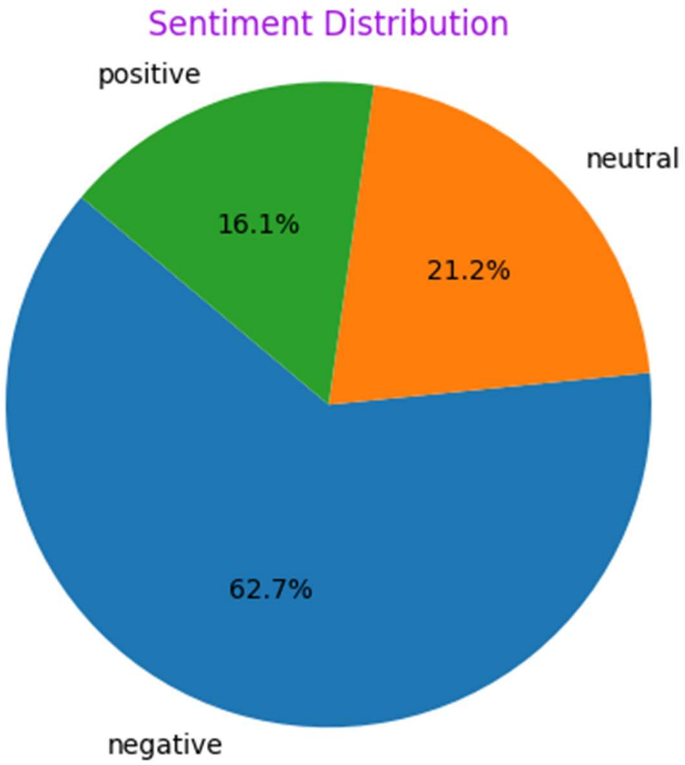
Data visulisation

Sentiment Distribution:

```
sentiment_counts = df['airline_sentiment'].value_counts()
labels = sentiment_counts.index
sizes = sentiment_counts.values
```

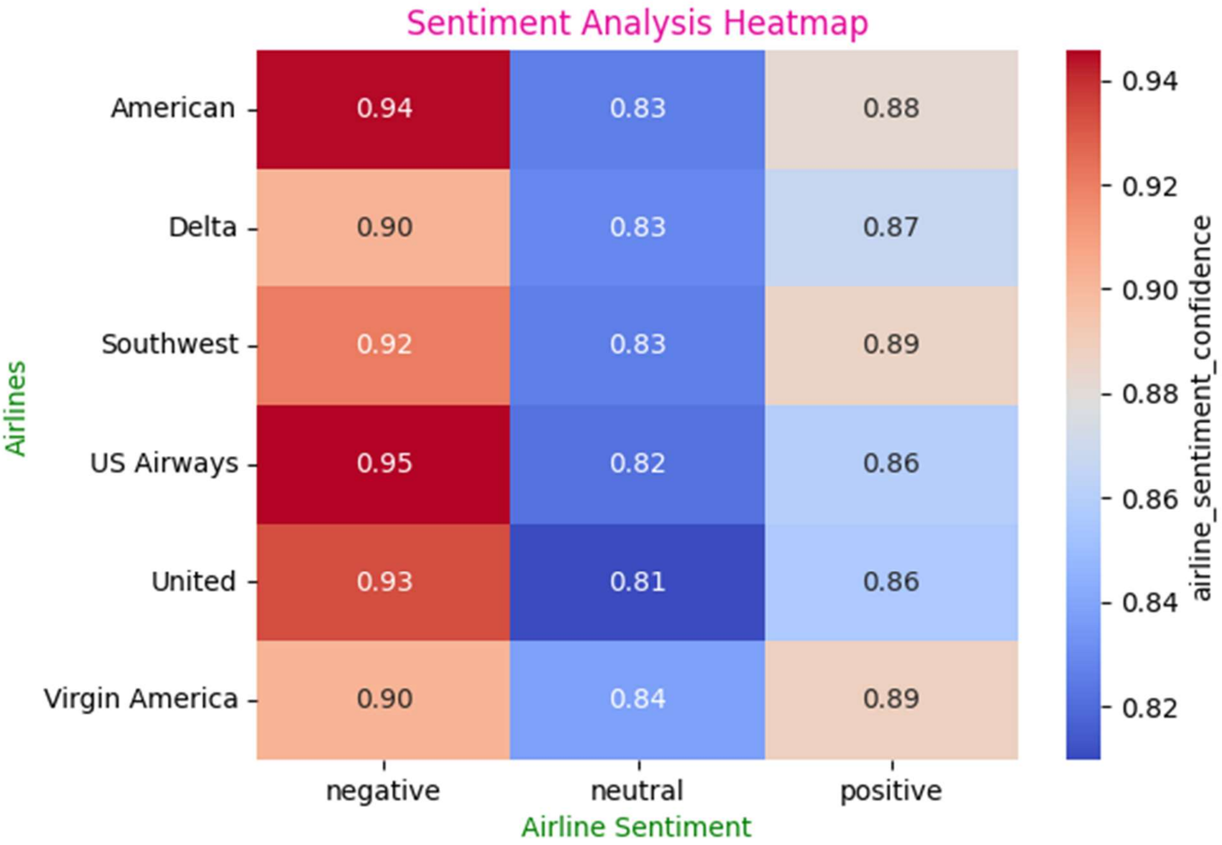
```
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title("Sentiment Distribution",color='#a114de')
plt.show()
```



A numeric feature representing the confidence level of classifying the tweet to one of the 3 classes. For each document or each sentence, the predicted scores associated with the labels (positive, negative, and neutral) add up to 1.

Sentiment_analysis 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()
```



A heat map is both more granular and more understandable. Information relating to population density is overlain directly on the map of the town instead of being siloed into a separate chart.

Airline sentiment confidence vs Nagative 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()
```

A negative confidence lower confidence limit suggests the use of an approximate method for calculating the standard error usually in combination with a small sample size.

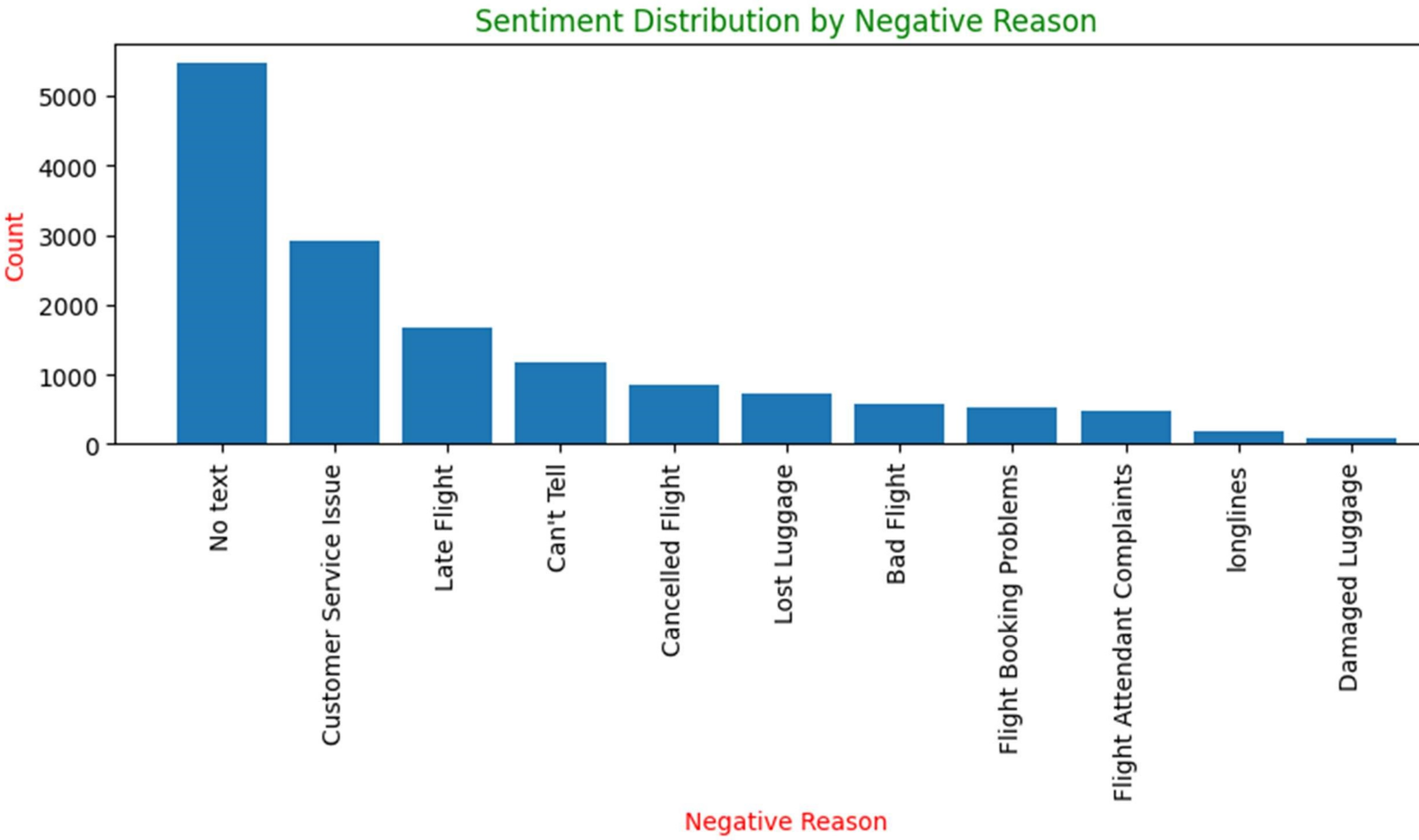
Sentiment Distribution by Navgative reson:

```
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()
```

Which comes make the prediction with the passenger command to the count it is displayed by the using sentiment distribution by negative reason.

No of tweets for airline:

```
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')
```

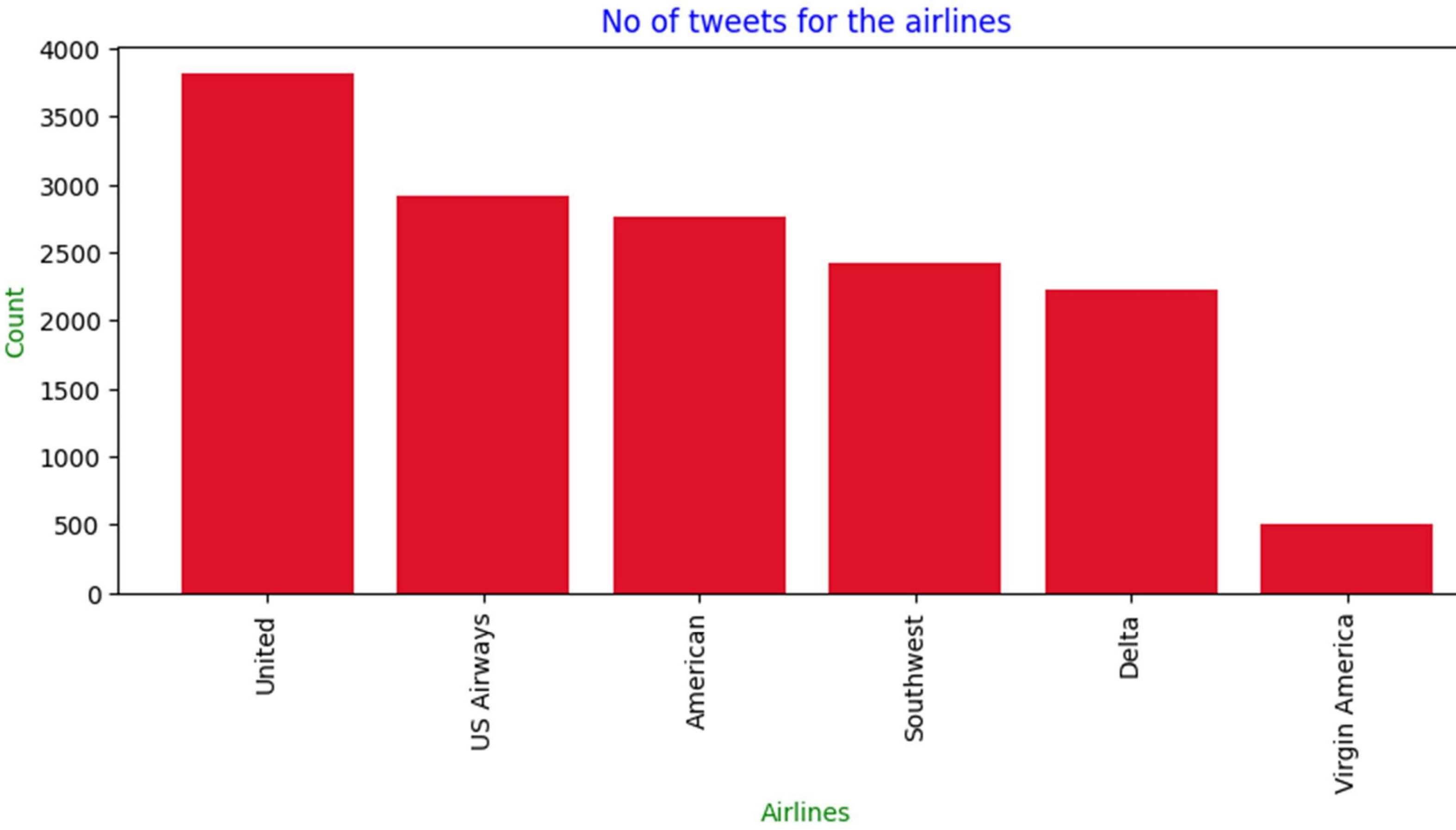


```
plt.xticks(rotation=90)
plt.show()
```

According to a recent survey for the number of people travel by airline in country, the United States is home to the most users, with at least 3900 there.

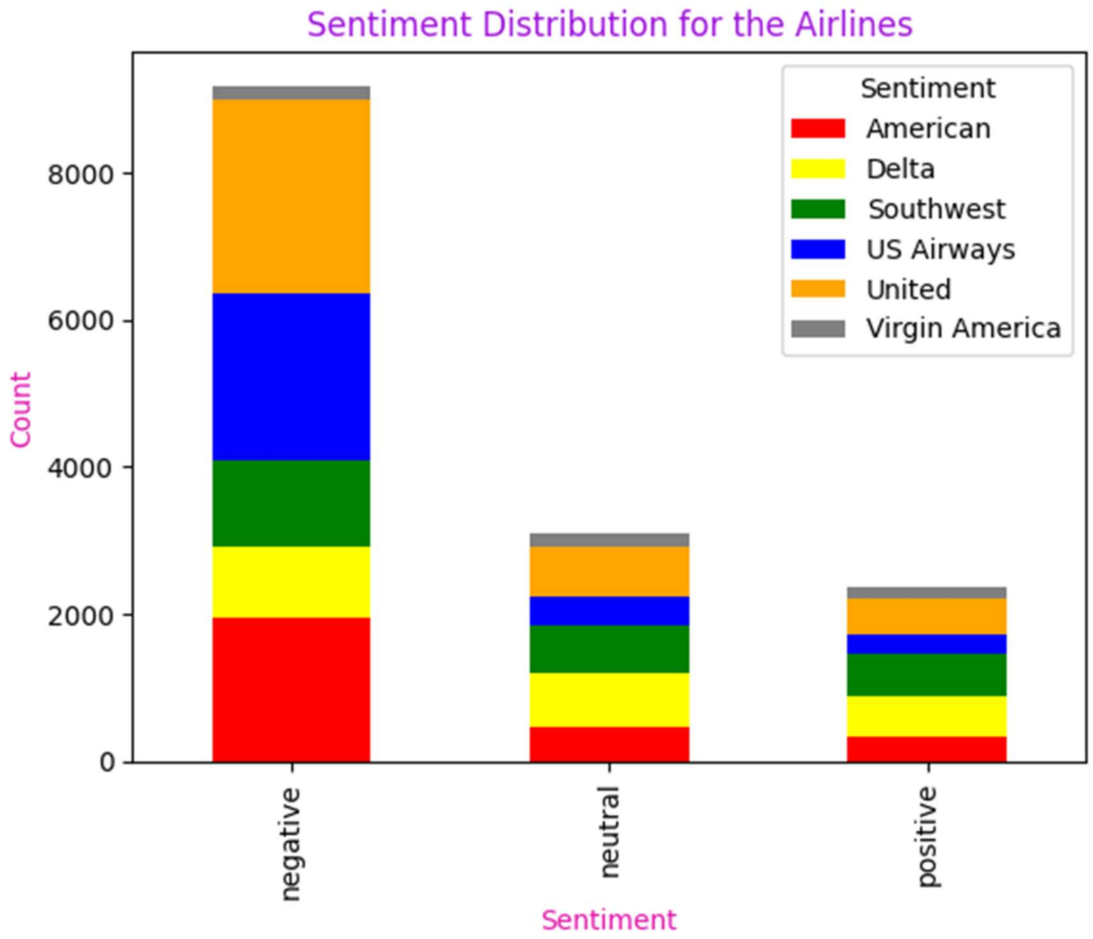
Sentiment Distribution for Airline:

```
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()
```

A numeric feature representing the confidence level of classifying the tweet to one of the 3 classes. For each document or each sentence, the predicted scores associated with the labels (positive, negative, and neutral) add up to 1.



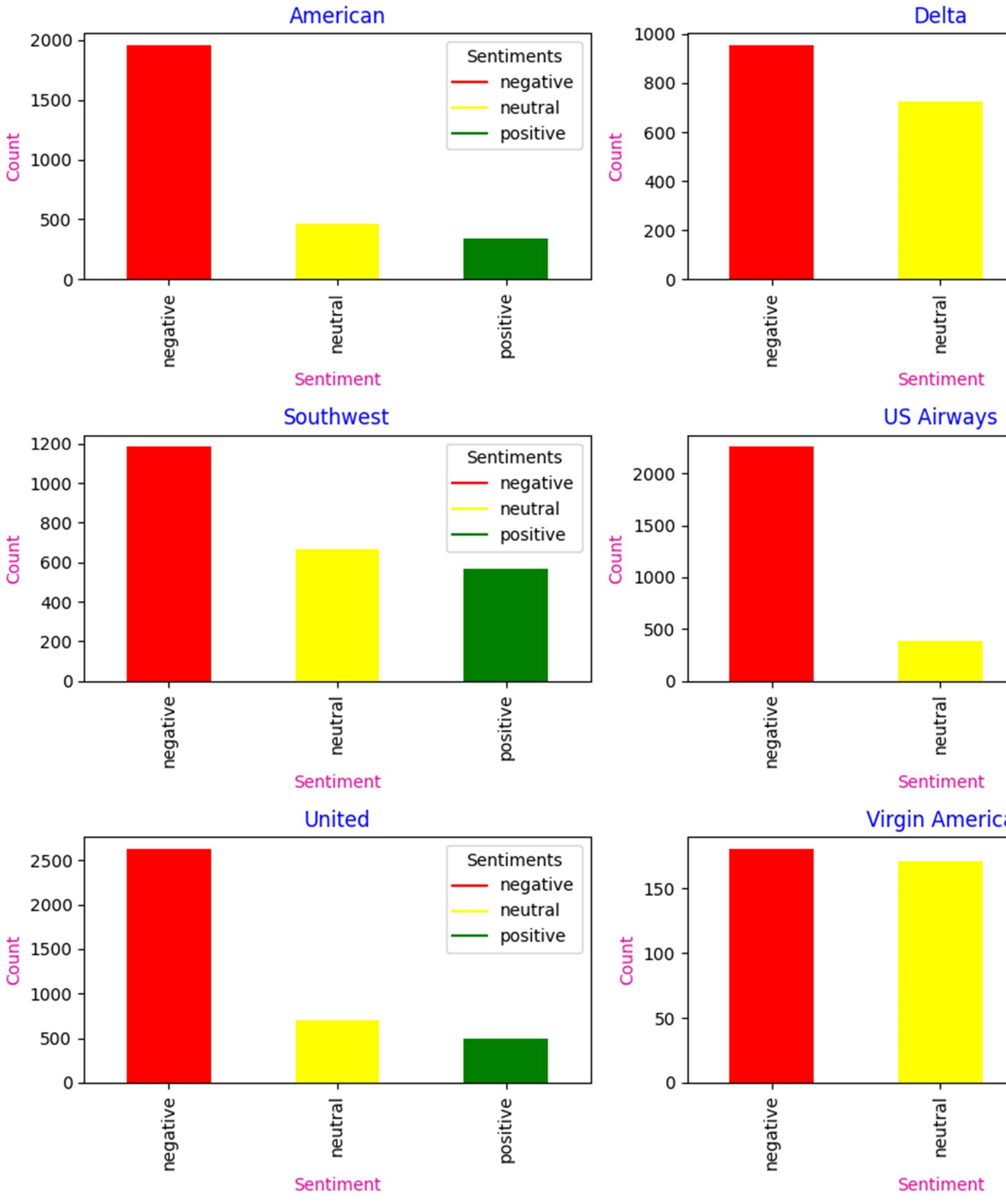
Sentiment for the countries:

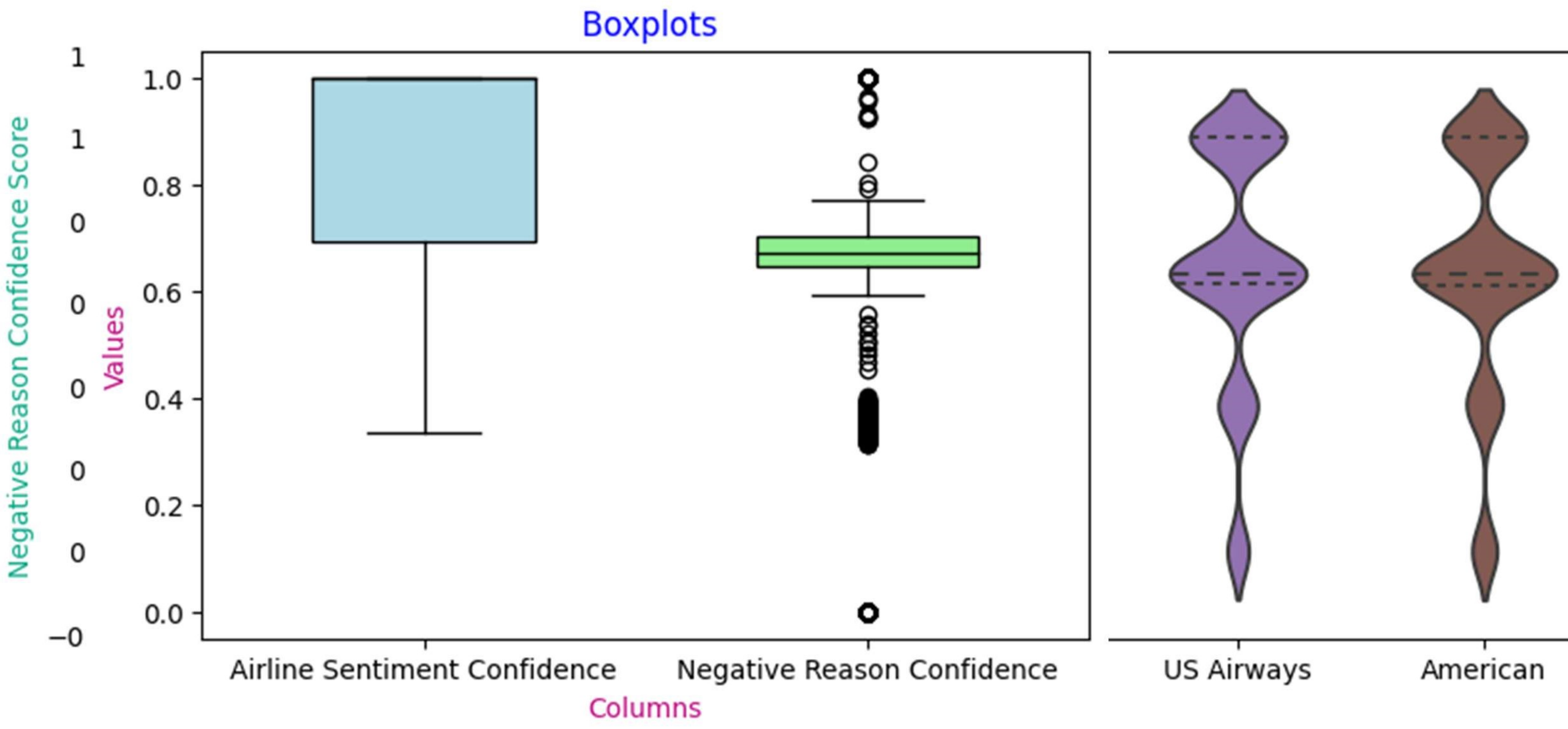
```
sentiment_counts = df.groupby(['airline', 'airline_sentiment']).size().unstack(fill_value=0)
unique_airlines = sentiment_counts.index
fig, axes = plt.subplots(3, 2, figsize=(10, 10))
axes = axes.flatten()
colors = ['red', 'yellow', 'green']
legend_dict = {
    'negative': 'red',
    'neutral': 'yellow',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

It is the mostly sentiment negative is high for alevel countries and is equally neural for Virgin America country.And the positive sentiment is high also this Virgin America country.


American
Sentiments
negative
neutral
positive
2000
1500
1000
500
0
Count
negative
neutral
positive
Sentiment
Delta
1000
800
600
400
200
0
Count
negative
neutral
Sentiment
Southwest
Sentiments
negative
neutral
positive
1200
1000
800
600
400
200
0
Count
negative
neutral
positive
Sentiment
US Airways
2000
1500
1000
500
0
Count
negative
neutral
Sentiment
United
Sentiments
negative
neutral
positive
2500
2000
1500
1000
500
0
Count
negative
neutral
positive
Sentiment
Virgin Americ
150
100
50
0
Count
negative
neutral
Sentiment



```
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()
```

Violin plots help to visualize the distribution of different numerical values. You can visualize data distributions by the ranges, medians and distributions of the data. The Violin plot is a Hybrid of Box plot and Kernel Density plot, as we can see peaks in the data.

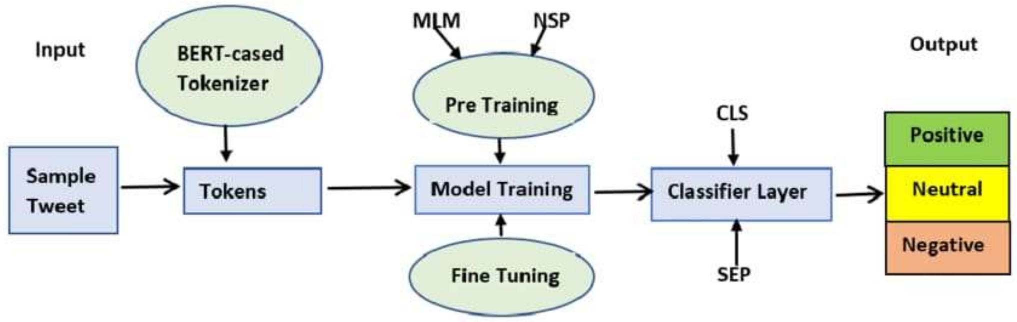
Airline sentiment confidence and Negative reason confidence in Box plot:

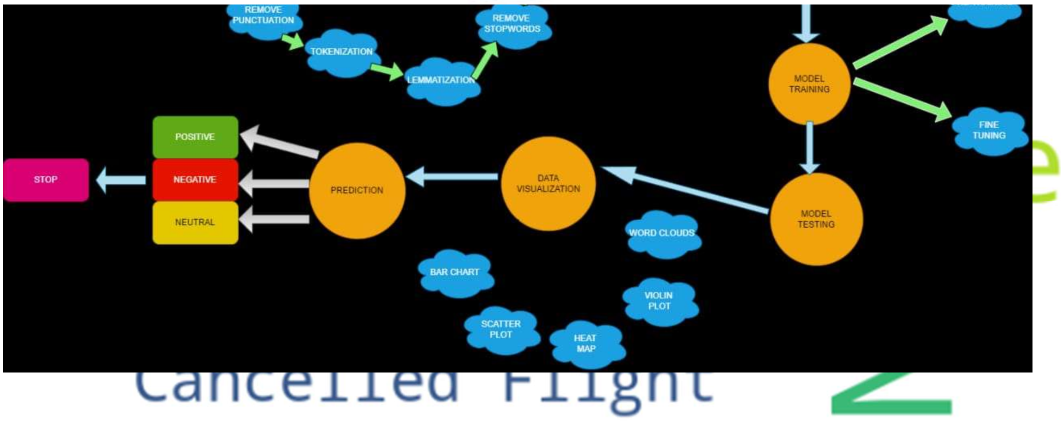
```
data1 = df['airline_sentiment_confidence']
data2 = df['negativereason_confidence']
plt.figure(figsize=(6, 4))
bp1 = plt.boxplot(data1, positions=[1], patch_artist=True, widths=0.5)
bp2 = plt.boxplot(data2, positions=[2], patch_artist=True, widths=0.5)
box_colors = ['lightblue', 'lightgreen']
whisker_color = '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()
```

We will try to gather our first insight by observing the centrality of the box plots. Centerline represents the median value for the house price in different areas. The spread of a box plot talks about the variance present in the data. More the spread, more the variance.

Text to the different styles:





```
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()
```

In the below picture represents the different style to the text it is using the negative reason confidence and biline.

Architecture diagram: