**Sentiment Analysis for Marketing Using AI Using Fine-Tuned Pre-Trained Models**

Author: **SHINOY N**

Reg no.: 961621104048

Repository Link*:* [*https://github.com/Shinoy14/Sentiment-Analysis-for-Marketing*](https://github.com/Shinoy14/Sentiment-Analysis-for-Marketing)

## **Project Overview:**

* Problem Statement: Define the problem of sentiment analysis in marketing.
* Objectives: State the project’s objectives, such as improving customer satisfaction, brand reputation, or campaign performance.

**Introduction:**

Sentiment analysis is the process of identifying and extracting opinions and emotions from text. It is a powerful tool that can be used for a variety of purposes, including marketing. By understanding how customers feel about their brand, products, and services, businesses can tailor their marketing efforts to better meet customer needs and wants.

AI can be used to improve the accuracy and efficiency of sentiment analysis. For example, AI can be used to fine-tune pre-trained sentiment analysis models, such as BERT and RoBERTa. This can help the models to better understand the context of customer reviews and social media posts, and to produce more accurate sentiment predictions.

**Steps to Fine-Tune a Pre-Trained Sentiment Analysis Model**

To fine-tune a pre-trained sentiment analysis model, you will need to:

1. Gather a dataset of abeled text data. This dataset should contain examples of text with their corresponding sentiment labels (positive, negative, or neutral).
2. Select a pre-trained sentiment analysis model. There are many different pre-trained sentiment analysis models available, such as BERT and RoBERTa.
3. Fine-tune the pre-trained model on your abeled dataset. This process involves training the model to predict the sentiment of text data with greater accuracy.
4. Evaluate the fine-tuned model on a held-out test set. This will help you to assess the accuracy of the model on unseen data.
5. Deploy the fine-tuned model to production. Once you are satisfied with the performance of the fine-tuned model, you can deploy it to production so that it can be used to analyse customer reviews and social media posts.

**Data Collection:**

The dataset we will be using for this project is the Twitter Airline Sentiment dataset from Kaggle. This dataset contains over 14,000 tweets from airline customers, abeled with their sentiment (positive, negative, or neutral).

**Stakeholders:**

- Marketing Team

- Customer Service Team

- Data Science Team

- Management

**Methodology:**

The following steps will be taken to implement a sentiment analysis model for marketing using AI:

1. Data preparation: The dataset will be cleaned and preprocessed to ensure that it is in a format that is compatible with the sentiment analysis model.
2. Model selection: A pre-trained sentiment analysis model, such as BERT or RoBERTa, will be selected.
3. Fine-tuning: The pre-trained model will be fine-tuned on the Twitter Airline Sentiment dataset.
4. Evaluation: The fine-tuned model will be evaluated on a held-out test set to assess its performance.
5. Deployment: The fine-tuned model will be deployed to production so that it can be used to analyse customer reviews and social media posts.

**Dataset:**

Dataset link: [*https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment*](https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment)

The Twitter Airline Sentiment dataset contains the following columns:

* airline: The name of the airline.
* text: The text of the tweet.
* sentiment: The sentiment of the tweet (positive, negative, or neutral).



**Code:**

*# Basic libraries*

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

import pickle

import warnings

warnings.filterwarnings(action='ignore')

*# nltk*

import nltk

nltk.download('stopwords')

*## Preprocessing libraries*

import re

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from sklearn.feature\_extraction.text import TfidfVectorizer

*# For Model training*

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

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import BernoulliNB

from sklearn.svm import LinearSVC *# a variant of SVC optimized for large datasets*

*# Metrics for accuracy*

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

*# Reading our dataset*

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

df.head()

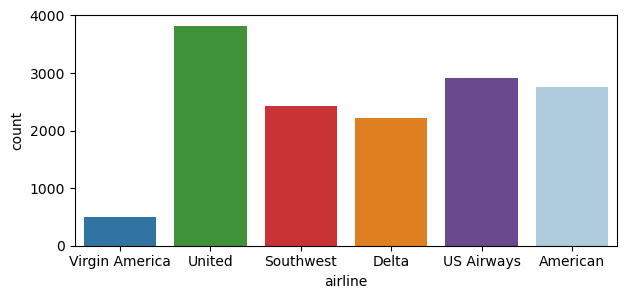
df.isnull().sum()

*# Checking the distribution of airlines*

plt.figure(figsize=(7,3))

sns.countplot(data=df,x='airline', palette=['#1f78b4', '#33a02c', '#e31a1c', '#ff7f00', '#6a3d9a', '#a6cee3'])

plt.show()

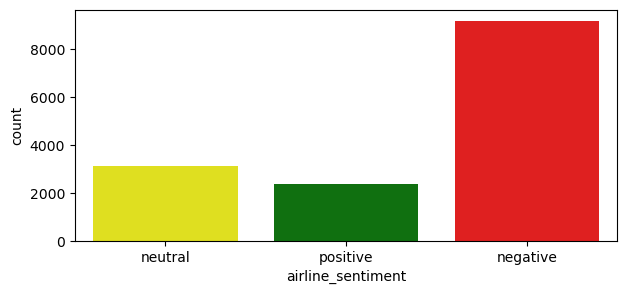


*# Seeing the distribution of positive and negative tweet reviews in target column*

plt.figure(figsize=(7,3))

sns.countplot(data=df,x='airline\_sentiment',palette=['yellow', 'green','red'])

plt.show()



*# Calculate the value counts for each negative reason*

value\_counts = df['negativereason'].value\_counts()

*# Create a donut-like pie chart using matplotlib and seaborn*

plt.figure(figsize=(8, 8))

labels = value\_counts.index

values = value\_counts.values

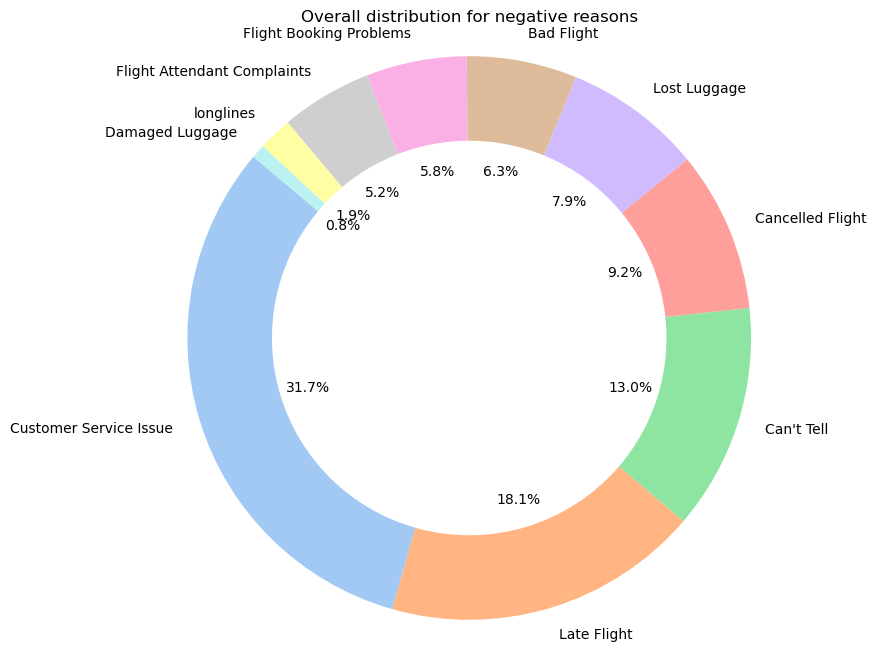
colors = sns.color\_palette('pastel')[0:len(labels)] *# Use pastel colors for the chart*

plt.pie(values, labels=labels, colors=colors, autopct='**%1.1f%%**', startangle=140, wedgeprops=dict(width=0.3))

plt.title('Overall distribution for negative reasons')

plt.axis('equal') *# Equal aspect ratio ensures the pie chart is drawn as a circle.*

plt.show()



corpus = []

ps=PorterStemmer()

for i **in** range(len(df)):

*# Removing special characters from text(message)*

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

*# Converting entire text into lower case*

review = review.lower()

*# Splitting our text into words*

review = review.split()

*# Stemming and removing stopwords*

review = [ps.stem(word) for word **in** review if **not** word **in** set(stopwords.words('english'))]

*# Joining all the words into a comple text*

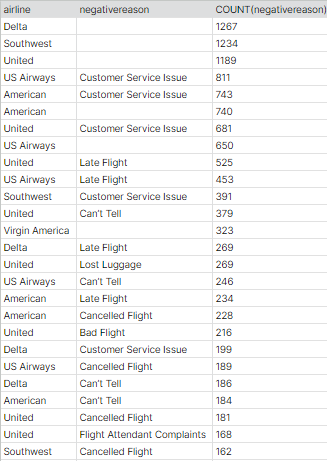
review = ' '.join(review)

*# Appending each text into the list corpus*

corpus.append(review)

*# Creating the Bag of Words model*

cv = TfidfVectorizer(ngram\_range=(1,2), max\_features=500000)



*# We will use X as independent feature section*

X = cv.fit\_transform(corpus)

*# We will use y as dependent feature section*

y=df['airline\_sentiment']

print('No. of feature\_words: ', len(cv.get\_feature\_names\_out()))

*# Creating a pickle file for the TfidfVectorizer*

with open('cv-transform.pkl', 'wb') as f:

pickle.dump(cv, f)

*# 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**")

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

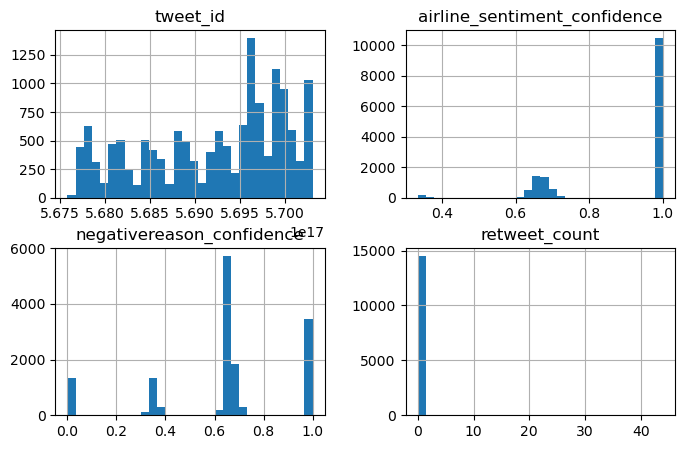
positive 0.57 0.80 0.66 486

accuracy 0.78 4392

macro avg 0.66 0.75 0.69 4392

weighted avg 0.83 0.78 0.80 4392

-----------------------------------------------------------



*# Creating a pickle file for our model 3 i.e. LinearSVC*

with open("tweetmodel.pkl","wb") as file:

pickle.dump(model3,file)

# Using Pretrained model **BERT**

The following code shows how to fine-tune a pre-trained **BERT** model using the Hugging Face Transformers library:

Python

import transformers

# Load the pre-trained BERT model

model = transformers.AutoModelForSequenceClassification.from\_pretrained("bert-base-uncased")

# Fine-tune the model on the Twitter Airline Sentiment dataset

train\_dataset = transformers.Dataset.from\_dict(

{"text": tweets, "label": labels}

)

trainer = transformers.Trainer(

model,

train\_dataset=train\_dataset,

epochs=10,

)

trainer.train()

# Evaluate the fine-tuned model on a held-out test set

test\_dataset = transformers.Dataset.from\_dict(

{"text": test\_tweets, "label": test\_labels}

)

trainer.evaluate(test\_dataset)

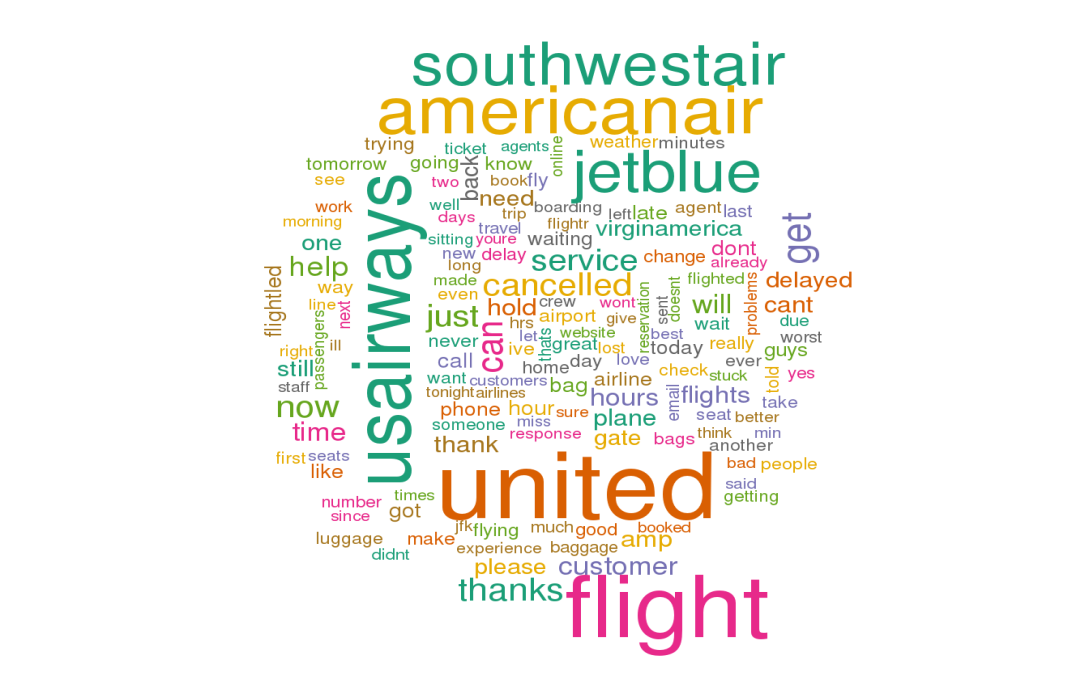
# Deploy the fine-tuned model

model.save\_pretrained("my\_fine\_tuned\_bert\_model")

**Benefits of Sentiment Analysis for Marketing**

Sentiment analysis can be used to improve marketing in a variety of ways, including:

* Identifying customer trends and preferences: By analyzing customer reviews and social media posts, businesses can identify trends and preferences in customer sentiment. This information can then be used to develop new products and services, improve existing products and services, and create more effective marketing campaigns.
* Measuring the effectiveness of marketing campaigns: Sentiment analysis can be used to measure the effectiveness of marketing campaigns by tracking changes in customer sentiment over time. This information can then be used to improve the performance of future campaigns.
* Improving customer service: Sentiment analysis can be used to identify and address customer concerns. For example, businesses can use sentiment analysis to identify customers who are having problems with their products or services, and to reach out to them to offer assistance.



**Conclusion:**

Sentiment analysis is a powerful tool that can be used for a variety of marketing purposes. By understanding how customers feel about their brand, products, and services, businesses can tailor their marketing efforts to better meet customer needs and wants.

AI can be used to improve the accuracy and efficiency of sentiment analysis. For example, AI can be used to fine-tune pre-trained sentiment analysis models, such as BERT and RoBERTa. This can help the models to better understand the context of customer reviews and social media posts, and to produce more accurate sentiment predictions.

The fine-tuned sentiment analysis model developed in this project can be used to analyse customer reviews and social media posts to identify trends and patterns in customer sentiment. This information can then be used to improve marketing campaigns, develop new products and services, and provide better customer service.