SENTIMENTAL ANALYSIS FOR MARKETING (AI PHASE-5)

Submitted by:
Meenakshi B (513421106028)
B.E ECE-III year
University College of Engineering Kanchipuram

INTRODUCTION

- Sentiments are feelings, opinions, emotions, likes/dislikes , good/bad.
- Sentimental analysis is Natural Language Processing(NPL) and Information Extraction Task that aims to obtain writer's feelings expressed in positive or negative comments, questions and requests by analysing a large number of documents.
- Sentimental analysis is a study of human behaviour in which we extract human opinion.
- It's also known as Opinion Mining.

- The aim of the project is to develop a model using NLP technique for sentimental analysis using datasets .
- By understanding what your target audience is thinking on a scale that only sentiment analysis can achieve, you can tweak a product, campaign, and more, to meet their needs and let your customers know you’re listening.
- The problem is to perform sentiment analysis on customer feedback to gain insights into competitor products. By understanding customer sentiments, companies can identify strengths and weaknesses in competing products, thereby improving their own offerings. This project requires utilizing various NLP methods to extract valuable insights from customer feedback.

TOOLS AND SOFTWARE USED IN THE PROCESS

- Python is the most used language for machine learning due to its extensive libraries and frameworks. You can use libraries like NumPy, pandas and more.
- Dataset is taken from Kaggle.
- Analysis is also done using MATLAB which is used to analyse and design systems.

DESIGN THINKING

1. **Empathize**: Understand the target audience - Begin by empathizing with your customers and understanding their needs, preferences, and pain points. Conduct user interviews, surveys, and observational research to gather insights on how they express their sentiments and emotions related to your product or service.
2. **Define** : Clearly articulate the problem you want to address with sentiment analysis. For example, it could be understanding customer reactions to a recent marketing campaign or product release.
3. **Ideate** : Gather a cross-functional team of marketers, data analysts, and data scientists to brainstorm possible solutions for sentiment analysis. Generate a wide range of ideas on how to collect, process, and analyze sentiment data.

4. **Prototype** : Create a sentiment analysis system - Build a prototype of your sentiment analysis system. This could involve selecting sentiment analysis tools and technologies, creating a data collection plan, and developing algorithms or models for sentiment analysis. Keep it simple but functional for testing.

5. **Test** : Test with users - Implement your prototype and gather sentiment data from users. Evaluate the accuracy and effectiveness of your sentiment analysis system. Collect feedback from users to make improvements.

6. **Iterate** : Refine and enhance - Based on the feedback and test results, iterate on your sentiment analysis system. Refine the algorithms, data sources, or tools used. Ensure that the sentiment analysis aligns with the specific marketing objectives.

7. **Implement** : Deploy the system - Once you have a working sentiment analysis system, integrate it into your marketing operations. This may involve real-time monitoring of social media, customer reviews, or other channels for sentiment analysis.

8. **Monitor and Analyze** : Continuously monitor sentiment - Regularly track and analyze sentiment data to gain insights into customer perceptions and emotions. Use dashboards and reporting tools to visualize and interpret sentiment trends.

9. **Act** : Take action - Use the insights from sentiment analysis to inform marketing strategies and decisions. For example, adjust marketing campaigns, product features, or customer support based on sentiment feedback.

10**. Feedback Loop** : Create a feedback loop - Continuously gather feedback from your marketing and data analysis teams to improve the sentiment analysis process. This loop ensures that the system remains up-to-date and effective.

11. **Scale** : Scale the sentiment analysis - If your initial sentiment analysis system is successful, consider scaling it to cover a broader range of marketing activities and channels.

12. **Document and Share** : Document the process and share findings - Maintain detailed documentation of your sentiment analysis approach, including algorithms and tools used. Share your findings and insights with stakeholders to facilitate informed marketing decisions.

To Summarize:

Data Collection : Identify a dataset containing customer reviews and sentiments about competitor products.

Data Preprocessing : Clean and preprocess the textual data for analysis.

Sentiment Analysis Techniques : Employ different NLP techniques like Bag of Words, Word Embeddings, or Transformer models for sentiment analysis.

Feature Extraction : Extract features and sentiments from the text

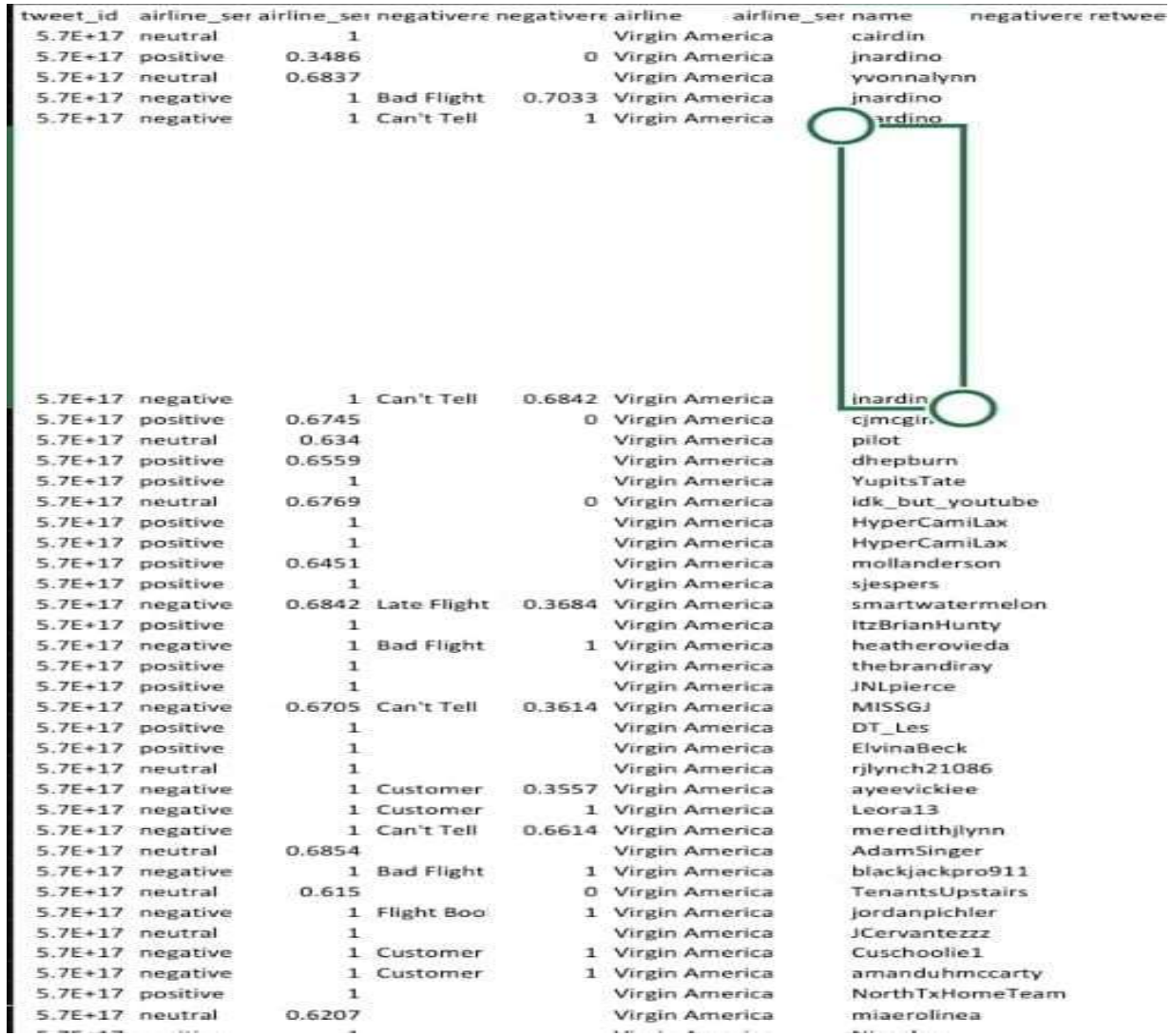
Data Visualization : Create visualizations to depict the sentiment distribution and analyze trends.

Insights Generation: Extract meaningful insights from the sentiment analysis results to guide business decisions.

PHASES OF DEVELOPMENT

1. In Phase 1 , we defined the problem definition and design thinking.
2. In Phase 2 , we described the innovative techniques such as ensemble methods and explore advanced techniques like fine-tuning pre-trained sentiment analysis model like BERT, RoBERTa .
3. In Phase 3 , we developed our project by loading and preprocessing the dataset.
4. In Phase 4 , we employed various NLP techniques and generating insights.
5. In Phase 5 , we documented the developed project.

Dataset link : https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment



tweet_id	airline_ser	airline_se	negativere	negativere	airline	airline_ser	name	negativere	retwee
5.7E+17	neutral	1			Virgin America		cairdin		
5.7E+17	positive	0.3486		0	Virgin America		jnardino		
5.7E+17	neutral	0.6837			Virgin America		yvonnalynn		
5.7E+17	negative	1	Bad Flight	0.7033	Virgin America		jnardino		
5.7E+17	negative	1	Can't Tell	1	Virgin America		[illegible]rdino		
5.7E+17	negative	1	Can't Tell	0.6842	Virgin America		inardin		
5.7E+17	positive	0.6745		0	Virgin America		cjmcgin		
5.7E+17	neutral	0.634			Virgin America		pilot		
5.7E+17	positive	0.6559			Virgin America		dhepburn		
5.7E+17	positive	1			Virgin America		YupitsTate		
5.7E+17	neutral	0.6769		0	Virgin America		idk_but_youtube		
5.7E+17	positive	1			Virgin America		HyperCamiLax		
5.7E+17	positive	1			Virgin America		HyperCamiLax		
5.7E+17	positive	0.6451			Virgin America		mollanderson		
5.7E+17	positive	1			Virgin America		sjespers		
5.7E+17	negative	0.6842	Late Flight	0.3684	Virgin America		smartwatermelon		
5.7E+17	positive	1			Virgin America		ItzBrianHunty		
5.7E+17	negative	1	Bad Flight	1	Virgin America		heatherovieda		
5.7E+17	positive	1			Virgin America		thebrandiray		
5.7E+17	positive	1			Virgin America		JNLpierce		
5.7E+17	negative	0.6705	Can't Tell	0.3614	Virgin America		MISSGJ		
5.7E+17	positive	1			Virgin America		DT_Les		
5.7E+17	positive	1			Virgin America		ElvinaBeck		
5.7E+17	neutral	1			Virgin America		rjlynch21086		
5.7E+17	negative	1	Customer	0.3557	Virgin America		ayeevickiee		
5.7E+17	negative	1	Customer	1	Virgin America		Leora13		
5.7E+17	negative	1	Can't Tell	0.6614	Virgin America		meredithjlynn		
5.7E+17	neutral	0.6854			Virgin America		AdamSinger		
5.7E+17	negative	1	Bad Flight	1	Virgin America		blackjackpro911		
5.7E+17	neutral	0.615		0	Virgin America		TenantsUpstairs		
5.7E+17	negative	1	Flight Boo	1	Virgin America		jordanpichler		
5.7E+17	neutral	1			Virgin America		JCervantezzz		
5.7E+17	negative	1	Customer	1	Virgin America		Cuschoolie1		
5.7E+17	negative	1	Customer	1	Virgin America		amanduhmccarty		
5.7E+17	positive	1			Virgin America		NorthTxHomeTeam		
5.7E+17	neutral	0.6207			Virgin America		miaerolinea		

DATA PROCESSING STEPS

- The code begins by importing the necessary libraries including pandas for data handling , matplotlib and seaborn for visualization and scikit-learn for machine learning.
- The airline tweet dataset is loaded from csv file.
- For cleaning the file

- Combine both test and training set so we can preprocess both together

- Remove reductant characters – numerics, special characters(not hashtags), short words, username(@user)

-Tokenise the processed tweet.

Stemming-strip suffixes to get the root word

- There are several types of sentiment analysis where the models focus on feelings and emotions , urgency and even intentions and polarity . The most popular types of sentiment analysis are:

- Fine-grained sentiment analysis
- Emotion detection
- Aspect based sentiment analysis
- Multilingual sentiment analysis

Sentiment analysis is critical because it helps businesses to understand the emotion and sentiments of their customers. Companies analyze customers' sentiment through social media conversations and reviews so they can make better-informed decisions. The Global Sentiment Analysis Software Market is projected to reach US$4.3 billion by the year 2027. Between 2017 and 2023, the global sentiment analysis market will increase by a CAGR of 14%

In[1]

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
print(os.listdir("../input"))
import re import nltk
from nltk.corpus import stopwords
from sklearn.model_selection import trai _test_split
from mlxtend.plotting import plot_confusion matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

In[2]:

```
df= pd.read_csv("../input/Tweets.csv")
df.head()
```

output

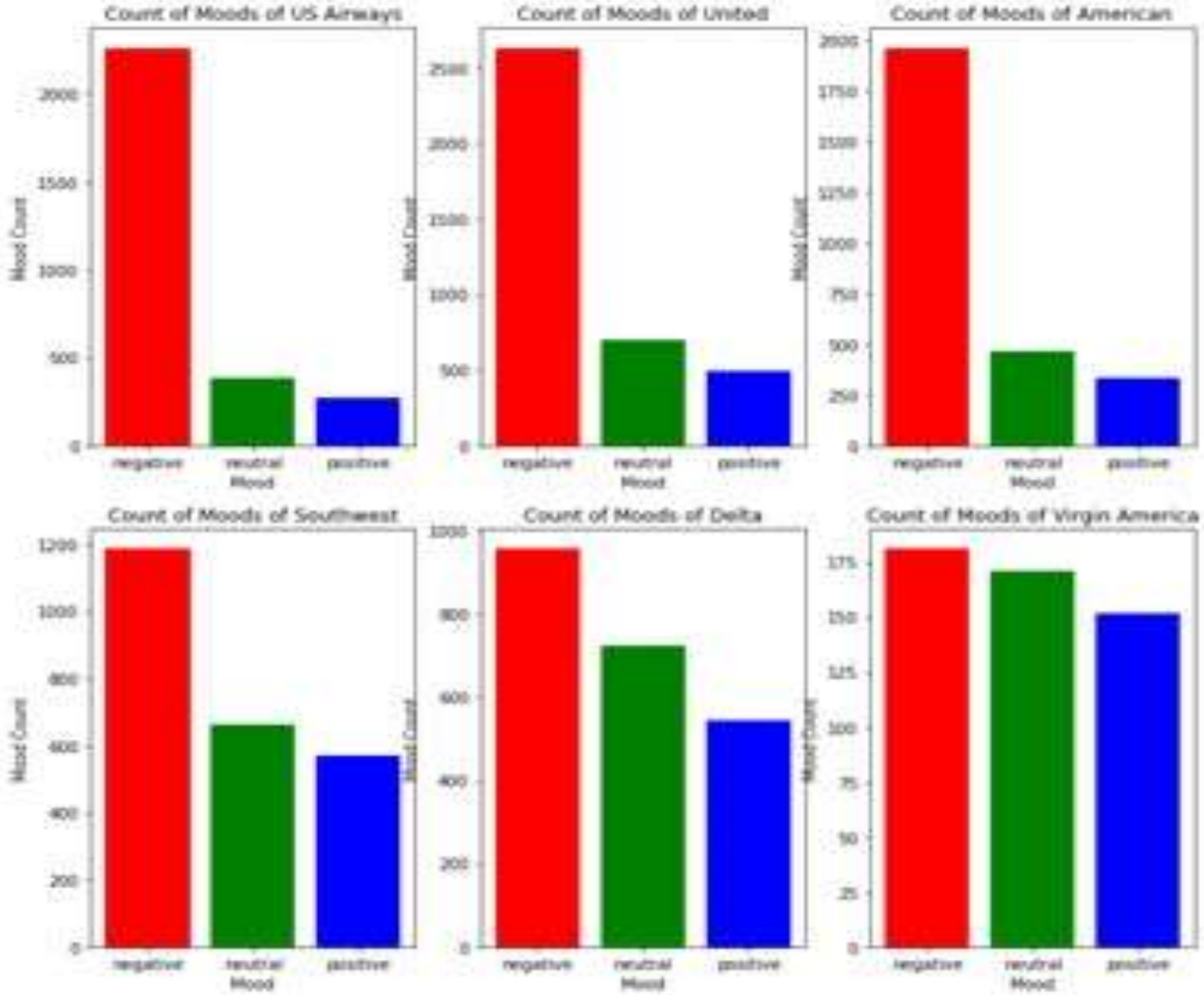
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	tweet_id	airline_se	airline_se	negativen	negativen	airline	airline_se	name	negativen	retweet_c	text	tweet_co	tweet_cre	tweet_loc	user_timezone						
2	5.7E+17	neutral	1			Virgin America		cairdin		0	@VirginAmerica Wh		########		Eastern Time (US & Canada)						
3	5.7E+17	positive	0.3486		0	Virgin America		jnardino		0	@VirginAmerica plu		########		Pacific Time (US & Canada)						
4	5.7E+17	neutral	0.6837			Virgin America		yvonnalynn		0	@VirginAmerica I di		########	Lets Play	Central Time (US & Canada)						
5	5.7E+17	negative	1	Bad Flight	0.7033	Virgin America		jnardino		0	@VirginAmerica it's		########		Pacific Time (US & Canada)						
6	5.7E+17	negative	1	Can't Tell	1	Virgin America		jnardino		0	@VirginAmerica and		########		Pacific Time (US & Canada)						
7	5.7E+17	negative	1	Can't Tell	0.6842	Virgin America		jnardino		0	@VirginA		########		Pacific Time (US & Canada)						
8	5.7E+17	positive	0.6745		0	Virgin America		cjmcginnis		0	@VirginAmerica yes		########	San Franci	Pacific Time (US & Canada)						
9	5.7E+17	neutral	0.634			Virgin America		pilot		0	@VirginAmerica Rea		########	Los Angel	Pacific Time (US & Canada)						
10	5.7E+17	positive	0.6559			Virgin America		dhepburn		0	@virginamerica Wel		########	San Diego	Pacific Time (US & Canada)						
11	5.7E+17	positive	1			Virgin America		YupItsTate		0	@VirginAmerica it w		########	Los Angel	Eastern Time (US & Canada)						
12	5.7E+17	neutral	0.6769		0	Virgin America		idk_but_youtube		0	@VirginAmerica did		########	1/1 loner	Eastern Time (US & Canada)						
13	5.7E+17	positive	1			Virgin America		HyperCamLas		0	@VirginAmerica I &I		########	NYC	America/New_York						
14	5.7E+17	positive	1			Virgin America		HyperCamLas		0	@VirginAmerica Thi		########	NYC	America/New_York						
15	5.7E+17	positive	0.6451			Virgin America		mollandenoor		0	@VirginAmerica @v		########		Eastern Time (US & Canada)						
16	5.7E+17	positive	1			Virgin America		sjespers		0	@VirginAmerica Tha		########	San Franci	Pacific Time (US & Canada)						
17	5.7E+17	negative	0.6842	Late Flight	0.3684	Virgin America		smartwatermelon		0	@VirginAmerica SFC		########	palo alto,	Pacific Time (US & Canada)						
18	5.7E+17	positive	1			Virgin America		itsBrianMurray		0	@VirginAmerica So +		########	west covi	Pacific Time (US & Canada)						
19	5.7E+17	negative	1	Bad Flight	1	Virgin America		heatherovieda		0	@VirginAmerica I fl		########	this place	Eastern Time (US & Canada)						
20	5.7E+17	positive	1			Virgin America		thebrandiray		0	I Abl_flying @VirginA		########	Somewhe	Atlantic Time (Canada)						
21	5.7E+17	positive	1			Virgin America		JNLpierce		0	@VirginAmerica you		########	Boston] \	Quito						
22	5.7E+17	negative	0.6705	Can't Tell	0.3614	Virgin America		MissGJ		0	@VirginAmerica whr		########								
23	5.7E+17	positive	1			Virgin America		DT_Les		0	@VirginA [#] 74804		########								
24	5.7E+17	positive	1			Virgin America		ElvinaBeck		0	@VirginAmerica I lo		########	Los Angel	Pacific Time (US & Canada)						
25	5.7E+17	neutral	1			Virgin America		rjlynch21088		0	@VirginAmerica will		########	Boston, M	Eastern Time (US & Canada)						

```
print("Total number of tweets for each airline \n
",df.groupby('airline')['airline_sentiment'].count().sort_values(ascending=False))
airlines= ['US Airways','United','American','Southwest','Delta','Virgin America’]
plt.figure(1,figsize=(12, 12))
 for i in airlines:
 indices= airlines.index(i)
plt.subplot(2,3,indices+1)
 new_df=df[df['airline']==i]
count=new_df['airline_sentiment'].value_counts()
Index = [1,2,3]
plt.bar(Index,count, color=['red', 'green', 'blue'])
plt.xticks(Index,['negative','neutral','positive’])
plt.ylabel('Mood Count’)
plt.xlabel('Mood’)
plt.title('Count of Moods of ‘+i)
```

OUTPUT:

Total number of tweets for each airline

```
airline
United           3822
US Airways       2913
American         2759
Southwest        2420
Delta            2222
Virgin America    504
Name: airline_sentiment, dtype: int64
```


Count of Moods of US Airways
Count of Moods of United
Count of Moods of American
Count of Moods of Southwest
Count of Moods of Delta
Count of Moods of Virgin America
Mood Count
Mood
negative
neutral
positive

POSITIVE SENTIMENTAL TWEETS

```
def freq(str):
  str = str.split()
   str2 = []
   for i in str:

        if i not in str2:


         str2.append(i)

   for i in range(0, len(str2)):
      if(str.count(str2[i])>50):
         print('Frequency of', str2[i], 'is :', str.count(str2[i]))

print(freq(cleaned_word))
```

- **OUTPUT**

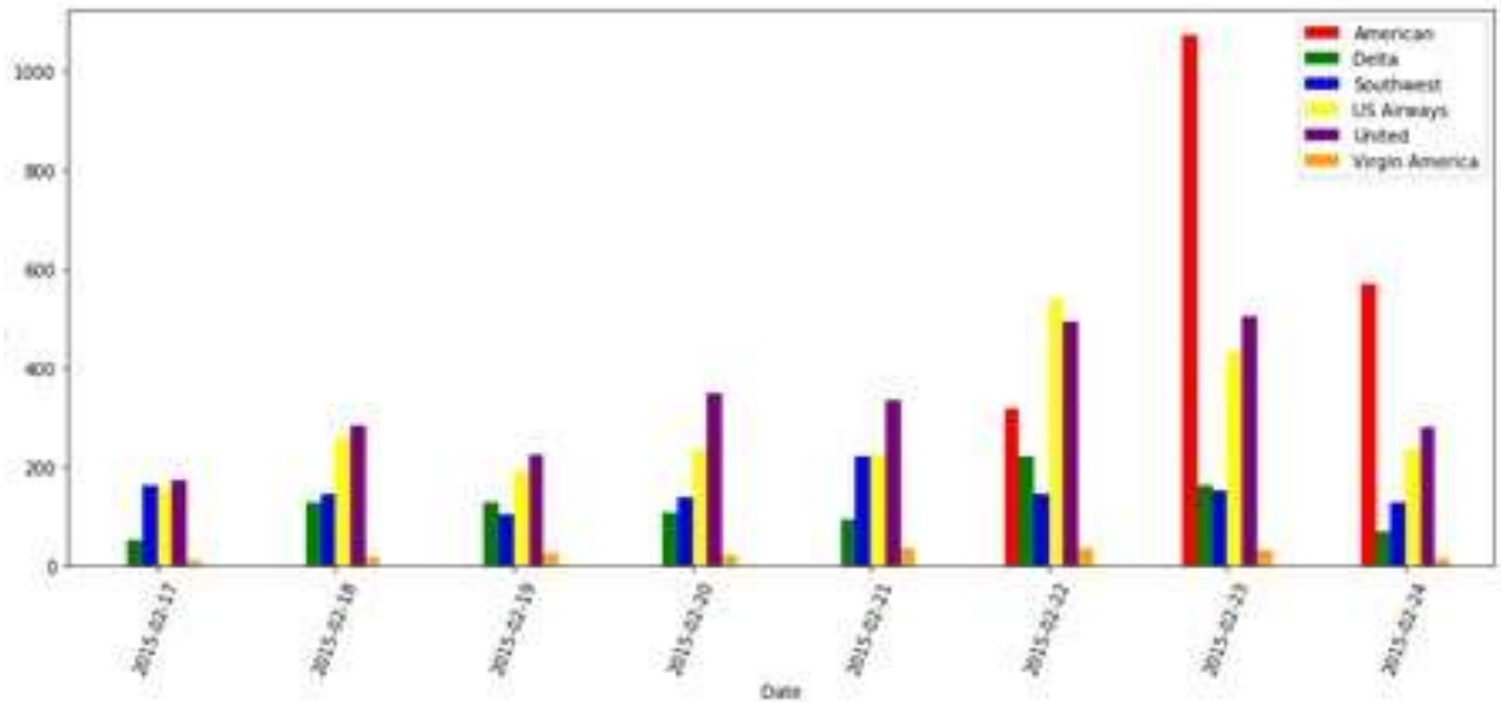
Frequency of to is : 923
Frequency of the is : 924
Frequency of time is : 59
Frequency of I is : 574
Frequency of fly is : 54
Frequency of this is : 143
Frequency of :) is : 96
Frequency of it is : 166
Frequency of was is : 226
Frequency of and is : 416
Frequency of an is : 74
Frequency of good is : 75
Frequency of so is : 163
Frequency of much is : 54
Frequency of is is : 219
Frequency of a is : 501
Frequency of great is : 144
Frequency of my is : 320
Frequency of & is : 77
Frequency of on is : 327
Frequency of I'm is : 67
Frequency of flying is : 59
Frequency of your is : 212
Frequency of all is : 92
Frequency of from is : 124
Frequency of Thanks! is : 69
Frequency of for is : 658
Frequency of flight is : 263
Frequency of but is : 91
Frequency of you is : 509
Frequency of would is : 56
Frequency of be is : 135
Frequency of with is : 195
Frequency of you. is : 77
Frequency of love is : 85
Frequency of You is : 62
Frequency of are is : 120
Frequency of of is : 236
Frequency of that is : 102
Frequency of in is : 309
Frequency of just is : 129
Frequency of very is : 55
Frequency of not is : 57
Frequency of been is : 52
Frequency of like is : 57
Frequency of we is : 75
Frequency of can is : 54
Frequency of crew is : 51
Frequency of - is : 87
Frequency of customer is : 101
Frequency of back is : 54
Frequency of us is : 62
Frequency of out is : 71
Frequency of best is : 63
Frequency of have is : 124
Frequency of Thank is : 231

NEGATIVE SENTIMENTAL TWEETS

```
day_df = day_df.loc(axis=0)[:,:,'negative']

#groupby and plot data
ax2 =
day_df.groupby(['tweet_created','airline']).sum
().unstack().plot(kind = 'bar', color=['red',
'green', 'blue','yellow','purple','orange'], figsize
= (15,6), rot = 70)
labels = ['American','Delta','Southwest','US
Airways','United','Virgin America']
ax2.legend(labels = labels)
ax2.set_xlabel('Date')
ax2.set_ylabel('Negative Tweets')
plt.show()
```

OUTPUT


American
Delta
Southwest
US Airways
United
Virgin America
1000
800
600
400
200
0
2015-02-17
2015-02-18
2015-02-19
2015-02-20
2015-02-21
2015-02-22
2015-02-23
2015-02-24
Date

FEATURE EXTRACTION

- In sentiment analysis, we detect tweets that have negative sentiment, i.e, racist, sexist or general hate speech. Here, tweets with a label ‘1’ denote a negative tweet, while '0' denotes the absence of hate speech in the tweet. We extract features using the following :
    1. Bag of Words Features
    2. TF-IDF features
    3. Word Embedding’s
- VISUALIZATION: The code creates a histogram to visualize the distribution of airline sentiments. It also creates a pie chart to visualize the sentiment distribution using percentages. We will analyze the text of the tweet and its relation to the sentiment with the following : Wordcloud : Most used words (have bigger fonts), for positive and negative tweets . Hashtags: Analyze the effect of hashtags on the tweet sentiment.

```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer f
from sklearn.linear_model
 import LogisticRegression
from sklearn.metrics import roc_auc_score, confusion_matrix
from sklearn.model_selection import train_test_split
sentiment_counts = df['airline_sentiment'].value_counts()
 plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts )
```

```
labels=sentiment_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Airline Sentiments') plt.axis('equal’)
hashtags = []
 for i in x:
 ht = re.findall(r"#(\w+)", i)
 hashtags.append(ht)
 return hashtags In [22]: linkcode
_non_negative = hashtag_extract(combine['tidy_tweet'][combine['label'] == 0
 HT_negative = hashtag_extract(combine['tidy_tweet'][combine['label'] == 1]
HT_non_negative = sum(HT_non_negative,[])
HT_negative = sum(HT_negative,[])
```

1**. Installing NLTK and downloading the data**

pip install nltk==3.3

Python3

import nltk

nltk.download('twitter_samples’)

2**. Tokenizing the data**

nano nlp_test.py

from nltk.corpus import twitter_samples

from nltk.corpus import twitter_samples

positive_tweets = twitter_samples.strings('positive_tweets.json') negative_tweets = twitter_samples.strings('negative_tweets.json’)

text = twitter_samples.strings('tweets.20150430-223406.json’)

python3

import nltk nltk.download('punkt’)

from nltk.corpus import twitter_samples

positive_tweets=twitter_samples.strings(‘positive_tweets.json’)

```
negative_tweets=twitter_samples.strings('negative_tweets.json')
text=twitter_samples.strings('tweets.20150430-223406.json')
tweet_tokens=twitter_samples.tokenized('positive_tweets.json')
```

OUTPUT:

```
['#FollowFriday','JJ'),
('@France_Inte','NNP'),
('@PKuchly57','NNP'),
('for','IN'),
('being','VBG'),
('in','IN'),
('my','PRP$'),
('Community','NN')]
```

3. **Normalizing the data**

```
python3
import nltk
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger’)
from nltk.tag import pos_tag
from nltk.corpus import twitter_samples
tweet_tokens = twitter_samples.tokenized('positive_tweets.json')
print(pos_tag(tweet_tokens[0]))
```

```
from nltk.tag import pos_tag
from nltk.stem.wordnet import WordNetLemmatizer
def lemmatize_sentence(tokens):
lemmatizer = WordNetLemmatizer()
lemmatized_sentence = []
for word, tag in pos_tag(tokens):
if tag.startswith('NN'):
pos = 'n'
elif tag.startswith('VB’):
pos = 'v'
else:
pos = 'a’
lemmatized_sentence.append(lemmatizer.lemmatize(word, pos))
return lemmatized_sentence
print(lemmatize_sentence(tweet_tokens[0]))
```

4. **Removing noise from data**

```
import re, string
def remove_noise(tweet_tokens, stop_words = ()):
    cleaned_tokens = []
    for token, tag in pos_tag(tweet_tokens):
        token = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+#]|[!*\(\),]|'\
                       '(?:%[0-9a-fA-F][0-9a-fA-F]))+','', token)
        token = re.sub("(@[A-Za-z0-9_]+)","", token)
        if tag.startswith("NN"):
            pos = 'n'
        elif tag.startswith('VB'):
            pos = 'v'
        else:
            pos = 'a'
```

```
lemmatizer = WordNetLemmatizer()
token = lemmatizer.lemmatize(token, pos)
if len(token) > 0 and token not in string.punctuation and
token.lower() not in stop_words:
cleaned_tokens.append(token.lower())
return cleaned_tokens
nltk.download('stopwords’)
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
print(remove_noise(tweet_tokens[0], stop_words))
```

OUTPUT:

```
'#followfriday', 'top', 'engage', 'member', 'community', 'week', ':)']
```

```
print(positive_tweet_tokens[500])
print(positive_cleaned_tokens_list[500])
```

OUTPUT:

```
['Dang', 'that', 'is', 'some', 'rad', '@AbzuGame', '#fanart', '!', ':D',
'https://t.co/bI8k8tb9ht']
['dang', 'rad', '#fanart', ':d’]
```

5.**Determining Word Density**

```
def get_all_words(cleaned_tokens_list):
    for tokens in cleaned_tokens_list:
        for token in tokens:
            yield token
all_pos_words = get_all_words(positive_cleaned_tokens_list)
```

OUTPUT

[(':)', 3691),
(':-)', 701),
(':d', 658),
('thanks', 388),
('follow', 357),
('love', 333),
('...', 290),
('good', 283),
('get', 263),
('thank', 253)]

6. Preparing data for the model

```
def get_tweets_for_model(cleaned_tokens_list):
    for tweet_tokens in cleaned_tokens_list:
        yield dict([token, True] for token in tweet_tokens)
positive_tokens_for_model =
get_tweets_for_model(positive_cleaned_tokens_list)
negative_tokens_for_model = get_tweets_for_mod
import random
positive_dataset = [(tweet_dict, "Positive")
                    for tweet_dict in positive_tokens_for_model]
negative_dataset = [(tweet_dict, "Negative")
                    for tweet_dict in negative_tokens_for_model]
dataset = positive_dataset + negative_dataset
random.shuffle(dataset)
train_data = dataset[:7000]
test_data = dataset[7000:]el(negative_cleaned_tokens_list)
```

7. **Building and testing the model**

```
from nltk import classify
from nltk import NaiveBayesClassifier
classifier = NaiveBayesClassifier.train(train_data)
print("Accuracy is:", classify.accuracy(classifier, test_data))
print(classifier.show_most_informative_features(10))
```

OUTPUT:

```
Accuracy is: 0.995666666666667
Most Informative Features
                   :( = True           Negati : Positi =   2085.6 : 1.0
                   :) = True           Positi : Negati =    986.0 : 1.0
                welcome = True           Positi : Negati =     37.2 : 1.0
                arrive = True           Positi : Negati =     31.3 : 1.0
                  sad = True           Negati : Positi =     25.9 : 1.0
               follower = True           Positi : Negati =     21.1 : 1.0
                  bam = True           Positi : Negati =     20.7 : 1.0
                 glad = True           Positi : Negati =     18.1 : 1.0
                  x15 = True           Negati : Positi =     15.9 : 1.0
            community = True           Positi : Negati =     14.1 : 1.0
'(?:%[0-9a-fA-F][0-9a-fA-F]))+','', token)
```

MACHINE LEARNING ALGORITHM

- Machine learning algorithms can model many features and adapt to adjusting input. That's why companies implement machine learning or deep learning algorithms to fasten business processes and get insights to develop new strategies.
- 4 machine learning approaches that can be applied to sentiment analysis:

1. **Supervised learning**： In supervised learning, the data is labeled manually by the annotators, and it is used to train the algorithm. Thus, the algorithm can classify incoming, unlabeled data based on pre-labeled data. This method outperforms both semi-supervised and unsupervised methods as it depends on data labeled manually by humans and includes fewer errors

Some supervised algorithms are as follows:

- Naive Bayes (NB)
- Logistic Regression (LogR)
- Maximum Entropy (ME)
- Support Vector Machines (SVM)
- K-Nearest Neighbor (kNN)
- Random Forest (RF)
- Decision Trees (DT)

2. **Semi-supervised learning** : Semi-supervised learning uses both labeled and unlabeled data, and because it doesn't require as much human intervention as supervised learning, it takes less time to conduct analysis. Unlabeled data assists in extracting language-invariant features, while labeled data is utilized as a classifier.

3. **Unsupervised learning** : Unsupervised learning is a lexical-based approach where the data is clustered based on shared characteristics, including word pairings or popular terms. It does not need training data or modeling and instead uses predefined lists or dictionaries.

4.**Deep learning algorithms** : Deep learning algorithms depend on neural networks and outperform other machine learning methods. However, they require a great amount of data to train the model. Thus, they give the best results when applied to large datasets. Some common deep-learning methods are:

- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Deep Belief Networks (DBN)
- Long-Short Term Memory (LSTM)

We have used semi-supervised learning and deep learning algorithms.

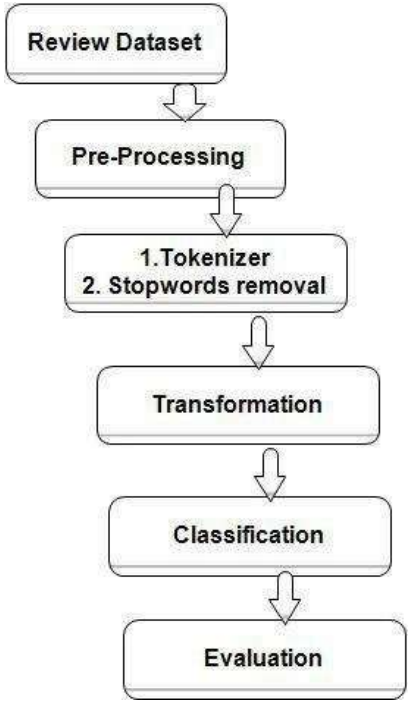
MODEL TRAINING

- There are different approaches to sentiment analysis, such as rule-based, lexicon-based, or machine learning-based. Each approach has its advantages and disadvantages, depending on the complexity, accuracy, and scalability of your project. For example, rule-based methods are easy to implement and customize, but they can be limited by the quality and coverage of the rules. Lexicon-based methods rely on predefined dictionaries of words and phrases with associated sentiment scores, but they can be affected by context and ambiguity. Machine learning methods can learn from data and adapt to new situations, but they require a lot of training and validation.

- To train your sentiment analysis model, you need to have a labeled dataset that contains text samples and their corresponding sentiment labels, such as positive, negative, or neutral. You can either use an existing dataset or create your own, depending on the availability and relevance of the data for your project. To create your own dataset, you need to collect text samples from your data sources, such as social media, surveys, reviews, or blogs, and annotate them manually or with the help of tools. You also need to ensure that your dataset is balanced, diverse, and representative of your target domain.
- To optimize your sentiment analysis model, you need to evaluate its performance and fine-tune its parameters. You can use various metrics to measure the accuracy, precision, recall, and f1-score of your model, depending on your objectives and data characteristics. You can also use cross-validation, grid search, or hyperparameter optimization techniques to find the optimal combination of parameters for your model.
- We have used BERT model by enhancing some of the features of the model.

EVALUATION METRICS

- As a classification problem, Sentiment Analysis uses the evaluation metrics of Precision.
- Average measures like macro, micro, and weighted F1-scores are useful for multi-class problems. Depending on the balance of classes of the dataset the most appropriate metric should be used.



- Depending on the type and level of sentiment analysis, there are different metrics that can be used to evaluate its accuracy and reliability. For instance, accuracy measures the percentage of correctly classified texts or sentences according to their sentiment polarity, while precision measures the percentage of correctly classified texts or sentences out of those that were classified as having a certain sentiment polarity. Additionally, recall measures the percentage of correctly classified texts or sentences out of those that actually have a certain sentiment polarity, and F1-score is the harmonic mean of precision and recall, ranging from 0 to 1, where 1 is the best score.
- Some possible strategies for achieving this include data cleaning and preprocessing, domain adaptation and customization, and metric selection and optimization. Data cleaning can improve the quality of the data used for training, testing, and evaluating sentiment analysis models and applications. Domain adaptation and customization can tailor the models to specific domains and contexts.

Sentimental Analysis

POSITIVE

- Low cost than traditional methods.
- Faster way of getting user data.
- Identifies an organization's strengths , weaknesses, opportunities and threats
- More accurate and insightful customer perceptions and feedback

NEGATIVE

- Relying exclusively on pre-built, generic sentiment analysis may not yield optimal results
- Regular update for the model is required.

Ensemble methods in deep learning are used to improve the performance of neural networks and can take many forms including

Stacking: Training multiple deep learning models and utilizing the outputs of each model to train a "meta-model", a machine learning model that takes other models' outputs as inputs. The meta-model takes the base model predictions as inputs and learns how to best combine them to make the final prediction. This approach can enhance the model's predictive power and capture complex relationships in the data.

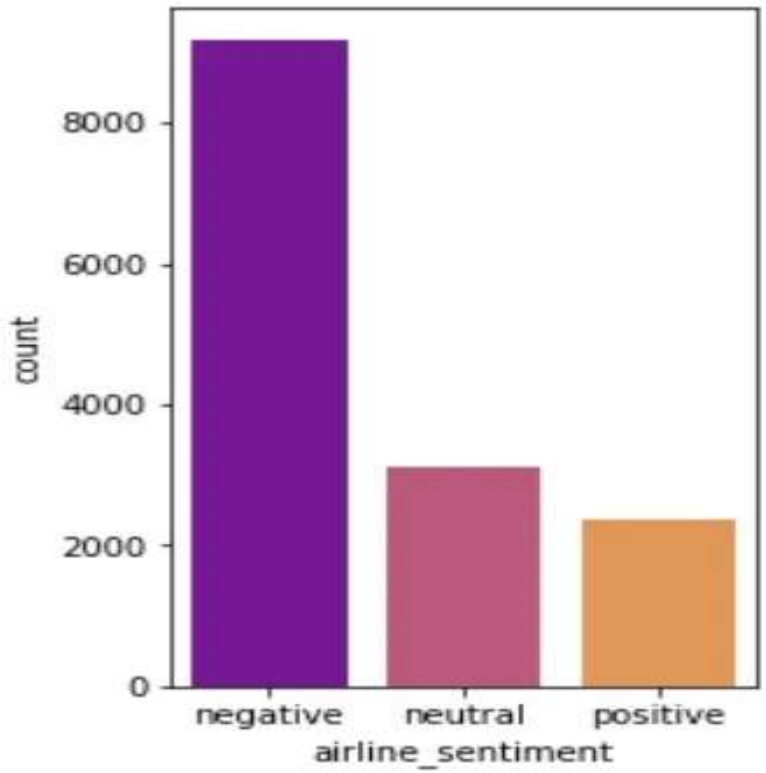
Bagging: Training multiple instances of the same model on different subsets of data and combining the model outputs through averaging or voting. This approach can improve the model's generalizability.

Model Averaging: Independently training multiple instances of the same deep learning model with different initializations (the initial values of the parameters or weights of a model before training), and averaging the model outputs to obtain a final prediction. This approach can reduce the impact of varying initializations among models and provide more stable predictions.

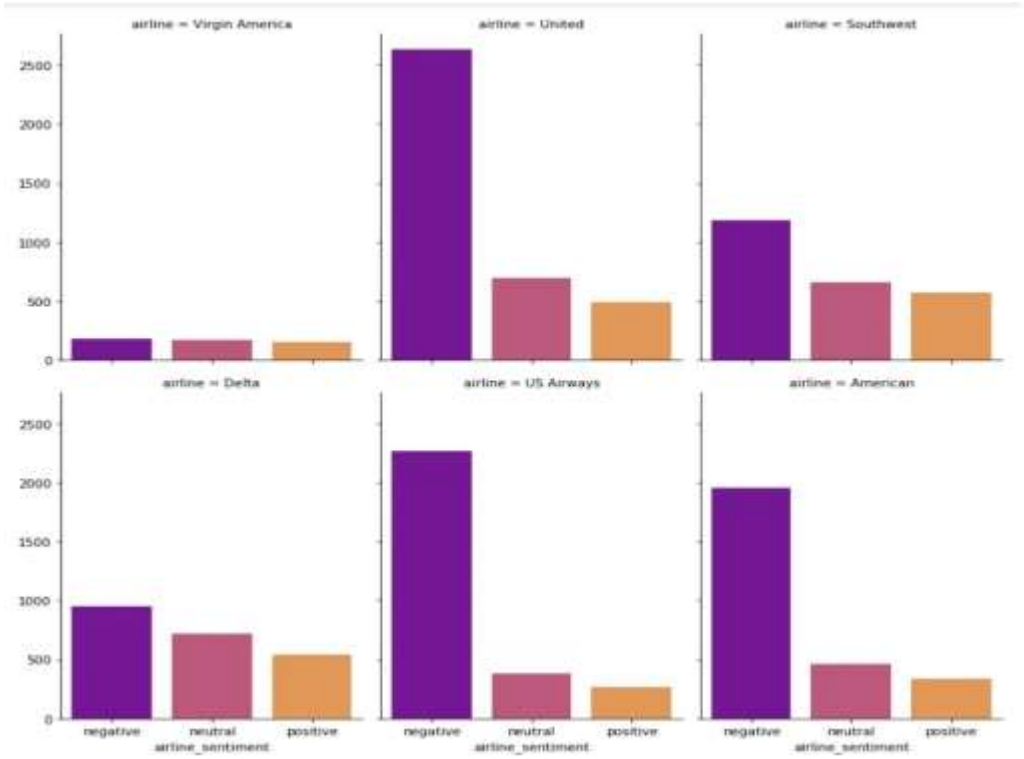
Boosting, a very common ensemble method in classical machine learning is not prevalent in deep learning. Boosting entails combining weaker machine learning models, such as decision trees in classical machine learning, to create a single strong model. While there are some recent examples of boosting in deep learning, deep learning models are often capable of achieving high accuracy without the need for boosting.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 15 columns):
tweet_id                      14640 non-null int64
airline_sentiment             14640 non-null object
airline_sentiment_confidence  14640 non-null float64
negativereason                9178 non-null object
negativereason_confidence     10522 non-null float64
airline                       14640 non-null object
airline_sentiment_gold        40 non-null object
name                          14640 non-null object
negativereason_gold           32 non-null object
retweet_count                 14640 non-null int64
text                          14640 non-null object
tweet_coord                   1019 non-null object
tweet_created                 14640 non-null object
tweet_location                9907 non-null object
user_timezone                 9820 non-null object
dtypes: float64(2), int64(2), object(11)
memory usage: 1.7+ MB
```

```
plt.figure(figsize=(3,5))
sns.countplot(tweets['airline_sentiment’],
order=tweets.airline_sentiment.value_counts().index,palette='plasma')
plt.show()
```



```
g = sns.FacetGrid(tweets, col="airline", col_wrap=3, height=5, aspect =0.7)
g = g.map(sns.countplot, "airline_sentiment")
order =tweets.airline_sentiment.value_counts().index, palette='plasma')
plt.show()
```



CONCLUSION

Thus we documented the step-by-step development of our project “Sentimental Analysis for Marketing”. We explained the important steps in developing our project and provided the code for analysis as well.We have mentioned all the phases of the project development