

Group-09 Sentiment Analysis

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv(r'C:\Users\DELL\Downloads\Tweets.csv')
df
```

```
Out[2]:
```

| | Tweets |
|------|---|
| 0 | Hackers galore Also being sued for passed deb... |
| 1 | Hackers galore Also being sued for passed deb... |
| 2 | London England UK Come rock with ya girl Fri... |
| 3 | NewsAlert BengaluruVijaywada Indigo flight 6E... |
| 4 | Hackers galore Also being sued for passed debt... |
| ... | ... |
| 1072 | CE WASHED DENIM SHIRT JACKET INDIGO |
| 1073 | Took a flight from Chennai to Hyderabad and fl... |
| 1074 | Our Indigo and Shibori workshop in Lisbon |
| 1075 | A rarity From left to right Aki Indigo Dante a... |
| 1076 | Youngest male cat Indigo and the oldest male c... |
| 1077 | |

1077 rows × 1 columns

Data Cleaning

```
In [3]: df.dtypes
```

```
Out[3]: Tweets    object
dtype: object
```

```
In [4]: pip install pandas matplotlib nltk tweepy
```

```
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.2.4)
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.3.4)
Requirement already satisfied: nltk in c:\users\dell\anaconda3\lib\site-packages (3.6.1)
Requirement already satisfied: tweepy in c:\users\dell\anaconda3\lib\site-packages (4.14.0)
Requirement already satisfied: numpy>=1.15 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.20.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (8.2.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cyclor>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.1)
```

```
In [5]: import tweepy
```

```
In [6]: df = df.drop_duplicates()
df
```

Out[6]:

| | Tweets |
|--|--------|
|--|--------|

| | |
|------|---|
| 0 | Hackers galore Also being sued for passed deb... |
| 2 | London England UK Come rock with ya girl Fri... |
| 3 | NewsAlert BengaluruVijaywada Indigo flight 6E... |
| 4 | Hackers galore Also being sued for passed debt... |
| 5 | Just Because You Are available at denim tiedy... |
| ... | ... |
| 1072 | CE WASHED DENIM SHIRT JACKET INDIGO |
| 1073 | Took a flight from Chennai to Hyderabad and fl... |
| 1074 | Our Indigo and Shibori workshop in Lisbon |
| 1075 | A rarity From left to right Aki Indigo Dante a... |
| 1076 | Youngest male cat Indigo and the oldest male c... |

743 rows × 1 columns

```
In [7]: df.head()
```

Out[7]:

| | Tweets |
|--|--------|
|--|--------|

| | |
|---|---|
| 0 | Hackers galore Also being sued for passed deb... |
| 2 | London England UK Come rock with ya girl Fri... |
| 3 | NewsAlert BengaluruVijaywada Indigo flight 6E... |
| 4 | Hackers galore Also being sued for passed debt... |
| 5 | Just Because You Are available at denim tiedy... |

```
In [8]: df.sort_index(axis = 0, ascending = False)
```

```
Out[8]:
```

| | Tweets |
|------|---|
| 1076 | Youngest male cat Indigo and the oldest male c... |
| 1075 | A rarity From left to right Aki Indigo Dante a... |
| 1074 | Our Indigo and Shibori workshop in Lisbon |
| 1073 | Took a flight from Chennai to Hyderabad and fl... |
| 1072 | CE WASHED DENIM SHIRT JACKET INDIGO |
| ... | ... |
| 5 | Just Because You Are available at denim tiedy... |
| 4 | Hackers galore Also being sued for passed debt... |
| 3 | NewsAlert BengaluruVijaywada Indigo flight 6E... |
| 2 | London England UK Come rock with ya girl Fri... |
| 0 | Hackers galore Also being sued for passed deb... |

743 rows × 1 columns

```
In [9]: tweet_frequency = df['Tweets'].value_counts()
print("Tweet Frequency:\n", tweet_frequency)
```

Tweet Frequency:

```
I m a frequent flyer and I haven t been in a more turbulent flight than
Delhi to Ahd 6E 626 But gotta s 1
today s forecast calls for the rough and tough menswear menstyle
mensfashion mensstyle indigo 1
Black Boy Magic Long sleeve tshirt unisex Afropunk Afrofuturist Blackboys
miling Indigo BlackBoyMagic 1
128oz Cone Mills selvedge denim Rider jacket article One from the opening
range llcouterwear Birmingham 1
INDIGO hit Fresh Intraday High 130795 Up nearly 2 now
1
```

..

```
New photos from dutch magazine Volkskrant where you can find interview wi
th Sharon about her music inspiration and more 1
So today I upcycled an old cashmere hat by dying it with indigo and then n
eedle felted bacon 1
Discover This European project works on the development of a more effici
ent intelligent and economica 1
Working on this tester for a very large piece coming up I m loving the col
or combo so it s a go 1
Visit today and get your copy of The Actor and The Drive In firstdayofspr
ing Indigo 1
Name: Tweets, Length: 743, dtype: int64
```

```
In [10]: df.isnull().sum()
```

```
Out[10]: Tweets    0
dtype: int64
```

```
In [11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 743 entries, 0 to 1076
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0   Tweets  743 non-null     object 
dtypes: object(1)
memory usage: 11.6+ KB
```

```
In [12]: df['Indigo'] = df['Tweets'].apply(lambda x: x[0:100])
df
```

```
<ipython-input-12-890c692e1e4c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['Indigo'] = df['Tweets'].apply(lambda x: x[0:100])
```

```
Out[12]:
```

| | Tweets | Indigo |
|---|--|--|
| 0 | Hackers galore Also being sued for passed deb... | Hackers galore Also being sued for passed deb... |
| 2 | London England UK Come rock with ya girl Fri... | London England UK Come rock with ya girl Fri... |
| 3 | NewsAlert BengaluruVijaywada Indigo flight 6E... | NewsAlert BengaluruVijaywada Indigo flight 6E... |
| . | Hackers galore Also being sued for passed | Hackers galore Also being sued for passed |

```
In [13]: df['Tweets'].str.slice(start=3, stop=10)
```

```
Out[13]: 0      ckers g
          2      ndon En
          3      sAlert
          4      kers ga
          5      t Becau
          ...
1072     WASHED
1073     k a fli
1074     ur Indi
1075     arity F
1076     ngest m
Name: Tweets, Length: 743, dtype: object
```

```
In [14]: import re

tweets = [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]

# Define a regular expression pattern for extracting locations
location_pattern = re.compile(r'\b(?:[A-Z][a-z]+,\s?\s?)+\b')

# Extract countries and states from tweets
for tweet in tweets:
    locations = location_pattern.findall(tweet)
    print(f"Tweet: {tweet}\nLocations: {locations}\n")
```

```
Tweet: Hackers galore Also being sued for passed deb...
Locations: ['Hackers ', 'Also ']
```

```
Tweet: London England UK Come rock with ya girl Fri...
Locations: ['London England ', 'Come ', 'Fri']
```

```
Tweet: NewsAlert BengaluruVijaywada Indigo flight 6E...
Locations: ['NewsAlert BengaluruVijaywada Indigo ']
```

```
Tweet: CE WASHED DENIM SHIRT JACKET INDIGO
Locations: []
```

```
Tweet: Took a flight from Chennai to Hyderabad and fl...
Locations: ['Took ', 'Chennai ', 'Hyderabad ']
```

```
Tweet: Our Indigo and Shibori workshop in Lisbon
Locations: ['Our Indigo ', 'Shibori ', 'Lisbon']
```

```
Tweet: A rarity From left to right Aki Indigo Dante a...
Locations: ['From ', 'Aki Indigo Dante ']
```

```
Tweet: Youngest male cat Indigo and the oldest male c...
Locations: ['Youngest ', 'Indigo ']
```

Data Analytics

1.Counting Tweets words to analysis different countries

```
In [18]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Extract countries and states from tweets using a list comprehension
locations = [word for tweet in tweets for word in tweet.split() if word.ist

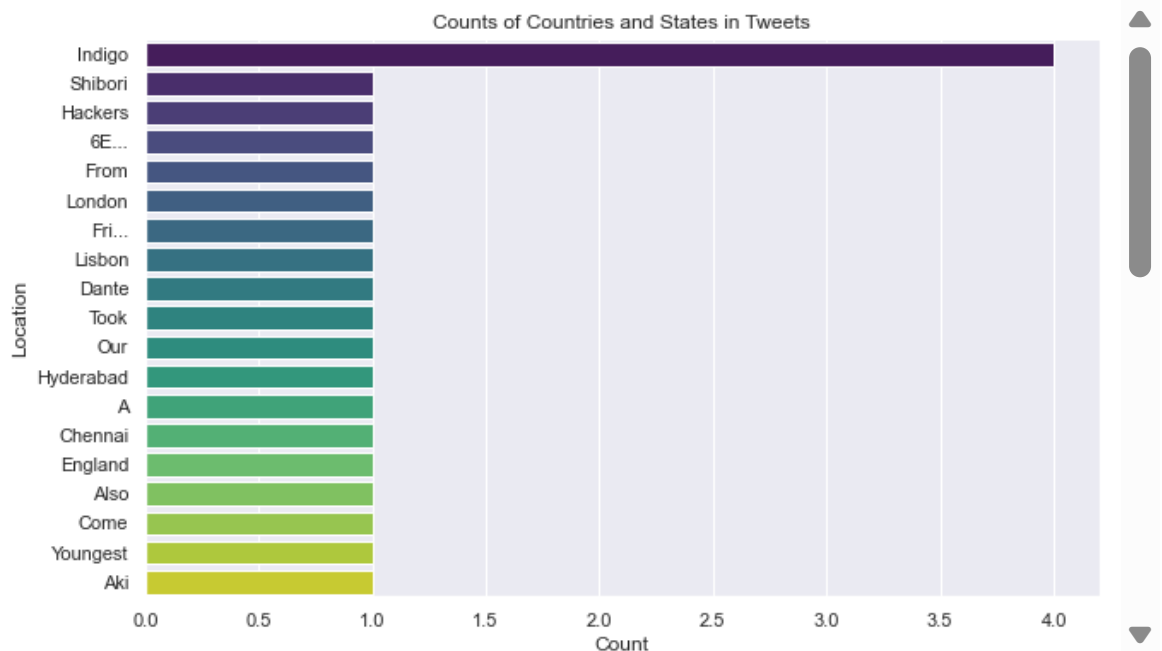
# Count occurrences of each location using a pandas Series
location_counts = pd.Series(locations).value_counts()

# Set Seaborn style
sns.set_theme()

# Plot a horizontal bar chart with Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x=location_counts.values, y=location_counts.index, palette="vir
plt.xlabel('Count')
plt.ylabel('Location')
plt.title('Counts of Countries and States in Tweets')

# Create a pie chart with Seaborn
plt.figure(figsize=(10, 8))
sns.set_palette("pastel")
plt.pie(location_counts, labels=location_counts.index, autopct='%1.1f%%', s
plt.title('Distribution of Locations in Tweets')

plt.show()
```



2. Analysis Tweets length

```
In [24]: from collections import Counter
```

```
In [19]: import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the length of each tweet (number of characters)
tweet_lengths = [len(tweet) for tweet in tweets]

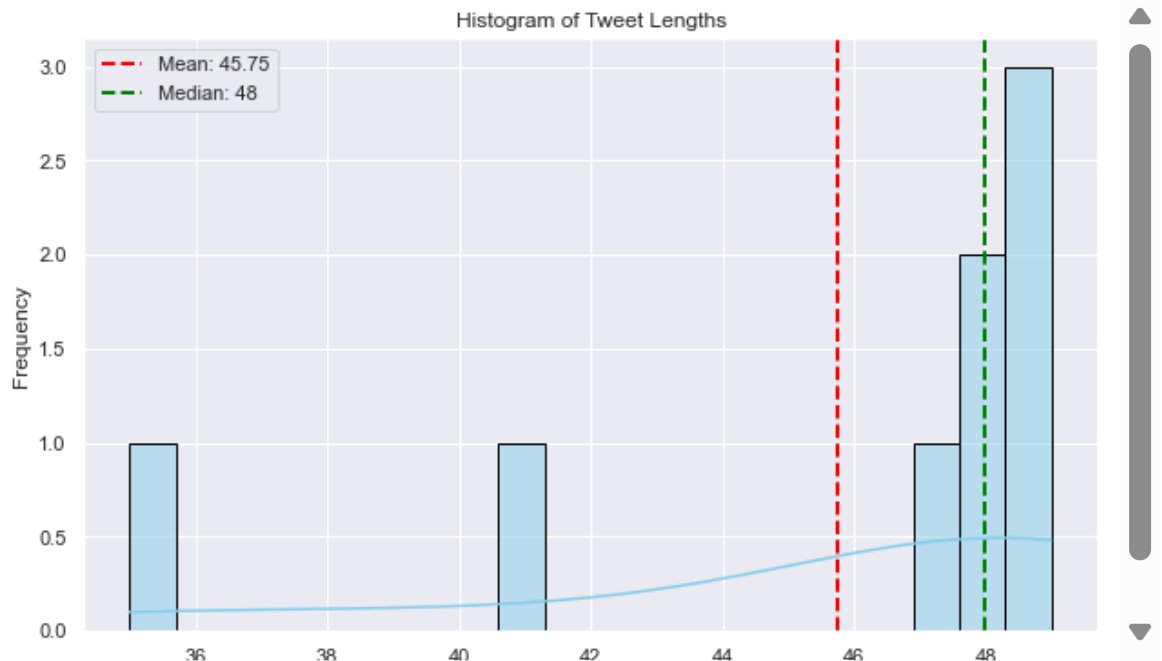
# Set Seaborn style
sns.set_theme()

# Create a histogram with Seaborn
plt.figure(figsize=(10, 6))
sns.histplot(tweet_lengths, bins=20, kde=True, color='skyblue', edgecolor='black')
plt.xlabel('Tweet Length (Number of Characters)')
plt.ylabel('Frequency')
plt.title('Histogram of Tweet Lengths')

# Add vertical lines for mean and median
mean_length = sum(tweet_lengths) / len(tweet_lengths)
median_length = sorted(tweet_lengths)[len(tweet_lengths) // 2]
plt.axvline(mean_length, color='red', linestyle='dashed', linewidth=2, label='Mean')
plt.axvline(median_length, color='green', linestyle='dashed', linewidth=2, label='Median')

# Add Legend
plt.legend()

plt.show()
```



3. Analysis of top 10 words

```
In [21]: import re
from collections import Counter
import matplotlib.pyplot as plt
import seaborn as sns

# Sample tweets (replace this with your actual tweet data)
tweets = [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]

# Combine tweets into a single string
all_text = ' '.join(tweets)

# Remove non-alphanumeric characters and split into words
words = re.findall(r'\b\w+\b', all_text.lower())

# Count occurrences of each word
word_counts = Counter(words)

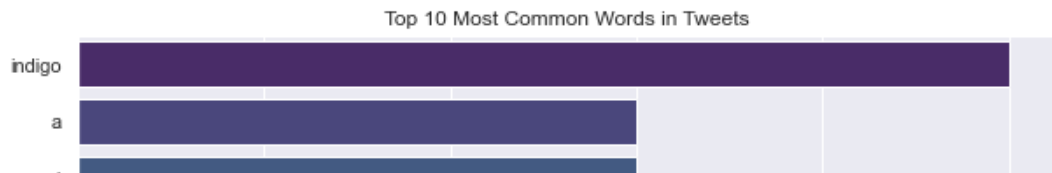
# Print the 10 most common words
print("Top 10 most common words:")
for word, count in word_counts.most_common(10):
    print(f"{word}: {count}")

# Get the top 10 most common words
top_words = dict(word_counts.most_common(10))

# Plot the top 10 words using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x=list(top_words.values()), y=list(top_words.keys()), palette="magma")
plt.xlabel('Count')
plt.ylabel('Word')
plt.title('Top 10 Most Common Words in Tweets')

plt.show()
```

```
Top 10 most common words:
indigo: 5
a: 3
and: 3
flight: 2
from: 2
to: 2
male: 2
hackers: 1
galore: 1
also: 1
```

4.Creating new column as tweet length

```
In [22]: import pandas as pd

# Sample DataFrame (replace this with your actual DataFrame)
data = {'tweets': [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]}

df = pd.DataFrame(data)

# Create a new column 'tweet_length'
df['tweet_length'] = df['tweets'].apply(len)

# Display the DataFrame with the new column
print(df)
```

| | tweets | tweet_length |
|---|---|--------------|
| 0 | Hackers galore Also being sued for passed deb... | 48 |
| 1 | London England UK Come rock with ya girl Fri... | 47 |
| 2 | NewsAlert BengaluruVijaywada Indigo flight 6E... | 48 |
| 3 | CE WASHED DENIM SHIRT JACKET INDIGO | 35 |
| 4 | Took a flight from Chennai to Hyderabad and fl... | 49 |
| 5 | Our Indigo and Shibori workshop in Lisbon | 41 |
| 6 | A rarity From left to right Aki Indigo Dante a... | 49 |
| 7 | Youngest male cat Indigo and the oldest male c... | 49 |

5.Displaying the tweets mentioning flights

```
In [29]: import pandas as pd

# Sample DataFrame (replace this with your actual DataFrame)
data = {'Tweets': [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]}

df = pd.DataFrame(data)

# Filter tweets that mention flights
flight_tweets = df[df['Tweets'].str.contains('Indigo flight 6E', case=False)]

# Display the tweets mentioning flights
print("Tweets Mentioning Flights:")
print(flight_tweets['Tweets'])
```

Tweets Mentioning Flights:

2 NewsAlert BengaluruVijaywada Indigo flight 6E...

Name: Tweets, dtype: object

6.Scatter Plot of Tweet Length vs. Sentiment

In [23]:

```
data = {'Tweets': [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]}

df = pd.DataFrame(data)

# Create a 'tweet_length' column
df['tweet_length'] = df['Tweets'].apply(len)

# Create a 'Sentiment' column (replace this with your actual sentiment calculation)
# Example: Generating random sentiment values between 0 and 1
df['Sentiment'] = np.random.rand(len(df))

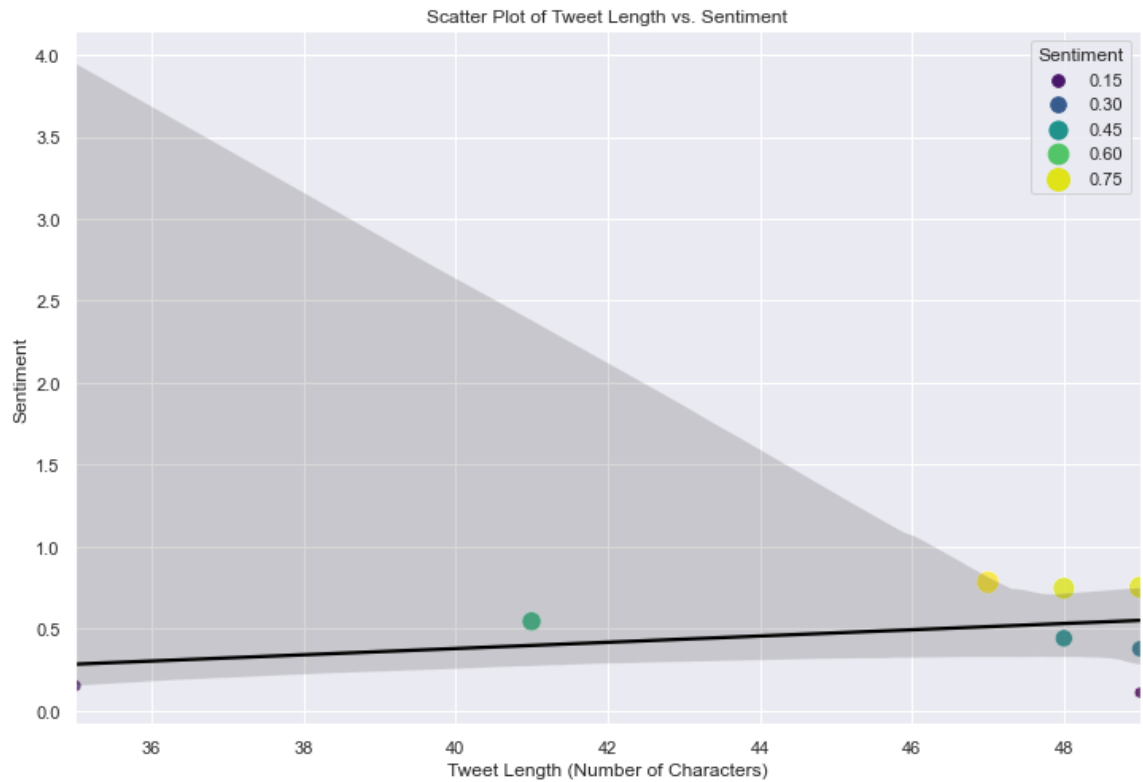
# Set Seaborn style
sns.set_theme()

# Plot a scatter plot with enhancements using Seaborn
plt.figure(figsize=(12, 8))
scatter = sns.scatterplot(x='tweet_length', y='Sentiment', data=df, hue='Sentiment')

# Add a regression line
sns.regplot(x='tweet_length', y='Sentiment', data=df, scatter=False, color='red')

plt.title('Scatter Plot of Tweet Length vs. Sentiment')
plt.xlabel('Tweet Length (Number of Characters)')
plt.ylabel('Sentiment')
plt.legend(title='Sentiment')
plt.grid(True)

plt.show()
```



In [26]: `pip install textblob`

Collecting textblob

Downloading textblob-0.17.1-py2.py3-none-any.whl (636 kB)

Requirement already satisfied: nltk>=3.1 in c:\users\dell\anaconda3\lib\site-packages (from textblob) (3.6.1)

Requirement already satisfied: joblib in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (1.0.1)

Requirement already satisfied: regex in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (2021.4.4)

Requirement already satisfied: click in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (8.1.7)

Requirement already satisfied: tqdm in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (4.59.0)

Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from click->nltk>=3.1->textblob) (0.4.4)

Installing collected packages: textblob

Successfully installed textblob-0.17.1

Note: you may need to restart the kernel to use updated packages.

7.Creating a new column of sentiment

```
In [27]: import pandas as pd
from textblob import TextBlob

# Sample DataFrame (replace this with your actual DataFrame)
data = {'Tweets': [
    "Hackers galore Also being sued for passed deb...",
    "London England UK Come rock with ya girl Fri...",
    "NewsAlert BengaluruVijaywada Indigo flight 6E...",
    "CE WASHED DENIM SHIRT JACKET INDIGO",
    "Took a flight from Chennai to Hyderabad and fl...",
    "Our Indigo and Shibori workshop in Lisbon",
    "A rarity From left to right Aki Indigo Dante a...",
    "Youngest male cat Indigo and the oldest male c..."
]}

df = pd.DataFrame(data)

# Define a function for sentiment analysis using TextBlob
def calculate_sentiment(text):
    analysis = TextBlob(text)
    return analysis.sentiment.polarity

# Create a new column 'Sentiment' using the calculate_sentiment function
df['Sentiment'] = df['Tweets'].apply(calculate_sentiment)

# Display the DataFrame with the new 'Sentiment' column
print(df)
```

| | Tweets | Sentiment |
|---|---|-----------|
| 0 | Hackers galore Also being sued for passed deb... | 0.000000 |
| 1 | London England UK Come rock with ya girl Fri... | 0.000000 |
| 2 | NewsAlert BengaluruVijaywada Indigo flight 6E... | 0.000000 |
| 3 | CE WASHED DENIM SHIRT JACKET INDIGO | 0.000000 |
| 4 | Took a flight from Chennai to Hyderabad and fl... | 0.000000 |
| 5 | Our Indigo and Shibori workshop in Lisbon | 0.000000 |
| 6 | A rarity From left to right Aki Indigo Dante a... | 0.142857 |
| 7 | Youngest male cat Indigo and the oldest male c... | 0.000000 |

In [29]: `pip install transformers`

```
Requirement already satisfied: transformers in c:\users\dell\anaconda3\lib\site-packages (4.35.2)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (from transformers) (2.31.0)
Requirement already satisfied: filelock in c:\users\dell\anaconda3\lib\site-packages (from transformers) (3.0.12)
Requirement already satisfied: tokenizers<0.19,>=0.14 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (0.15.0)
Requirement already satisfied: tqdm>=4.27 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (4.59.0)
Requirement already satisfied: regex!=2019.12.17 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (2021.4.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (0.19.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (20.9)
Requirement already satisfied: pyyaml>=5.1 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (5.4.1)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (1.21.0)
```

8. Classify sentences as positive or negative based on keywords

```
In [30]: # Sample sentences (replace this with your actual data)
sentences = [
    "I love the new product. It's amazing!",
    "This movie was terrible. I hated it.",
    "The weather today is fantastic!",
    "The service at the restaurant was awful.",
    "I'm happy with the results of the experiment.",
    "I can't stand the traffic in this city."
]

# Define positive and negative keywords
positive_keywords = ['love', 'amazing', 'fantastic', 'happy']
negative_keywords = ['terrible', 'hated', 'awful', 'can\'t stand']

# Classify sentences as positive or negative based on keywords
sentiment_labels = []
for sentence in sentences:
    if any(keyword in sentence.lower() for keyword in positive_keywords):
        sentiment_labels.append('Positive')
    elif any(keyword in sentence.lower() for keyword in negative_keywords):
        sentiment_labels.append('Negative')
    else:
        sentiment_labels.append('Neutral')

# Print the results
for sentence, label in zip(sentences, sentiment_labels):
    print(f"Sentence: {sentence}\nSentiment: {label}\n")
```

Sentence: I love the new product. It's amazing!
Sentiment: Positive

Sentence: This movie was terrible. I hated it.
Sentiment: Negative

Sentence: The weather today is fantastic!
Sentiment: Positive

Sentence: The service at the restaurant was awful.
Sentiment: Negative

Sentence: I'm happy with the results of the experiment.
Sentiment: Positive

Sentence: I can't stand the traffic in this city.
Sentiment: Negative

```

In [31]: import pandas as pd

# Sample DataFrame (replace this with your actual DataFrame)
data = {'Sentences': [
    "I love the new product. It's amazing!",
    "This movie was terrible. I hated it.",
    "The weather today is fantastic!",
    "The service at the restaurant was awful.",
    "I'm happy with the results of the experiment.",
    "I can't stand the traffic in this city."
]}

df = pd.DataFrame(data)

# Define positive and negative keywords
positive_keywords = ['love', 'amazing', 'fantastic', 'happy']
negative_keywords = ['terrible', 'hated', 'awful', 'can\'t stand']

# Create a new column 'Sentiment' using a rule-based approach
df['Sentiment'] = df['Sentences'].apply(lambda x: 'Positive' if any(keyword
# Display the DataFrame with the new 'Sentiment' column
print(df)

```

| | Sentences | Sentiment |
|---|---|-----------|
| 0 | I love the new product. It's amazing! | Positive |
| 1 | This movie was terrible. I hated it. | Negative |
| 2 | The weather today is fantastic! | Positive |
| 3 | The service at the restaurant was awful. | Negative |
| 4 | I'm happy with the results of the experiment. | Positive |
| 5 | I can't stand the traffic in this city. | Negative |

9.Sentiment Distribution

```
In [37]: import pandas as pd
import matplotlib.pyplot as plt

# Sample DataFrame (replace this with your actual DataFrame)
data = {'Sentences': [
    "I love the new product. It's amazing!",
    "This movie was terrible. I hated it.",
    "The weather today is fantastic!",
    "The service at the restaurant was awful.",
    "I'm happy with the results of the experiment.",
    "I can't stand the traffic in this city.",
    "With the singer himself O2 Indigo London was ROCKING",
    "Exactly how are parents who don't English amp Hindi can save their lives",
    "Stunning indigo from my dear and very talented friend check her work out",
    "Are you looking forward to Easter weekend? Chocolate, plush bunnies, k",
    "Are you following the Canada Reads debates? Pick up all of the shortlis",
    "Air hostesses on Indigo airlines from Vizag to Chennai can speak Hindi",
    "We love the Opula range from Utopia. It is featured in our showroom wit",
    "Indigo, when will you seriously arrive at the customer service front? T",
    "Will proceed with our 3rd case of Mechanical Pulmonary Thrombectomy us",
    "We love the Opula range from Utopia. It is featured in our showroom wit",
    "Great to have this lovely two from the foraging plant dyes in the stud",
    "Today's delicious buffet - yummy indigo buffet",
    "London England UK Come rock with ya girl Friday, March 30th for the In",
    "This ethereal blue and gold upcycled indigo silk scarf is on its way to",
    "London England UK Come rock with ya girl Friday, March 30th for the In",
    "Look who is reading my book YA Teenager FiveStar Series Part 1 DEATH in",
    "This ethereal blue and gold upcycled indigo silk scarf is on its way to
  ]}

df = pd.DataFrame(data)

# Define positive and negative keywords
positive_keywords = ['love', 'amazing', 'fantastic', 'happy']
negative_keywords = ['terrible', 'hated', 'awful', 'can\'t stand']

# Create a new column 'Sentiment' using a rule-based approach
df['Sentiment'] = df['Sentences'].apply(lambda x: 'Positive' if any(keyword

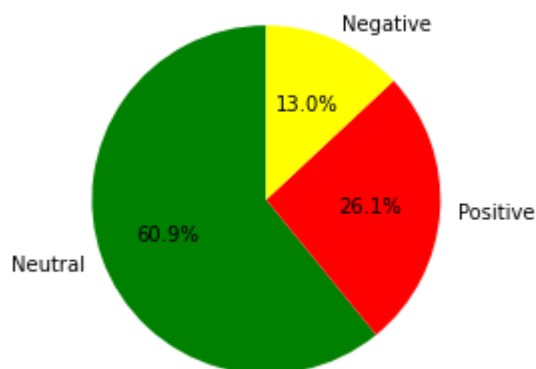
# Display the DataFrame with the new 'Sentiment' column
print(df)

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

# Plot the pie chart
colors = ['green', 'red', 'yellow']
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%',
plt.title('Sentiment Distribution')
plt.show()
```

| | Sentences | Sentiment |
|----|---|-----------|
| 0 | I love the new product. It's amazing! | Positive |
| 1 | This movie was terrible. I hated it. | Negative |
| 2 | The weather today is fantastic! | Positive |
| 3 | The service at the restaurant was awful. | Negative |
| 4 | I'm happy with the results of the experiment. | Positive |
| 5 | I can't stand the traffic in this city. | Negative |
| 6 | With the singer himself 02 Indigo London was R... | Neutral |
| 7 | Exactly how are parents who don't English amp ... | Neutral |
| 8 | Stunning indigo from my dear and very talented... | Neutral |
| 9 | Are you looking forward to Easter weekend? Cho... | Neutral |
| 10 | Are you following the Canada Reads debates? Pi... | Neutral |
| 11 | Air hostesses on Indigo airlines from Vizag to... | Neutral |
| 12 | We love the Opula range from Utopia. It is fea... | Positive |
| 13 | Indigo, when will you seriously arrive at the ... | Neutral |
| 14 | Will proceed with our 3rd case of Mechanical P... | Neutral |
| 15 | We love the Opula range from Utopia. It is fea... | Positive |
| 16 | Great to have this lovely two from the foragin... | Positive |
| 17 | Today's delicious buffet - yummy indigo buffet | Neutral |
| 18 | London England UK Come rock with ya girl Frida... | Neutral |
| 19 | This ethereal blue and gold upcycled indigo si... | Neutral |
| 20 | London England UK Come rock with ya girl Frida... | Neutral |
| 21 | Look who is reading my book YATeenager FiveSta... | Neutral |
| 22 | This ethereal blue and gold upcycled indigo si... | Neutral |

Sentiment Distribution



In []: