-}IDF = TF * IDF$$

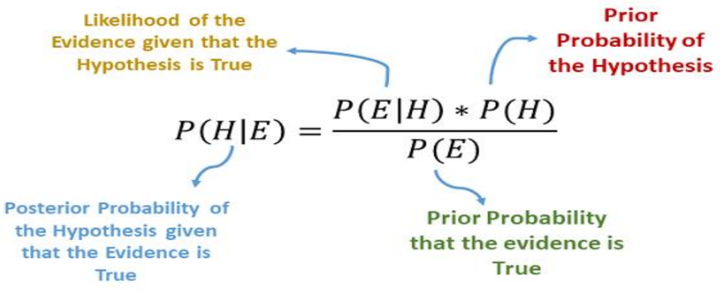
❖ BoW: Bag of Words is an algorithm which turns text into vectors on the basis of the count of each word.



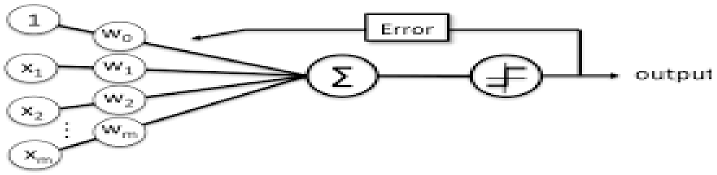
	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

- MODELS

  - Naive Bayes

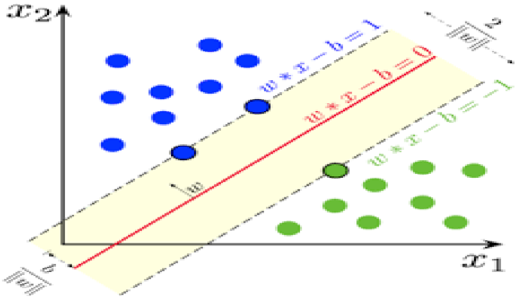

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$
Posterior Probability of the Hypothesis given that the Evidence is True
Prior Probability that the evidence is True

  - Perceptron: h(x_i) = sign ($w^T x_i$+ b)

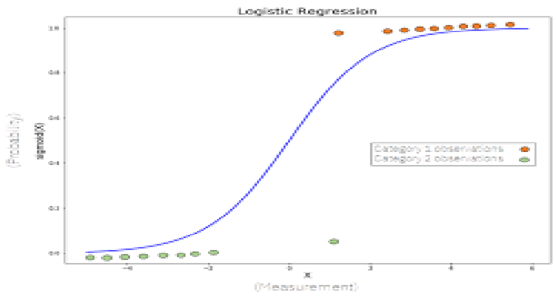


Schematic of a perceptron classifier.

  - SVM



  - Logistic Regression



- PYTHON

  - ❖ Version: 3.11

  - ❖ Requirements:
    pandas==2.1.0
    regex==2023.8.8
    nltk==3.8.1
    scikit-learn==1.3.0

  - ❖ Model Scores:

Algorithm	Feature	Accuracy	Precision	Recall	F1
Perceptron	TFIDF	0.795	0.802	0.789	0.795
Perceptron	BOW	0.796	0.798	0.796	0.797
Logistic Regression	TFIDF	0.827	0.831	0.825	0.828
Logistic Regression	BOW	0.845	0.834	0.864	0.849
Naïve Bayes	TFIDF	0.819	0.846	0.784	0.814
Naïve Bayes	BOW	0.809	0.789	0.850	0.818
SVM	TFIDF	0.847	0.855	0.840	0.847
SVM	BOW	0.833	0.820	0.857	0.838