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

df = pd.read_csv('twitter_training.csv')

df
```

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...	...	...	...	...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74681 rows × 4 columns

Next steps: Generate code with df View recommended plots

```
df.head
```

pandas.core.generic.NDFrame.head
`def head(n: int=5) -> NDFrameT`

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py
Return the first `n` rows.

This function returns the first `n` rows for the object based
on position. It is useful for quickly testing if your object
has the right type of data in it.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74681 entries, 0 to 74680
Data columns (total 4 columns):
 #   Column                                               Non-Null Count  Dtype 
---  ------                                               --------------  ----- 
 0   2401                                                 74681 non-null  int64 
 1   Borderlands                                          74681 non-null  object
 2   Positive                                             74681 non-null  object
 3   im getting on borderlands and i will murder you all ,  73995 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

```
df.shape
```

```
(74681, 4)
```

```
df.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None, datetime_is_numeric: bool_t=False) ->
NDFrameT
    - 'all' : All columns of the input will be included in the output.
    - A list-like of dtypes : Limits the results to the
      provided data types.
      To limit the result to numeric types submit
      ``numpy.number``. To limit it instead to object columns submit
      the ``numpy.object`` data type. Strings
      can also be used in the style of
```

```
col_names = ['ID', 'Borderlands', 'Positive', 'Content']
data = pd.read_csv('twitter_training.csv', names=col_names)
```

```
#checking null values
```

```
data.isnull().sum()
```

```
ID              0
Borderlands     0
Positive        0
Content       686
dtype: int64
```

```
data.dropna(axis=0 , inplace=True)
```

```
data.isna().sum()
```

```
ID             0
Borderlands    0
Positive       0
Content        0
dtype: int64
```

```
#checking duplicates
data.duplicated().sum()
```

```
2341
```

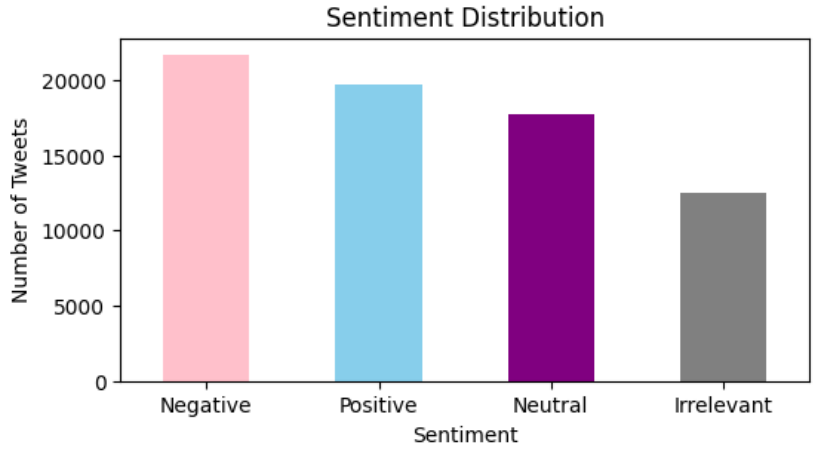
```
data.drop_duplicates(inplace=True)
data.duplicated().sum()
```

```
0
```

```
sentiment_counts = data['Positive'].value_counts()
sentiment_counts
```

```
Negative      21698
Positive      19713
Neutral       17707
Irrelevant    12537
Name: Positive, dtype: int64
```

```
#Barplot visualization
plt.figure(figsize=(6, 3))
sentiment_counts.plot(kind='bar', color=['pink', 'skyblue', 'purple', 'grey'])
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Number of Tweets')
plt.xticks(rotation=0)
plt.show()
```

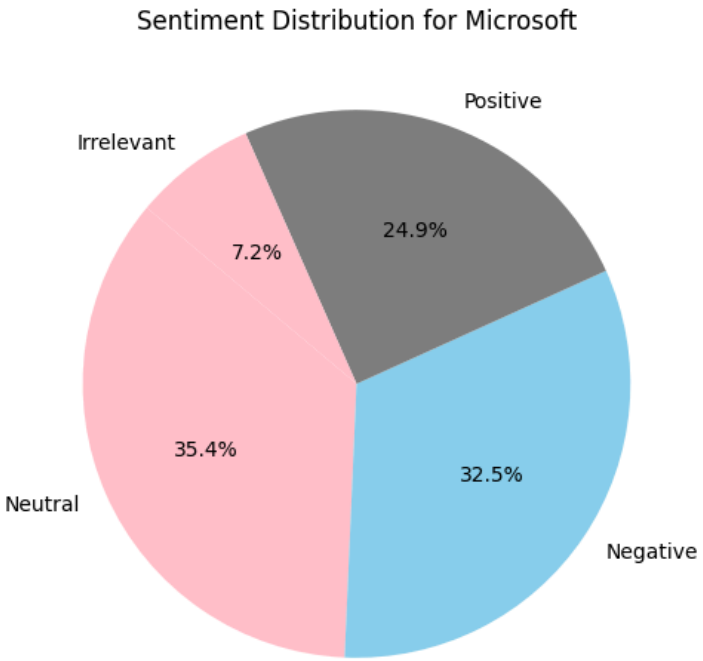



```
brand_data = data[data['Borderlands'].str.contains('Microsoft', case=False)]
brand_sentiment_counts = brand_data['Positive'].value_counts()
brand_sentiment_counts
```

```
Neutral       816
Negative      748
Positive      573
Irrelevant    167
Name: Positive, dtype: int64
```

```
custom_colors = ['pink', 'skyblue', 'grey']  # Example custom colors for positive, neutral, and negative sentiments

plt.figure(figsize=(6, 6))
plt.pie(brand_sentiment_counts, labels=brand_sentiment_counts.index, autopct='%1.1f%%', startangle=140, colors=custom_colors)
plt.title('Sentiment Distribution for Microsoft')
plt.show()
```


Sentiment Distribution for Microsoft
Positive
Irrelevant
24.9%
7.2%
35.4%
32.5%
Neutral
Negative