PROJECT: SENTIMENT ANALYSIS FOR MARKETING

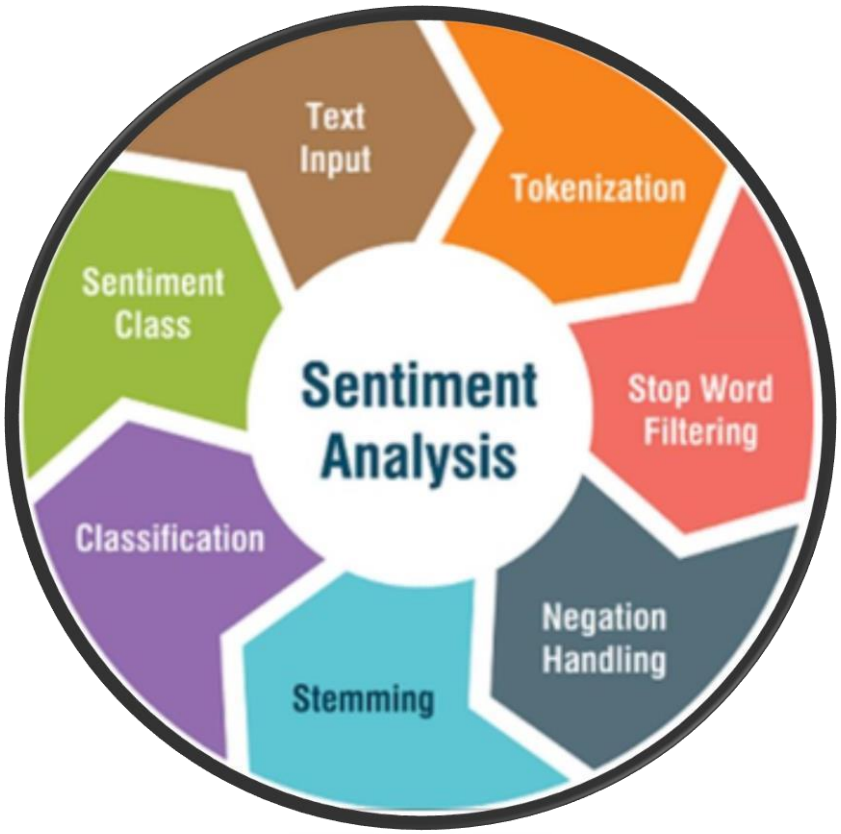
PRESENTED BY

M. BAIRAVI

821021104013

PHASE 4 : DEVELOPMENT PART 2

Start building the Sentiment Analysis for Marketing to analysis



Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online.

APPLICABILITY:

AI powered to enhance products by understanding Customers likes and dislikes.

TABLE OF CONTENTS:

INTRODUCTION:

- Sentiment Analysis, often referred to as opinion mining, is a powerful technique within the field of Natural Language Processing (NLP).
- At its core, sentiment analysis involves teaching machines to understand and interpret human emotions and opinions expressed within text data.
- By analyzing the sentiment behind words and phrases, Al models can classify text as positive, negative, or neutral, thus providing valuable insights into people's attitudes, feelings, and reactions.

TRAINING AI MODELS FOR SENTIMENT ANALYSIS:

Employing NLP techniques:

Employing Natural Language Processing (NLP) techniques for sentiment analysis involves using computational methods to understand and extract sentiment or emotional information from textual data. Here are the key steps and techniques commonly used in NLP-based sentiment analysis.

1.Text Preprocessing:

Tokenization:

Breaking down text into individual words or tokens.

Lowercasing:

Converting all text to lowercase for consistency.

Stop Word Removal:

Removing common words (e.g., "and," "the," "is") that don't carry much sentiment information.

2. Bag of Words:

Representing text data as a vector where each word is treated as a feature, and the count of words in a document is used to create a numerical representation.

3. Term Frequency-Inverse Document Frequency (TF-IDF):

Assigning a weight to each word in a document based on its importance within that document and across a collection of documents.

4. Word Embeddings:

Utilizing pre-trained word vectors (e.g., Word2Vec, GloVe) that capture semantic relationships between words and represent words in a continuous vector space.

5. Sentiment Lexicans:

Using sentiment lexicons or dictionaries that associate words with positive or negative sentiment scores. These can be employed in rule-based sentiment analysis.

6. Rule-Based Model:

Developing sentiment analysis models based on predefined rules, regular expressions, and lexicons. For example, assigning a positive or negative score to words and aggregating these scores for sentiment determination.

7. Machine Learning Models:

Training supervised machine learning models to predict sentiment labels (e.g., positive, negative, neutral). Common algorithms include:

- **Naïve Bayes**
- **Support Vector Machines**
- **Random Forest**

- **Recurrent Neural Networks (RNN)**
- **Long Short-term Memory (LSTM)**

8. Aspect-Based Sentiment Analysis:

Identifying and analysing sentiment at a more granular level, focusing on specific aspects or features within a piece of text.

9. Emotion Detection:

Going beyond simple positive/negative sentiment and identifying specific emotions expressed in text (e.g., joy, anger, sadness).

10. Deep Learning and Transformers:

Using deep learning models, such as Transformers, to capture complex contextual relationships in text and improve sentiment analysis accuracy.

11. Ensemble Methods:

Combining multiple sentiment analysis models or techniques to improve overall sentiment prediction performance.

These NLP techniques can be used individually or in combination to perform sentiment analysis on various types of text data, such as customer reviews, social media posts, product descriptions, and more.

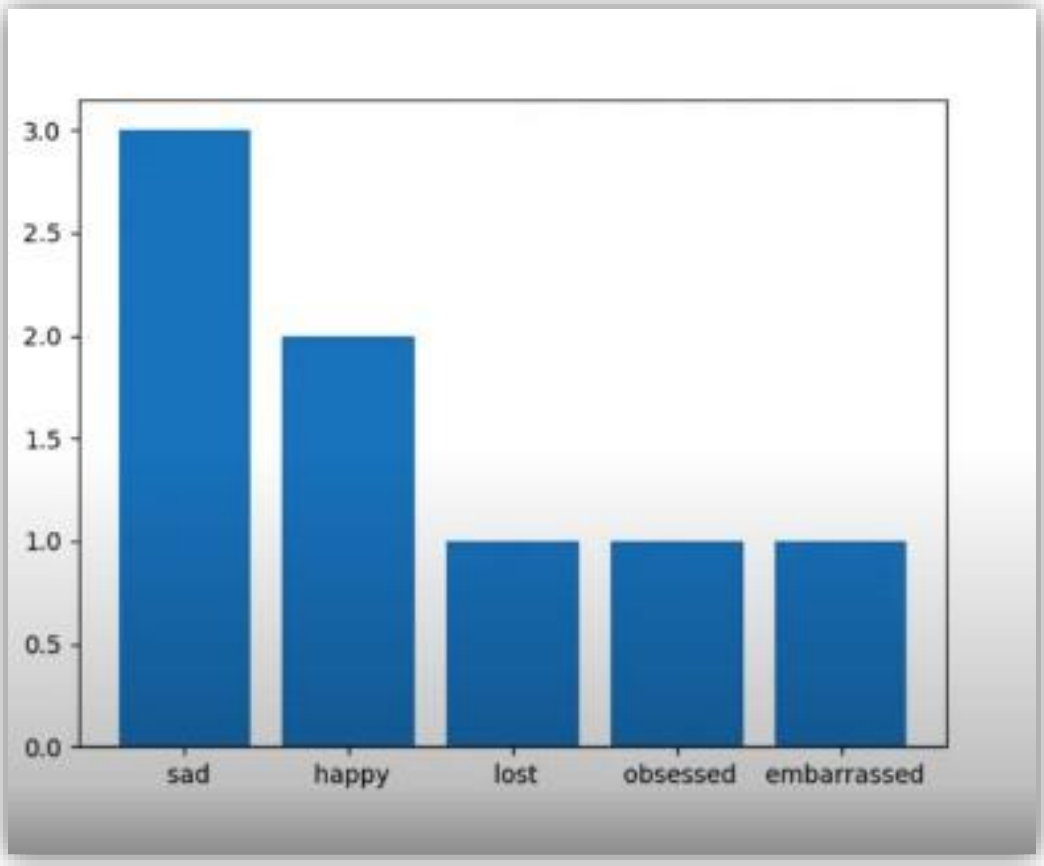
GENERATING INSIGHTS :

Generating insights for sentiment analysis involves extracting valuable information from text data to understand the opinions, emotions, or attitudes expressed within the text. Here's a step-by-step guide on how to generate insights for sentiment analysis:

- Data collection
- Preprocessing
- Feature Extraction
- Sentiment labelling
- Model Selection
- Model Training and Testing
- Interpretation and Visualization
- Fine-tuning and Improvement
- Deployment

EMOTIONS ANALYZED:

Counter ({ ‘SAD': 3,'HAPPY': 2, 'LOST': 1, 'OBSESSED': 1, 'EMBARRASSED': 1 })



SENTIMENT EMOTION ANALYSIS:

IMPORTING LIBRARIES:

IN [1]:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import nltk
```

READING THE DATA:

https://github.com/attreyabhatt/Sentiment-Analysis/blob/master/read.txt

CLEANING TEXT FOR NLP:

IN [2]:

```
import  string
```

```
Text = open (‘read.txt’,
encoding=’utf8’).read()

Lower_case=text.lower()

Cleaned_text=lower_case.translate
(str.maketrans (‘ ‘,’ ‘,string.punctuation))

Print (cleaned_text)
```

Sample output :

TOKENIZED WORDS AND STOP WORDS:

IN [3]:

```
Tokenised_words=cleaned_text.split()

Print (Tokenised_words)
```

Sample output :

IN [4]:

Stop_words =[“I”, ”me”, ”my”, ”myself”, ”we”, “our”, “ours”, “ourselves”, “you”, “a”, “your”, “yours”, “yourself”, “yourselves”, “he”, “him”, “his”, “himself”, “she”, “her”, “hers”, “herself”, “it”, “its”, “itself”, “they”, “them”, “their”, “theirs”, “themselves”, “what”, “which”, “who”, “whom”, “this”, “that”, “there”, “those”, “am”, “is”, “are”, “was”, “were”, “be”, “been”, “being”, “do”, “has”, “have”, “having”, “had”, “does”, “did”, “doing”, “an”, “the”, “and”, “if”, “or”, “but”, “because”, “until”, “while”, “as”, “of”, “at”, “by”, “for”, “with”, “about”, “against”, “between”, “through”, “into”,

“during”, “before”, “after”, “above”, “to”, “before”, “from”, “up”, “down”, “in”, “out”, “on”, “off”, “over”, “under”, “again”, “them”, “further”, “once”, “here”, “when”, “where”, “why”, “how”, “all”, “any”, “few”, “both”, “each”, “more”, “most”, “other”, “some”, “such”, “no”, “nor”, “not”, “own”, “only”, “same”, “so”, “then”, “too”, “very”, “can”, “will”, “just”, “shall”, “may”, “could”, “would”, “should”, “might”, “now”]

REMOVING STOP WORDS FROM THE TOKENISED WORDS LIST:

In [5]:

```
Final_words= [ ] for word in
    tokenized_words:
        if word not in stop_words:
```

Final_words.append(word)

Print (Final_words)

Sample output:

```
C:\Users\lenovo\PycharmProjects\SentimentAnalysis\venv\Scripts\python.exe
['i', 'love', 'python']
['love', 'python']
```

NLP EMOTION ALGORITHM:

Step 1:

Check if the word in the final word list is also present in emotion.txt

➔ Open the emotion file.

➔ Loop through each line and clear it.

➔ Extract the word and emotion using split.

Step 2:

If the word is present -> Add the emotion to emotion_list

Step 3:

Finally count each emotion in the emotion list.

Emotion of text:

https://github.com/attreyabhatt/Sentimen t-Analysis/blob/master/emotions.txt

In [6]:

```
emotion_list = [] with open
('emotions.txt', 'r' ) as file:
    for line in file:
        clear_line= line.replace("\n", ' ')
        .replace(" , " , ' ').replace(" ' ", ' ')
        .strip() word,emotion =
        clear_line.split(' : ')
```

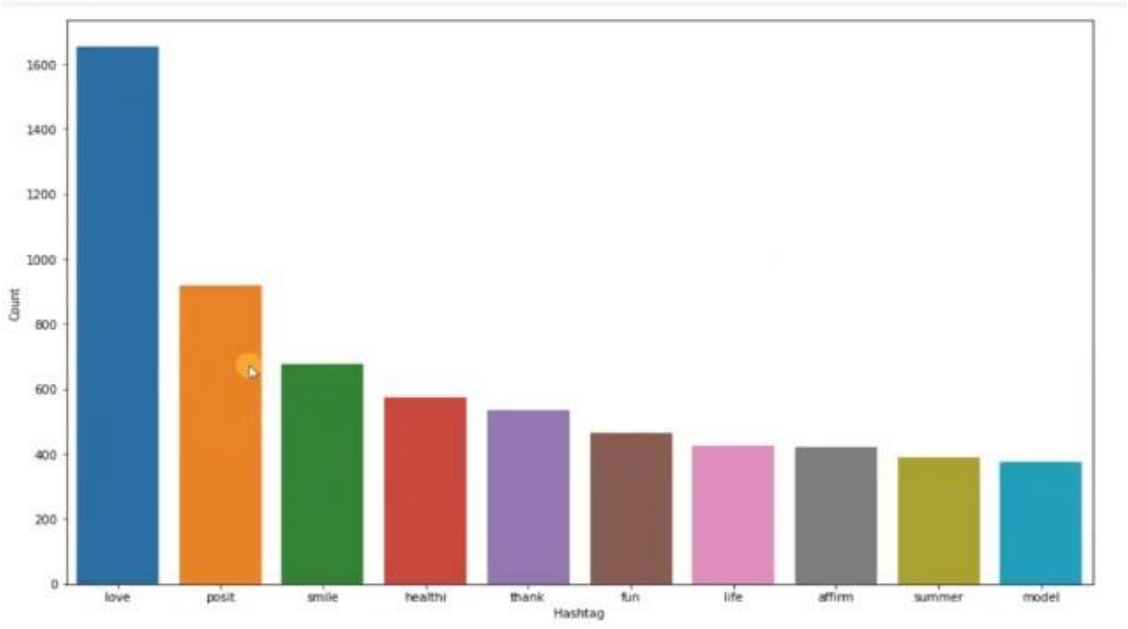
```
            if word in final_words:
                emotion_list.append(emotion)
print (emotion_list)
 w =Counter(emotion_list)
print (w)
```

sample output:

```
C:\Users\lenovo\PycharmProjects\SentimentAnalysis\venv\Scripts\python.exe C:/Users/lenovo/PycharmProjects
['i', 'feel', 'victimized', 'because', 'my', 'curent', 'boyfriend', 'cheated', 'on', 'me']
['feel', 'victimized', 'curent', 'boyfriend', 'cheated']

Process finished with exit code 0
```

CORRELATION MATRIX:

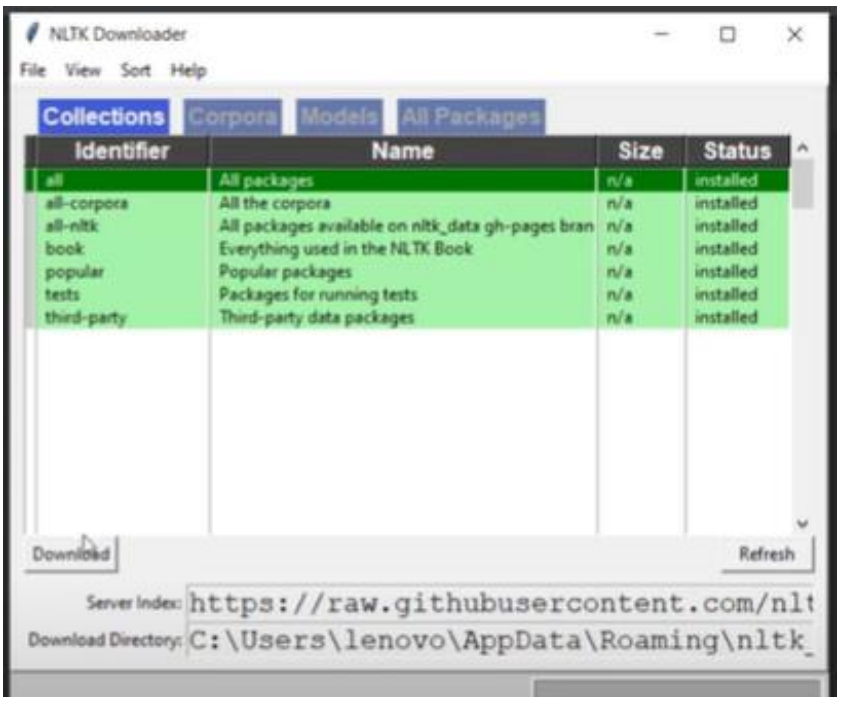


Large values in this matrix indicate serious collinearity between the variables involved. However, the nonexistence of extreme correlations does not imply lack of collinearity.

INSTALLING NLTK:

IN [8]:

```
import nltk
nltk.download()
```



DETECT SENTIMENT POSITIVE OR NEGATIVE:

IN [8]:

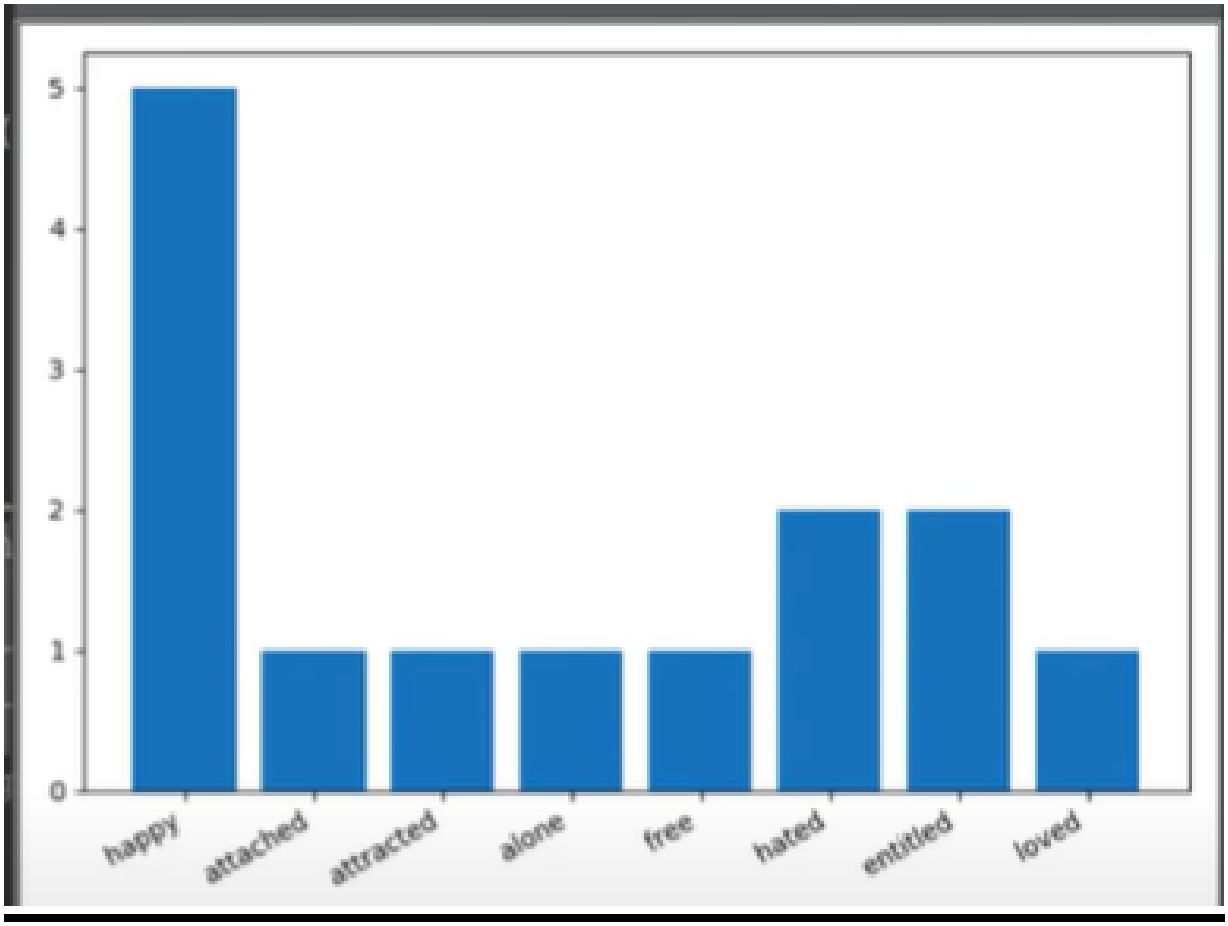
```
def sentiment_analyse(sentiment_text):
    score = SentimentIntensityAnalyser()
    .polarity_scores(sentiment_text)
neg = score['neg']
pos = score['pos']
if neg > pos:
    print ("Negative Sentiment")
elif pos > neg:
    print ("Positive Sentiment")
else:
    print ("Neutral Vibe")
```

Sample output:

```
[' happy', ' happy', ' attached', ' happy', ' attracted', ' alone', ' free', ' hated', ' happy', ' entitled', ' happy'
Counter({' happy': 5, ' hated': 2, ' entitled': 2, ' attached': 1, ' attracted': 1, ' alone': 1, ' free': 1, ' loved':
{'neg': 0.091, 'neu': 0.748, 'pos': 0.161, 'compound': 0.9996}
```

```
[' happy', ' happy', ' attached', ' happy', ' attracted', ' alone', ' free', ' hated', ' happy', ' entitled',
Counter({' happy': 5, ' hated': 2, ' entitled': 2, ' attached': 1, ' attracted': 1, ' alone': 1, ' free': 1,
Positive Sentiment
```

Graph:



CONCLUSION:

In conclusion, sentiment analysis is a crucial tool for modern marketing, enabling businesses to not only understand their customers' feelings but also to adapt and thrive in an everchanging marketplace. By harnessing the power of sentiment analysis, companies can foster stronger customer relationships, gain a competitive edge, and make more informed, data-driven marketing decisions.

* * * * * *