

PHASE-3

Data Visualization

Sentiment Analysis for Marketing:

Date	17 October 2023
TEAM ID	Proj_212173_Team_1
Project Name	Sentiment analysis for Marketing

Data visualization is the graphical representation of data to help people understand and interpret information more effectively. It involves creating visual representations of data, such as charts, graphs, maps, and other visual elements, to reveal patterns, trends, relationships, and insights.

Data visualization is a valuable tool for analyzing and presenting the results of sentiment analysis, which involves assessing and understanding the emotions and opinions expressed in text data.

Here are some common data visualizations used for sentiment analysis:

1. Heatmap
2. Bar chart
3. Pie chart
4. Scatterplot
5. Wordcloud

These visualizations can help you identify trends, patterns, and outliers in sentiment data and make data-driven decisions based on the analysis results.

Program:

Libraries used in Data Visualization:

Matplotlib- Matplotlib is a cross-platform, data visualization and graphical plotting library (histograms, scatter plots, bar charts, etc.) for Python and its numerical extension NumPy.

Seaborn- Seaborn is a library for making statistical graphics in Python.

Various plots:

Bar chart- The bar chart displays data using a number of bars, each representing a particular category.

Pie chart- The basic task of sentiment analysis is to determine the sentiment polarity (positivity, neutrality or negativity) of a piece text.

Heatmap- Heatmaps in Sentiment Analysis are used to visualize the lexicon's intensity of negative and positive scores and its impact on the sentence's sentiment.

Scatterplot- A Scatter Plot is widely used by seasoned data visualization experts to display the causal relationships between two variables.

Word cloud- A word cloud is a visual representation of text data, where the size of each word or phrase is determined by its frequency in the given text.

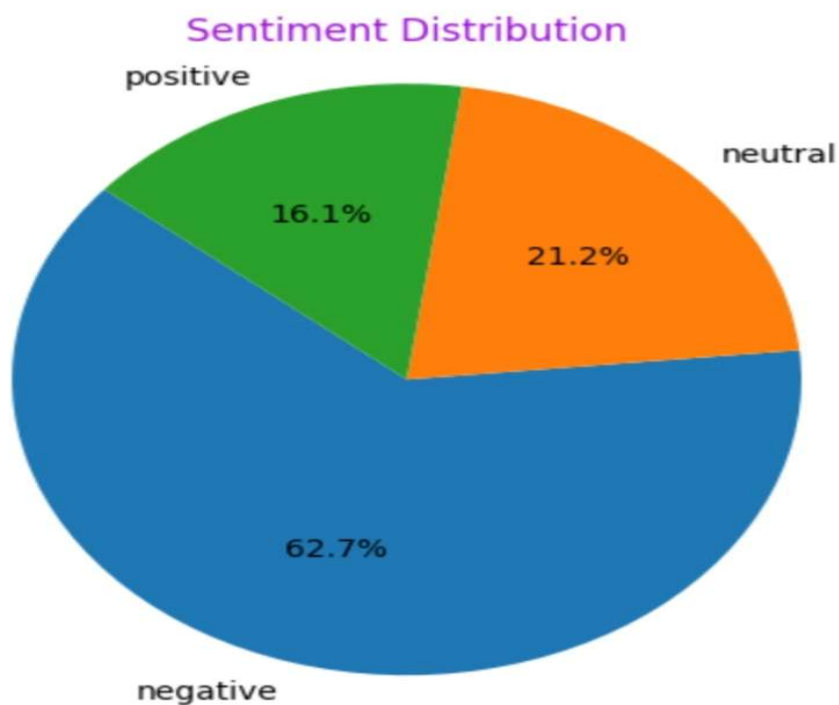
Violin plot- A violin plot for sentiment analysis is a data visualization that's used to compare and visualize the distribution of sentiment scores or confidence levels across different categories or groups

Pie chart for Sentiment analysis:



```
sentiment_counts = df['airline_sentiment'].value_counts()
labels = sentiment_counts.index
sizes = sentiment_counts.values
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title("Sentiment Distribution",color='#a114de')
plt.show()
```

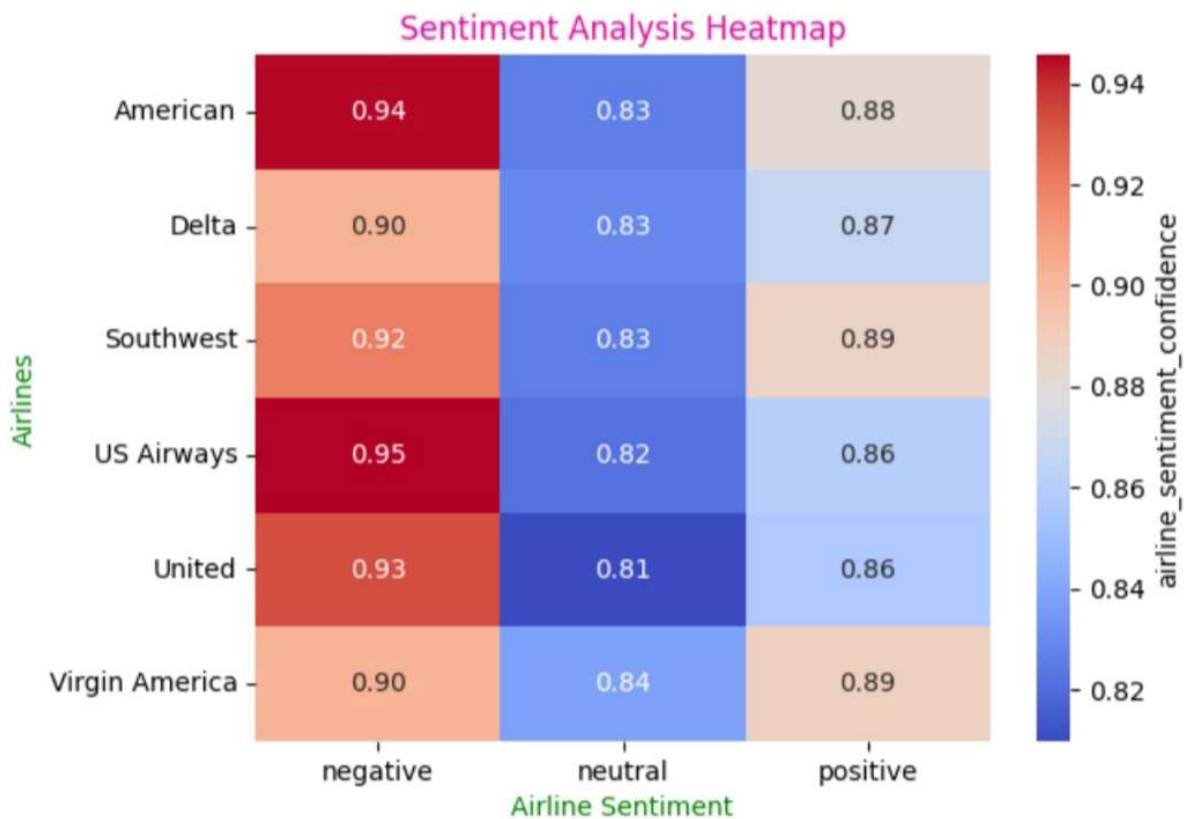
This code creates a pie chart to visualize the distribution of sentiments in a dataset.



Heatmap:

```
[ ] heatmap_data = df.pivot_table(index='airline', columns='airline_sentiment', values='airline_sentiment_confidence', aggfunc='mean')
sns.heatmap(heatmap_data, cmap="coolwarm", annot=True, fmt=".2f", cbar_kws={'label': 'airline_sentiment_confidence'})
plt.xlabel('Airline Sentiment',color='green')
plt.ylabel('Airlines',color='green')
plt.title('Sentiment Analysis Heatmap',color='#e6079b')
plt.show()
```

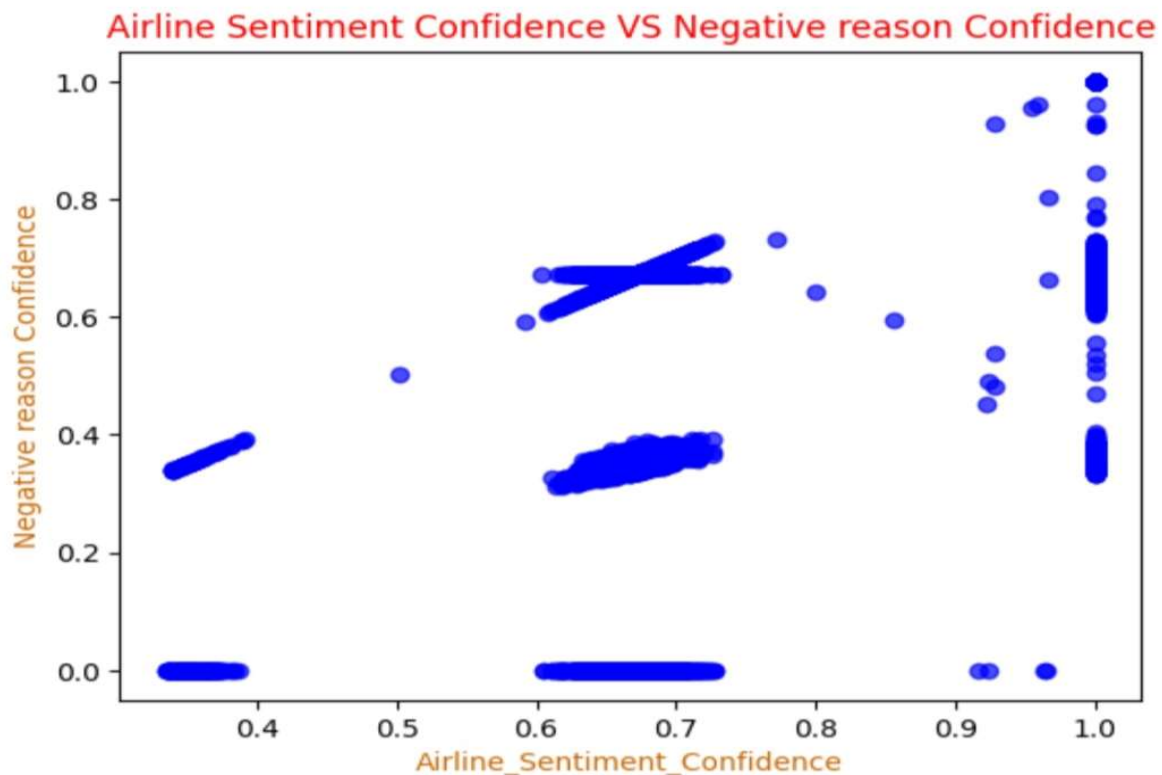
This code creates a heatmap to visualize the sentiment analysis results for different airlines



Scatterplot for Airline Sentiment confidence and Negative reason confidence:

```
▶ x=df['airline_sentiment_confidence']  
y=df['negativereason_confidence']  
plt.scatter(x, y, marker='o', color='blue', alpha=0.7)  
plt.xlabel('Airline_Sentiment_Confidence',color='#c96806')  
plt.ylabel('Negative reason Confidence',color='#c96806')  
plt.title('Airline Sentiment Confidence VS Negative reason Confidence',color='red')  
plt.show()
```

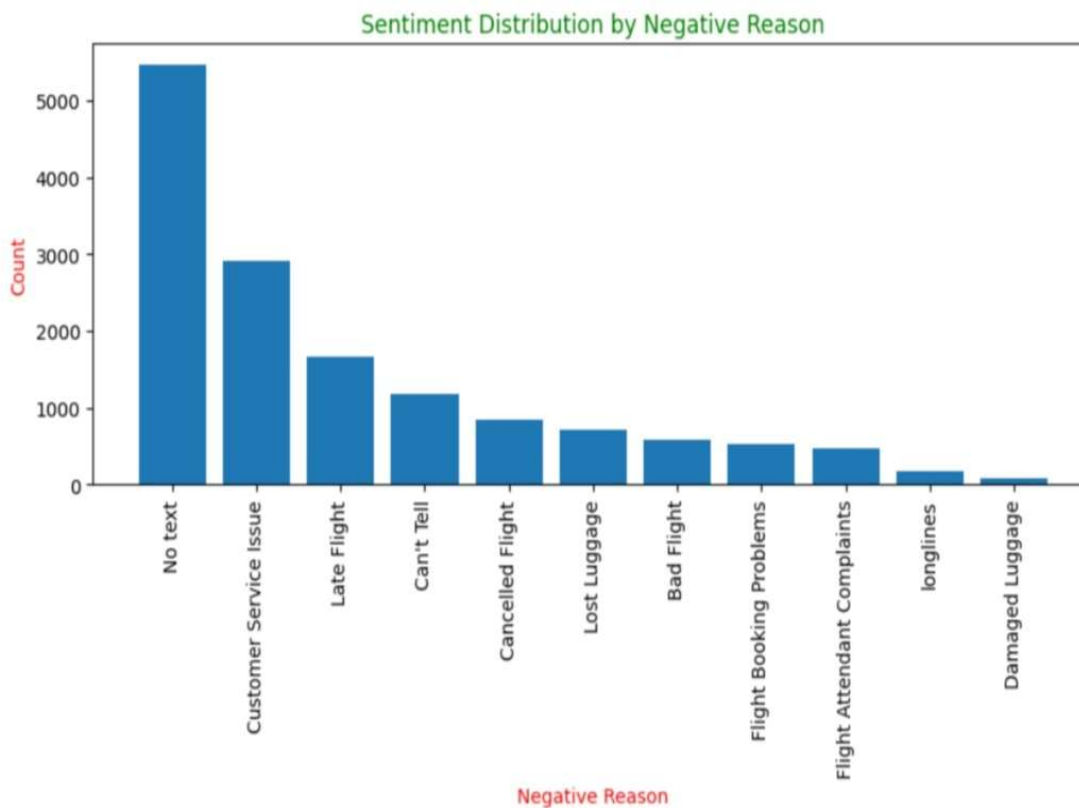
This code creates a scatter plot to visualize the relationship between two variables: 'Airline_Sentiment_Confidence' (x-axis) and 'Negative reason Confidence' (y-axis).



Bar chart Sentiment distribution by negative reason:

```
negative_reason_counts = df['negativereason'].value_counts()
x = negative_reason_counts.index
y = negative_reason_counts.values
plt.figure(figsize=(10, 3))
plt.bar(x,y)
plt.xlabel('Negative Reason',color='red')
plt.ylabel('Count',color='red')
plt.title('Sentiment Distribution by Negative Reason',color='green')
plt.xticks(rotation=90)
plt.show()
```

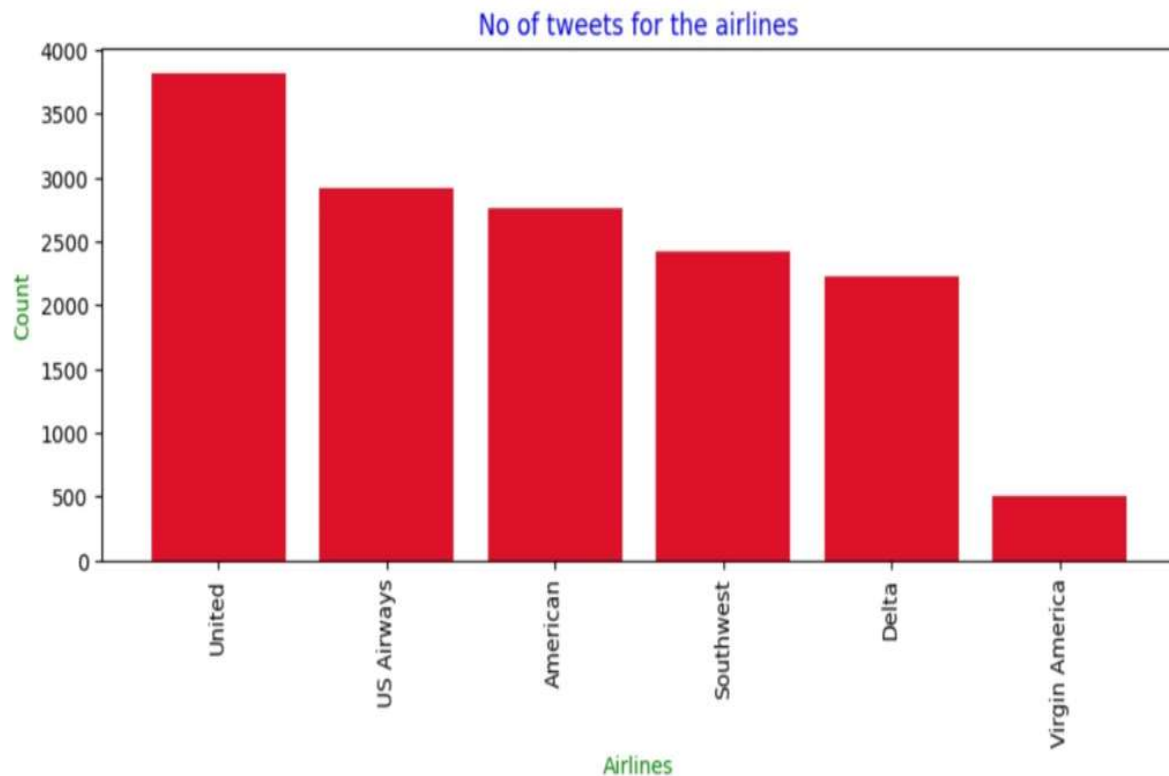
This code creates a bar chart to visualize the distribution of sentiment by negative reason



Bar chart for Number of tweets for the airlines:

```
[ ] airline_counts=df['airline'].value_counts()
x=airline_counts.index
y=airline_counts.values
plt.figure(figsize=(10, 4))
plt.bar(x,y,color='#de122a')
plt.xlabel('Airlines',color='green')
plt.ylabel('Count',color='green')
plt.title('No of tweets for the airlines',color='blue')
plt.xticks(rotation=90)
plt.show()
```

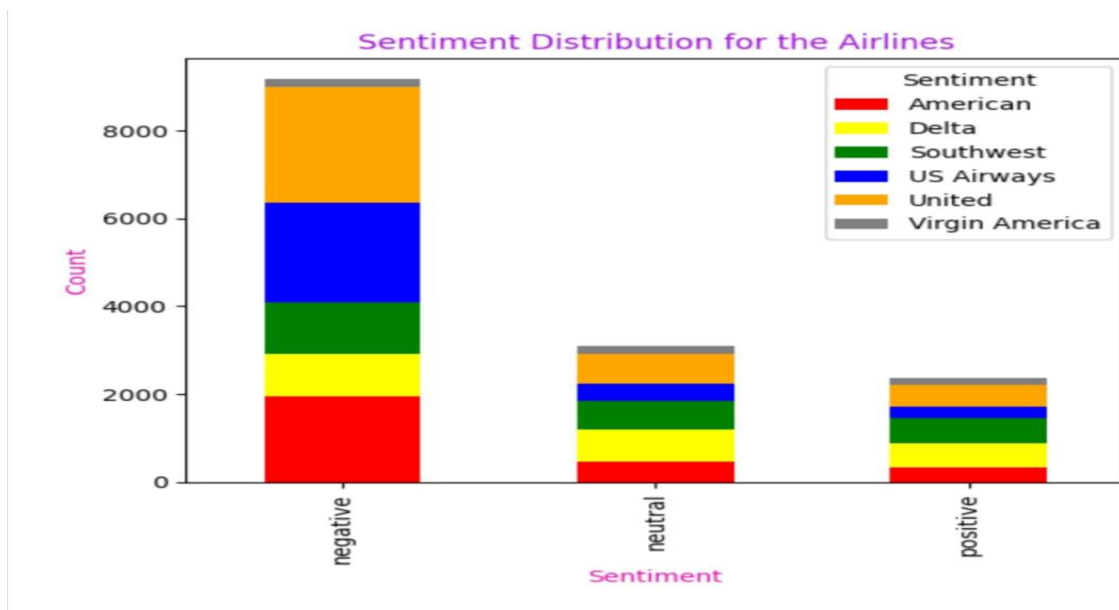
This code creates a bar chart to visualize the number of tweets for each airline.



Bar chart for Sentiment distribution for the airlines:

```
[ ]  
sentiment_counts = df.groupby(['airline_sentiment', 'airline']).size().unstack(fill_value=0)  
colors = ['red', 'yellow', 'green', 'blue', 'orange', 'grey']  
sentiment_counts.plot(kind='bar', stacked=True, color=colors)  
plt.xlabel('Sentiment', color='#e310ab')  
plt.ylabel('Count', color='#e310ab')  
plt.title('Sentiment Distribution for the Airlines', color='#a114de')  
plt.legend(title='Sentiment', loc='upper right')  
plt.show()
```

This code creates a stacked bar chart to visualize the sentiment distribution for different airlines.



Subplots:

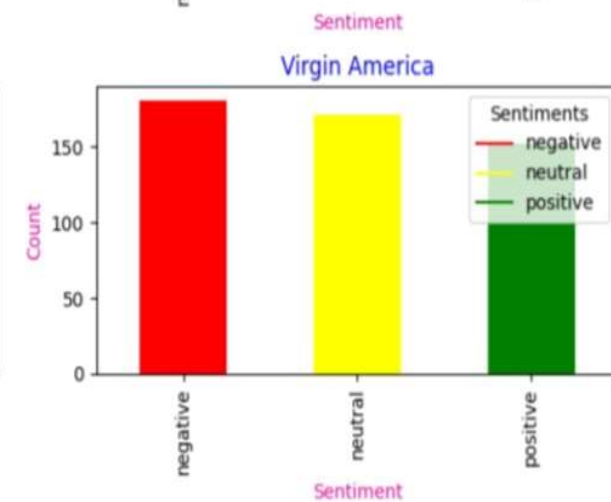
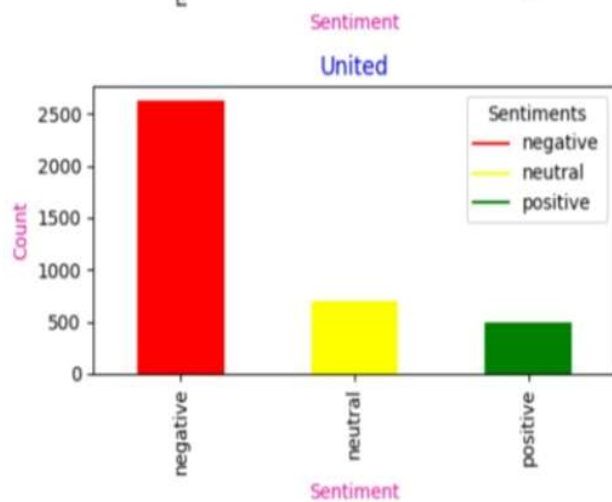
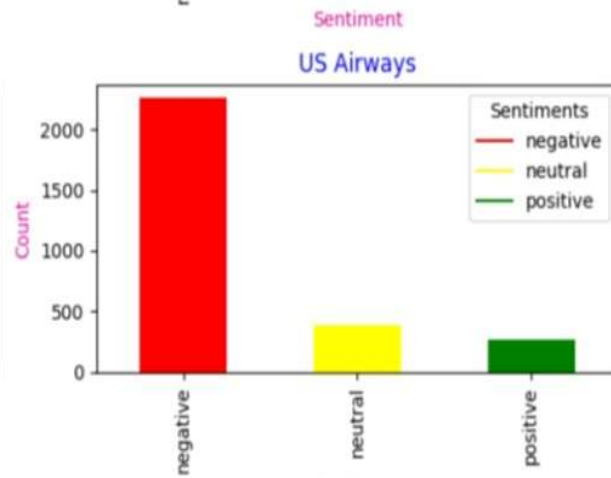
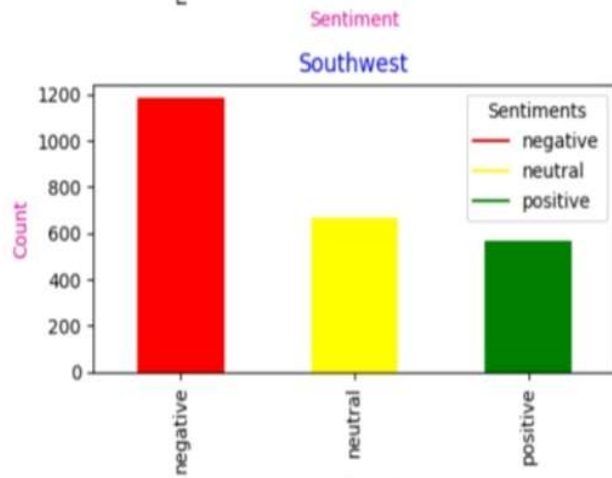
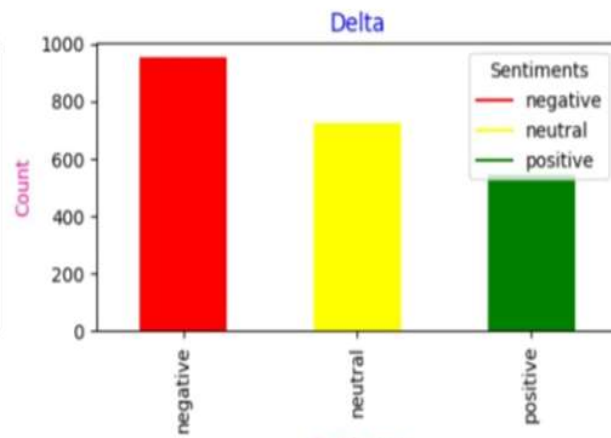
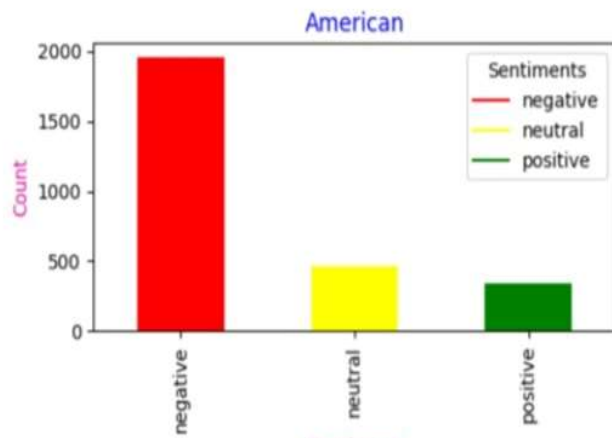
```
[ ] sentiment_counts = df.groupby(['airline', 'airline_sentiment']).size().unstack(fill_value=0)
unique_airlines = sentiment_counts.index
fig, axes = plt.subplots(3, 2, figsize=(10, 10))
axes = axes.flatten()
colors = ['red', 'yellow', 'green']
legend_dict = {
    'negative': 'red',
    'neutral': 'yellow',
    'positive': 'green'
}

for i, j in enumerate(unique_airlines):
    sentiment_counts.loc[j].plot(kind='bar', stacked=True, ax=axes[i], color=[legend_dict[c] for c in sentiment_counts.columns])
    axes[i].set_title(j, color='blue')
    axes[i].set_xlabel('Sentiment', color='#e6079b')
    axes[i].set_ylabel('Count', color='#e6079b')
    legend_handles = [plt.Line2D([0], [0], color=legend_dict[sentiment], label=sentiment) for sentiment in sentiment_counts.columns]
    axes[i].legend(handles=legend_handles, title='Sentiments', loc='upper right')

plt.tight_layout()
plt.show()
```

The resulting visualization consists of stacked bar charts for each airline, illustrating the distribution of sentiment categories ('negative', 'neutral', 'positive') within each airline.

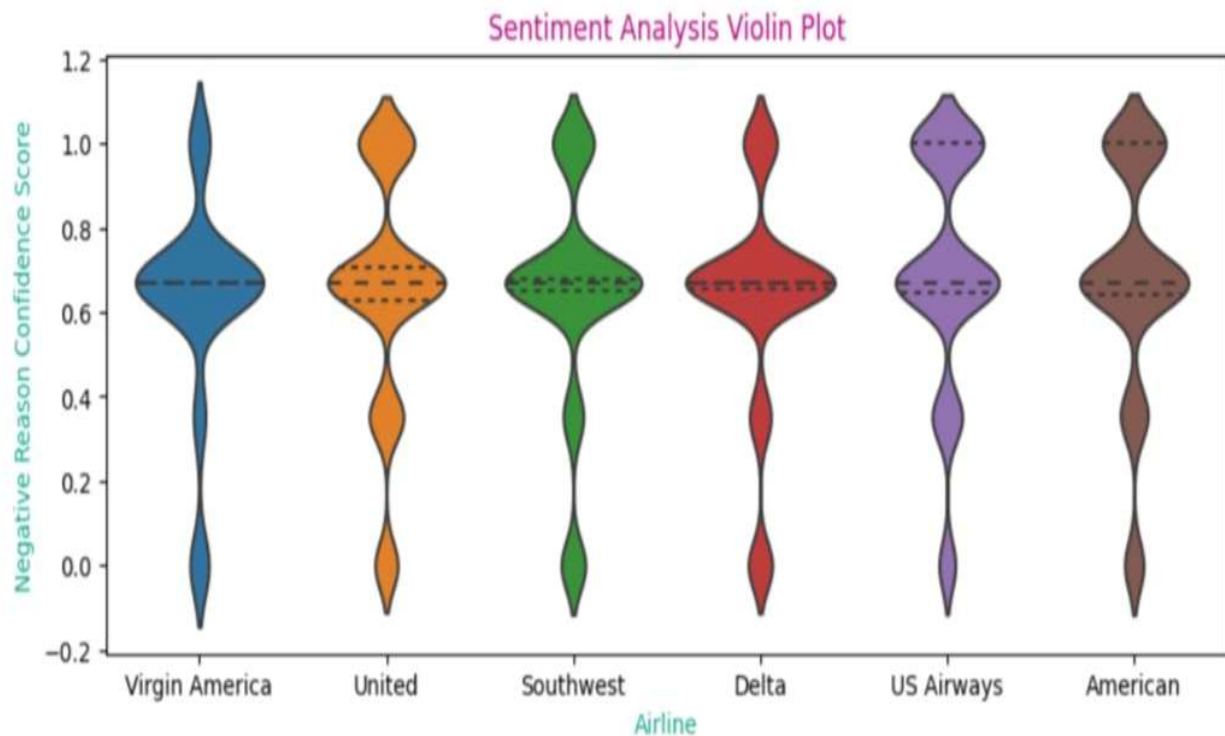
Output for the above code:



Violin plot for Sentiment Analysis:

```
[ ] plt.figure(figsize=(10, 4))
sns.violinplot(x='airline', y='negativereason_confidence', data=df, inner='quartile')
plt.xlabel('Airline',color='#0ca889')
plt.ylabel('Negative Reason Confidence Score',color='#0ca889')
plt.title('Sentiment Analysis Violin Plot',color='#bd0981')
plt.show()
```

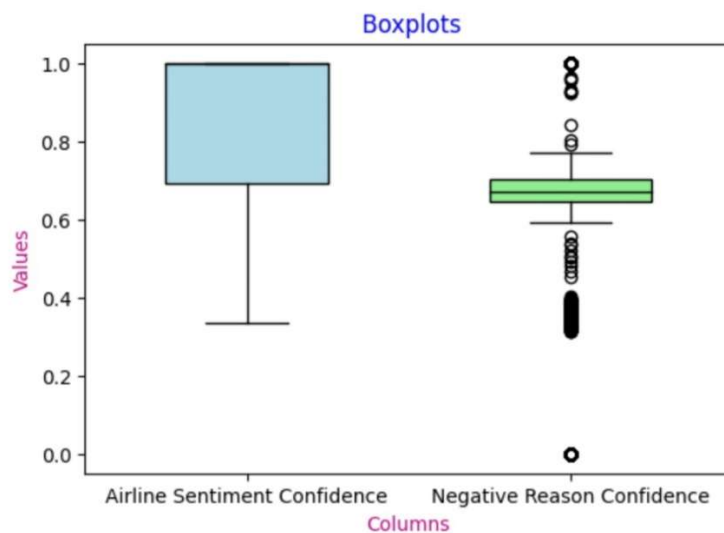
This code creates a violin plot using the Seaborn library to visualize the distribution of 'negativereason_confidence' scores for different airlines.



Boxplots:

```
[ ]  
data1 = df['airline_sentiment_confidence']  
data2 = df['negativereason_confidence']  
plt.figure(figsize=(6, 4))  
bp1 = plt.boxplot(data1, positions=[1], patch_artist=True, widths=0.5)  
bp2 = plt.boxplot(data2, positions=[2], patch_artist=True, widths=0.5)  
box_colors = ['lightblue', 'lightgreen']  
whisker_color = 'black'  
  
for bplot, color in zip([bp1, bp2], box_colors):  
    for element in ['boxes', 'whiskers', 'medians', 'fliers']:  
        plt.setp(bplot[element], color=whisker_color)  
        if element == 'boxes':  
            plt.setp(bplot[element], facecolor=color)  
plt.xticks([1, 2], ['Airline Sentiment Confidence', 'Negative Reason Confidence'])  
plt.xlabel('Columns', color='#bd0981')  
plt.ylabel('Values', color='#bd0981')  
plt.title(' Boxplots', color='blue')  
plt.show()
```

This code displays side-by-side boxplots for the 'airline_sentiment_confidence' and 'negativereason_confidence' columns, allowing for a visual comparison of their distribution and statistical characteristics.



Word cloud :

```
from collections import Counter
word_counts = Counter(df['negativereason'])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_counts)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

This code visually represents the most frequently occurring words or phrases in the 'negativereason' column of the DataFrame.



Architecture diagram for Sentiment analysis for Marketing:

