```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
df = pd.read_csv('Tweets.csv')
df.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	air
0	570306133677760513	neutral	1.0000	NaN	NaN	\ Am
1	570301130888122368	positive	0.3486	NaN	0.0000	\ Am
2	570301083672813571	neutral	0.6837	NaN	NaN	\ Am
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	\ Am
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	\ Am

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
print(df.columns)
```

```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline',
       'airline_sentiment_gold', 'name', 'negativereason_gold',
       'retweet_count', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

```
# Keep only needed columns
df = df[['text', 'airline_sentiment']]

# Check for null values
print(df.isnull().sum())
```

```
text                0
airline_sentiment    0
dtype: int64
```

```
def get_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity

df['polarity'] = df['text'].apply(get_sentiment)
df.head()
```

	text	airline_sentiment	polarity
0	@VirginAmerica What @dhepburn said.	neutral	0.000000
1	@VirginAmerica plus you've added commercials t...	positive	0.000000
2	@VirginAmerica I didn't today... Must mean I n...	neutral	-0.390625
3	@VirginAmerica it's really aggressive to blast...	negative	0.006250
4	@VirginAmerica and it's a really big bad thing...	negative	-0.350000

Next steps: Generate code with df | View recommended plots | New interactive sheet

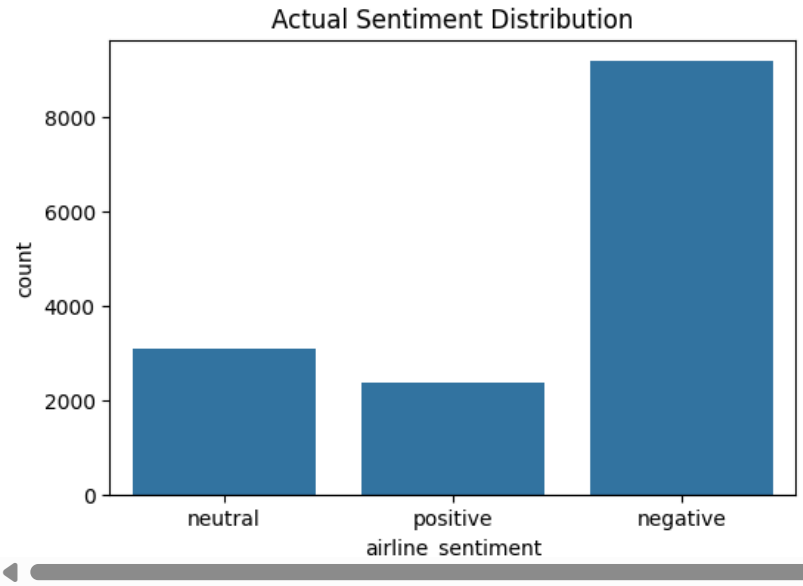
```
def classify_sentiment(polarity):
    if polarity > 0:
        return 'positive'
    elif polarity < 0:
        return 'negative'
    else:
        return 'neutral'

df['predicted_sentiment'] = df['polarity'].apply(classify_sentiment)
df.head()
```

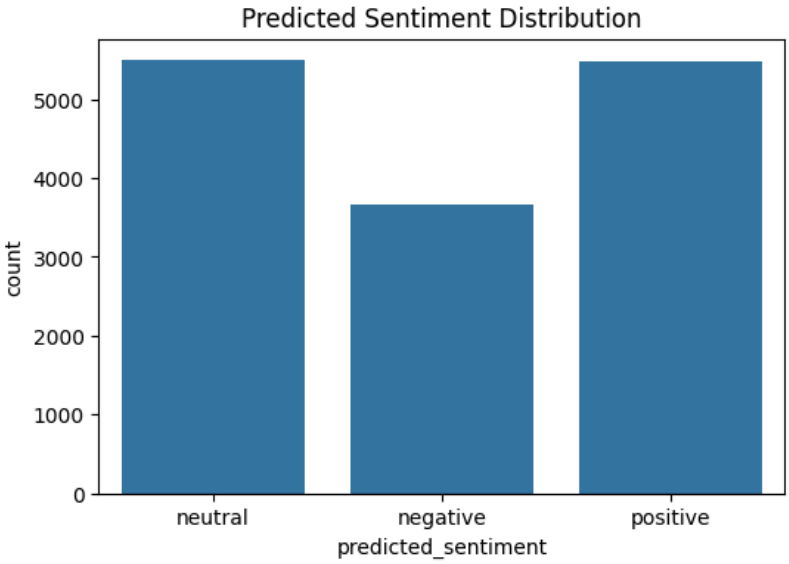
	text	airline_sentiment	polarity	predicted_sentiment
0	@VirginAmerica What @dhepburn said.	neutral	0.000000	neutral
1	@VirginAmerica plus you've added commercials t...	positive	0.000000	neutral
2	@VirginAmerica I didn't today... Must mean I n...	neutral	-0.390625	negative
3	@VirginAmerica it's really aggressive to blast...	negative	0.006250	positive
4	@VirginAmerica and it's a really big bad thing...	negative	-0.350000	negative

Next steps: Generate code with df | View recommended plots | New interactive sheet

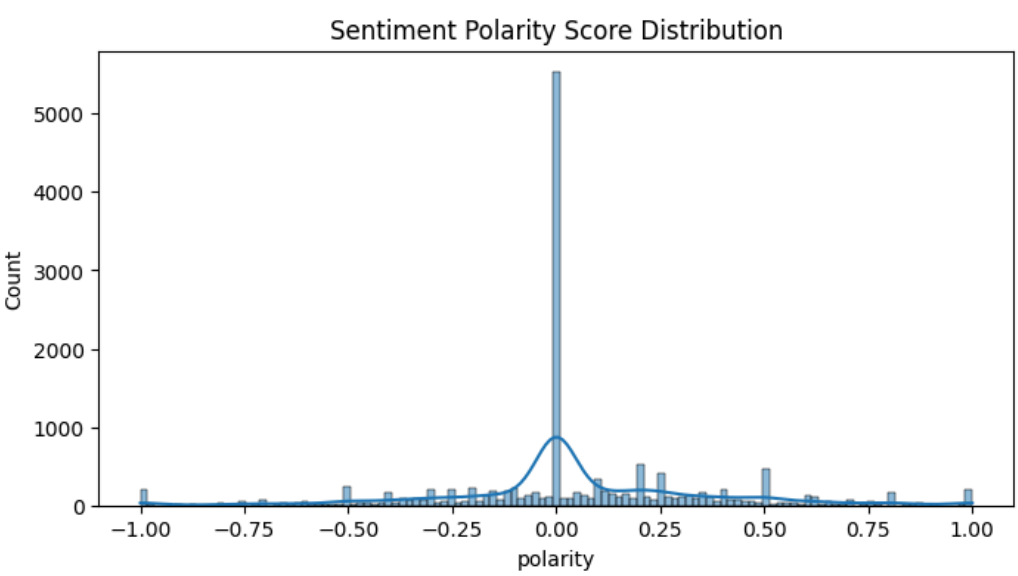
```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='airline_sentiment')
plt.title("Actual Sentiment Distribution")
plt.show()
```



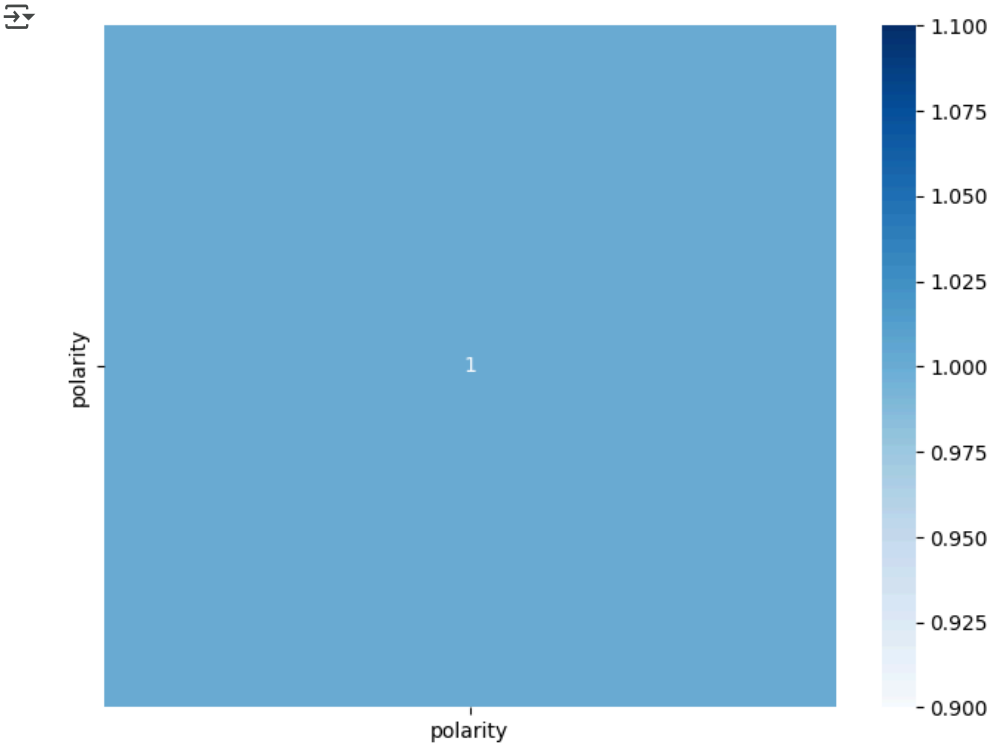

```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='predicted_sentiment')
plt.title("Predicted Sentiment Distribution")
plt.show()
```




```
plt.figure(figsize=(8,4))
sns.histplot(df['polarity'], kde=True)
plt.title("Sentiment Polarity Score Distribution")
plt.show()
```




```
# Check numeric columns only
numeric_df = df.select_dtypes(include=['float64', 'int64'])
plt.figure(figsize=(8,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='Blues')
plt.show()
```


polarity
1
polarity
1.100
1.075
1.050
1.025
1.000
0.975
0.950
0.925
0.900