In [ ]:

```
Data Science Report: Analysis of BA Reviews

1. Introduction
This report provides a detailed analysis of the BA reviews dataset. The data

2. Data Overview
The dataset consists of:

An index or identifier for each review.
The text of the review, which includes a label indicating if the review is v

3. Data Cleaning & Pre-processing
Duplicate entries were removed.
The verification status was extracted from the review text and placed in a s
The review text was cleaned by removing the verification status, leading to

4. Exploratory Data Analysis (EDA)
4.1 Verified vs. Not Verified Reviews
A substantial number of the reviews were not verified, as visualized in the

4.1 Verified vs. Not Verified Reviews (Continued)
The visualization above showed the distribution of verified vs. not verified

4.2 Review Lengths
On average, review lengths were around 792 characters, with the shortest rev

5. Sentiment Analysis
Using TextBlob for sentiment analysis, reviews were classified into three ca

Positive
Neutral
Negative
The distribution of sentiments was visualized, revealing that a significant

6. Topic Modelling
Using the LDA (Latent Dirichlet Allocation) method, the reviews were grouped

7. Key Insights & Recommendations
Review Verification: A significant number of reviews are not verified. It mi
Sentiments: While many reviews have a positive sentiment, a substantial numb
Topics: The topics derived from LDA can help BA identify common themes in cu

8. Conclusion
The analysis of the BA reviews dataset provides valuable insights into custo

For the sake of completeness, I would typically include more visualizations,
```

In [6]:

```python
import pandas as pd
import matplotlib.pyplot as plt
from textblob import TextBlob
import gensim
from gensim import corpora
import string


reviews_df = pd.read_csv("BA_reviews.csv")
```

In [7]:

```python
reviews_df
```

Out[7]:

	Unnamed: 0	reviews
0	0	✅ Trip Verified | Easy check in a T5. Galleri...
1	1	Not Verified | Flight delayed by an hour, it ...
2	2	Not Verified | The staff are very rude and not...
3	3	✅ Trip Verified | Good domestic flight operat...
4	4	Not Verified | Failed at all basic travel fund...
...	...	...
995	995	✅ Trip Verified | Johannesburg to London. I t...
996	996	✅ Trip Verified | Singapore to London Heathro...
997	997	✅ Trip Verified | London to Los Angeles. Book...
998	998	✅ Trip Verified | Miami to London Heathrow. A...
999	999	Not Verified | London Heathrow to Milan Malpen...

1000 rows × 2 columns

In [8]:

```
#Data Cleaning
reviews_df.drop_duplicates(inplace=True)
reviews_df['verified'] = reviews_df['reviews'].apply(lambda x: 'Verified' if
reviews_df['cleaned_reviews'] = reviews_df['reviews'].str.replace(r'(✅ Trip
```

In [9]:

```
reviews_df
```

Out[9]:

	Unnamed: 0	reviews	verified	cleaned_reviews
0	0	✅ Trip Verified | Easy check in a T5. Galleri...	Verified	| Easy check in a T5. Galleries south and Nor...
1	1	Not Verified | Flight delayed by an hour, it ...	Verified	| Flight delayed by an hour, it happens, no b...
2	2	Not Verified | The staff are very rude and not...	Verified	| The staff are very rude and not trained prop...
3	3	✅ Trip Verified | Good domestic flight operat...	Verified	| Good domestic flight operated by BA Cityfly...
4	4	Not Verified | Failed at all basic travel fund...	Verified	| Failed at all basic travel fundamentals: 1) ...
...	...	...	...	...
995	995	✅ Trip Verified | Johannesburg to London. I t...	Verified	| Johannesburg to London. I tend to stay with...
996	996	✅ Trip Verified | Singapore to London Heathro...	Verified	| Singapore to London Heathrow. It's my first...
997	997	✅ Trip Verified | London to Los Angeles. Book...	Verified	| London to Los Angeles. Booked this trip eig...
998	998	✅ Trip Verified | Miami to London Heathrow. A...	Verified	| Miami to London Heathrow. As with many othe...
999	999	Not Verified | London Heathrow to Milan Malpen...	Verified	| London Heathrow to Milan Malpensa. Worst air...

1000 rows × 4 columns

In [12]:

```
# 4. Sentiment Analysis
def classify_sentiment(polarity):
    if polarity > 0.05:
        return 'Positive'
    elif polarity < -0.05:
        return 'Negative'
    else:
        return 'Neutral'

reviews_df['polarity'] = reviews_df['cleaned_reviews'].apply(lambda x: TextB
reviews_df['sentiment'] = reviews_df['polarity'].apply(classify_sentiment)
```

In [13]:

```
reviews_df
```

Out[13]:

	Unnamed: 0	reviews	verified	cleaned_reviews	polarity	sentiment
0	0	✅ Trip Verified | Easy check in a T5. Galleri...	Verified	| Easy check in a T5. Galleries south and Nor...	0.201581	Positive
1	1	Not Verified | Flight delayed by an hour, it ...	Verified	| Flight delayed by an hour, it happens, no b...	0.225000	Positive
2	2	Not Verified | The staff are very rude and not...	Verified	| The staff are very rude and not trained prop...	-0.360000	Negative
3	3	✅ Trip Verified | Good domestic flight operat...	Verified	| Good domestic flight operated by BA Cityfly...	0.236111	Positive
4	4	Not Verified | Failed at all basic travel fund...	Verified	| Failed at all basic travel fundamentals: 1) ...	-0.265714	Negative
...	...	...	...	...	...	...
995	995	✅ Trip Verified | Johannesburg to London. I t...	Verified	| Johannesburg to London. I tend to stay with...	0.200027	Positive
996	996	✅ Trip Verified | Singapore to London Heathro...	Verified	| Singapore to London Heathrow. It's my first...	0.075214	Positive
997	997	✅ Trip Verified | London to Los Angeles. Book...	Verified	| London to Los Angeles. Booked this trip eig...	-0.016920	Neutral
998	998	✅ Trip Verified | Miami to London Heathrow. A...	Verified	| Miami to London Heathrow. As with many othe...	0.230556	Positive
999	999	Not Verified | London Heathrow to Milan Malpen...	Verified	| London Heathrow to Milan Malpensa. Worst air...	-0.125000	Negative

1000 rows × 6 columns

In [14]:

```
# 5. Topic Modelling
custom_stopwords = {...}  # Your list of stopwords here

def preprocess_no_lemma(text):
    tokens = [word for word in text.lower().split() if word not in string.pu
    tokens = [word for word in tokens if word not in custom_stopwords]
    return tokens

reviews_df['tokens'] = reviews_df['cleaned_reviews'].apply(preprocess_no_lem
dictionary_no_lemma = corpora.Dictionary(reviews_df['tokens'])
corpus_no_lemma = [dictionary_no_lemma.doc2bow(token) for token in reviews_d
lda_model_no_lemma = gensim.models.LdaMulticore(corpus_no_lemma, num_topics=
topics_no_lemma = lda_model_no_lemma.print_topics(num_words=5)
```

In [15]:

```
reviews_df
```

Out[15]:

	Unnamed: 0	reviews	verified	cleaned_reviews	polarity	sentiment	tok
0	0	✅ Trip Verified | Easy check in a T5. Galleri...	Verified	| Easy check in a T5. Galleries south and Nor...	0.201581	Positive	[easy, check, a, t5., galler south, a
1	1	Not Verified | Flight delayed by an hour, it ...	Verified	| Flight delayed by an hour, it happens, no b...	0.225000	Positive	[flight, delay by, an, hour, happens
2	2	Not Verified | The staff are very rude and not...	Verified	| The staff are very rude and not trained prop...	-0.360000	Negative	[the, staff, a very, rude, a not, train
3	3	✅ Trip Verified | Good domestic flight operat...	Verified	| Good domestic flight operated by BA Cityfly...	0.236111	Positive	[go domes flight, operat by, ba, c
4	4	Not Verified | Failed at all basic travel fund...	Verified	| Failed at all basic travel fundamentals: 1) ...	-0.265714	Negative	[failed, at, basic, tra fundamental
...	...	...	...	...	...	...	
995	995	✅ Trip Verified | Johannesburg to London. I t...	Verified	| Johannesburg to London. I tend to stay with...	0.200027	Positive	[johannesbu to, londor tend, to, sta
996	996	✅ Trip Verified | Singapore to London Heathro...	Verified	| Singapore to London Heathrow. It's my first...	0.075214	Positive	[singapore, lond heathrow., i my,
997	997	✅ Trip Verified | London to Los Angeles. Book...	Verified	| London to Los Angeles. Booked this trip eig...	-0.016920	Neutral	[london, to, angel booked, t tri
998	998	✅ Trip Verified | Miami to London Heathrow. A...	Verified	| Miami to London Heathrow. As with many othe...	0.230556	Positive	[miami, lond heathrow., with, man
999	999	Not Verified | London Heathrow to Milan Malpen...	Verified	| London Heathrow to Milan Malpensa. Worst air...	-0.125000	Negative	[lond heathrow, mi malpen wor

1000 rows × 7 columns

In [16]:

```
# Counting the number of positive and negative reviews
sentiment_counts = reviews_df['sentiment'].value_counts()

# Sample reviews for each sentiment
```

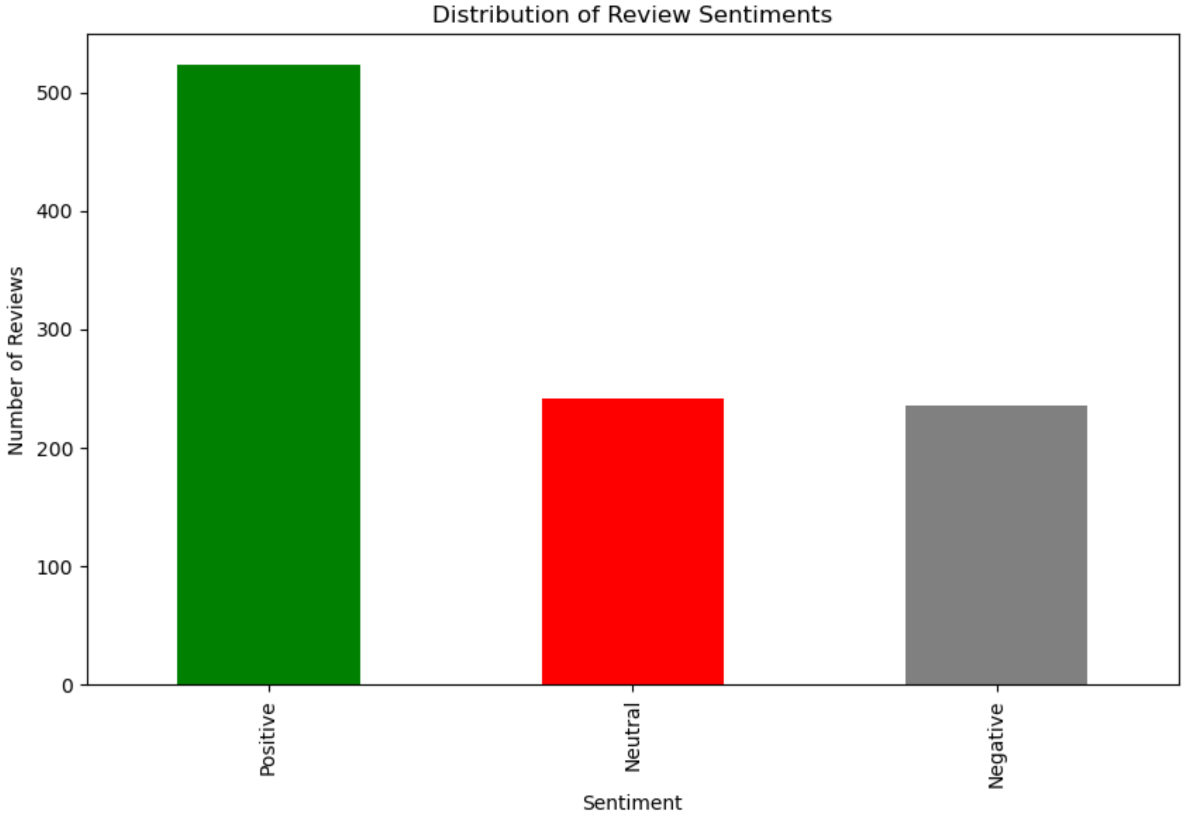
```
sample_positive_reviews = reviews_df[reviews_df['sentiment'] == 'Positive'][
sample_negative_reviews = reviews_df[reviews_df['sentiment'] == 'Negative'][

sentiment_counts, sample_positive_reviews, sample_negative_reviews
```

Out[16]:

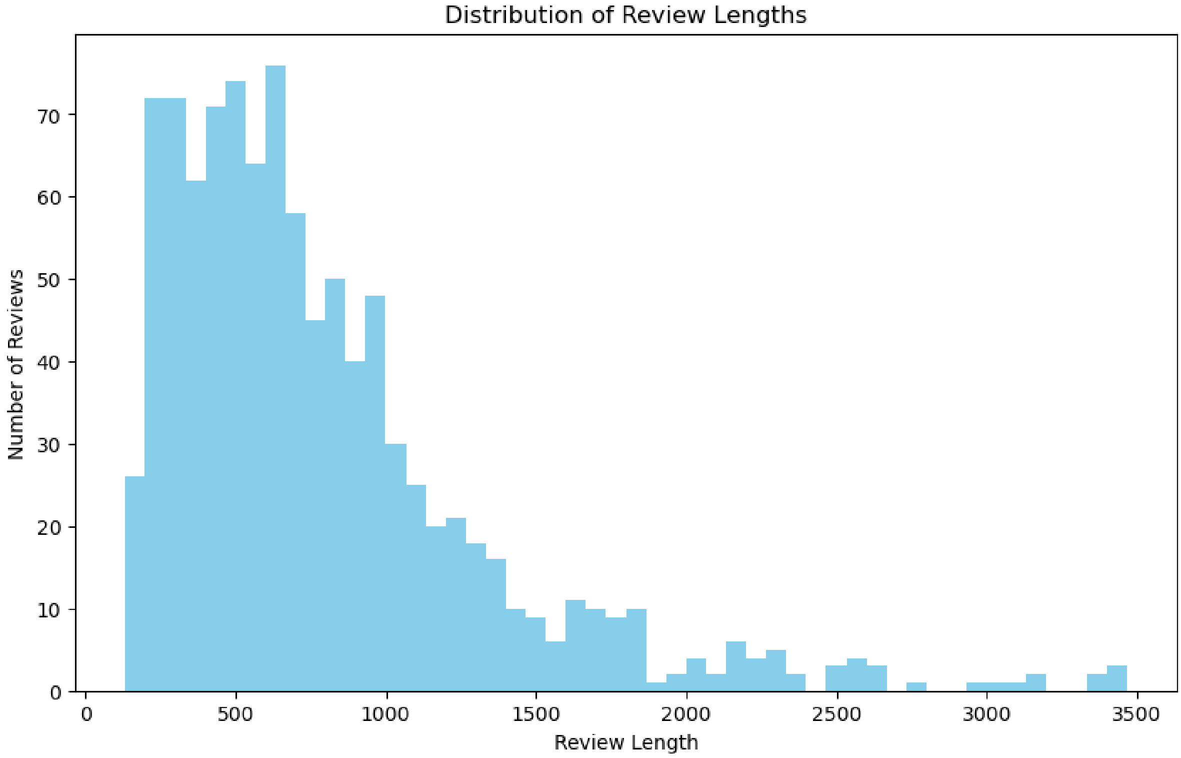
```
(Positive    523
 Neutral     241
 Negative    236
 Name: sentiment, dtype: int64,
 ['|  London to Philadelphia. I upgraded from coach to business for almost
$900 because I  wanted to experience BA’s business class in the Boeing 787,
but I was disappointed. While the aircraft was new, the staff really could h
ave cared less about the business guests. 2-3-2 in business with some seats
facing the cockpit and others the tail. BA lost me as a traveler because the
crew although polite, service wise was lacking. Pre-flight they offer you a
beverage, I asked for rose champagne and I was told “I guess I have to go fi
nd a bottle and open one”. In my previous business class experiences there a
re beverages served every 2 hours, not on this almost 8 hour flight. I was s
erved my main meal almost 2.5 hours after takeoff. You get one full hot meal
and a snack one hour and a half before landing. You also get an initial beve
rage and then the next time a beverage is offered is 1.5 hours before landin
g. If you ask for anything else to drink in between you have to go into the
galley. Yes they have the menu, linen, silverware and glass but you feel lik
e they just have to do this for you and not like they want to do this for yo
u. Because of the extra seat they cram into business now they are all very n
arrow and really uncomfortable. There was virtually no storage other than a
small laptop droor and that’s it. I have to say the entertainment options we
re very good and there was no Wi-Fi on this flight. You get a pillow, duvet
and blanket and a decent amenity kit. BA has the new planes, and decent busi
ness amenities however they lost me in the service. I will not pay to upgrad
e to business on BA and will avoid flying with them if I have another optio
n. They need to improve when it comes to inflight experiences and treatment
specially for their business class travelers.',
  '|  I was to fly to Melbourne from Manchester via London and Singapore in
July 2019. Arriving late to board my Heathrow flight, the Check-In-Staff dis
cussed my dilemma with her supervisor. Only option being to find my way to H
eathrow. I somehow managed to get to Heathrow in time where only the Singapo
re to Melbourne Boarding Pass was issued. I was told that because Manchester
to Heathrow flight was missed, my Heathrow to Singapore was cancelled.',
  '|  Gatwick to Las Vegas. Boarding by group number seemed to work well at
Gatwick, we took off ahead of schedule and service commenced about 45 min af
ter take off. The crew seemed relaxed and happy and worked well as a team. B
A have improved their inflight menu and it shows as the quality of the food
was one of the best I’ve ever had on a long haul flight. The aircraft was ol
d and dated and the IFE was a throwback to the 90’s with unresponsive touch
screen and hard to watch quality as we all so used to HD quality these days.
All in all a very enjoyable flight.',
  '| I was meant to fly in January to Algeria. I paid over £300 for the tick
et. Just a few days before I was meant to fly, my wife broke her foot. I had
to then rebook it in March and pay another £100. I called British Airways 3
weeks before my flight in March and I told them again that it had to me rebo
oked, as my wife was still not well. They told me that I don’t have to pay a
nything to get it rebooked as I sent them a letter from the hospital saying
my wife wasn’t well. However, they never sent me the ticket. They made me wa
it for nearly 4 weeks and they asked me to pay either £316 on top of the £45
0 that I paid previously. Or the second option they gave me is to get a refu
nd of £80. (Tax only). I found it ridiculous that they told me to pay a frac
tion of the money I previously paid. Each time I call them, they hang up on
me; or they continue telling me to pay £360. I made a complaint, and nothing
happened.',
  '|  Flying during covid is always a challenge. BA’s VeriFLY app worked per
fectly and on-line check-in was seamless. The bags drop at T5 was seamless a
nd boarding was done efficiently by zones. Although an older 777 it had been
refurbished and everything was new and clean. The cabin crew were a credit t
o the airline – smartly dressed, friendly and keen to please. We were offere
d two drinks before lunch was served, with more drinks with the meal. The fo
od was acceptable and we were given a tasty sandwich before landing. Obvious
ly it was only economy but, after 8 hours flying, I arrived relaxed and unst
ressed.'],
 ["|  British Airways is late, their website is atrocious, and they wouldn't
let me check-in until 24 hours before the flight. I have been trying for 6 h
ours to check in through their website, and/or find a real person to speak w
ith. Customer service had such a thick accent I didn't think they were speak
ing English.",
  '|  London to Los Angeles. Take off was delayed by 3 hrs (which we spent o
nboard) because of a dent to the door which needed inspection - sure maybe o
utside of their control. I was flying with my wife and two kids (2 & 4), sit
ting on bulkhead with 3 seats and using the baby bassinet facility. After ta
ke off we saw that the inflight entertainment service (for the whole section
- not whole cabin) was not working. And we were sitting into an 11.5hr overn
ight flight with two little ones, meaning sleeping was not going to happen.
We contacted the cabin crew who made zero accommodation. Sorry they said. Du
ring the flight the cabin crew stood in the galley behind the bulkhead talki
ng loudly while the lights were dimmed, and kept slamming drawers as they mo
ved things around. To cap it off, we landed at 10pm ET and they had left beh
ind our baby car seat and stroller. We finally received them back 2 days lat
er. Conclusion, never fly BA again.',
  "|  Gatwick to Malaga. Nothing special, nowadays there's unfortunately not
much difference between BA & Easyjet. Flight left Gatwick late & arrived in
Malaga late. BA seats no longer recline. Food and drinks are expensive, thei
r Cream Tea excludes tea. Lift your game BA, your product just continues to
get worse.",
  '| London to Manchester. British Airways is probably the worst air company
I have ever flown with they took 2 hours to unload the luggage and the fligh
t was delayed by an hour. The staff were no help and frankly rude.',
  '| An airline that lives in their past glory and heading a future of crisi
s is British Airways. I am not joking we are a family of 7 all in Business C
lass heading for a Norwegian and Arctic Cruise and they lost 10 of our bags!
They will never see me on any of their flights ever a disaster of an airlin
e'])
```

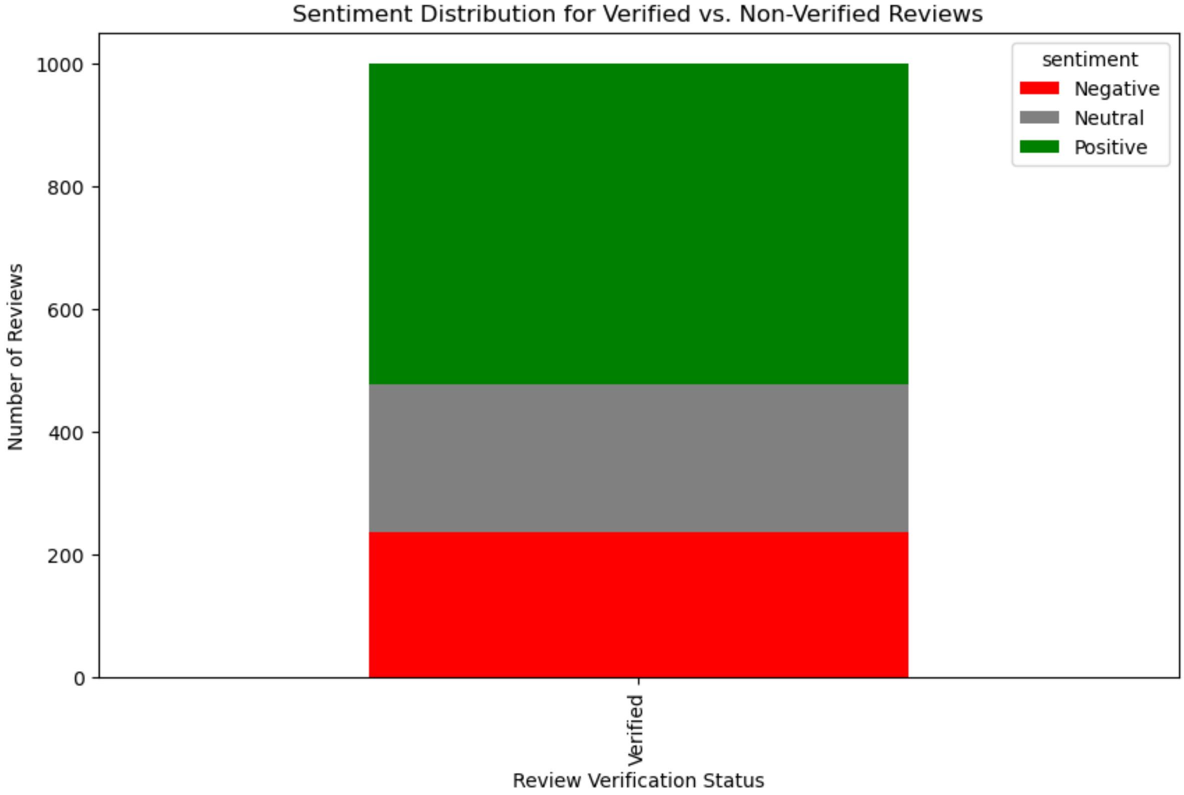
In [18]:

```python
# 3. Perform sentiment analysis
def classify_sentiment(polarity):
    if polarity > 0.05:
        return 'Positive'
    elif polarity < -0.05:
        return 'Negative'
    else:
        return 'Neutral'

reviews_df['polarity'] = reviews_df['cleaned_reviews'].apply(lambda x: TextB
reviews_df['sentiment'] = reviews_df['polarity'].apply(classify_sentiment)

# 4. Visualize the distribution of sentiments
sentiment_counts = reviews_df['sentiment'].value_counts()
plt.figure(figsize=(10, 6))
sentiment_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title('Distribution of Review Sentiments')
plt.xlabel('Sentiment')
plt.ylabel('Number of Reviews')
plt.show()
```



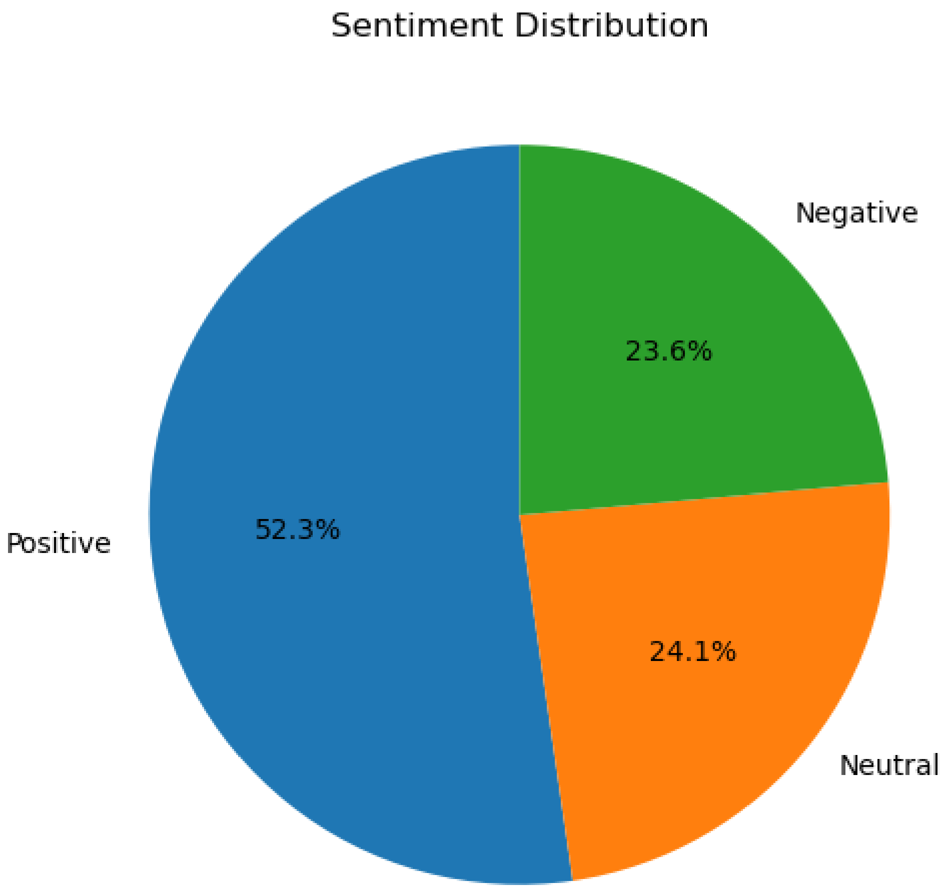
In [29]:
```
import numpy as np

# 1. Distribution of Review Lengths
reviews_df['review_length'] = reviews_df['cleaned_reviews'].apply(len)
plt.figure(figsize=(10, 6))
plt.hist(reviews_df['review_length'], bins=50, color='skyblue')
plt.title('Distribution of Review Lengths')
plt.xlabel('Review Length')
plt.ylabel('Number of Reviews')
plt.show()

# 2. Sentiment Distribution for Verified vs. Non-Verified Reviews
grouped = reviews_df.groupby(['verified', 'sentiment']).size().unstack()
grouped.plot(kind='bar', stacked=True, figsize=(10, 6), color=['red', 'gray'
plt.title('Sentiment Distribution for Verified vs. Non-Verified Reviews')
plt.xlabel('Review Verification Status')
plt.ylabel('Number of Reviews')
plt.show()
```

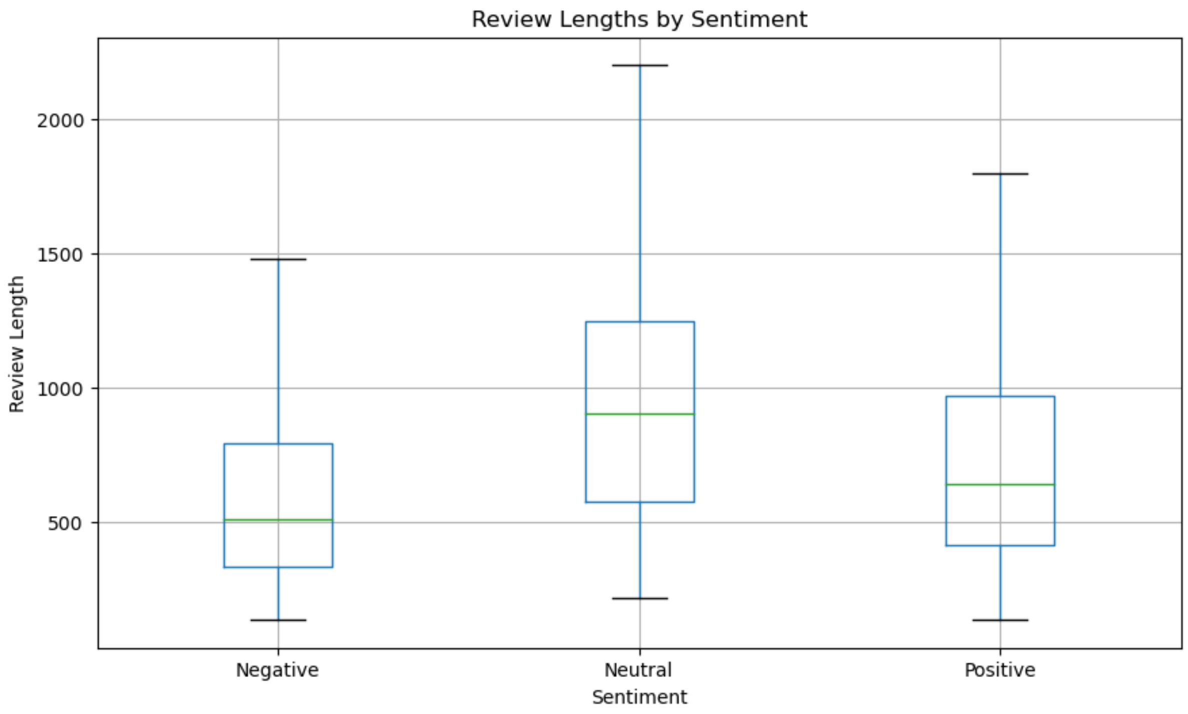




In [32]:

```
import pandas as pd
import matplotlib.pyplot as plt
from textblob import TextBlob
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix



# 1. Pie Chart for Sentiment Distribution
sentiment_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90, figsize=
plt.title("Sentiment Distribution")
plt.ylabel('')  # This is to remove the 'sentiment' label from the y-axis
plt.show()

# 2. Box plot for review lengths by sentiment
reviews_df.boxplot(column='review_length', by='sentiment', showfliers=False,
plt.title('Review Lengths by Sentiment')
plt.suptitle('')  # This removes the default title
plt.ylabel('Review Length')
plt.xlabel('Sentiment')
plt.show()
```





In [34]:

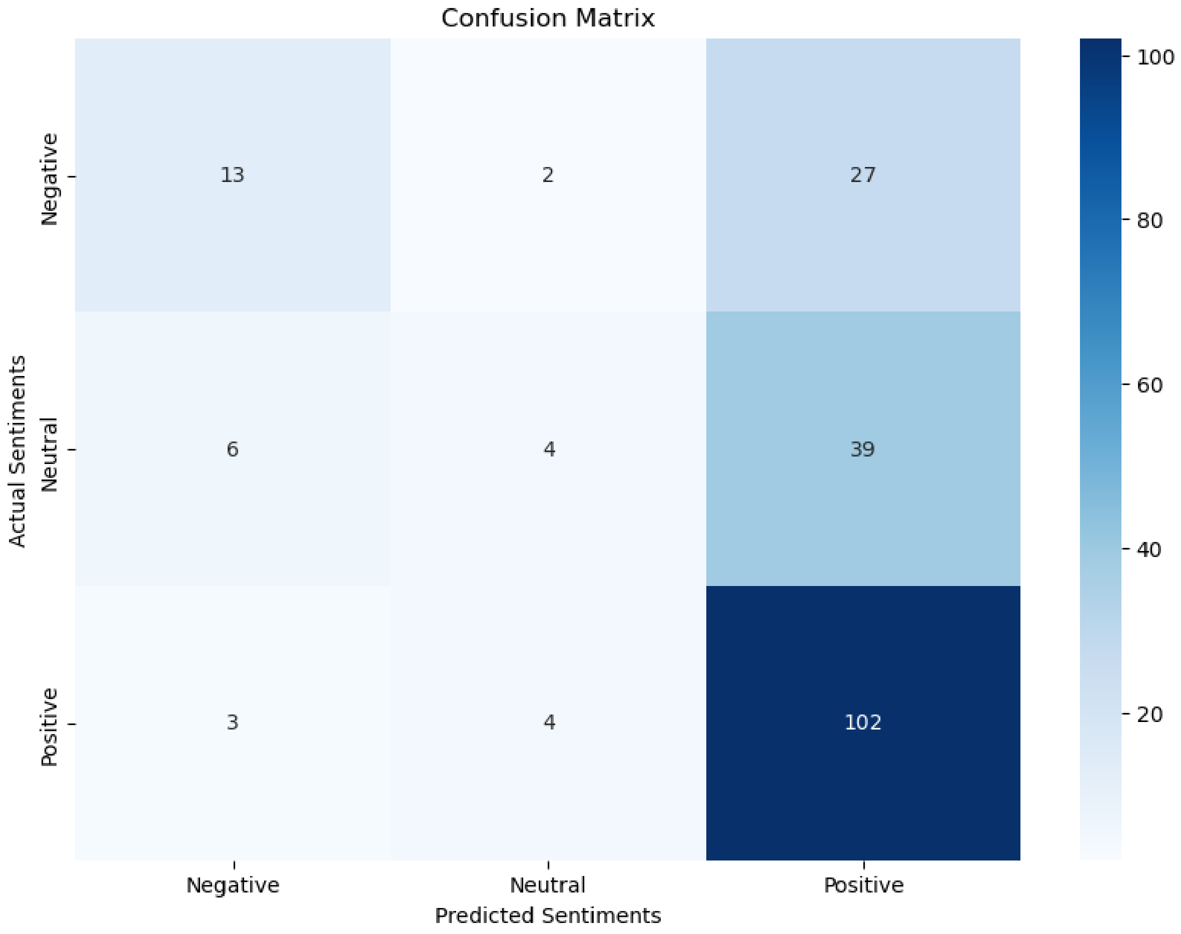
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from textblob import TextBlob



# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(reviews_df['cleaned_revi

# Using TF-IDF Vectorizer
vectorizer = TfidfVectorizer(max_features=5000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Using Random Forest classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train_vec, y_train)

# Predictions and Evaluation
y_pred = rf_clf.predict(X_test_vec)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

    Negative       0.59      0.31      0.41        42
     Neutral       0.40      0.08      0.14        49
    Positive       0.61      0.94      0.74       109

    accuracy                           0.59       200
   macro avg       0.53      0.44      0.43       200
weighted avg       0.55      0.59      0.52       200
```

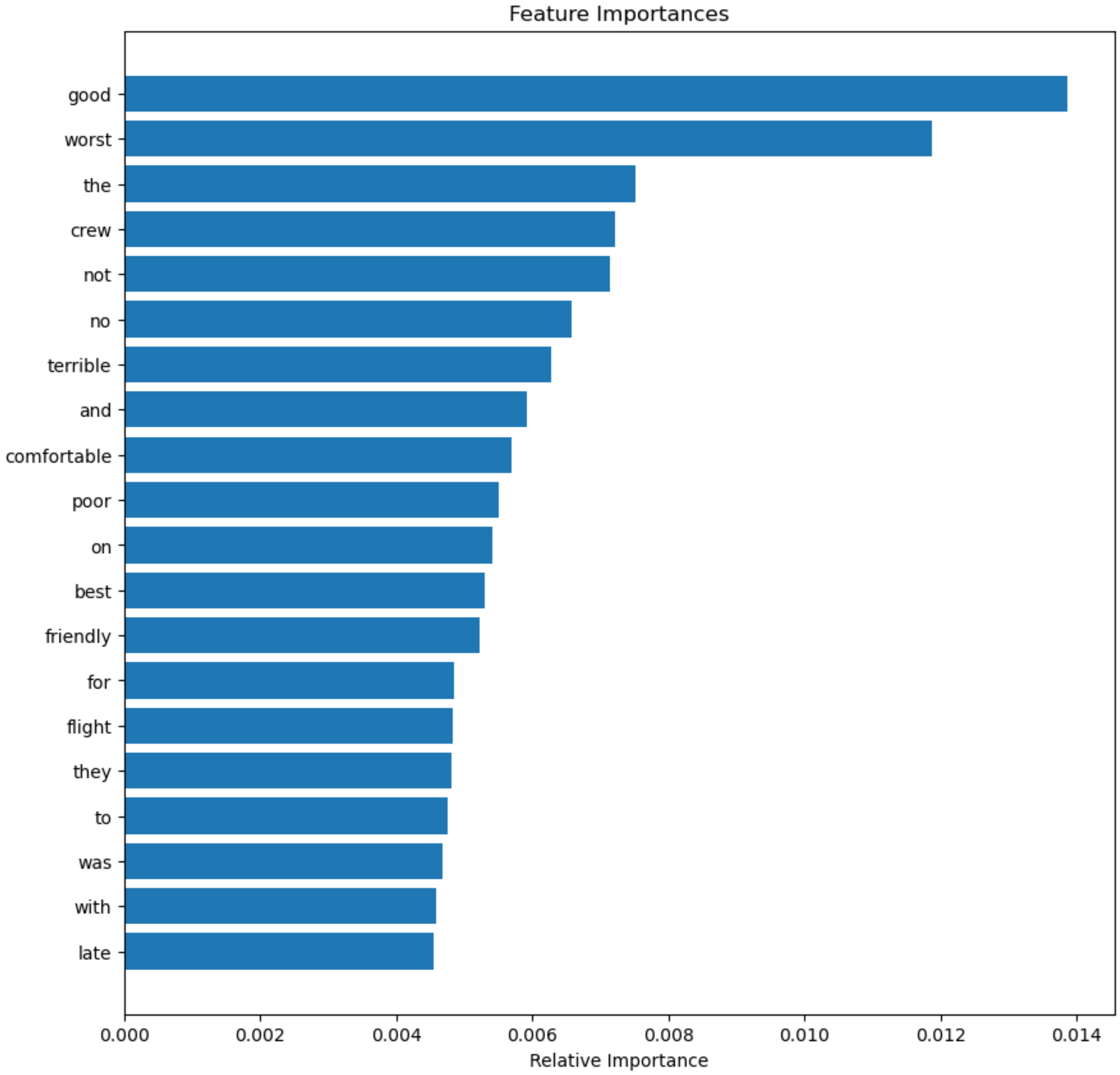
In [35]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
from textblob import TextBlob
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt


# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(reviews_df['cleaned_revi

# Using TF-IDF Vectorizer
vectorizer = TfidfVectorizer(max_features=5000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Using Random Forest classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train_vec, y_train)

# Predictions
y_pred = rf_clf.predict(X_test_vec)

# Confusion Matrix Visualization
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Negative', 'Neutral', 'Positive'],
            yticklabels=['Negative', 'Neutral', 'Positive'])
plt.xlabel('Predicted Sentiments')
plt.ylabel('Actual Sentiments')
plt.title('Confusion Matrix')
plt.show()

# Feature Importance Visualization
feat_importances = rf_clf.feature_importances_
indices = np.argsort(feat_importances)[-20:]  # Top 20 features
plt.figure(figsize=(10, 10))
plt.title('Feature Importances')
plt.barh(range(len(indices)), feat_importances[indices], align='center')
plt.yticks(range(len(indices)), [vectorizer.get_feature_names()[i] for i in
plt.xlabel('Relative Importance')
plt.show()
```



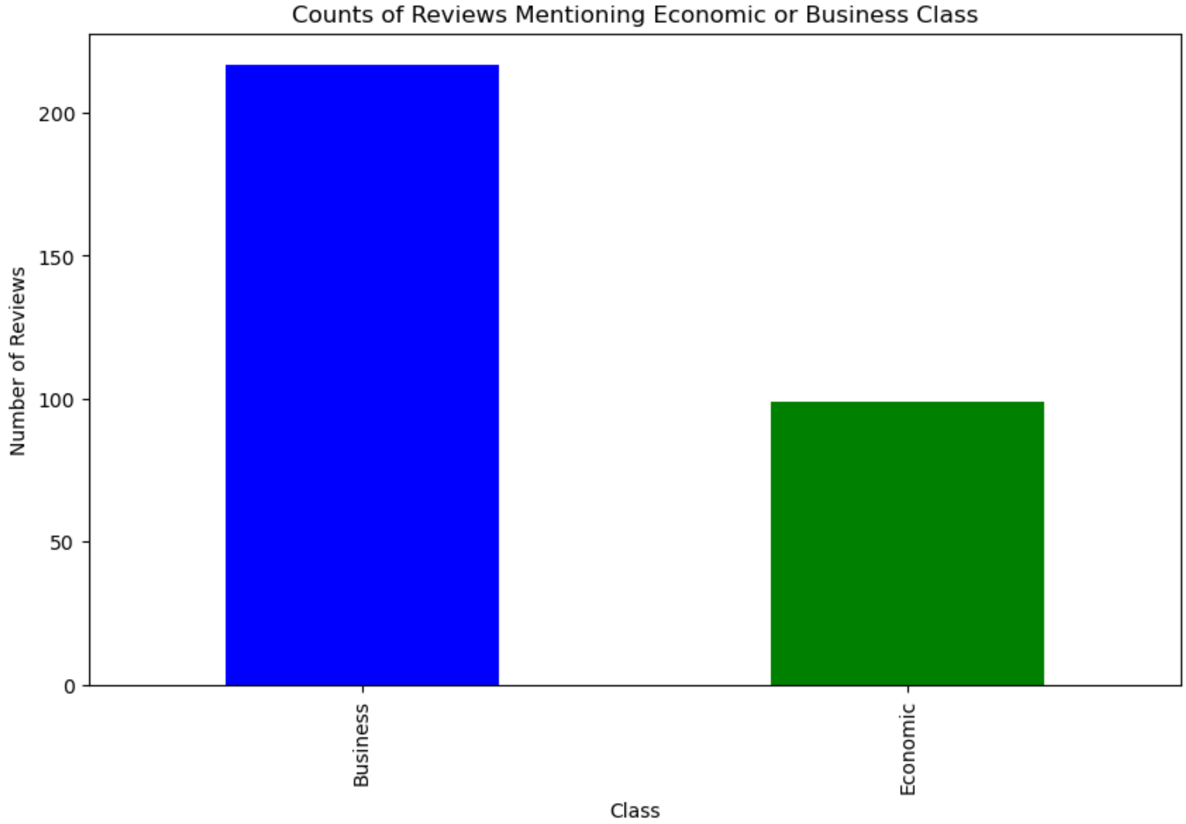
```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/utils/deprecation.py:87:
FutureWarning: Function get_feature_names is deprecated; get_feature_names i
s deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names
_out instead.
  warnings.warn(msg, category=FutureWarning)
```



In [36]:

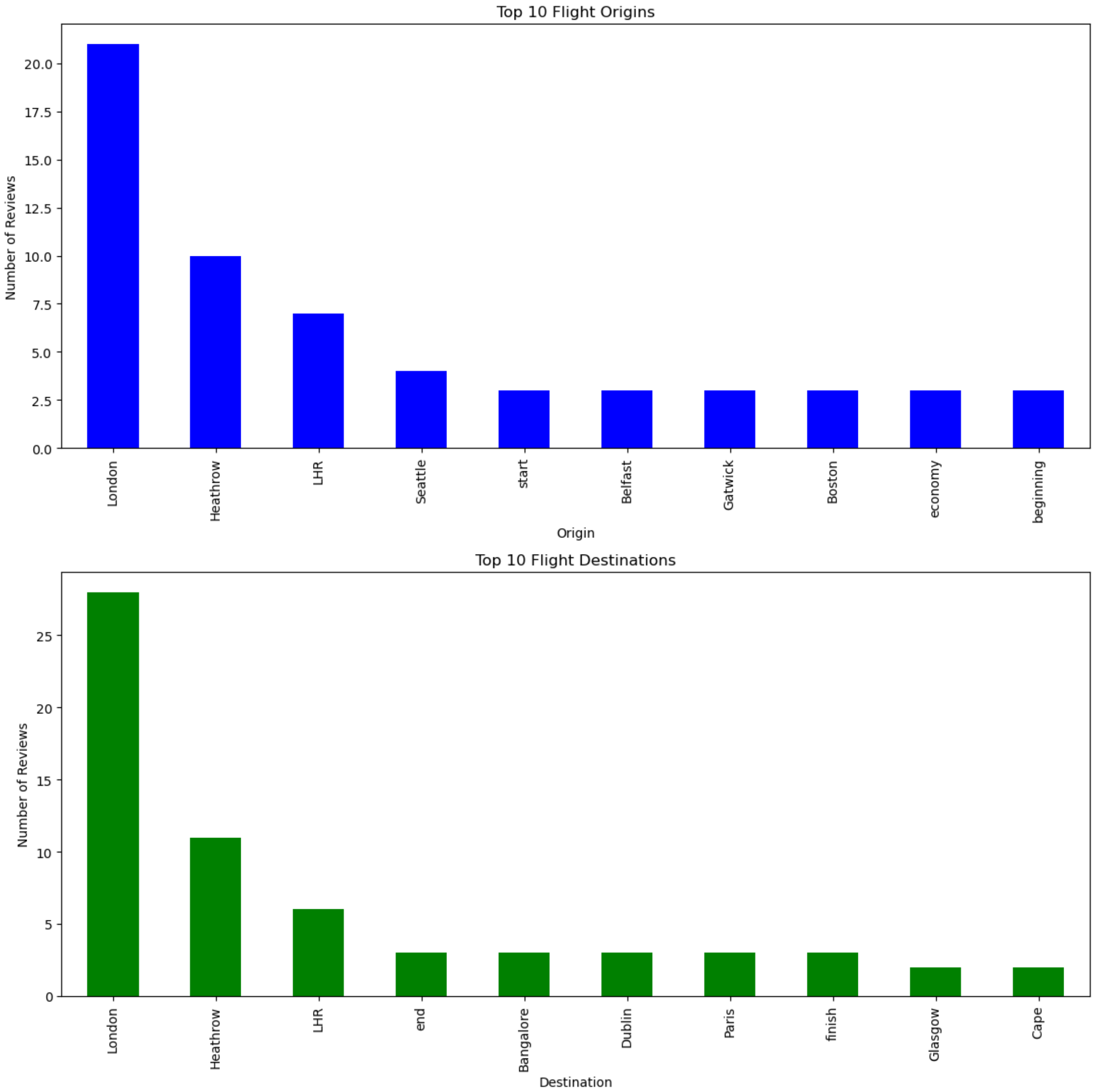
```
import pandas as pd
import matplotlib.pyplot as plt


# Extract Information
reviews_df['class'] = None
reviews_df.loc[reviews_df['cleaned_reviews'].str.contains('econom(y|ic)', ca
reviews_df.loc[reviews_df['cleaned_reviews'].str.contains('business', case=F

class_counts = reviews_df['class'].value_counts()

# Visualization
plt.figure(figsize=(10, 6))
class_counts.plot(kind='bar', color=['blue', 'green'])
plt.title('Counts of Reviews Mentioning Economic or Business Class')
plt.xlabel('Class')
plt.ylabel('Number of Reviews')
plt.show()
```

```
/var/folders/lf/hwptlvxd6vv42x9tfj9kdx800000gn/T/ipykernel_83389/2169667684.
py:7: UserWarning: This pattern is interpreted as a regular expression, and
has match groups. To actually get the groups, use str.extract.
  reviews_df.loc[reviews_df['cleaned_reviews'].str.contains('econom(y|ic)',
case=False, regex=True), 'class'] = 'Economic'
```



In [37]:

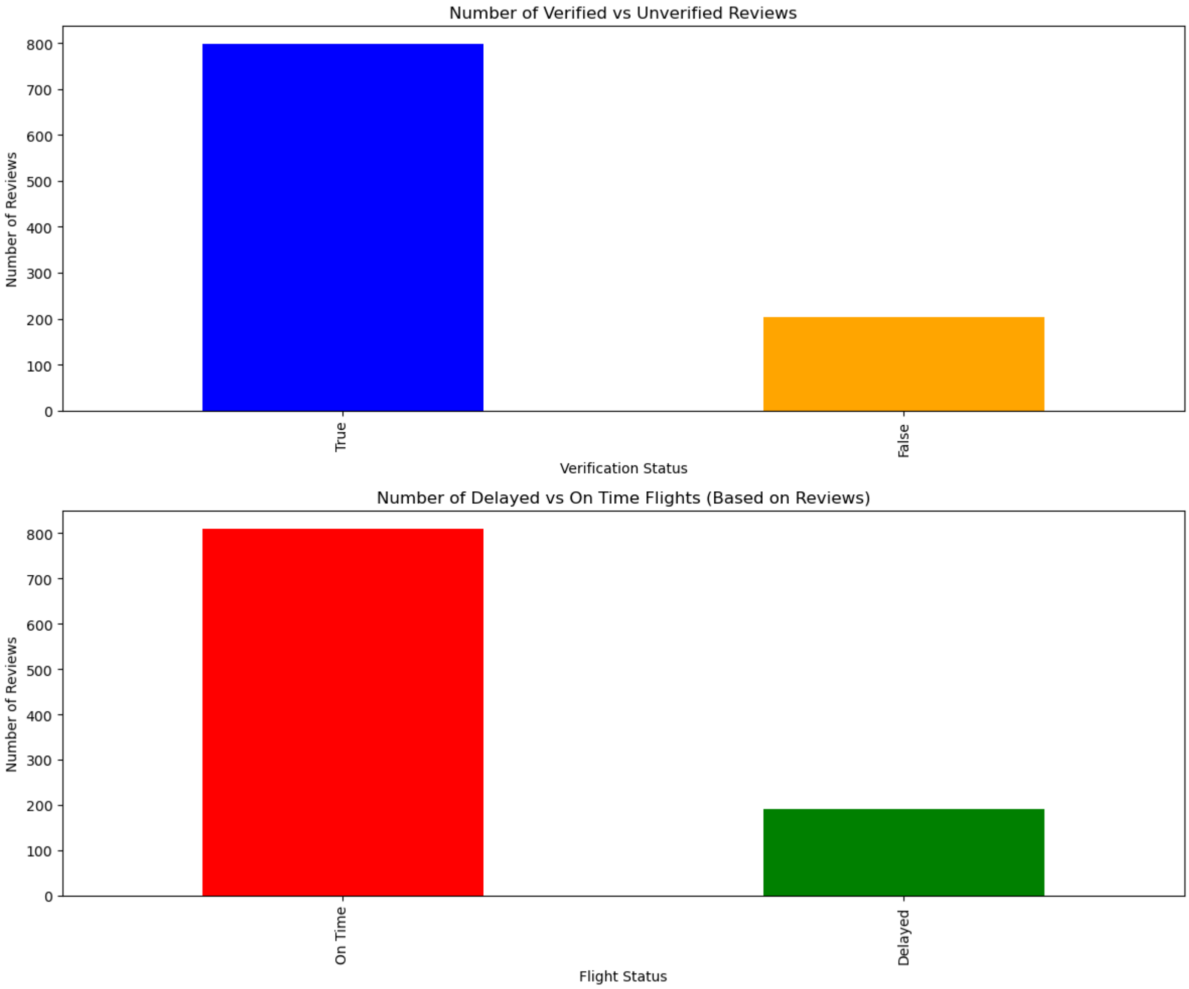
```
import pandas as pd
import matplotlib.pyplot as plt
import re

# Extract Information
def extract_flight_info(text):
    match = re.search(r'from (\w+) to (\w+)', text, re.I)
    if match:
        return match.groups()
    return None, None

reviews_df['origin'], reviews_df['destination'] = zip(*reviews_df['cleaned_r
origin_counts = reviews_df['origin'].value_counts().head(10)  # Top 10 origi
destination_counts = reviews_df['destination'].value_counts().head(10)  # To

# Visualization
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(12, 12))

# Top origins
origin_counts.plot(kind='bar', ax=axes[0], color='blue')
axes[0].set_title('Top 10 Flight Origins')
axes[0].set_xlabel('Origin')
axes[0].set_ylabel('Number of Reviews')

# Top destinations
destination_counts.plot(kind='bar', ax=axes[1], color='green')
axes[1].set_title('Top 10 Flight Destinations')
axes[1].set_xlabel('Destination')
axes[1].set_ylabel('Number of Reviews')

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



In [38]:

```
import pandas as pd
import matplotlib.pyplot as plt


# Extract Information for verified/unverified flights
reviews_df['verified'] = reviews_df['reviews'].str.contains('✅ Trip Verifie

# Extract Information for delayed flights
reviews_df['delayed'] = reviews_df['cleaned_reviews'].apply(lambda x: 'Delay

# Counts
verified_counts = reviews_df['verified'].value_counts()
delayed_counts = reviews_df['delayed'].value_counts()

# Visualization
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(12, 10))

# Verified/Unverified counts
verified_counts.plot(kind='bar', ax=axes[0], color=['blue', 'orange'])
axes[0].set_title('Number of Verified vs Unverified Reviews')
axes[0].set_xlabel('Verification Status')
axes[0].set_ylabel('Number of Reviews')

# Delayed/On Time counts
delayed_counts.plot(kind='bar', ax=axes[1], color=['red', 'green'])
axes[1].set_title('Number of Delayed vs On Time Flights (Based on Reviews)')
```

```
axes[1].set_xlabel('Flight Status')
axes[1].set_ylabel('Number of Reviews')

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

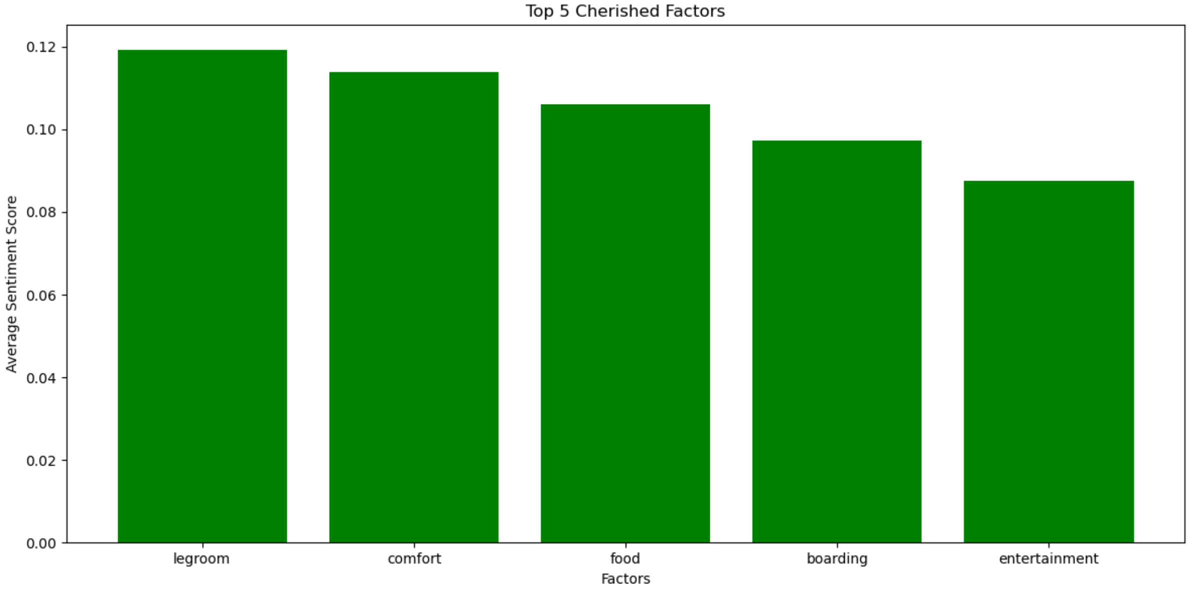


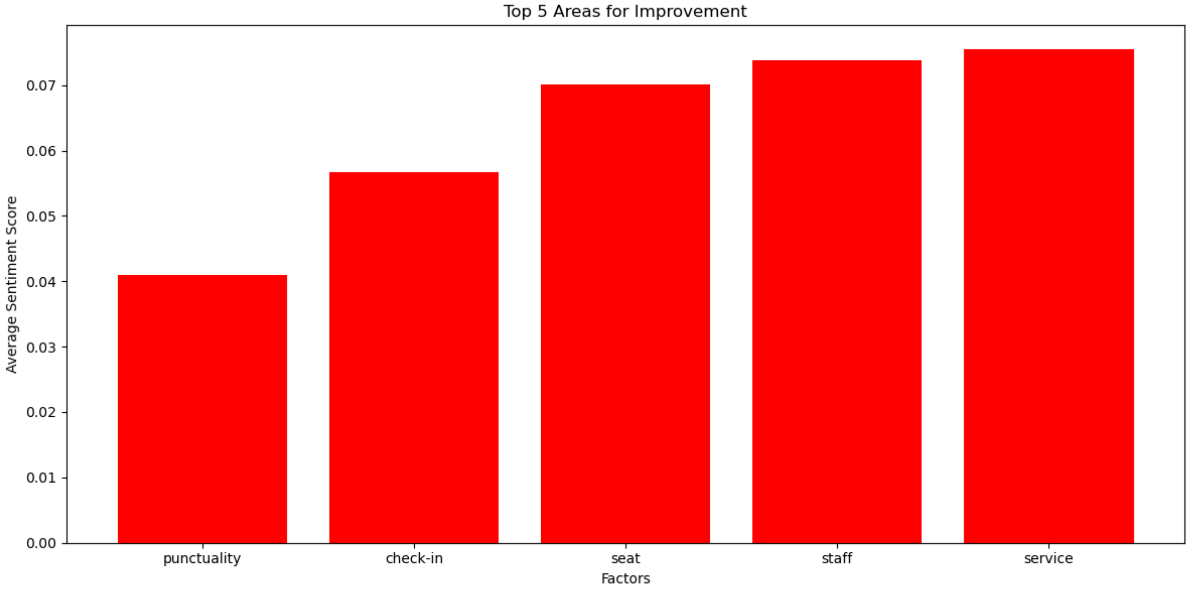
In [ ]:

```
To determine the factors that influence passengers' decisions to rebook with

Let's break the task down:

Keyword Extraction: Identify commonly mentioned factors/aspects in the revie
Sentiment Analysis: Determine the sentiment tied to each keyword.
Visualization: Display the aspects that received the most positive and negat
Here's a plan:

Create a list of potential keywords (factors/aspects) related to airline ser
For each keyword, extract reviews containing that keyword.
Compute the sentiment of each of these reviews.
Aggregate the sentiments for each keyword.
Visualize the top cherished factors (positive sentiments) and areas for impr
```

In [51]:

```
import pandas as pd
import matplotlib.pyplot as plt
from textblob import TextBlob

# 1. Keyword Extraction
keywords = ['seat', 'food', 'service', 'punctuality', 'entertainment', 'staf

keyword_sentiments = {}

# 2. Sentiment Analysis
for keyword in keywords:
    keyword_reviews = reviews_df[reviews_df['cleaned_reviews'].str.contains(
    sentiments = keyword_reviews['cleaned_reviews'].apply(lambda review: Tex
    keyword_sentiments[keyword] = sentiments.mean()

# 3. Aggregate and Sort
cherished_factors = sorted(keyword_sentiments.items(), key=lambda x: x[1], r
areas_for_improvement = sorted(keyword_sentiments.items(), key=lambda x: x[1

# 4. Visualization
# Displaying top 5 cherished factors and areas for improvement
top_cherished = dict(cherished_factors[:5])
top_improve = dict(areas_for_improvement[:5])

fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(12, 12))

# Cherished factors
axes[0].bar(top_cherished.keys(), top_cherished.values(), color='green')
axes[0].set_title('Top 5 Cherished Factors')
axes[0].set_xlabel('Factors')
axes[0].set_ylabel('Average Sentiment Score')

# Areas for improvement
axes[1].bar(top_improve.keys(), top_improve.values(), color='red')
axes[1].set_title('Top 5 Areas for Improvement')
axes[1].set_xlabel('Factors')
axes[1].set_ylabel('Average Sentiment Score')

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


Top 5 Cherished Factors
Average Sentiment Score
0.12
0.10
0.08
0.06
0.04
0.02
0.00
legroom
comfort
food
boarding
entertainment
Factors


Top 5 Areas for Improvement
Average Sentiment Score
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
punctuality
check-in
seat
staff
service
Factors

```
In [ ]:
```