**SENTIMENT ANALYSIS FOR MARKETING**

**BATCH MEMBER**

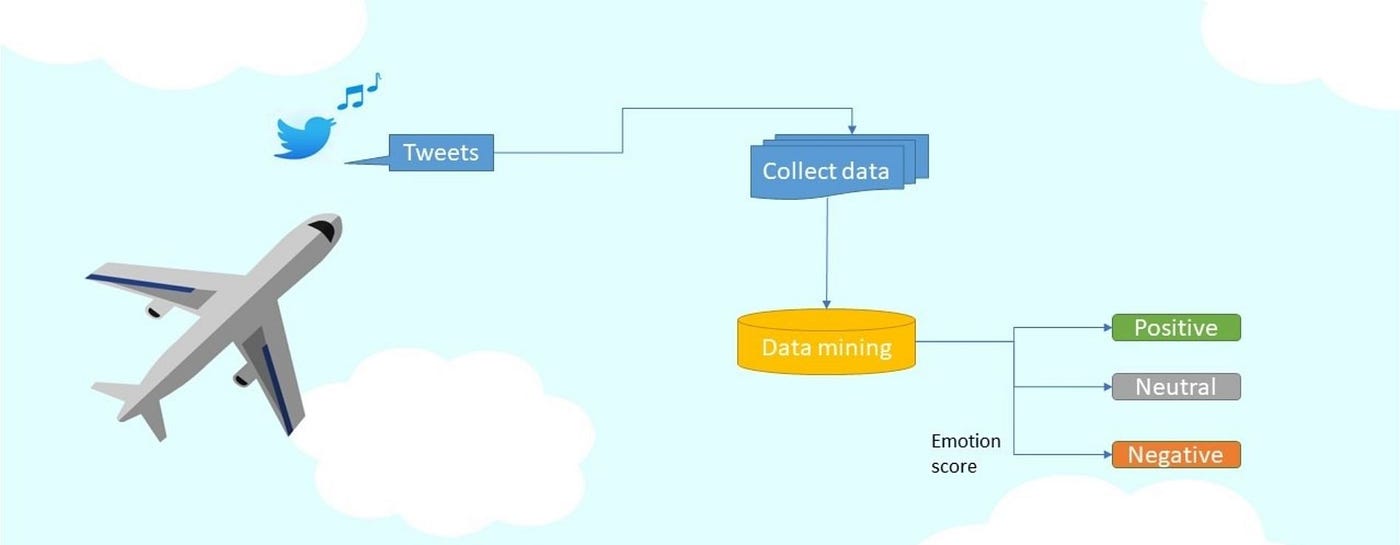
**815621104028: PUVIYARASI EZHILARASAN**

**Phase 3 submission document**

**Project Title**: Sentiment Analysis for Marketing

**Phase 3: Development Part 1**

**Topic:** Start building the sentiment analysis solution by loading dataset and preprocessing the data.

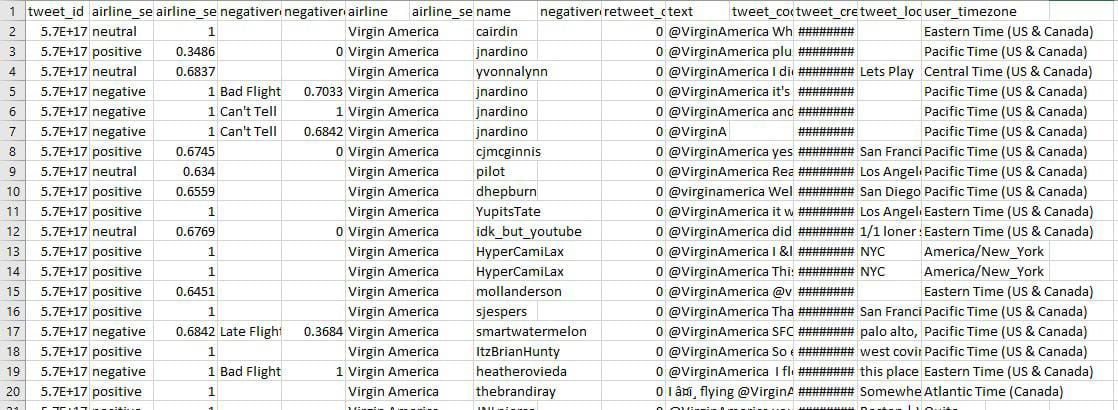


Sentiment Analysis for Marketing

**Introduction:**

* Sentiment analysis in marketing is a powerful application of artificial intelligence (AI) that involves using natural language processing (NLP) techniques to determine the sentiment or emotional tone expressed in text data, such as customer reviews, social media posts, and other user-generated content.
* This analysis can help businesses better understand how their customers feel about their products, services, or brand, and it has several important implications for marketing.
* Monitoring sentiment in online discussions helps businesses manage their online reputation. By identifying negative sentiment early, companies can address customer concerns and resolve issues promptly, mitigating potential damage to their brand.
* This introduction will guide you through the initial steps of the process. We'll explore how to import essential libraries, load the marketing dataset, and perform critical preprocessing steps. Data preprocessing is crucial as it helps clean, format, and prepare the data for further analysis. This includes handling missing values, encoding categorical variables, and ensuring that the data is appropriately scaled.

**Given data set:**



**Necessary step to follow:**

**1.Import Libraries:**

Start by importing the necessary libraries:

**Program:**

import pandas as pd

import numpy as np

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

from sklearn. preprocessing import StandardScaler

**2.Load the Dataset:**

Load your dataset into a Pandas Data Frame. You can typically find sentiment analysis datasets in CSV format, but you can adapt this code to other formats as needed.

**Program:**

df = pd.read\_csv(' E:\USA\_Analysing.csv ')

Pd.read()

**3. Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

**Program:**

# Check for missing values

print(df.isnull().sum())

# Explore statistics print (df.describe())

# Visualize the data (e.g., histograms, scatter plots, etc.)

**Importance of loading and processing dataset:**

* Datasets provide the raw material for training sentiment analysis models. Without appropriate data, you cannot build an effective model.
* Loading and processing data allow you to convert text data into a format that can be used by machine learning algorithms, typically numerical representations like word embeddings or tokenized text.

**Challenges involved in loading and preprocessing a sentiment analysis dataset**;

* **Handling Imbalanced Data:**

Sentiment datasets may have an imbalanced distribution of sentiment classes (e.g., more neutral than positive or negative), which can affect model performance.

* **Encoding Labels:**

Categorizing sentiments: Converting textual labels (e.g., "positive," "negative," "neutral") into numerical values or one-hot encoding for model compatibility.

* **Handling Imbalanced Data:**

Sentiment datasets may have an imbalanced distribution of sentiment classes (e.g., more neutral than positive or negative), which can affect model performance.

The process of labeling data for sentiment analysis can be subjective, leading to labeling errors or inconsistencies.

**How to overcome the challenges of loading and preprocessing a twitter us airline sentiment dataset:**

There are a number of things that can be done to overcome the challenges of loading and preprocessing a sentiment analysis dataset, including:

* **Handling Sarcasm and Irony:**

Twitter is known for its use of sarcasm and irony. Advanced sentiment analysis models should be capable of detecting these nuances. Investing in NLP models that can handle such complexities can be beneficial.

* **Customer Support and Feedback:**

Set up automated alerts for negative sentiment tweets to address customer complaints and issues promptly.

Engage with customers on Twitter to address their concerns, provide assistance, and resolve problems.

**1.Loading the dataset:**

* Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.
* The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks.

**a.Identify the dataset:**

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

**b.Load the dataset:**

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-infunction in the machine learning library, or it may involve writing your own code.

**c.Preprocess the dataset:**

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Here, how to load a dataset using machine learning in Python

**Program:**

*# Basic Operation*

import pandas as pd

import numpy as np

*# Text Preprocessing & Cleaning*

from sklearn.feature\_extraction.text import TfidfVectorizer

from nltk.corpus import stopwords

import re

from sklearn.model\_selection import train\_test\_split *# Split Data*

from imblearn.over\_sampling import SMOTE *# Handling Imbalanced*

*# Model Building*

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import MultinomialNB

from xgboost import XGBClassifier

from sklearn.svm import SVC

from sklearn.metrics import classification\_report , confusion\_matrix , accuracy\_score *# Performance Metrics*

*# Data Visualization*

import matplotlib.pyplot as plt

from wordcloud import WordCloud

from termcolor import cprint

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

%matplotlib inline

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

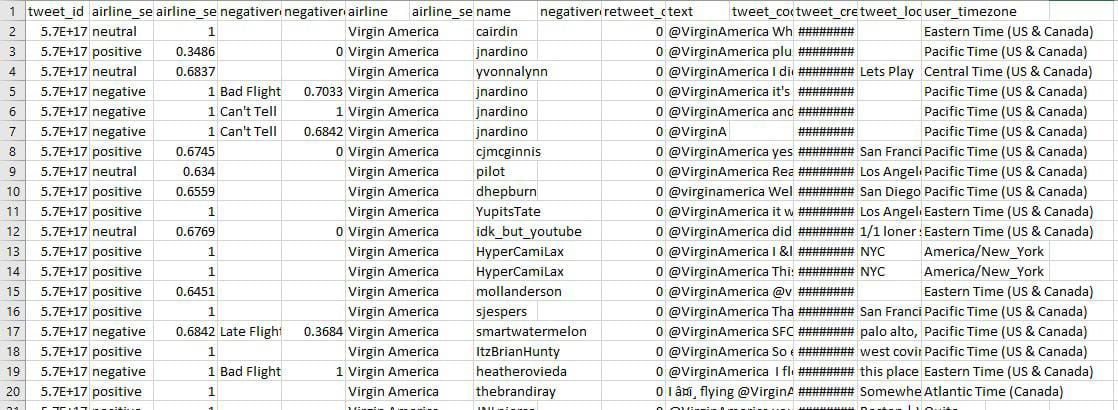
**Loading Dataset:**

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

**Data Exploration:**

**Dataset:**

**Output:**



**2.Preprocessing the dataset:**

Data preprocessing is the process of cleaning, transforming, and integrating data in order to make it ready for analysis.

**Data visualization:**

In[1]:cprint("Total number of sentiments of tweets :",'green')

print(df.airline\_sentiment.value\_counts())

plt.figure(figsize = (10, 8))

ax = sns.countplot(x = 'airline\_sentiment', data = df)

ax.set\_title(label = 'Total number of sentiments of tweets', fontsize = 20)

plt.show()

**output:**

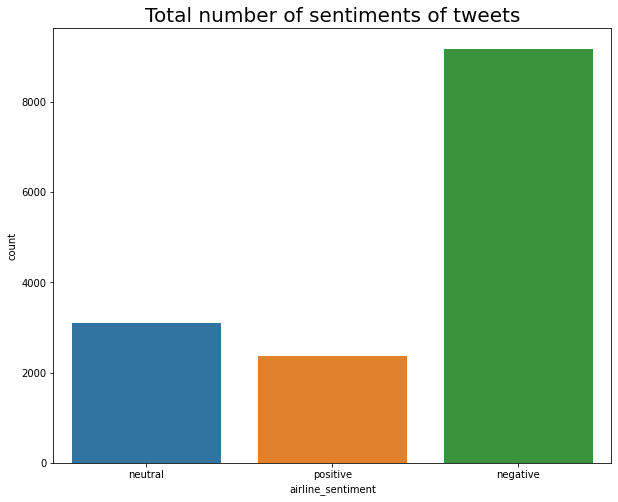
Total number of sentiments of tweets :

negative 9178

neutral 3099

positive 2363

Name: airline\_sentiment, dtype: int64



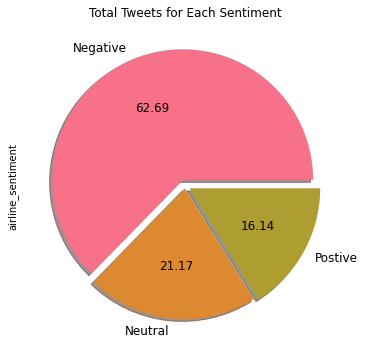
In[2]:

ax.set\_title(label = 'Total number of sentiments of tweets:')

colors=sns.color\_palette('husl',10)

pd.Series(df['airline\_sentiment']).value\_counts().plot(kind='pie',colors=colors,labels=['Negative','Neutral','Postive'],explode=[0.05,0.02,0.04],shadow=True,autopct='**%.2f**',fontsize=12,figsize=(6,6),title="Total Tweets for Each Sentiment")

plt.show()



In[3]:

colors=sns.color\_palette('husl',10)

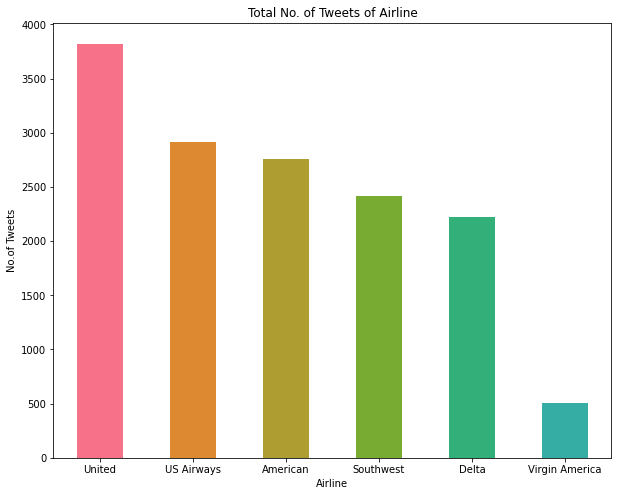
pd.Series(df['airline']).value\_counts().plot(kind="bar",color=colors,figsize=(10,8),fontsize=10,rot=0,title='Total No. of Tweets of Airline')

plt.xlabel('Airline',fontsize=10)

plt.ylabel('No.of Tweets',fontsize=10)

out[3] :

Text(0, 0.5, 'No.of Tweets')



In[4]:

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

def NCount(Airline):

airlineName =df[df['airline']==Airline]

count= airlineName['negativereason'].value\_counts()

Unique\_reason= df['negativereason'].unique()

Unique\_reason=[x for x **in** Unique\_reason if str(x) != 'nan']

Reason\_frame=pd.DataFrame({'Reasons': def Plot\_Reason(airline):

Reason\_frame['count']=Reason\_frame['Reasons'].apply(lambda x: count[x])

return Reason\_frame

a= NCount(airline)

count=a['count']

Id = range(1,(len(a)+1))

plt.bar(Id,count, color=['darkviolet','yellow','blue','lime','pink','crimson','gold','cyan','orange','purple'])

plt.xticks(Id,a['Reasons'],rotation=90)

plt.title('Count of Reasons for '+ airline)

Unique\_reason})

plt.figure(2,figsize=(16, 14))

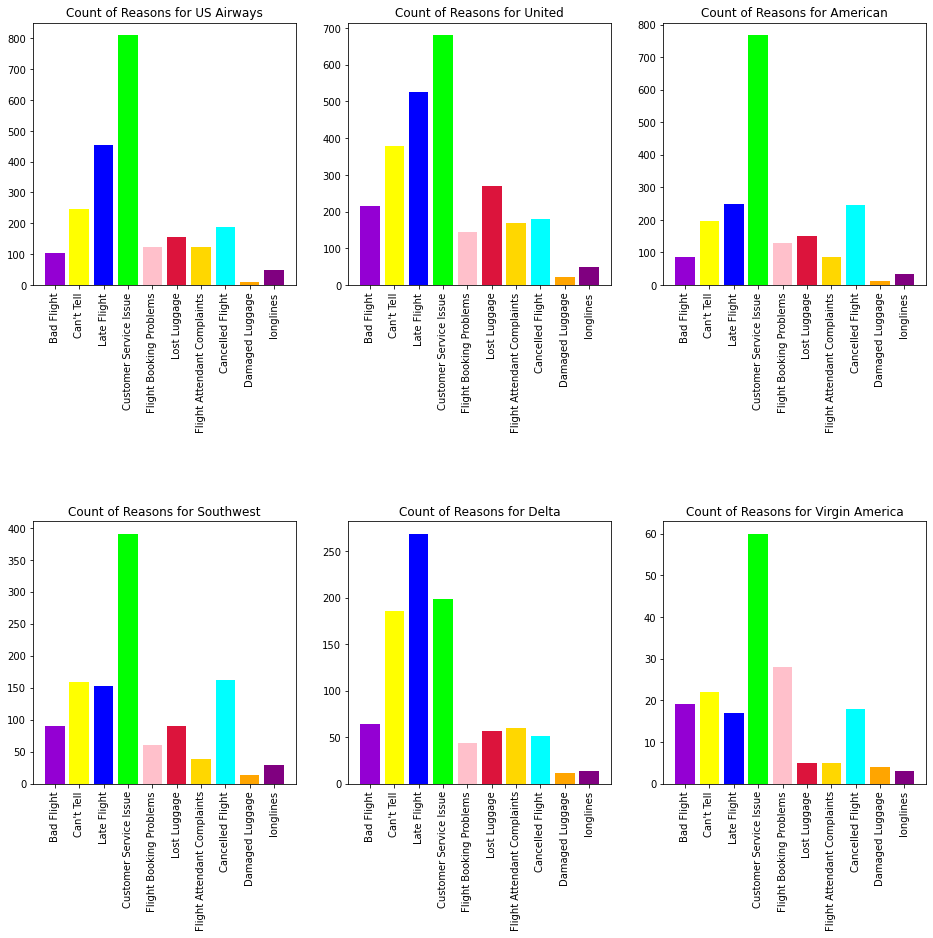
for i **in** airlines:

indices= airlines.index(i)

plt.subplot(2,3,indices+1)

plt.subplots\_adjust(hspace=0.9)

Plot\_Reason(i )



In[5]:

print("Classification Report:**\n**----------------------**\n**", cr)

cm = confusion\_matrix(y\_test,random\_forest\_classifier\_prediction)

*# plot confusion matrix*

plt.figure(figsize=(10,6))

sentiment\_classes = ['Negative', 'Neutral', 'Positive']

sns.heatmap(cm, cmap=plt.cm.Blues, annot=True, fmt='d',

xticklabels=sentiment\_classes,

yticklabels=sentiment\_classes)

plt.title('Confusion matrix', fontsize=16)

plt.xlabel('Actual label', fontsize=12)

plt.ylabel('Predicted label', fontsize=12)

plt.show()

Classification Report:

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

precision recall f1-score support

0 0.97 1.00 0.98 2343

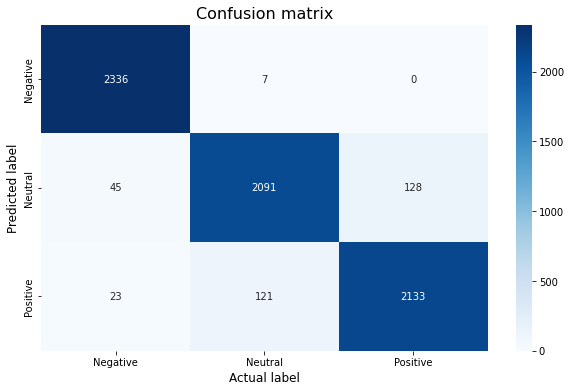
1 0.94 0.92 0.93 2264

2 0.94 0.94 0.94 2277

accuracy 0.95 6884

macro avg 0.95 0.95 0.95 6884

weighted avg 0.95 0.95 0.95 6884



**Conclusion:**

* Twitter sentiment analysis to track specific keywords and topics to detect customer trends and interests. Understanding what customers like, what their behaviors are, and how this changes over time is essential if you are planning to launch a new product.
* . Collecting large amounts of unstructured data from various sources. Tracking real-time customer feedback and sentiment about an organization's brand, products and services. Providing feedback on ways to improve products, services and customer experience.
* Sentiment analysis is used to analyze the Airline reviews dataset in this paper. To test the performance of sentiment analysis, many Machine Learning (ML) algorithms have been utilized, such as Naive Bayes, Support Vector Machine, and Decision Tree (DT), and each of these approaches has produced distinct results.
* With these foundational steps completed, our dataset is now primed for the subsequent stages of building a twitter us airline sentiment model.