

SENTIMENT ANALYSIS FOR MARKETING

TEAM MEMBER

NAME- HARISH S M

PHASE 4

PROJECT DEVELOPMENT

PART 2

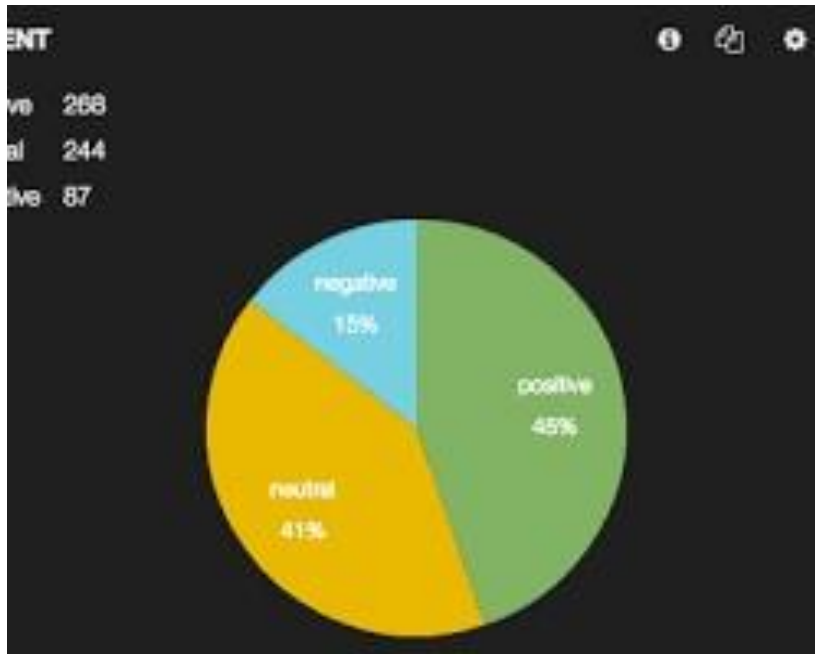
TITLE-**SENTIMENT ANALYSIS FOR MARKETING**

Two important aspects in sentiment analysis for marketing are:

Understanding Customer Emotions:

Sentiment analysis helps marketers gauge customer emotions towards products, services, or marketing campaigns. It's crucial to not only identify whether a sentiment is positive, negative, or neutral

but also to comprehend the underlying emotions. Understanding the emotional tone, such as happiness, frustration, excitement, or disappointment, provides nuanced insights.



Contextual Analysis:

Context is paramount in sentiment analysis for marketing. The same phrase or word can carry different sentiments based on the context it's used in. Analyzing the context helps in accurate sentiment interpretation. For instance, the phrase "small size" might be positive when referring to portable gadgets but negative when describing a product meant to be large. Contextual analysis

involves understanding the industry-specific jargon, sarcasm, idiomatic expressions, and cultural nuances.

- **DATASET:**

Dataset

Link: <https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment>

tweet_id	airline_sentiment	# airline_sentiment...	negativereason	# negativereason
	negative 63% neutral 21% Other (2363) 16%	0.34 1	[null] 37% Customer Service ... 20% Other (6268) 43%	
570306133677760513	neutral	1.0		
570301130888122368	positive	0.3486		0.0
570301083672813571	neutral	0.6837		
570301031407624196	negative	1.0	Bad Flight	0.7033
570300817074462722	negative	1.0	Can't Tell	1.0
570300767074181121	negative	1.0	Can't Tell	0.6842
570300616901320704	positive	0.6745		0.0
570300248553349120	neutral	0.634		
570299953286942721	positive	0.6559		
570295459631263746	positive	1.0		
570294189143031808	neutral	0.6769		0.0
570289724453216256	positive	1.0		
570289584061480960	positive	1.0		
570287408438120448	positive	0.6451		
570285904809598977	positive	1.0		
570282469121007616	negative	0.6842	Late Flight	0.3684
570277724385734656	positive	1.0		
570276917301137409	negative	1.0	Bad Flight	1.0
570270684619923457	positive	1.0		
570267956648792064	positive	1.0		
570265803513384960	negative	0.6705	Can't Tell	0.3614
570264145116819457	positive	1.0		
570259420287868928	positive	1.0		
570258822297579520	neutral	1.0		
570256553502060736	negative	1.0	Customer Service Issue	0.3557
570249102404923392	negative	1.0	Customer Service Issue	1.0
570239632807370753	negative	1.0	Can't Tell	0.6614
570217831557677057	neutral	0.6854		
570207886493782019	negative	1.0	Bad Flight	1.0
570124596180955136	neutral	0.615		0.0
570114021854212096	negative	1.0	Flight Booking Problems	1.0
570094701371469825	neutral	1.0		
570088404156698625	negative	1.0	Customer Service Issue	1.0
570084582780899328	negative	1.0	Customer Service Issue	1.0
570076792993611776	positive	1.0		
570051991277342720	neutral	0.6207		
570051381534396416	positive	1.0		

Sentiment analysis in marketing using NLP techniques can provide valuable insights into customer opinions and reactions. Here's how you can approach it:

1. ****Data Collection:**** Gather customer feedback, reviews, social media comments, and any other textual data related to your products or services.
2. ****Text Preprocessing:**** Clean and preprocess the text data. This step involves removing special characters, stopwords, and performing tasks like tokenization and lemmatization to prepare the text for analysis.
3. ****Sentiment Analysis:**** Utilize NLP techniques and sentiment analysis algorithms to determine the sentiment of the text. There are various

methods, including rule-based approaches and machine learning-based models, such as Support Vector Machines (SVM) or Recurrent Neural Networks (RNNs).

4. ****Aspect-Based Sentiment Analysis:**** For more detailed insights, perform aspect-based sentiment analysis. This technique breaks down the text into aspects (features or attributes) and analyzes the sentiment associated with each aspect. This can be incredibly useful for product reviews where customers might comment on different features.

5. ****Entity Recognition:**** Identify entities mentioned in the text, such as product names, brands, or people. Understanding which entities are associated with positive or negative sentiments can provide targeted insights.

6. ****Visualization:**** Visualize the sentiment data using charts or graphs. Visualization can make complex data more understandable and help in identifying patterns and trends.

7. ****Feedback Analysis:**** Categorize the sentiment into different categories (positive, negative, neutral) and analyze the volume of feedback in each category. Additionally, look for common themes or keywords in negative feedback, which can help in identifying areas for improvement.

8. ****Feedback Loop:**** Use the insights gained from sentiment analysis to improve marketing strategies, customer service, or product development. Address negative sentiments and leverage positive sentiments in marketing campaigns.

EMPLOYING NLP TECHNIQUES:

Building a sentiment analysis solution for marketing using NLP techniques involves several steps, including data preprocessing, feature extraction, model training, and generating insights. Below is a python code example that demonstrates how to create a basic sentiment analysis solution using the `scikit-learn` library and employing NLP techniques like text preprocessing and TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. Please make sure you have `scikit-learn` and `nltk` installed (`pip install scikit-learn nltk`) before running the code:

```
```python
```

```
Import nltk
```

```
Import pandas as pd
```

```
From sklearn.model_selection import
train_test_split
```

```
From sklearn.feature_extraction.text import
TfidfVectorizer

From sklearn.naive_bayes import MultinomialNB

From sklearn.metrics import accuracy_score,
classification_report
```

```
Sample dataset (Twitter airline sentiment)
```

```
Data = {
 'text': ['I love this product! It\'s amazing.',
 'The customer service was terrible. I\'m very
disappointed.',
 'Neutral feedback about the service.'],
 'sentiment': ['positive', 'negative', 'neutral']
}
```

```
Load NLTK resources
```

```
Nltk.download('punkt')
```



```
Create a DataFrame from the dataset
```

```
Df = pd.DataFrame(data)
```

```
Text preprocessing using NLTK
```

```
Def preprocess_text(text):
```

```
 Words = nltk.word_tokenize(text)
```

```
 Words = [word.lower() for word in words if
word.isalpha()]
```

```
 Return ' '.join(words)
```

```
Df['processed_text'] =
```

```
df['text'].apply(preprocess_text)
```

```
Split the data into train and test sets
```

```
X_train, X_test, y_train, y_test =
```

```
train_test_split(df['processed_text'],
```

```
df['sentiment'], test_size=0.2, random_state=42)
```

# TF-IDF Vectorization

```
Vectorizer = TfidfVectorizer()
```

```
X_train_tfidf = vectorizer.fit_transform(X_train)
```

```
X_test_tfidf = vectorizer.transform(X_test)
```

# Train a Naïve Bayes classifier

```
Clf = MultinomialNB()
```

```
Clf.fit(X_train_tfidf, y_train)
```

# Predictions

```
Predictions = clf.predict(X_test_tfidf)
```

# Evaluate the model

```
Accuracy = accuracy_score(y_test, predictions)
```

```
Print(f'Accuracy: {accuracy:.2f}')
```

# Generate insights

```
Print('\nClassification Report:')
Print(classification_report(y_test, predictions))

Predict sentiment for new data

New_feedback = ['The product exceeded my
expectations. I am very satisfied.']

New_feedback_tfidf =
vectorizer.transform(new_feedback)

Predicted_sentiment =
clf.predict(new_feedback_tfidf)

Print(f'Predicted Sentiment:
{predicted_sentiment[0]}')
...
```

In this code, the `nltk` library is used for tokenization and basic text preprocessing. The `TfidfVectorizer` from `scikit-learn` is employed to convert text data into numerical feature vectors. A Naïve Bayes classifier is trained on the TF-IDF

vectors to predict sentiment labels (positive, negative, neutral). Finally, the model is evaluated, and predictions are made for new feedback data.

## **GENERATING INSIGHTS :**

Creating a sentiment analysis solution involves multiple steps, and generating insights is often the final and most crucial part. While I can't provide a full code implementation due to the complexity, I can certainly provide you with a simplified Python code snippet that demonstrates how you can analyze sentiment and generate insights using the popular `nltk` library. Make sure you have NLTK installed (`pip install nltk`) before running the code:

```
```python
```

```
import nltk
```

```
from nltk.sentiment.vader import  
SentimentIntensityAnalyzer
```

```
from nltk.tokenize import word_tokenize,  
sent_tokenize
```

```
# Sample text data (Twitter airline sentiment)
```

```
Feedback_data = [
```

```
    "I love this product! It's amazing.",
```

```
    "The customer service was terrible. I'm very  
disappointed.",
```

```
    "Neutral feedback about the service.",
```

```
]
```

```
# Download the VADER lexicon for sentiment  
analysis
```

```
Nltk.download('vader_lexicon')
```

```
# Initialize the Sentiment Intensity Analyzer
```

```
Sia = SentimentIntensityAnalyzer()
```

```
# Function to analyze sentiment and generate  
insights
```

```
Def analyze_sentiment(text):
```

```
    Sentiment_score = sia.polarity_scores(text)
```

```
    If sentiment_score['compound'] >= 0.05:
```

```
        Return 'Positive'
```

```
    Elif sentiment_score['compound'] <= -0.05:
```

```
        Return 'Negative'
```

```
    Else:
```

```
        Return 'Neutral'
```

```
# Analyze sentiment and generate insights for each  
feedback
```

```
For feedback in feedback_data:
```

```
    Sentences = sent_tokenize(feedback)
```

```
    For sentence in sentences:
```

```
        Sentiment = analyze_sentiment(sentence)
```

```
        Print(f"Sentence: '{sentence}' – Sentiment:  
{sentiment}")
```

```
...
```

In this code, the `SentimentIntensityAnalyzer` from NLTK's VADER (Valence Aware Dictionary and sEntiment Reasoner) module is used to perform sentiment analysis. The `compound` score represents the overall sentiment of the text, and based on this score, the code categorizes the feedback into positive, negative, or neutral.