PROJECT 5 : SENTIMENT ANALYSIS FOR MARKETING

PRESENTED BY

M. BAIRAVI

S. DURGA

S. SAHANA

V.SUBASRI

D.VARDHANI

PHASE 5: PROJECT DOCUMENTATION AND SUBMISSION

INTRODUCTION:

Start building the Sentiment Analysis for Marketing to analysis customers sentiments for competitor products.

PROBLEM STATEMENT :

To develop an accurate sentiment analysis system that can effectively classify the sentiment expressed in textual data (such as reviews, social media posts, or customer feedback) into positive, negative, or neutral categories, aiming to assist business in understanding public opinion, customer satisfaction, and overall sentiment trends.

DESIGN THINKING PROCESS :

- Design thinking for sentiment analysis involves a human-centric approach to developing an effective sentiment analysis system for marketing purposes.
- It encompasses empathizing with the marketing team's needs, defining clear objectives for sentiment analysis, brainstorming innovative methods,

prototyping, testing with real data, implementing the solution, and iterating based on continuous feedback.

- This iterative process ensures a sentiment analysis system that accurately captures and interprets customer sentiments from various sources, aiding marketing strategies and decision-making.

Key principles:

- **User-Centric Approach:** Prioritize the needs and experiences of the users who will interact with the sentiment analysis tool.
- **Iterative Process:** Embrace an iterative and flexible approach, allowing room for continuous improvements based on user feedback.

- **Collaboration:** Involve cross-disciplinary teams to bring diverse perspectives and expertise to the solution.

DOMAIN USED:

In the context of sentiment analysis, the use of AI (Artificial Intelligence) spans across various domains and plays a crucial role in extracting meaningful insights from textual data to determine sentiment. AI techniques are applied within sentiment analysis to interpret, classify, and understand sentiments expressed within texts, be it in social media posts, customer reviews, surveys, or any other form of written communication. These AI methodologies contribute to the analysis and categorization of sentiment, offering a more nuanced

understanding of human emotions and opinions.

PHASES OF DEVELOPMENT :

PHASE 1:

In this phase, we clearly define about the tools and techniques used in this project and explain about how the sentimental analysis can testify the customers reviews or commands or feedbacks for online products and also define the clear problem statement to adapt to changing business needs and emerging technologies in the field of sentiment analysis.

PHASE 2 :

In phase 2 we explain about the preprocessing of text data that involves task like tokenization, removing stop words, etc...

Phase 3 :

In phase 3 we implement some training AI models and import some essential libraries like numpy, pandas, seaborn, matplotlib.

Phase 4 :

In phase 4 we employing NLP techniques, machine learning models, generating insights and import Natural Language Toolkit (NLTK).

In Phase 5:

We explained completely about the project details like the data preprocessing steps we used for sentiment analysis techniques and also describe the dataset we used and compile all coding files.

DESCRIBE DATASETS:

The dataset used for sentiment analysis can vary based on the context and the specific goals of the analysis. These datasets can vary in size with some containing thousands or millions of data points. They may also describe come pre-labelled with sentiments (Positive, Negative, Neutral) or require manual labelling for supervised learning tasks.

DATA PREPROCESSING:

Data preprocessing is a critical step in sentiment analysis as it involves cleaning and organizing the text data to make it suitable for analysis. Here are the key preprocessing steps for sentiment analysis:

1.Text Lowercasing:

Convert all text to lowercase to ensure consistency. This prevents the algorithm from treating words with different cases as different entities.

2.Tokenization:

Break the text into smaller units, usually words or phrases, known as tokens. This process helps in understanding the structure of the text.

3.Removing Punctuation:

Eliminate punctuation marks such as commas, periods, and exclamation points. Punctuation doesn't usually carry sentiment and can be removed to improve analysis accuracy.

4.Removing Stop Words:

Common words that don't carry significant sentiment or meaning (e.g., "and," "the," "is") can be removed. Libraries like NLTK (Natural Language Toolkit) provide predefined stop word lists.

5.Removing Special Characters :

Remove special characters, emojis that might not contribute to sentiment analysis.

6.Removing Numerical Values:

Remove the numerical values that might not be contribute to sentiment analysis.

7.Stemming or Lemmatization:

Reducing words to their root form. Stemming removes prefixes and suffixes (e.g., "running" becomes "run"), while lemmatization considers the word's meaning and converts it to its base or dictionary form (e.g., "better" becomes "good").

8. Handling Negations:

Special consideration for negations is important in sentiment analysis. For instance, "not happy" should be processed in a way that reflects the opposite sentiment of "happy."

9. Normalization:

Ensuring consistent representation, like converting abbreviations to their full form (e.g., "u" to "you").

10. Token Cleaning and Filtering:

Check for any remaining anomalies, irregularities, or noise in the data that might affect sentiment analysis.

It's important to note that the preprocessing steps may vary based on the specific nature of the text data and the analysis goals. Implementing these steps is crucial to ensure that the data is clean, consistent, and ready for sentiment analysis models to derive meaningful insights. The choice of techniques used for preprocessing can

significantly impact the accuracy of sentiment analysis models.

SENTIMENT ANALYSIS TECHNIQUES:

Sentiment analysis, also known as opinion mining, involves using natural language processing, machine learning, and text analysis techniques to identify, extract, quantify, and study subjective information from textual data. There are various techniques used for sentiment analysis:

1.Lexicon-Based Approaches:

These methods utilize sentiment lexicons or dictionaries containing words or phrases mapped to sentiment scores (positive, negative, neutral). The overall sentiment of a piece of text is determined by

aggregating the sentiment scores of individual words. Examples include AFINN, SentiWordNet, and VADER (Valence Aware Dictionary and Sentiment Reasoner).

2.Machine Learning-Based Approaches:

Supervised learning: This method involves training a model on labeled data (text data labeled with sentiment) to predict the sentiment of unseen text. Common algorithms include Support Vector Machines (SVM), Naive Bayes, Logistic Regression, and Random Forest.

Unsupervised learning: It involves techniques that don't rely on labeled data. Clustering algorithms (e.g., k-means), topic modeling (e.g., Latent Dirichlet Allocation - LDA), and neural network-

based models like Word2Vec and Doc2Vec can be used for this purpose.

3.Hybrid Approaches:

These methods combine both lexicon-based and machine learning-based techniques to improve accuracy and overcome limitations. For instance, using lexicon-based methods to inform or enhance the performance of machine learning models.

4.Aspect-Based Sentiment Analysis:

This technique aims to identify sentiments not just at the document or sentence level but also at a more granular level, focusing on specific aspects or entities

within the text. It helps in understanding the sentiment towards different features or topics.

5.Deep Learning-Based Approaches:

Neural network architectures like Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), and Transformer models (such as BERT, GPT, and their variants) have shown significant performance in sentiment analysis tasks due to their ability to capture complex relationships in text data.

6.Emotion Detection:

Goes beyond basic sentiment analysis to identify specific emotions expressed in text, such as joy, anger, sadness, etc.

7. Rule-Based Approaches:

These approaches use predefined rules or patterns to identify sentiment. These rules can range from simple heuristic-based rules to more complex linguistic or grammar-based rules.

Each technique has its strengths and weaknesses, and the choice of method depends on factors like available data, task requirements, domain specificity, and the level of accuracy needed. Often, a combination of these techniques or an ensemble approach might yield better results. Additionally, the choice of technique may vary based on whether the analysis is at document, sentence, or aspect level, as well as the complexity and nuances of the text data being analyzed.

IMPORTING ESSENTIAL LIBRARIES:

1.”STRING” Package:

In computer programming, the "string" package, often referred to as a "string library" or "string module," refers to a set of functions, methods, or tools designed to handle operations related to strings, which are sequences of characters.

Different programming languages provide their own string packages or libraries that offer various functionalities for manipulating, formatting, and managing string data.

In this analysis, String package is mainly used for to convert the string into lower case (i.e., String Case Conversion.

```
python

import string
```

2.”Matplotlib” package:

Matplotlib is a comprehensive data visualization library for Python. It allows you to create a wide range of plots and graphs, from simple bar charts to complex heatmaps and 3D plots. Matplotlib provides a flexible and powerful environment for creating visualizations.

Here's an overview of some of the key features and functionalities of Matplotlib:

- ✓ Pyplot
- ✓ Figure
- ✓ Axes

Before importing the ”matplotlib” package, we need to download it.

```
python -m pip install -U pip
python -m pip install -U matplotlib
```

Usage:

```python
import matplotlib.pyplot as plt
```

Matplotlib is highly extensible and can be used in conjunction with other libraries like Pandas for data manipulation and Seaborn for higher-level statistical graphics. Its flexibility and wide range of functionalities make it a popular choice for data visualization in Python.

3. "NLTK" Library:

NLTK (Natural Language Toolkit) is a powerful platform for building Python programs to work with human language data. It provides easy-to-use interfaces to

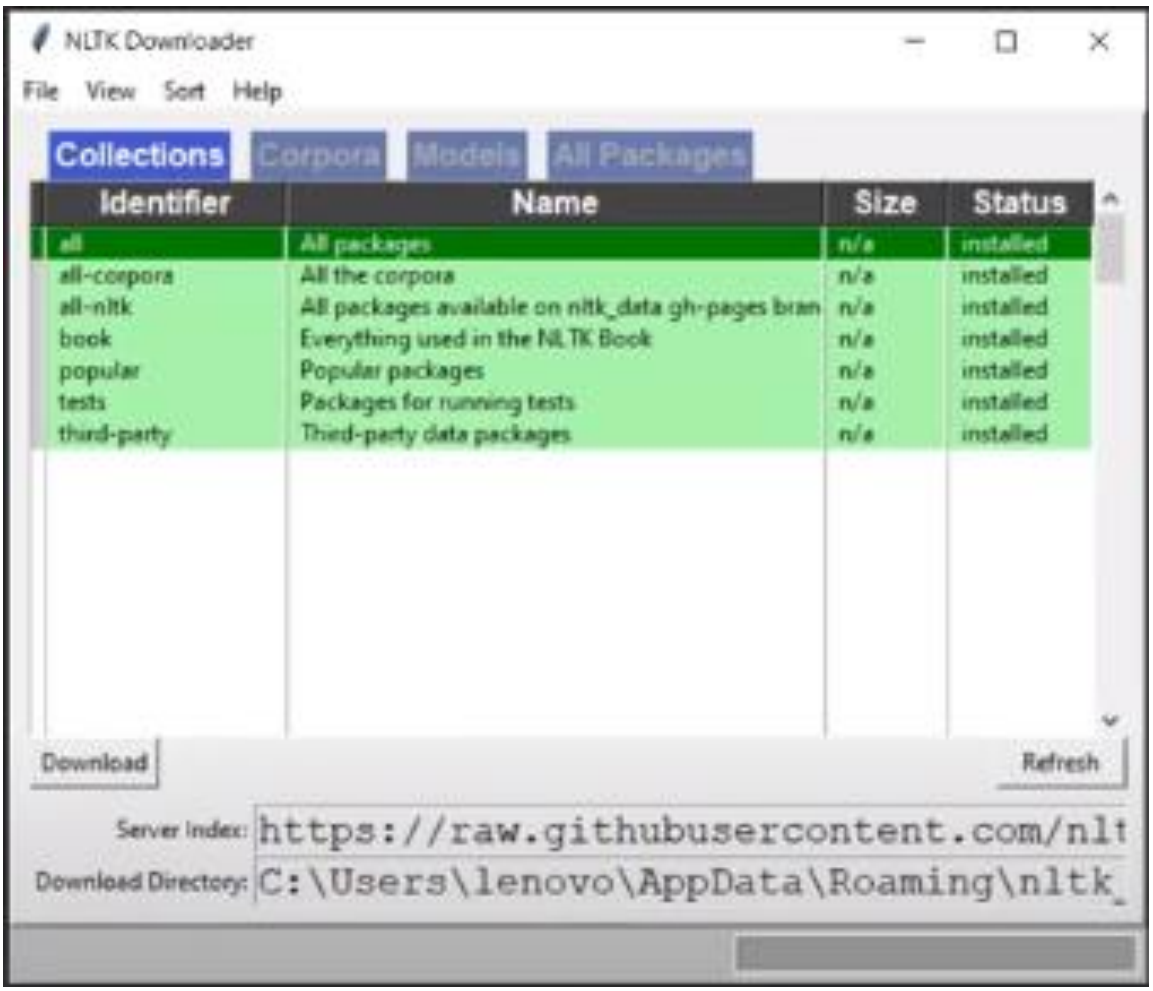
numerous corpora and lexical resources, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Installation and Resources:

install NLTK via ‘pip’ using:

```bash
pip install nltk
```

Once installed, you might need to download additional resources or corpora using **nltk.download()** to access specific functionalities.



Usage :

```
python

import nltk
from nltk.tokenize import word_tokenize
```

NLTK is widely used in academia, research, and industry for text analysis, natural

language processing, and machine learning tasks due to its extensive functionalities and broad range of text processing tools.

PLATFORM USED:

Visual Studio Code (VS Code) is a popular, free source-code editor developed by Microsoft. It's widely used by developers, data scientists, and researchers for various programming and analysis tasks, including sentiment analysis and natural language processing.

When performing sentiment analysis or any data analysis task within VS Code, We can utilize its features to write code, test and execute scripts, visualize results, and manage our project, all within a single, user-friendly environment. The extensibility and robust ecosystem of VS Code make it a popular

choice for many data-related tasks, including natural language processing and sentiment analysis.

CODINGS:

```
import string
from collections import Counter

import matplotlib.pyplot as plt
import nltk
nltk.download()
from nltk.corpus import stopwords
from nltk.sentiment.vader import
SentimentIntensityAnalyzer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
```

 # Read the text data

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

```
text = open('read.txt', encoding='utf-8').read()
lower_case = text.lower()
cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))
```

 # Using word_tokenize because it's faster than split()

```
tokenized_words = word_tokenize(cleaned_text, "english")
```

 # Removing Stop Words

```
final_words = []
for word in tokenized_words:
    if word not in stopwords.words('english'):
        final_words.append(word)
```

 # Lemmatization - From plural to single + Base form of a word (example better-> good)

```
lemma_words = []
for word in final_words:
    word = WordNetLemmatizer().lemmatize(word)
    lemma_words.append(word)
```

#Emotions for the text

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

```
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 lemma_words:
            emotion_list.append(emotion)

print(emotion_list)

w = Counter(emotion_list)
print(w)
```

#Analyse the data

(Classify the text whether it is positive or negative or neutral)

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

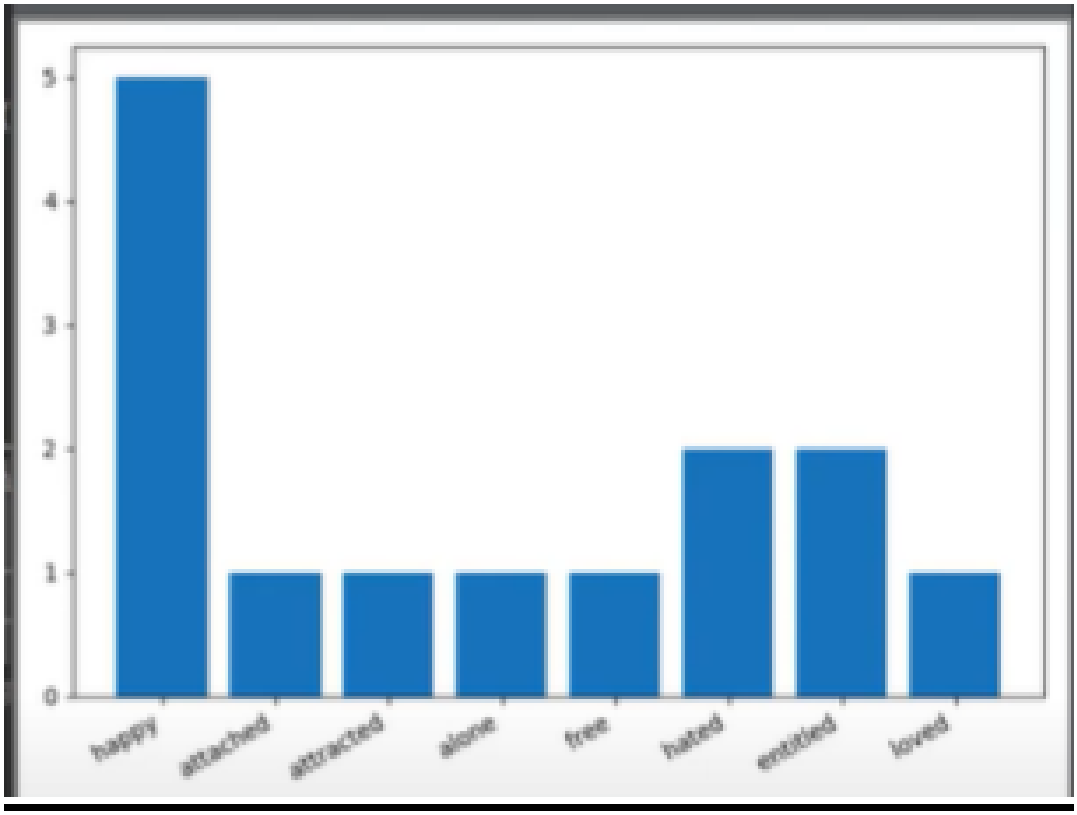
```
sentiment_analyse(cleaned_text)
```

#Display the graph

```
fig, ax1 = plt.subplots()
ax1.bar(w.keys(), w.values())
fig.autofmt_xdate()
plt.savefig('graph.png')
plt.show()
```

OUTPUT:

GRAPH:



In sentiment analysis, a "positive" output typically indicates that the sentiment expressed within the analyzed text is favorable, affirmative, or optimistic. When a sentiment analysis model processes text and classifies it as "positive," it suggests that the overall sentiment conveyed in the text is characterized by positive emotions or opinions.

CONCLUSION:

In conclusion, sentiment analysis in marketing is a powerful tool that assists in understanding, analyzing, and responding to the sentiments of consumers. By leveraging these insights, marketers can build stronger relationships with their audience, refine products and services, and craft more effective, customer-centric marketing strategies. The evolving landscape of sentiment analysis tools and techniques continues to offer promising opportunities for businesses to better connect with and serve their customers.

****