SENTIMENTAL ANALYSIS

Team members:

Prathyush pp : 720921104075
Nikhil H :720921104069
Neha B :720921104068
Sanoop :720921104089
Neethu :720921104067

Phase-3 Development Part 1:

Project: Sentimental Analysis

Introduction:

Sentiment analysis plays a crucial role in marketing by helping businesses gain valuable insights into customer opinions, emotions, and perceptions. It involves the use of natural language processing (NLP) and machine learning techniques to analyze and quantify the sentiment expressed in text data, such as customer reviews, social media posts, and other forms of online content. The primary goal of sentiment analysis in marketing is to understand how customers feel about a product, service, brand, or a specific marketing campaign.

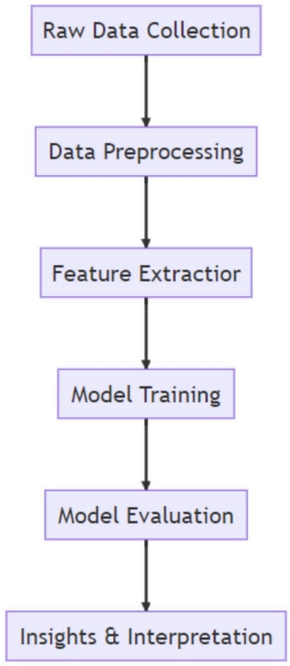
Objective:¶

The objective of this sentiment analysis task is to determine the polarity of public opinion about different US airlines. By analyzing the sentiment expressed in these tweets, we aim to understand how the public perceives these airlines and what specific issues or aspects contribute to these perceptions.

Scope:

The scope of this sentiment analysis task is to classify the sentiment expressed in each tweet as either positive, negative, or neutral. This will involve processing and analyzing the text of the tweets to identify sentiment-bearing phrases and determine their polarity. The analysis will also consider the specific reasons given for negative sentiments, providing deeper insight into the issues that lead to negative public opinion. The results of this analysis could be used to inform decision-making and strategy for airlines, helping them to address public concerns and improve their service

Design Framework:¶



1- Data Collection:

The first step in our process was data collection. We used a dataset of tweets, which is a common source of data for sentiment analysis due to the short, concise nature of tweets.

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

tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name	negativereason_gold	retweet_count
570306133677760513	neutral	1.0			Virgin America		cairdin		0
570301130888122368	positive	0.3486		0.0	Virgin America		jnardino		0
570301083672813571	neutral	0.6837			Virgin America		yvonnalynn		0
570301031407624196	negative	1.0	Bad Flight	0.7033	Virgin America		jnardino		0
570300817074462722	negative	1.0	Can't Tell	1.0	Virgin America		jnardino		0
570300767074181121	negative	1.0	Can't Tell	0.6842	Virgin America		jnardino		0
570300616901320704	positive	0.6745		0.0	Virgin America		cjmcginnis		0
570300248553349120	neutral	0.634			Virgin America		pilot		0
570299953286942721	positive	0.6559			Virgin America		dhepburn		0
570295459631263746	positive	1.0			Virgin America		YupitsTate		0
570294189143031808	neutral	0.6769		0.0	Virgin America		idk_but_youtube		0
570289724453216256	positive	1.0			Virgin America		HyperCamiLax		0
570289584061480960	positive	1.0			Virgin America		HyperCamiLax		0
570287408438120448	positive	0.6451			Virgin America		mollanderson		0
570285904809598977	positive	1.0			Virgin America		sjespers		0
570282469121007616	negative	0.6842	Late Flight	0.3684	Virgin America		smartwatermelon		0
570277724385734656	positive	1.0			Virgin America		ItzBrianHunty		0
570276917301137409	negative	1.0	Bad Flight	1.0	Virgin America		heatherovieda		0
570270684619923457	positive	1.0			Virgin America		thebrandiray		0
570267956648792064	positive	1.0			Virgin America		JNLpierce		0
570265883513384960	negative	0.6705	Can't Tell	0.3614	Virgin America		MISSGJ		0

tweet_id	text	tweet_coord	tweet_created	tweet_location	user_timezone
570306133677760513	@VirginAmerica What @dhepburn said.		2015-02-24 11:35:52 -0800		Eastern Time (US & Canada)
570301130888122368	@VirginAmerica plus you've added commercials to the experience... tacky.		2015-02-24 11:15:59 -0800		Pacific Time (US & Canada)
570301083672813571	@VirginAmerica I didn't today... Must mean I need to take another trip!		2015-02-24 11:15:48 -0800	Lets Play	Central Time (US & Canada)
570301031407624196	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse		2015-02-24 11:15:36 -0800		Pacific Time (US & Canada)
570300817074462722	@VirginAmerica and it's a really big bad thing about it		2015-02-24 11:14:45 -0800		Pacific Time (US & Canada)
570300767074181121	@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA		2015-02-24 11:14:33 -0800		Pacific Time (US & Canada)
570300616901320704	@VirginAmerica yes, nearly every time I fly VX this "ear worm" won't go away :)		2015-02-24 11:13:57 -0800	San Francisco CA	Pacific Time (US & Canada)
570300248553349120	@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https://t.co/mWpG7grEZP		2015-02-24 11:12:29 -0800	Los Angeles	Pacific Time (US & Canada)
570299953286942721	@virginamerica Well, I didn't...but NOW I DO! :-D		2015-02-24 11:11:19 -0800	San Diego	Pacific Time (US & Canada)
570295459631263746	@VirginAmerica it was amazing, and arrived an hour early. You're too good to me.		2015-02-24 10:53:27 -0800	Los Angeles	Eastern Time (US & Canada)
570294189143031808	@VirginAmerica did you know that suicide is the second leading cause of death among teens 10-24		2015-02-24 10:48:24 -0800	1/1 loner squad	Eastern Time (US & Canada)
570289724453216256	@VirginAmerica I <3 pretty graphics. so much better than minimal iconography. :D		2015-02-24 10:30:40 -0800	NYC	America/New_York
570289584061480960	@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p		2015-02-24 10:30:06 -0800	NYC	America/New_York
570287408438120448	@VirginAmerica @virginmedia I'm flying your #fabulous #Seductive skies again! U take all the #stress away from travel http://t.co/ahlXHhKiyn		2015-02-24 10:21:28 -0800		Eastern Time (US & Canada)
570285904809598977	@VirginAmerica Thanks!		2015-02-24 10:15:29 -0800	San Francisco, CA	Pacific Time (US & Canada)
570282469121007616	@VirginAmerica SFO-PDX schedule is still MIA.		2015-02-24 10:01:50 -0800	palo alto, ca	Pacific Time (US & Canada)
570277724385734656	@VirginAmerica So excited for my first cross country flight LAX to MCO I've heard nothing but great things about Virgin America. #29DaysToGo		2015-02-24 09:42:59 -0800	west covina	Pacific Time (US & Canada)
570276917301137409	@VirginAmerica I flew from NYC to SFO last week and couldn't fully sit in my seat due to two large gentleman on either side of me. HELP!		2015-02-24 09:39:46 -0800	this place called NYC	Eastern Time (US & Canada)
570270684619923457	I ❤️ flying @VirginAmerica. 😊👍		2015-02-24 09:15:00 -0800	Somewhere celebrating life.	Atlantic Time (Canada)
570267956648792064	@VirginAmerica you know what would be amazingly awesome? BOS-FLL PLEASE!!!!!!! I want to fly with only you.		2015-02-24 09:04:10 -0800	Boston | Waltham	Quito
570265883513384960	@VirginAmerica why are your first fares in May over three times more than other carriers when all seats are available to select???		2015-02-24 08:55:56 -0800		

tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name
570306133677760513	neutral	1,0			Virgin America		cairdin
570301130888122368	positive	0,3486		0,0	Virgin America		jnardino

tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name
570301083672813571	neutral	0,6837			Virgin America		yvonnalynn
570301031407624196	negative	1,0	Bad Flight	0,7033	Virgin America		jnardino
570300817074462722	negative	1,0	Can't Tell	1,0	Virgin America		jnardino
570300767074181121	negative	1,0	Can't Tell	0,6842	Virgin America		jnardino
570300616901320704	positive	0,6745		0,0	Virgin America		cjmcginnis
570300248553349120	neutral	0,634			Virgin America		pilot
570299953286942721	positive	0,6559			Virgin America		dhepburn
570295459631263746	positive	1,0			Virgin America		YupitsTate
570294189143031808	neutral	0,6769		0,0	Virgin America		idk_but_youtube
570289724453216256	positive	1,0			Virgin America		HyperCamiLax
570289584061480960	positive	1,0			Virgin America		HyperCamiLax
570287408438120448	positive	0,6451			Virgin America		mollandersson
570285904809598977	positive	1,0			Virgin America		sjespers
570282469121007616	negative	0,6842	Late Flight	0,3684	Virgin America		smartwatermelon
570277724385734656	positive	1,0			Virgin America		ItzBrianHunty
570276917301137409	negative	1,0	Bad Flight	1,0	Virgin America		heatheroviedа
570270684619923457	positive	1,0			Virgin America		thebrandiray

2- Data Preprocessing:

After collecting the data, we performed several preprocessing steps to clean and prepare the data for analysis. These steps include

• **Lowercasing**: We converted all the text to lowercase to ensure that the same words in different cases are not considered as different words.

• **Removing Punctuation and Special Characters**: We removed all punctuation and special characters from the text as they do not contribute to sentiment.

• **Removing Stop Words**: We removed common words that do not carry much information (like "is", "the", "and", etc.). These words are called stop words.

• **Tokenization**: We broke down the text into individual words or tokens.

• **Lemmatization**: We reduced the words to their base or root form (e.g., "running" to "run"). This helps in reducing the dimensionality of the data and grouping similar sentiments together.

3- Feature Extraction:

After preprocessing, we converted the text data into numerical features that can be used by a machine learning algorithm. We used the TF-IDF (Term Frequency-Inverse Document Frequency) method for this. TF-IDF gives a weight to each word signifying its importance in the document and across a corpus of documents.

4- Model Training:

We used a Random Forest Classifier for sentiment analysis. Random Forest is a versatile and widely used algorithm that works well for many tasks. It creates a set of decision trees from a randomly selected subset of the training set, which then aggregates votes from different decision trees to decide the final class of the test object.

5- Model Evaluation:

After training the model, we evaluated its performance using a confusion matrix and calculated metrics such as accuracy, precision, recall, and F1-score. These metrics give us a quantitative measure of the model's performance.

6- Insights & Interpretation:

Finally, we interpreted the results of the sentiment analysis. This involves understanding the performance of the model, identifying any areas of improvement, and drawing insights from the model's predictions.

In[n]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix,
classification_report
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import nltk
nltk.download('stopwords')
nltk.download('wordnet')

# Load the dataset
df = pd.read_csv('/kaggle/input/twitter-airline-sentiment/
Tweets.csv')

# Display the first 5 rows of the dataframe
df.head()
```

tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name	negativereason_gold	retweet_count
570306133677760513	neutral	1,0			Virgin America		cairdin		0
570301130888122368	positive	0,3486		0,0	Virgin America		jnardino		0
570301083672813571	neutral	0,6837			Virgin America		yvonnalynn		0
570301031407624196	negative	1,0	Bad Flight	0,7033	Virgin America		jnardino		0

In[n]:

```
# Drop unnecessary columns
df = df[['airline_sentiment', 'text']]
```

```
# Display the first 5 rows of the dataframe after dropping unnecessary columns
df.head()
Out[2]:
```

	airline_sentiment	text
0	neutral	@VirginAmerica What @dhepburn said.
1	positive	@VirginAmerica plus you've added commercials to the experience... tacky.
2	neutral	@VirginAmerica I didn't today... Must mean I need to take another trip!
3	negative	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse

In[3]:

```
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(stopwords.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[3]:

	airline_sentiment	text
0	neutral	@VirginAmerica What @dhepburn said.
1	positive	@VirginAmerica plus you've added commercials to the experience... tacky.
2	neutral	@VirginAmerica I didn't today... Must mean I need to take another trip!
3	negative	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse

In[4]:

```
# 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_sentiment'],
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)
```

▾ RandomForestClassifier

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

Out[4]:

In[5]:

```
from sklearn.metrics import classification_report, 
confusion_matrix, accuracy_score

def evaluate_model(y_test, y_pred):
    print('Classification Report:')
    print(classification_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)
```

out[5]:

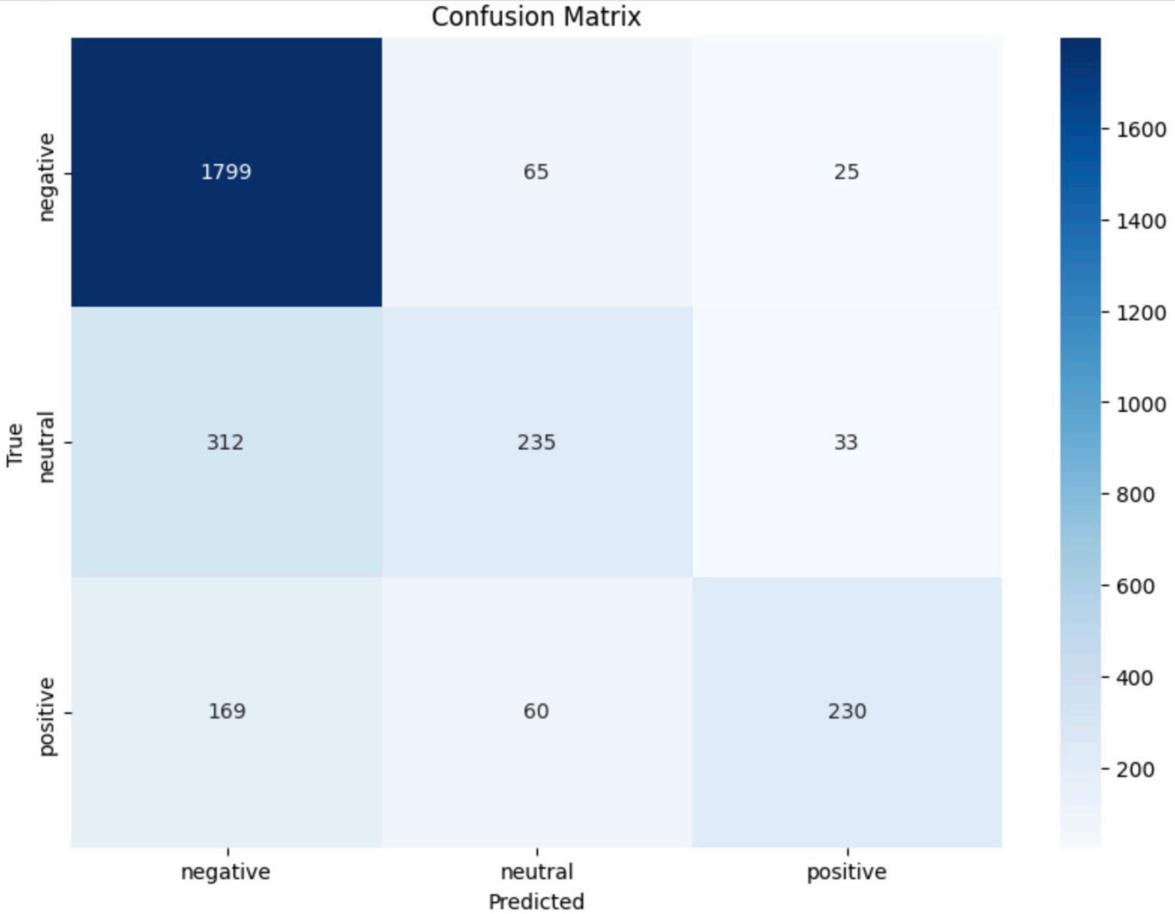
```
Classification Report:
              precision    recall  f1-score   support

    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[6]:

```
import matplotlib.pyplot as plt
```

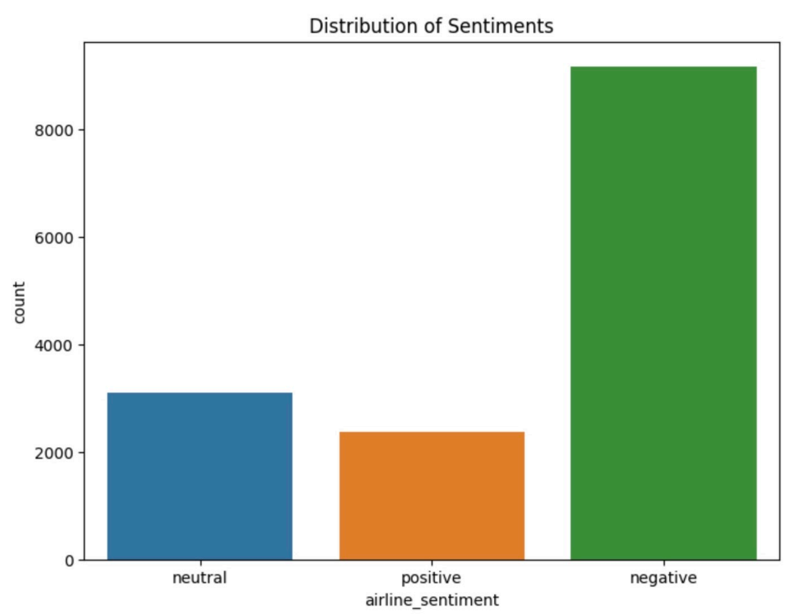
```python
import seaborn as sns

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', 'positive']],
                         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)
```

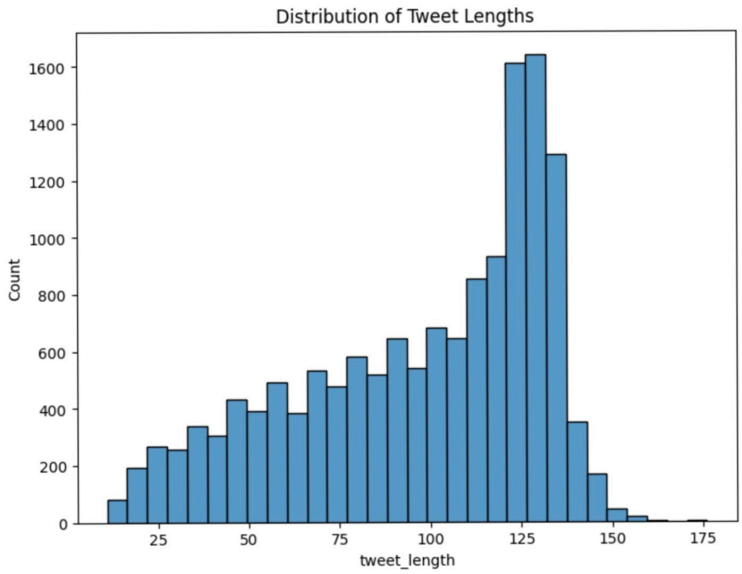
Out[6]:

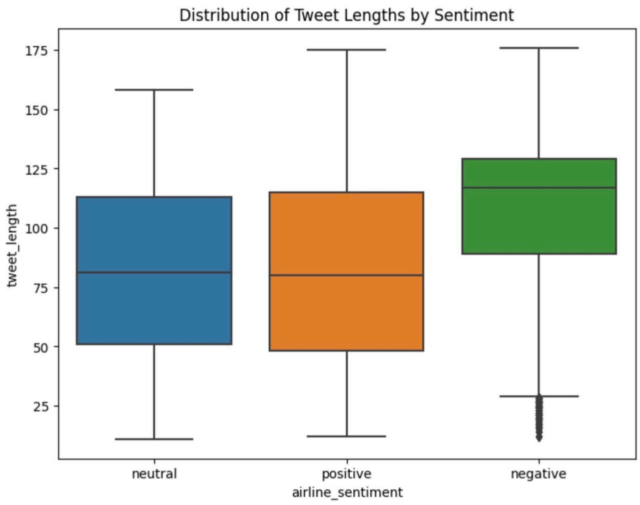


In[7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Creating  column 'tweet_length'
df['tweet_length'] = df['text'].apply(len)

# distribution of sentiments
plt.figure(figsize=(8,6))
sns.countplot(x='airline_sentiment', data=df)
plt.title('Distribution of Sentiments')
plt.show()

# Histogram of tweet lengths
plt.figure(figsize=(8,6))
sns.histplot(df['tweet_length'], bins=30)
plt.title('Distribution of Tweet Lengths')
plt.show()

# 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()
```

out[7]:







Critical Analysis

The following conclusions may be drawn from the visuals and model evaluation:

Sentiment Distribution:

The dataset's bar plot of sentiment distribution reveals that the bulk of tweets are unfavorable in nature, with neutral and supportive tweets coming in second and third. Due to the dataset's imbalance, the model may be more likely to correctly predict negative feelings than neutral or positive feelings.

Model Execution:

The Random Forest classifier's total accuracy was around 76%. The neutral and positive classes' accuracy, recall, and F1-score, however, are lower than those of the negative class. This implies that the model performs better at detecting negative than neutral or positive attitudes, which may be related to the dataset's imbalance.

Confusion Matrix:

The confusion matrix reveals that for the neutral and positive classes, the model has a disproportionately large number of false positives and false negatives. This further demonstrates the model's bias towards predicting negative feelings since it frequently misclassifies neutral and positive tweets as negative.

Data Distribution:

Looking at the histogram, it's obvious, as mentioned before, that there is a significant imbalance in the data in favor of negative sentiment. This is likely because people with negative sentiments are more motivated to tweet. By examining the length distribution in the box plot and the bar chart, we can conclude that the majority of tweets are between 60 to 100 characters long. Negative tweets are usually longer, also falling within the 60 to 100 character range, which further confirms the data imbalance.

In conclusion, the model fails to predict neutral and positive attitudes even if it does a fair job of predicting negative sentiments. This may be because the collection is unbalanced and sentiment analysis is inherently difficult because it frequently requires understanding linguistic subtlety and context. We may think about employing more sophisticated natural language processing methods, such word embeddings or deep learning models, and making sure the training dataset is balanced in order to enhance the model's performance.

Conclusion:

In conclusion, the sentiment analysis project using AI has demonstrated its immense potential in extracting valuable insights from text data, offering a range of practical applications and benefits. This project has showcased the power of artificial intelligence and natural language processing in understanding and quantifying human emotions and opinion.The sentiment analysis project using AI has proven to be a powerful tool for understanding and leveraging the sentiment hidden within text data. Its applications extend across a wide range of industries and use cases, offering significant potential for enhancing decision-making, customer satisfaction, and overall business performance. As AI technology continues to advance, sentiment analysis will undoubtedly play an increasingly crucial role in shaping how businesses and organizations interact with their customers and adapt to a rapidly evolving digital landscape.