**SENTIMENT ANALYSIS FOR MARKETING**

**INTRODUCTION:**

Sentiment analysis is a natural language processing (NLP) technique that is used to identify and extract opinions and emotions from text data. It can be used to analyze a variety of different types of data, including social media posts, customer reviews, and product descriptions.

Sentiment analysis modules for marketing can be used to analyze customer feedback, social media mentions, and other forms of text data to understand how people feel about a brand, product, or service. This information can be used to improve marketing campaigns, customer service, and product development.

**DESIGN FRAMEWORK:**

**Data Collection:**

The first step in the process is data collection.A dataset of tweets, which is a common source of data for sentiment analysis due to the short, concise nature of tweets is used.

**Data Preprocessing:**

After collecting the data, several preprocessing steps to clean and prepare the data for analysis is performed. These steps include:

• **Lowercasing**: Conversion of all the text to lowercase to ensure that the same words in different cases are not considered as different words.

• **Removing Punctuation and Special Characters**: Removing all punctuation and special characters from the text as they do not contribute to sentiment.

• **Removing Stop Words**: Removing common words that do not carry much information (like "is", "the", "and", etc.). These words are called stop words.

• **Tokenization**:Breaking down the text into individual words or tokens.

• **Lemmatization**: Reducing the words to their base or root form (e.g., "running" to "run"). This helps in reducing the dimensionality of the data and grouping similar sentiments together.

**Feature Extraction:**

After preprocessing, convert the text data into numerical features that can be used by a machine learning algorithm.Use the TF-IDF (Term Frequency-Inverse Document Frequency) method for this. TF-IDF gives a weight to each word signifying its importance in the document and across a corpus of documents.

**Model Training:**

Random Forest Classifier for sentiment analysis is used. Random Forest is a versatile and widely used algorithm that works well for many tasks. It creates a set of decision trees from a randomly selected subset of the training set, which then aggregates votes from different decision trees to decide the final class of the test object.

**Model Evaluation:**

After training the model, its performance is evaluated using a confusion matrix and calculated metrics such as accuracy, precision, recall, and F1-score. These metrics gives us a quantitative measure of the model's performance.

**Insights & Interpretation:**

Finally, the results of the sentiment analysis is interpreted. This involves understanding the performance of the model, identifying any areas of improvement, and drawing insights from the model's predictions.

These innovation ideas can enhance digital marketing by harnessing the power of sentiment analysis to better understand and connect with target audiences

**1. Emotion-Driven Content Optimization**: Develop a sentiment analysis tool that suggests content changes based on the emotional tone of user comments and social media interactions. This can help digital marketers tailor their content to elicit desired emotions.

**2. Real-time Sentiment Monitoring**: Create a platform that provides real-time sentiment analysis of brand mentions across various digital channels. This tool could alert marketers to trends or potential PR issues as they emerge.

**3. Sentiment-Enhanced Chatbots:** Integrate sentiment analysis into chatbots to enable more empathetic and context-aware responses. This can improve user interactions and customer satisfaction.

**4. Sentiment Scoring for Product Development**: Use sentiment analysis to score and prioritize customer feedback and suggestions for new product or feature development.

**5. Customer Support Sentiment Analysis**: Implement sentiment analysis in customer support to identify frustrated customers in real-time and prioritize their cases for immediate resolution.

**6. Competitive Sentiment Analysis:** Provide tools to analyze the sentiment surrounding competitors, allowing marketers to identify weaknesses and opportunities for differentiation.

**7. Sentiment-Personalized Recommendations:** Develop recommendation engines that suggest products, content, or experiences tailored to a user's current emotional state.

**8. Sentiment Trends Forecasting:** Utilize historical sentiment data to predict future trends and customer sentiments, allowing for proactive marketing and product adjustments.

In [1]:

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, classification\_report

import matplotlib.pyplot as plt

import seaborn as sns

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

import re

import nltk

nltk.download('stopwords')

nltk.download('wordnet')

*# Load the dataset*

df = pd.read\_csv('/kaggle/input/twitter-airline-sentiment/Tweets.csv')

*# Display the first 5 rows of the dataframe*

df.head()

/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and <{np\_maxversion}"

[nltk\_data] Downloading package stopwords to /usr/share/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

[nltk\_data] Downloading package wordnet to /usr/share/nltk\_data...

[nltk\_data] Package wordnet is already up-to-date!

In [2]:

*# Drop unnecessary columns*

df = df[['airline\_sentiment', 'text']]

*# Display the first 5 rows of the dataframe after dropping unnecessary columns*

df.head()

In [3]:

*# Function to preprocess the text*

def preprocess\_text(text):

*# Remove punctuations and numbers*

text = re.sub('[^a-zA-Z]', ' ', text)

*# Single character removal*

text = re.sub(r'\s+[a-zA-Z]\s+', ' ', text)

*# Removing multiple spaces*

text = re.sub(r'\s+', ' ', text)

*# Converting to Lowercase*

text = text.lower()

*# Lemmatization*

*#text = text.split()*

*#lemmatizer = WordNetLemmatizer()*

*#text = [lemmatizer.lemmatize(word) for word in text if not word in set(stopwords.words('english'))]*

*#text = ' '.join(text)*

return text

*# Apply the preprocessing to the 'text' column*

df['text'] = df['text'].apply(preprocess\_text)

*# Display the first 5 rows of the dataframe after preprocessing*

df.head()

In [4]:

*# Splitting the data into training and testing sets*

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['airline\_sentiment'], test\_size=0.2, random\_state=42)

*# Feature Extraction*

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max\_features=2500, min\_df=7, max\_df=0.8)

X\_train = vectorizer.fit\_transform(X\_train).toarray()

X\_test = vectorizer.transform(X\_test).toarray()

*# Model Training*

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n\_estimators=1000, random\_state=0)

classifier.fit(X\_train, y\_train)