**SENTIMENT ANALYSIS FOR MARKETING**

***INTRODUCTION :***

Sentiment analysis is a natural language processing (NLP) technique used to identify and extract opinions and emotions from text. It is a powerful tool that can be used to understand customer sentiment towards products, brands, and marketing campaigns.

* To train a sentiment analysis model for marketing, you will need a dataset of labeled text data. This data should consist of text samples that have been labeled as positive, negative, or neutral.
* Once you have trained a sentiment analysis model, it is important to evaluate its performance. This will help you to determine how accurate and reliable your model .

***DATASET :***

This includes twitter Us airline sentiment dataset which is taken from the kaggle.

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

***MACHINE LEARNING ALGORITHM :***

There are a variety of different machine learning algorithms that can be used for sentiment analysis in marketing. Some of the most popular algorithms are ;

* Naïve Bayes
* Support Vector Machines (SVMs)
* Recurrent Neural Networks (RNNs)

These approaches use machine learning algorithms to train a model to identify sentiment in text. The model is trained on a dataset of labeled data, where each piece of text is labeled with its sentiment (e.g., positive, negative, or neutral). Once the model is trained, it can be used to predict the sentiment of new pieces of text.

***TRAINING THE MODEL :***

This involves feeding your labeled training data to the machine learning algorithm. The algorithm will learn to classify new text samples as positive, negative, or neutral based on the patterns it learns from the training data.

**PROGRAM**

# Train Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.30, random\_state = 0)

# Training using three algorithms, let’s see which will give us better result

Model1=LogisticRegression()

Model2=BernoulliNB()

Model3=LinearSVC()

Model=[model1, model2, model3]

I = 0

For algo in model:

I += 1

Print(“M-O-D-E-L :”,i)

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

Y\_pred=algo.predict(X\_test)

# Checking the accuracy

Print(“Confusion matrix : \n”,confusion\_matrix(y\_pred,y\_test))

Print(“Accuracy score : “,accuracy\_score(y\_pred,y\_test))

Print(“Classification Report : \n”,classification\_report(y\_pred,y\_test))

Print(“-----------------------------------------------------------\n”)

**OUTPUT :**

M-O-D-E-L : 1

Confusion matrix :

[[2694 532 285]

[ 77 351 81]

[ 17 36 319]]

Accuracy score : 0.7659380692167578

Classification Report :

Precision recall f1-score support

Negative 0.97 0.77 0.86 3511

Neutral 0.38 0.69 0.49 509

Positive 0.47 0.86 0.60 372

Accuracy 0.77 4392

Macro avg 0.60 0.77 0.65 4392

Weighted avg 0.86 0.77 0.79 4392

M-O-D-E-L : 2

Confusion matrix :

[[2780 850 670]

[ 8 69 13]

[ 0 0 2]]

Accuracy score : 0.6491347905282332

Classification Report :

Precision recall f1-score support

Negative 1.00 0.65 0.78 4300

Neutral 0.08 0.77 0.14 90

Positive 0.00 1.00 0.01 2

Accuracy 0.65 4392

Macro avg 0.36 0.80 0.31 4392

Weighted avg 0.98 0.65 0.77 4392

M-O-D-E-L : 3

Confusion matrix :

[[2620 428 197]

[ 135 426 100]

[ 33 65 388]]

Accuracy score : 0.7818761384335154

Classification Report :

Precision recall f1-score support

Negative 0.94 0.81 0.87 3245

Neutral 0.46 0.64 0.54 661

Positive 0.57 0.80 0.66 486

Accuracy 0.78 4392

Macro avg 0.66 0.75 0.69 4392

Weighted avg 0.83 0.78 0.80 4392

***EVALUATION :***

Once your model is trained, it is important to evaluate its performance. This will help you to determine how accurate and reliable your model is. You can evaluate your model by feeding it a held-out test set of labeled data and comparing its predictions to the known labels.

**PROGRAM :**

Import pandas as pd

From sklearn.feature\_extraction.text import TfidfVectorizer

From sklearn.linear\_model import SGDClassifier

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

From sklearn.metrics import accuracy\_score

# Load the dataset

Df = pd.read\_csv(‘Tweets.csv’)

# Preprocess the data

# Remove punctuation and stop words

# Convert the text to lowercase

Df[‘tweet\_text’] = df[‘tweet\_text’].apply(lambda x: x.lower().strip())

Df[‘tweet\_text’] = df[‘tweet\_text’].apply(lambda x: x.translate(str.maketrans(‘’, ‘’, string.punctuation)))

Df[‘tweet\_text’] = df[‘tweet\_text’].apply(lambda x: ‘ ‘.join([word for word in x.split() if word not in stopwords.words(‘english’)]))

# Create the TF-IDF vectorizer

Vectorizer = TfidfVectorizer()

# Transform the text into TF-IDF vectors

X = vectorizer.fit\_transform(df[‘tweet\_text’])

# Split the data into training and test sets

Y = df[‘airline\_sentiment’]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

# Create the SVM classifier

Clf = SGDClassifier(loss=’hinge’, alpha=0.0001, max\_iter=1000)

# Train the model

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

# Make predictions on the test set

Y\_pred = clf.predict(X\_test)

# Evaluate the model

Accuracy = accuracy\_score(y\_test, y\_pred) \* 100

Print(‘Accuracy:’, accuracy, ‘%’)

# Use the trained model to analyze tweets from customers about your airline

# For example, you can collect tweets that mention your airline’s name or hashtags

# Then, you can use the trained model to classify the tweets as positive, negative, or neutral

# This information can be used to improve your airline’s marketing strategy.

***CONCLUSION :***

summary of the training and evaluation process for sentiment analysis in marketing:

* Collect a dataset of labeled text data.
* Preprocess your data
* Choose a machine learning algorithm
* Train your model.
* Evaluate your model.
* Deploy your model.

By following these steps, you can train and deploy a sentiment analysis model that will help you to better understand customer sentiment and improve your marketing efforts.This will help you to determine how accurate and reliable your model is. If you are satisfied with the performance of your model, you can deploy it to production. This means making it available to users so that they can use it to classify new text samples.