SENTIMENT ANALYSIS FOR MARKETING

BATCH MEMBER

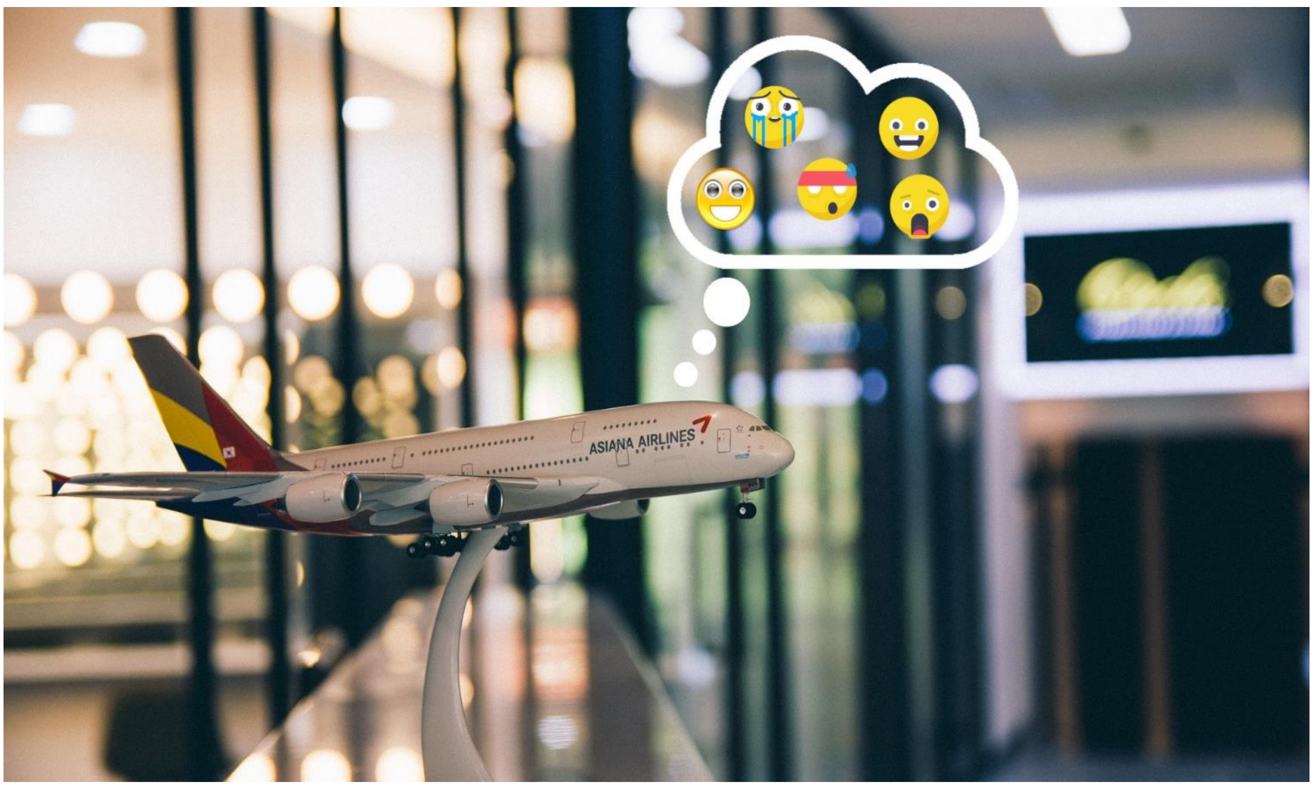
815621104028: PUVIYARASI EZHILARASAN

Phase 5 submission document

Project Title: Sentiment Analysis for Marketing

Phase 5: Project Documentation & Submission

Topic: In this section we will document the complete project and prepare it for submission.



SENTIMENT ANALYSIS FOR MARKETING

INTRODUCTION:

- Sentiment analysis, often referred to as opinion mining, is a field of study that analyzes people's sentiments, attitudes, emotions, and opinions from digital text.

- For marketing, this technology has become an invaluable tool, enabling businesses to understand consumer feelings towards products or brands and adjust strategies accordingly.

- By harnessing the power of machine learning and natural language processing, sentiment analysis can process vast amounts of data from social media, reviews, forums, and more, providing actionable insights in real time.

Dataset Link:

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

Given data set:

1	tweet_id	airline_se	airline_se	negativer	negativer	airline	airline_se	name	negativer	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 anc		#########		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		HyperCamiLax		0	@VirginAmerica I &l		#########	NYC	America/New_York
14	5.7E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica Thi:		#########	NYC	America/New_York
15	5.7E+17	positive	0.6451			Virgin America		mollanderson		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		ItzBrianHunty		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 â¤ï¸ flying @VirginA		#########	Somewhe	Atlantic Time (Canada)

Here's a list of tools and software commonly used in the process:

The tools and software that facilitate sentiment analysis in marketing include:

1. Natural Language Processing (NLP) Libraries:
    - Such as NLTK, spaCy, and TextBlob, which are crucial for text analysis and sentiment classification.
2. Machine Learning Platforms:
    - TensorFlow, PyTorch, and Scikit-learn are popular for building sentiment analysis models.
3. Data Visualization Tools:
    - Tableau, Power BI, and Google Data Studio help in presenting the sentiment analysis results in an understandable format.

4. APIs for Sentiment Analysis:
    - Services like IBM Watson, Google Cloud Natural Language API, and Microsoft Azure Text Analytics provide pre-trained models for sentiment analysis.

5. Social Media Analytics Tools:
    - Hootsuite, Brand watch, and Sprout Social are tailored for marketers to monitor sentiment on social media platforms.

6. Customer Feedback Tools:
    - Qualtrics, SurveyMonkey, and Zendesk gather and analyze customer feedback for sentiment.

1.DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT

1. Empathy:

- User Persona: Understand the end-users of the sentiment analysis insights.
- Customer Journey Mapping: Identify touchpoints where sentiment analysis can drive decision-making.

2. Define:

- Problem Refinement: Break down the overarching problem into manageable sub-problems.
- Stakeholder Analysis: Identify and involve key stakeholders in the feedback analysis process.

3. Ideate:

- Brainstorming Sessions: Encourage creative thinking for innovative solutions.
- Feature Prioritization: Prioritize key features for sentiment analysis that align with business goals.

4. Prototype:

- Technical Architecture: Design the technical framework for sentiment analysis.
- User Interface Mockups: Develop a prototype of the user interface for accessing insights.

5. Test:

- Pilot Testing: Conduct small-scale tests to validate the effectiveness of the sentiment analysis model.
- Feedback Loop: Establish a continuous feedback loop for refining the model.

6. NLP Techniques:

- Tokenization: Break down textual data into tokens for analysis.
- Sentiment Analysis Models: Implement machine learning models for sentiment classification.
- Feature Extraction: Identify key features contributing to sentiment.

7. Data Collection:

- Sources: Gather data from customer reviews, surveys, and social media.
- Data Cleaning: Ensure the quality and reliability of the collected data.

8. Technology Stack:

- NLP Libraries: Utilize established libraries like NLTK, spaCy, or TensorFlow.
- Data Storage: Choose appropriate databases for efficient data storage.

9. Data Gathering:

- Initiate the collection of customer feedback data from various sources.

10. Iterative Refinement:

- Establish an iterative process for refining the models based on ongoing feedback and performance evaluation.

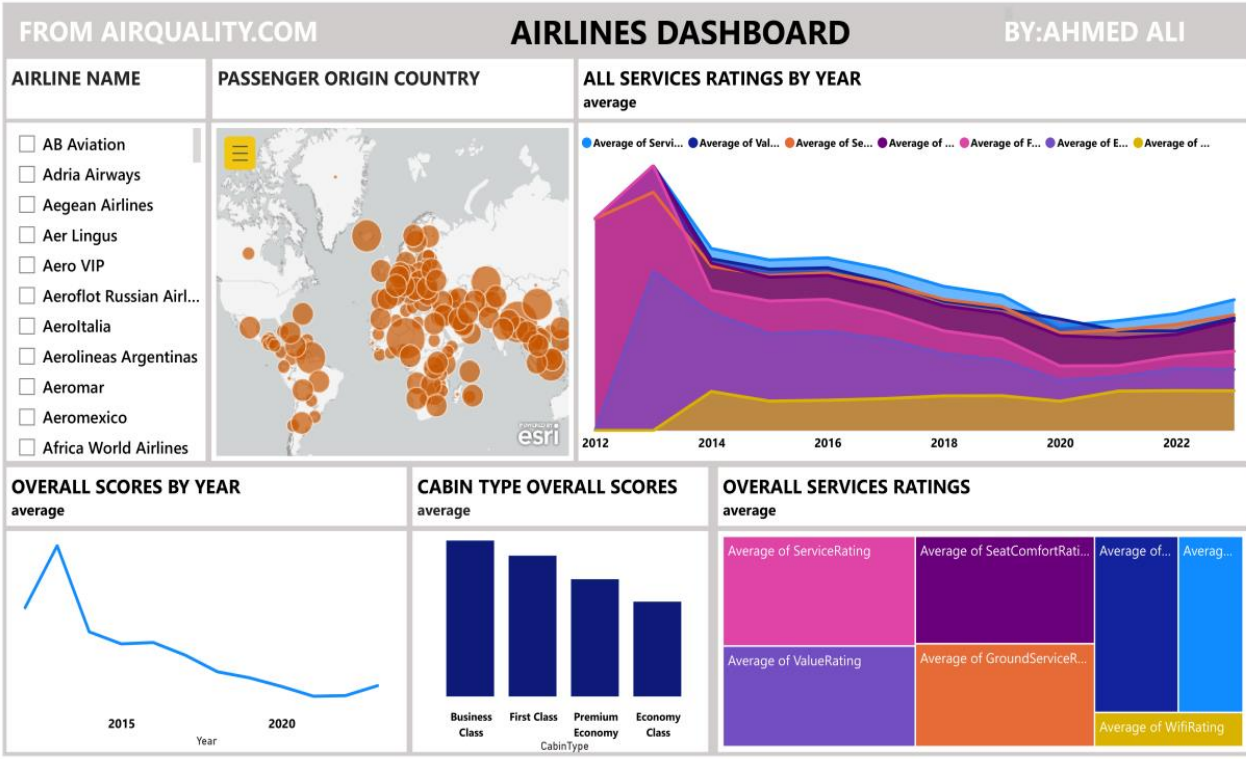
2.DESIGN INTO INNOVATION

Data Source:

A robust data source for sentiment analysis should be accurate, complete, covering a wide range of feedback, and accessible. The given database is a rich source of customer feedback on airline services, which can serve as a basis for our sentiment analysis.

Model Comparison:

Compare various models' performance using the above metrics and select the model that exhibits the highest accuracy and F1 score, ensuring a balanced performance across different sentiment classes.



Programs:

1.Sentiment Analysis using VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool.

from nltk.sentiment.vader import SentimentIntensityAnalyzer

sid = SentimentIntensityAnalyzer()

data['sentiment_scores'] = data['cleaned_text'].apply(lambda x: sid.polarity_scores(x))

data['sentiment_scores'] = data['cleaned_text'].apply(lambda x: sid.polarity_scores(x)) data['compound_score'] = data['sentiment_scores'].apply(lambda x: x['compound'])

data['vader_sentiment'] = data['compound_score'].apply(lambda x: 'positive' if x > 0 else ('neutral' if x == 0 else 'negative'))

2.Visualize the most common words in positive and negative sentiments.

from wordcloud import WordCloud

positive_text = ' '.join(data[data['vader_sentiment'] == 'positive']['cleaned_text'])

negative_text = ' '.join(data[data['vader_sentiment'] == 'negative']['cleaned_text'])

fig, ax = plt.subplots(1, 2, figsize=(20, 10))

wordcloud_pos = WordCloud(width=600, height=400).generate(positive_text)

wordcloud_neg = WordCloud(width=600, height=400).generate(negative_text)
ax[0].imshow(wordcloud_pos, interpolation='bilinear')

ax[0].set_title('Word Cloud of Positive Sentiments')

ax[0].axis('off') ax[1].imshow(wordcloud_neg, interpolation='bilinear')

ax[1].set_title('Word Cloud of Negative Sentiments')

ax[1].axis('off')

plt.show()

Feature Engineering:

Create new features or transform existing ones to capture valuable information from text data.

from sklearn.feature_extraction.text import TfidfVectorizer

TF-IDF Vectorization

vectorizer = TfidfVectorizer()

X = vectorizer.fit_transform(data['text'])

Model Evaluation and Selection:

Metrics:

- Precision, Recall, and F1-score are crucial metrics for evaluating the performance of our sentiment analysis models, especially in scenarios where classes are imbalanced.
- Area Under the ROC Curve (AUC-ROC) is another vital metric that measures the performance of the classification model at various threshold settings.

Feature Importance:

- Analyze the coefficients or feature importances of the model to understand which words or n-grams have the most significant impact on sentiment.
- Visualize the feature importances using bar plots or other suitable visualizations to make the interpretation intuitive.

Web Application:

Develop a web application using frameworks like Flask or Django to deploy the sentiment analysis model.

Create a user-friendly interface where users can input text data, submit it, and receive sentiment analysis results.

3.BUILD LOADING AND PREPROCESSING THEDATASET

1. 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.

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:

```
# 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_s
core # 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:

1	tweet_id	airline_se	airline_se	negativer	negativer	airline	airline_se	name	negativer	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		HyperCamiLax		0	@VirginAmerica I &I		########	NYC	America/New_York
14	5.7E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica Thi		########	NYC	America/New_York
15	5.7E+17	positive	0.6451			Virgin America		mollanderson		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		ItzBrianHunty		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 â¤ï¸ flying @VirginA		########	Somewhe	Atlantic Time (Canada)

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:

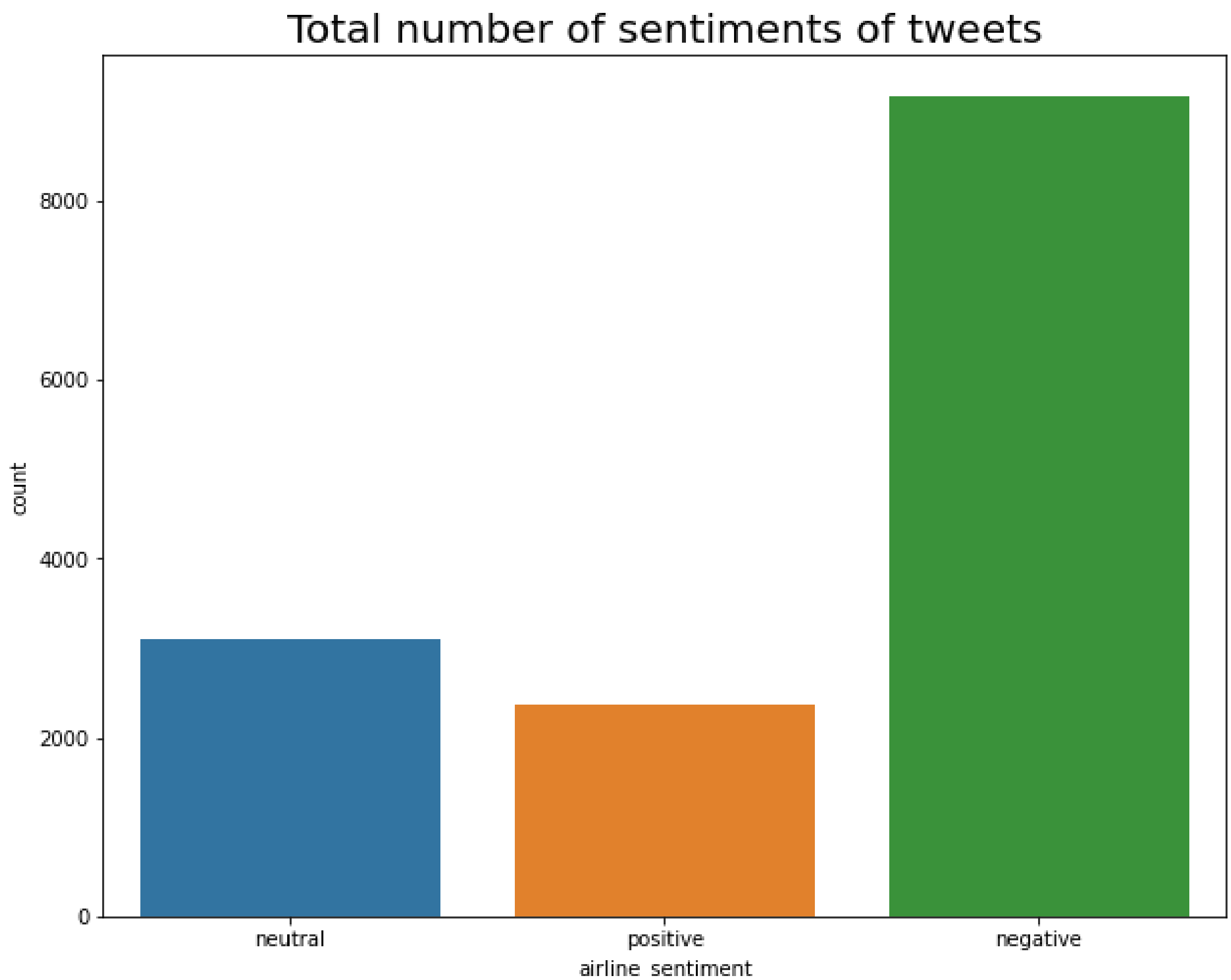
In[1]:

```
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()
```

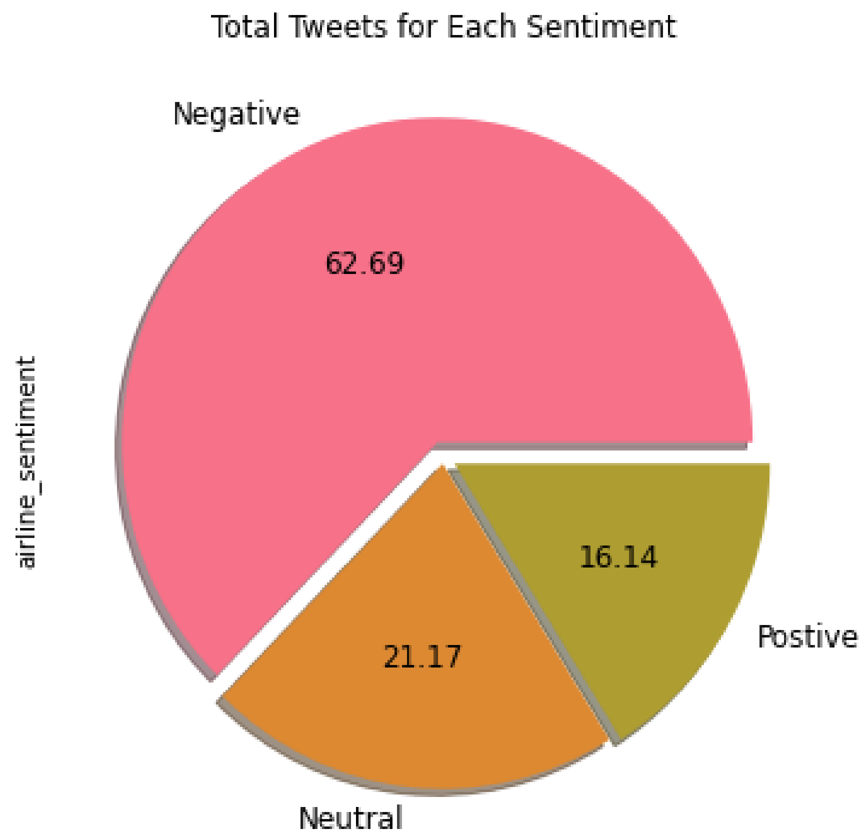
output:

```
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,label
s=['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()
```

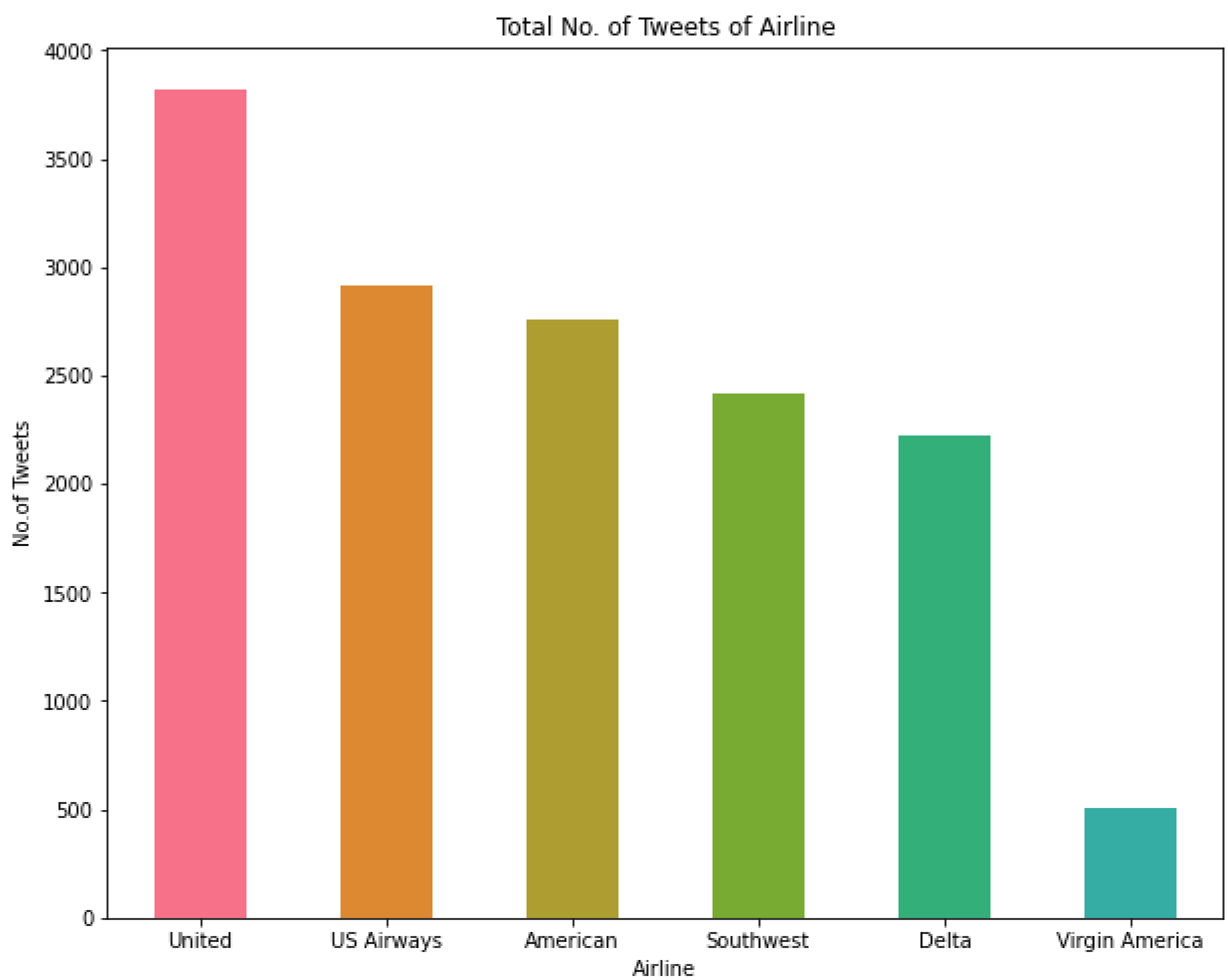


In[3]:

```
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] :

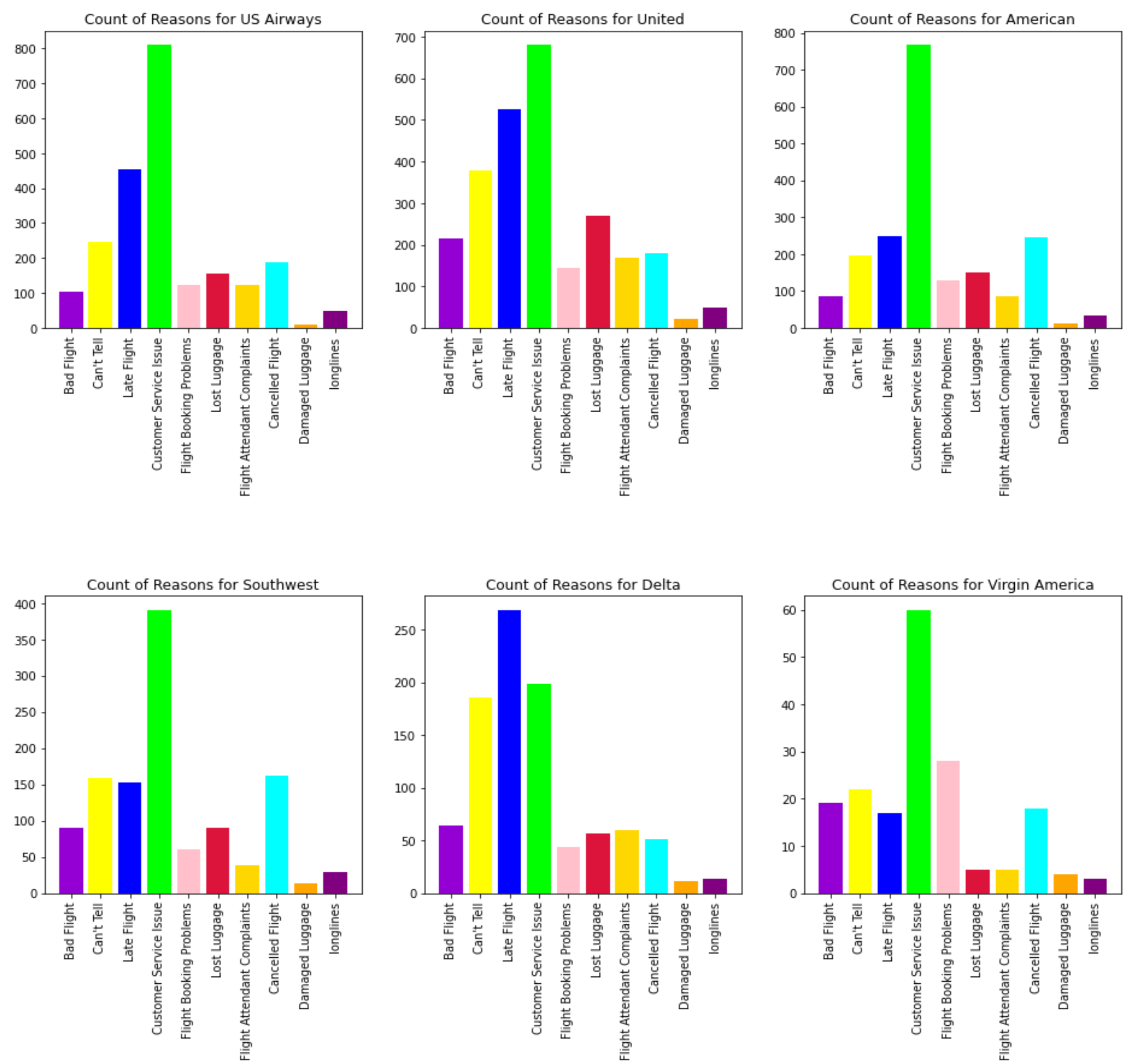
```
Text(0, 0.5, 'No.of Tweets')
```



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 )
```

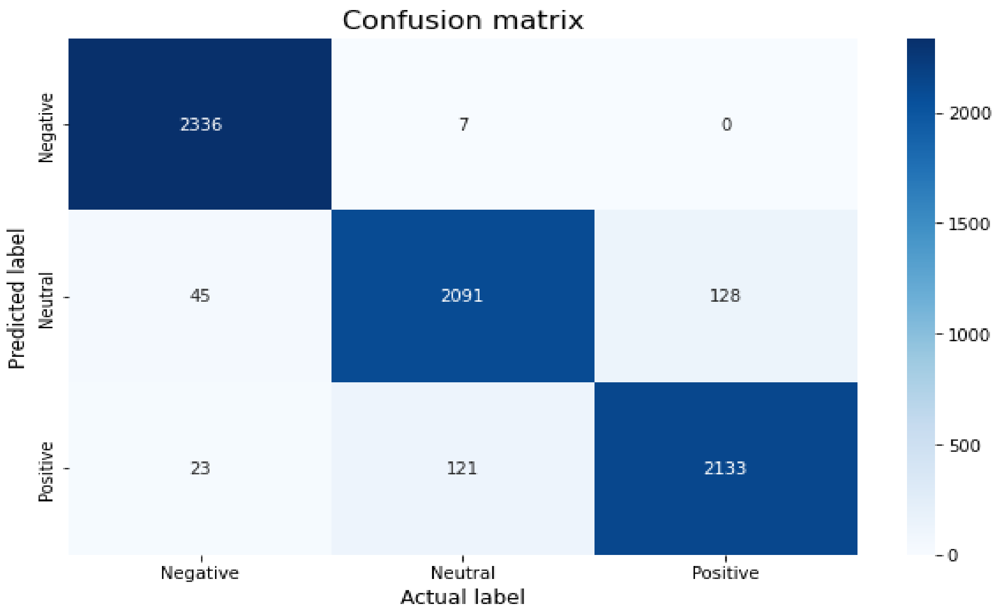


In[5]:

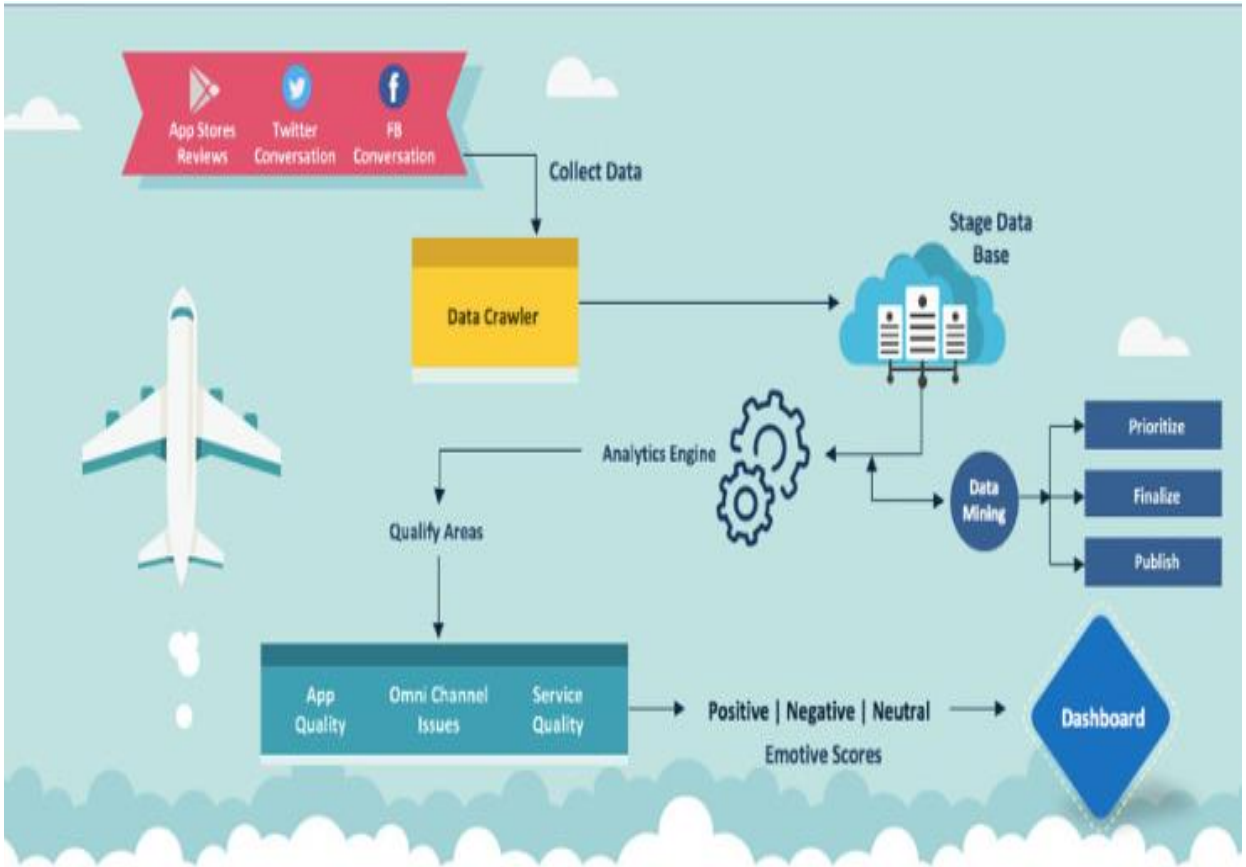
```
print("Classification Report:\n----------------------\n", cr)

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
```



4.PERFORMING DIFFERENT ACTIVITIES LIKE EMPLOYING NLP TECHNIQUES AND GENERATING INSIGHTS.



Employing NLP Techniques:

1. **Tokenization:** Break down the text into smaller pieces, like words or sentences.

2. **POS Tagging:** Identify the part of speech for each token.

3. **Named Entity Recognition:** Detect and classify entities like product names, brands, or user names.

4. **Sentiment Lexicons:** Use pre-existing lexicons that have sentiment scores for words, helping in sentiment determination.

5. **Deep Learning:** Implement techniques like LSTM or BERT for sentiment classification, especially when context matters.

Employing NLP Techniques:

In[1]:

```
text = "I love this product! It's amazing."
sentiment_score = afinn.score(text)

if sentiment_score > 0:
    sentiment = "positive"
elif sentiment_score < 0:
    sentiment = "negative"
else:
    sentiment = "neutral"

print(f"Sentiment: {sentiment}, Score: {sentiment_score}")
```

In[2]:

```
# Identify and print the most positive words
print('Most Positive Words')
for word, index in word_index_map.items():
    weight = model.coef_[0][index]
    if weight > threshold:
        print(word, weight)
```

```
Most Positive Words
great 5.516378614880334
```

```
virginamerica 3.4165631737297506
thank 8.172492647617368
southwestair 2.728627527382746
jetblue 3.1586422137139065
thanks 8.083441401654769
good 2.805464965619352
love 4.449114200749592
best 3.8620140153411207
appreciate 2.336612511736386
awesome 4.091284298701974
nice 2.16154339981104
thx 2.4222423243948117
amazing 3.6943805117897175
excellent 2.6209683927563843
worries 2.7557781608971568
wonderful 2.240905852132964
kudos 2.87036770762045
```

In[3]:

```python
# Vectorize text data using TF-IDF
vectorizer = TfidfVectorizer(max_features=2000)
x_train = vectorizer.fit_transform(df_train['text'])
x_test = vectorizer.transform(df_test['text'])
y_train = df_train['target']
y_test = df_test['target']
```

In[4]:

```python
# Vectorize text data for the binary sentiment classification
x_train = vectorizer.fit_transform(df_b_train['text'])
x_test = vectorizer.transform(df_b_test['text'])
y_train = df_b_train['target']
y_test = df_b_test['target']
```

In[5]:

```python
# 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_senti
ment'], 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)
```

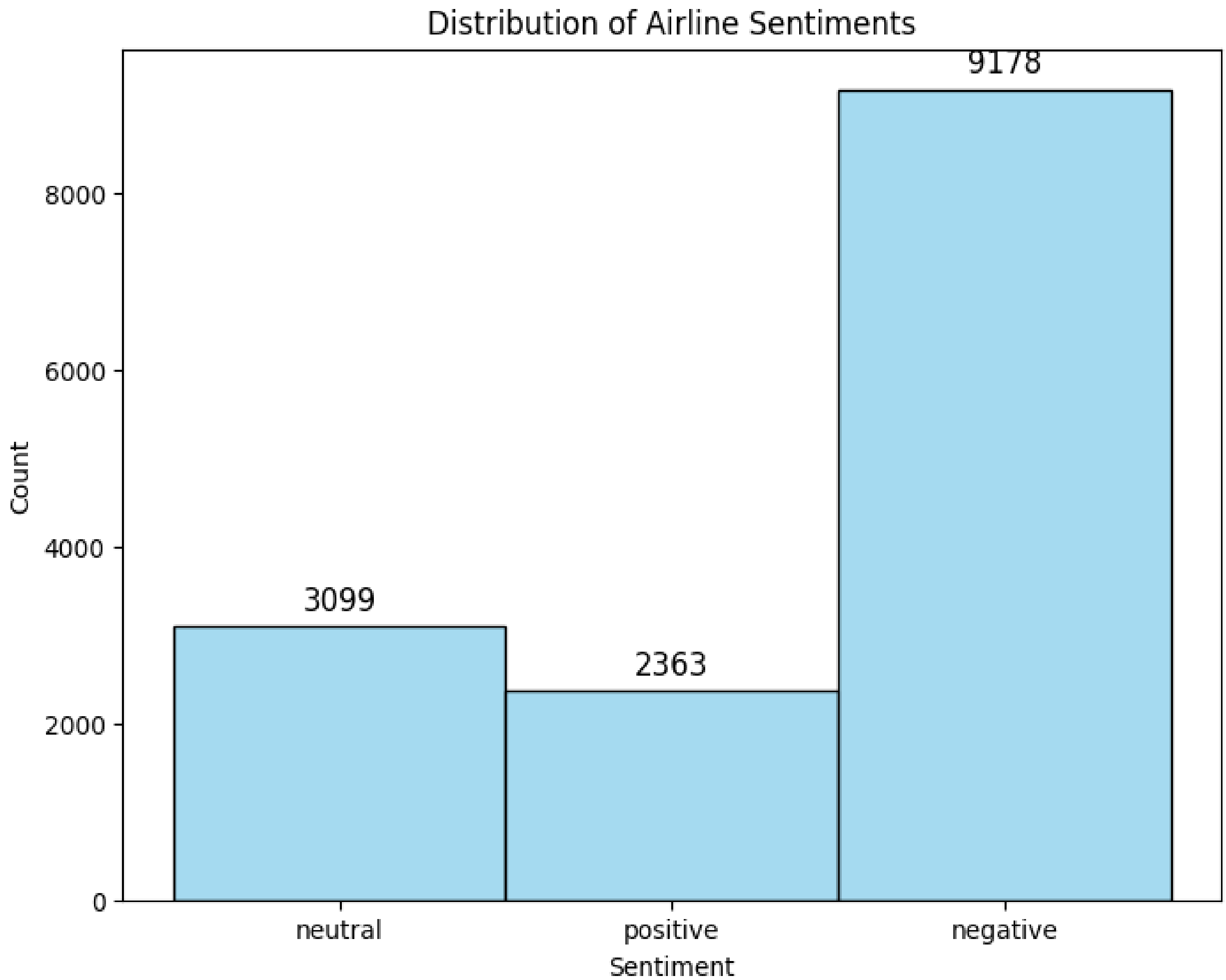
Out[5]:

RandomForestClassifier

```
RandomForestClassifier(n_estimators=1000, random_state=0)
```

In[6]:

```
# Count the occurrences of each sentiment category
sentiment_counts = df['airline_sentiment'].value_counts()

# Visualize the distribution using a histogram with counts on bars
plt.figure(figsize=(8, 6))
ax = sns.histplot(df['airline_sentiment'], bins=3, color='skyblue', discrete=True
)
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.title('Distribution of Airline Sentiments')

# Add counts on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_heigh
t()), ha='center', va='center', fontsize=12, xytext=(0, 10), textcoords='offset p
oints')

plt.xticks()
plt.show()
```



Generating Insights:

1.Sentiment Distribution: Analyze the overall sentiment spread, understanding if most feedback is positive, negative, or neutral.

2.Temporal Analysis: Check sentiment trends over time to identify any changes or anomalies.

3.Product-specific Insights: Delve deeper into sentiments about specific products or services, aiding in product improvement or feature addition.

4.Competitor Analysis: By analyzing sentiments about competitors, derive strategies to gain a competitive edge.

5.Target Audience Sentiments: Understanding the sentiments of different consumer demographics can guide targeted marketing strategies.

In[7]:

```
# 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(stopw
ords.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()
```

Out[7]:

In[8]:

	airline_sentiment	text
0	neutral	virginamerica what dhepburn said
1	positive	virginamerica plus you ve added commercials t...
2	neutral	virginamerica didn today must mean need to ta...
3	negative	virginamerica it really aggressive to blast o...
4	negative	virginamerica and it a really big bad thing a...

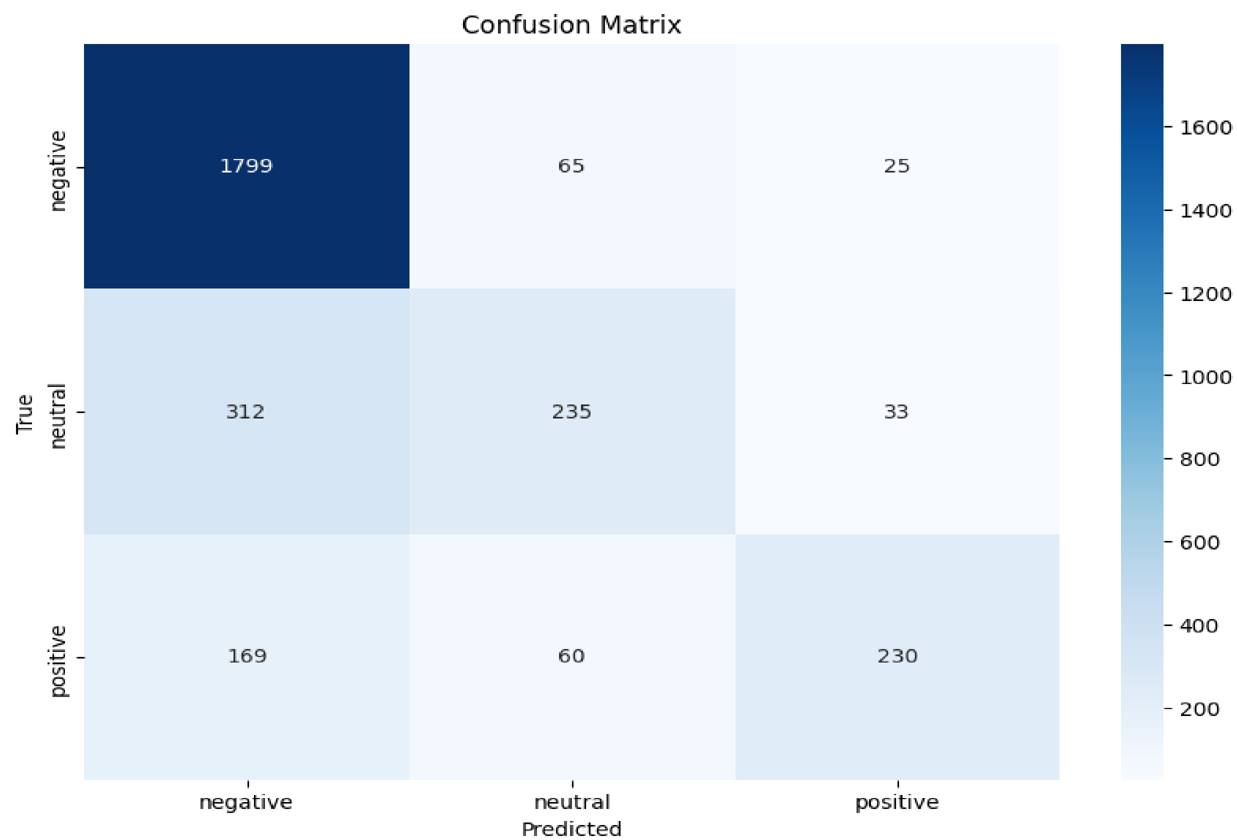
```
from sklearn .metrics imp
ort classifi cation_repor
t, confusion _matrix, acc
uracy_score

def evaluate _model(y_tes
t, y_pred):
    print('C lassificatio
n Report:')
    print(cl assification
_report(y_test, y_pred))
    print('Confusion Matrix:')
    print(confusion_matrix(y_test, y_pred))
    print('Accuracy Score:')
    print(accuracy_score(y_test, y_pred))

y_pred = classifier.predict(X_test)
evaluate_model(y_test, y_pred)
```

```
Classification Report:
              precision    recall  f1-score   support

    negative       0.79      0.95      0.86      1889
     neutral       0.65      0.41      0.50       580
    positive       0.80      0.50      0.62       459

    accuracy                           0.77      2928
   macro avg       0.75      0.62      0.66      2928
weighted avg       0.76      0.77      0.75      2928

Confusion Matrix:
[[1799   65   25]
```

```
 [ 312  235   33]
 [ 169   60  230]]
Accuracy Score:
0.773224043715847
```

In[9]:

```
import matplotlib.pyplot as plt
import seaborn as sns

def plot_confusion_matrix(y_test, y_pred):
    cm = confusion_matrix(y_test, y_pred)
    df_cm = pd.DataFrame(cm, index = [i for i in ['negative', 'neutral', 'positiv
e']],
                  columns = [i for i in ['negative', 'neutral', 'positive']])
    plt.figure(figsize = (10,7))
    sns.heatmap(df_cm, annot=True, fmt='d', cmap='Blues')
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

plot_confusion_matrix(y_test, y_pred)
```



In[10]:

```
# Train a logistic regression model
model = LogisticRegression(max_iter=500)
model.fit(x_train, y_train)
```

Out[10]:

```
LogisticRegression
 LogisticRegression(max_iter=500)
```

Model Evaluation:

1.Accuracy: Measure the fraction of predictions the model got right.

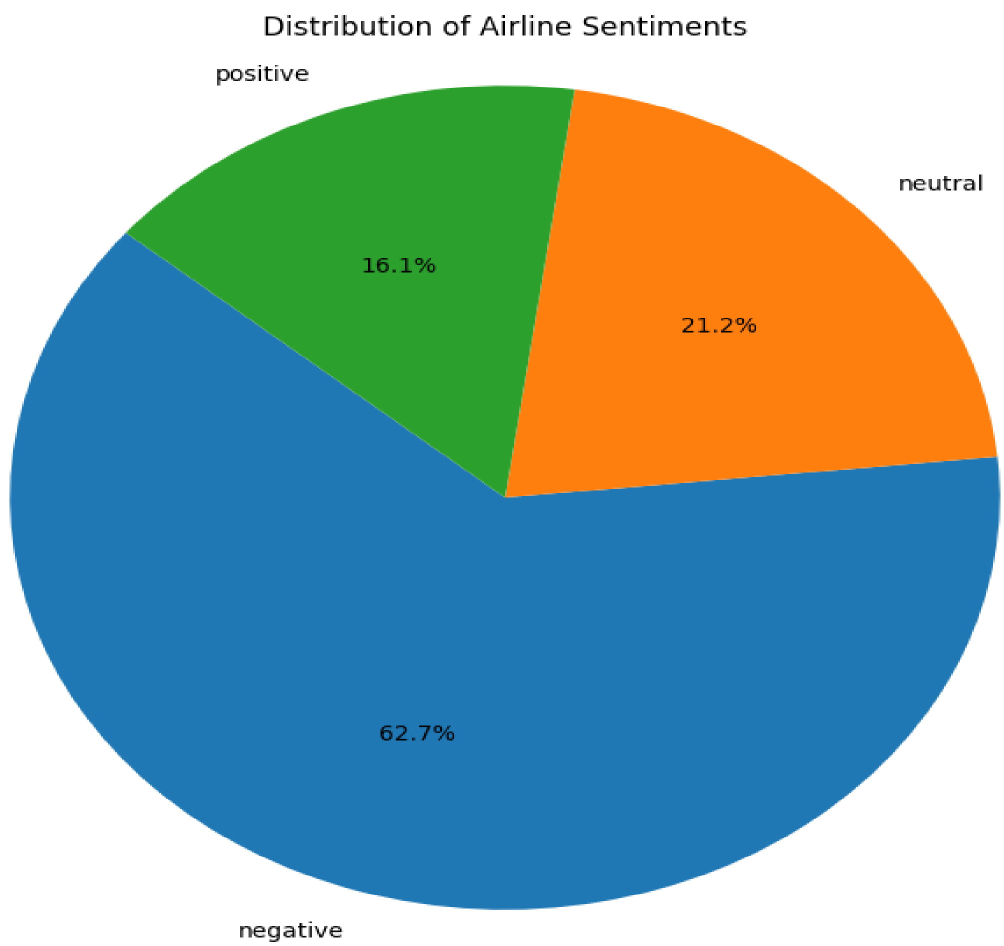
2.Precision & Recall: Gauge the model's ability to minimize false positives and false negatives.

3.F1-score: A balance between precision and recall.

4.Confusion Matrix: A summary of prediction results on a classification problem, highlighting false positives, false negatives, true positives, and true negatives.

In[11]:

```
# Visualize the distribution of airline sentiments using a pie chart
sentiment_counts = df['airline_sentiment'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%', start
angle=140)
plt.title('Distribution of Airline Sentiments')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

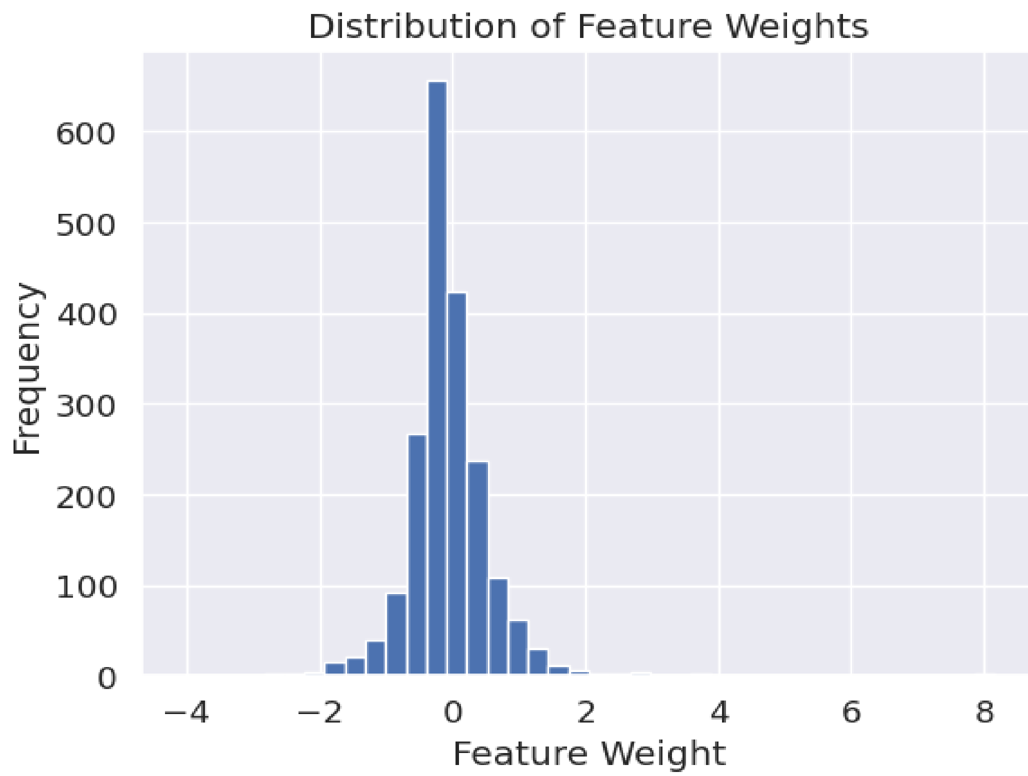


In[12]:

```
# Define a threshold for identifying most positive and most negative words
threshold = 2
```

In[13]:

```
# Plot a histogram of feature weights (coefficients)
plt.hist(model.coef_[0], bins=40)
plt.xlabel('Feature Weight')
plt.ylabel('Frequency')
plt.title('Distribution of Feature Weights')
plt.show()
```



In[14]:

```
from textblob import TextBlob


text = "I love this product! It's amazing."
blob = TextBlob(text)
sentiment = blob.sentiment


if sentiment.polarity > 0:
    sentiment_label = "positive"
elif sentiment.polarity < 0:
    sentiment_label = "negative"
```

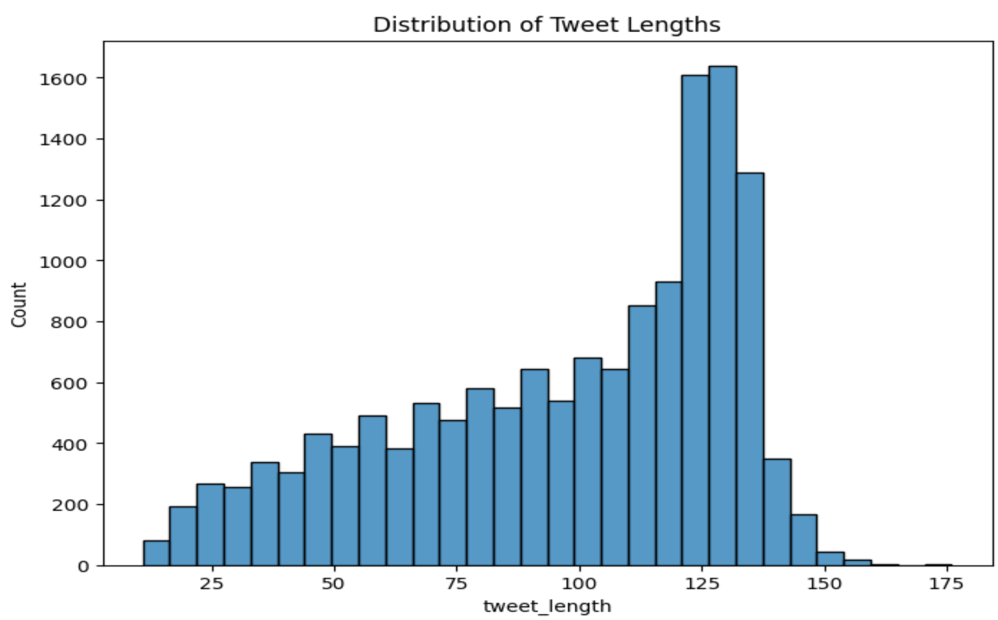
else:

    sentiment_label = "neutral"

print(f"Sentiment: {sentiment_label}, Polarity: {sentiment.polarity}")

In[15]:

```
# Boxplot of tweet lengths
plt.figure(figsize=(8,6))
sns.boxplot(x='airline_sentiment', y='tweet_length', data=df)
plt.title('Distribution of Tweet Lengths by Sentiment')
plt.show()
```



In[16]:

```
labels = list(crosstab_neg_reasons.columns)
values = [crosstab_neg_reasons[col_name].sum() for col_name in labels]

# Use `hole` to create a donut-like pie chart
fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
fig.update_layout(title='Overall distribution for negative reasons')
fig.show()
```

In[17]:

```
df.airline.value_counts()
```

Out[17]:

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

ADVANTAGES:

Twitter sentiment related to US airlines can provide several advantages:

1.Real-time Feedback:

- Twitter offers real-time feedback from passengers, enabling airlines to respond quickly to issues and improve customer service on the fly. Passengers can share their experiences, both positive and negative, immediately after their flights.

2. Customer Engagement:

- Airlines can engage with customers directly on Twitter, addressing their concerns, providing information, and building a more positive brand image.

3. Competitive Analysis:

- Airlines can monitor Twitter sentiment not only for their own brand but also for their competitors. This can help them identify areas where they can gain a competitive advantage and improve their services.

4. Trend Analysis:

- By analyzing trends in Twitter sentiment, airlines can identify emerging issues or opportunities. For example, they can spot common complaints and work to address them proactively.

5.Public Relations:

- Managing sentiment on Twitter is crucial for public relations. Airlines can use Twitter to communicate important updates, crisis management, and share positive news.

6.Marketing Insights:

- Airlines can gain insights into the preferences and interests of their passengers through Twitter sentiment analysis. This can inform marketing strategies and help target the right audience.

7.Customer Support:

- Twitter can be used as an additional channel for customer support. Passengers can ask questions and seek assistance, and airlines can respond in a timely manner.

8.Brand Loyalty:

- Effective engagement on Twitter can build brand loyalty among customers who feel heard and valued by the airline. This can lead to repeat business and positive word-of-mouth.

BENEFITS:

1. **Enhanced Understanding of Customer Sentiment:**
   Marketers can gauge public opinion on products, campaigns, and brands with high precision.
2. **Data-Driven Decision Making:**
   Insights from sentiment analysis inform marketing strategies, campaign adjustments, and product developments.
3. **Real-Time Analysis:**
   Sentiment can be monitored in real time, enabling quick responses to customer feedback.
4. **Adaptability to Market Conditions:**
   Machine learning models can evolve with market sentiments, keeping the analysis current.
5. **Market Transparency:**
   Helps in understanding the true perception of the brand, reducing gaps between brand messaging and public opinion.
6. **Risk Mitigation:**
   Can alert to negative sentiment early, allowing for proactive management of potential PR crise**s.**
7. **Customization**:
   Tailored to specific industries or markets for nuanced understanding.
8. **Cost Efficiency:**
   Automating sentiment analysis can reduce costs related to market research.
9. **Scalability:**

Applicable to small and large datasets, scalable according to business size and need.

10.Objectivity:

Offers an unbiased view of customer sentiment.

CONCLUSION:

Sentiment analysis in marketing is a cutting-edge approach that significantly enhances understanding of consumer behavior. This exploration underscores the technology's ability to deliver precise, data-driven sentiment insights, which are essential for successful marketing strategies. Key takeaways include:

- **Refined Consumer Insights:** The nuanced analysis of customer sentiment provides a wealth of knowledge, enabling marketers to fine-tune communications and product offerings.
- **Strategic Advantage:** Sentiment analysis offers marketers the ability to make informed decisions based on real-time data, leading to more effective campaigns and initiatives.
- **Market Responsiveness:** Real-time sentiment tracking allows for agile responses to consumer feedback, enhancing customer experience and brand reputation.
- **Considerations for Implementation:** Marketers must be aware of the challenges, including data privacy, model interpretability, and the need for continuous model maintenance.

As sentiment analysis technology advances, we can anticipate even more sophisticated tools that offer deeper insights and more actionable intelligence. For businesses and marketers, staying abreast of these developments will be crucial to harnessing the full potential of sentiment analysis, ensuring that marketing efforts are both responsive and resonant with the target audience. The future of marketing is data-driven, and sentiment analysis is a cornerstone in this evolving landscape.

PREPARED BY,

PUVIYARASI EZHILARASAN