PROGRAM :

[1]:
```
# Data Analysis
import pandas as pd
import numpy as np

# Data Visualization
from matplotlib import pyplot as plt
import seaborn as sns

# Machine Learning
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

# NLP
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from wordcloud import WordCloud, STOPWORDS
import re

# Warning
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
train_df = pd.read_csv('Tweets.csv')
print(f'Train data shape: {train_df.shape}')
train_df.head()
```

Train data shape: (14640, 15)

[2]:

	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
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	NaN	cairdin	NaN	0	@VirginAmerica What @dhepburn said.	NaN	2015-02-24 11:35:52 -0800	NaN	Eastern Time (US & Canada)
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica plus you've added commercials t...	NaN	2015-02-24 11:15:59 -0800	NaN	Pacific Time (US & Canada)
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn	NaN	0	@VirginAmerica I didn't today... Must mean I n...	NaN	2015-02-24 11:15:48 -0800	Lets Play	Central Time (US & Canada)
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica it's really aggressive to blast...	NaN	2015-02-24 11:15:36 -0800	NaN	Pacific Time (US & Canada)
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica and it's a really big bad thing...	NaN	2015-02-24 11:14:45 -0800	NaN	Pacific Time (US & Canada)

[3]:
```
test_df = pd.read_csv('Tweets.csv')
print(f'Test data shape: {test_df.shape}')
test_df.head()
```

Test data shape: (14640, 15)

[3]:

	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
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	NaN	cairdin	NaN	0	@VirginAmerica What @dhepburn said.	NaN	2015-02-24 11:35:52 -0800	NaN	Eastern Time (US & Canada)
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica plus you've added commercials t...	NaN	2015-02-24 11:15:59 -0800	NaN	Pacific Time (US & Canada)
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn	NaN	0	@VirginAmerica I didn't today... Must mean I n...	NaN	2015-02-24 11:15:48 -0800	Lets Play	Central Time (US & Canada)
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica it's really aggressive to blast...	NaN	2015-02-24 11:15:36 -0800	NaN	Pacific Time (US & Canada)
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica and it's a really big bad thing...	NaN	2015-02-24 11:14:45 -0800	NaN	Pacific Time (US & Canada)

```
  negativereason  negativereason_confidence         airline     \
0            NaN                        NaN  Virgin America
1            NaN                     0.0000  Virgin America
2            NaN                        NaN  Virgin America
3     Bad Flight                     0.7033  Virgin America
4     Can't Tell                     1.0000  Virgin America

  airline_sentiment_gold          name negativereason_gold  retweet_count  \
0                    NaN       cairdin                 NaN              0
1                    NaN     jnardino                 NaN              0
2                    NaN   yvonnalynn                 NaN              0
3                    NaN     jnardino                 NaN              0
4                    NaN     jnardino                 NaN              0

                                                text tweet_coord  \
0                @VirginAmerica What @dhepburn said.         NaN
1  @VirginAmerica plus you've added commercials t...         NaN
2  @VirginAmerica I didn't today... Must mean I n...         NaN
3  @VirginAmerica it's really aggressive to blast...         NaN
4  @VirginAmerica and it's a really big bad thing...         NaN

              tweet_created  tweet_location               user_timezone
0  2015-02-24 11:35:52 -0800             NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800             NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800       Lets Play  Central Time (US & Canada)
3  2015-02-24 11:15:36 -0800             NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800             NaN  Pacific Time (US & Canada)
```

[4]:
```
train_df.duplicated().sum()
```

[4]: 36

[5]:
```
train_df.dtypes
```

[5]:
```
tweet_id                        int64
airline_sentiment              object
airline_sentiment_confidence  float64
negativereason                 object
negativereason_confidence     float64
airline                        object
airline_sentiment_gold         object
name                           object
negativereason_gold            object
retweet_count                   int64
text                           object
tweet_coord                    object
```

```
tweet_created                 object
tweet_location                object
user_timezone                 object
dtype: object
```

[6]:
```
# Missing values check
print(f'Missing values in train data:\n{train_df.isnull().sum()}')
print('-'*40)
```

```
Missing values in train data:
tweet_id                          0
airline_sentiment                 0
airline_sentiment_confidence      0
negativereason                 5462
negativereason_confidence      4118
airline                           0
airline_sentiment_gold        14600
name                              0
negativereason_gold           14608
retweet_count                     0
text                              0
tweet_coord                   13621
tweet_created                     0
tweet_location                 4733
user_timezone                  4820
dtype: int64
----------------------------------------
```

[7]:
```
stopwords = set(STOPWORDS)

# Removing 'user' word as it does not hold any importance in our context
stopwords.add('user')

negative_tweets = train_df['text'][train_df['airline']==1].to_string()
wordcloud_negative = WordCloud(width = 800, height = 800,
                               background_color ='white', stopwords = stopwords,
                               min_font_size = 10).generate(negative_tweets)

positive_tweets = train_df['text'][train_df['airline']==0].to_string()
wordcloud_positive = WordCloud(width = 800, height = 800,
                               background_color ='white', stopwords = stopwords,
                               min_font_size = 10).generate(positive_tweets)

# Plotting the WordCloud images
plt.figure(figsize=(14, 6), facecolor=None)

plt.subplot(1, 2, 1)
```

```
plt.imshow(wordcloud_negative)
plt.axis("off")
plt.title("Negative Tweets", fontdict={'fontsize': 20})

plt.subplot(1, 2, 2)
plt.imshow(wordcloud_positive)
plt.axis("off")
plt.title("Positive Tweets", fontdict={'fontsize': 20})

plt.tight_layout()
plt.show()

plt.show()
```

Negative Tweets Positive Tweets

Series

Series

[8]:
```
# Feature Engineering
train_df_fe = train_df.copy()
train_df_fe["tweet_length"] = train_df_fe["text"].str.len()
train_df_fe["num_hashtags"] = train_df_fe["text"].str.count("#")
train_df_fe["num_exclamation_marks"] = train_df_fe["text"].str.count("\!")
train_df_fe["num_question_marks"] = train_df_fe["text"].str.count("\?")
train_df_fe["total_tags"] = train_df_fe["text"].str.count("@")
train_df_fe["num_punctuations"] = train_df_fe["text"].str.count("[.,:;]")
train_df_fe["num_question_marks"] = train_df_fe["text"].str.count("[*&$%]")
train_df_fe["num_words"] = train_df_fe["text"].apply(lambda x: len(x.split()))
train_df_fe.head()
```

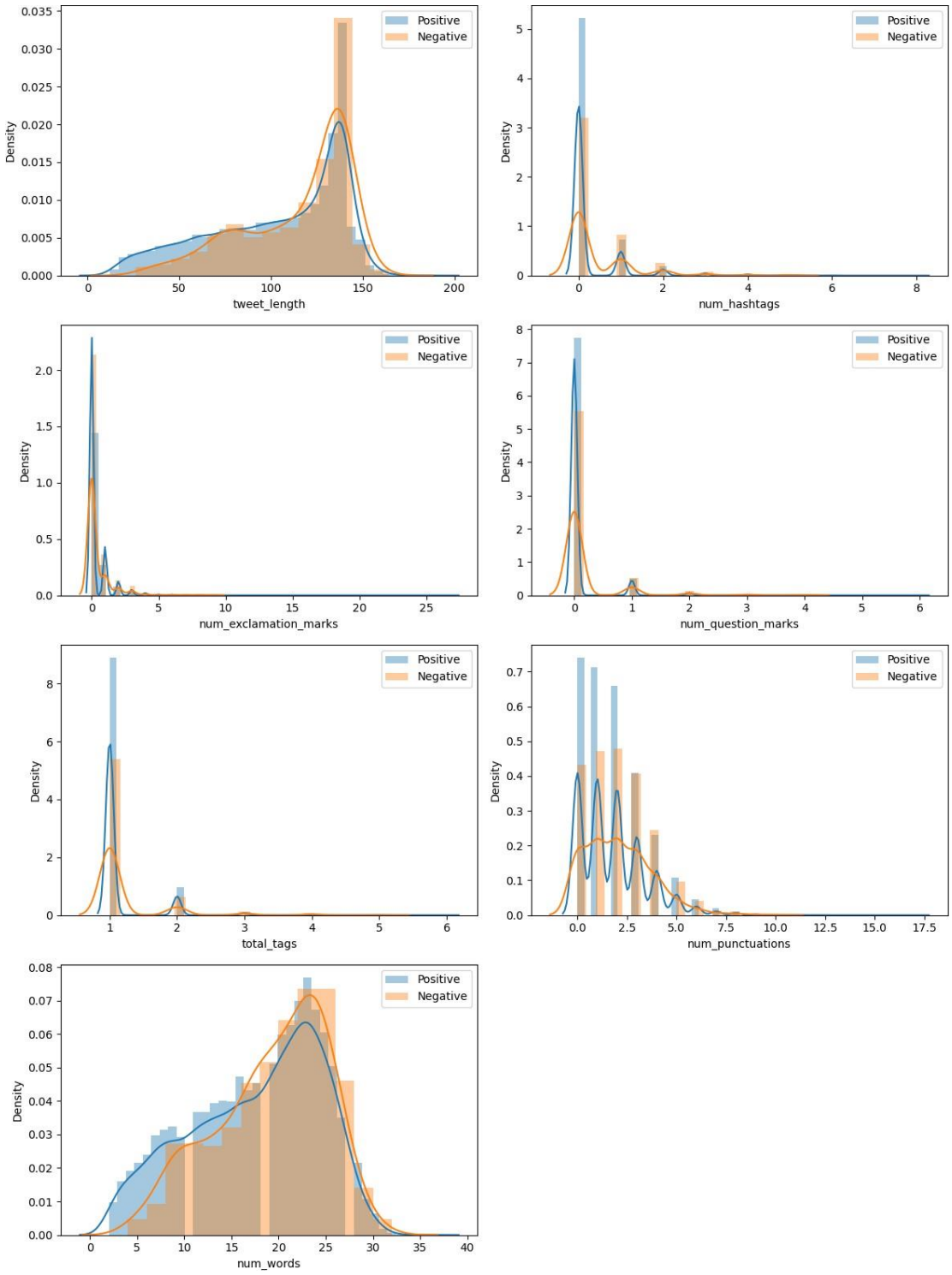
[8]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	\
0	570306133677760513	neutral	1.0000	
1	570301130888122368	positive	0.3486	

```
2  570301083672813571           neutral                        0.6837
3  570301031407624196          negative                        1.0000
4  570300817074462722          negative                        1.0000

  negativereason  negativereason_confidence         airline     \
0            NaN                       NaN  Virgin America
1            NaN                    0.0000  Virgin America
2            NaN                       NaN  Virgin America
3     Bad Flight                    0.7033  Virgin America
4     Can't Tell                    1.0000  Virgin America

  airline_sentiment_gold          name negativereason_gold   retweet_count  ... \
0                    NaN       cairdin                 NaN               0  ...
1                    NaN     jnardino                  NaN               0  ...
2                    NaN   yvonnalynn                  NaN               0  ...
3                    NaN     jnardino                  NaN               0  ...
4                    NaN     jnardino                  NaN               0  ...

              tweet_created  tweet_location               user_timezone  \
0  2015-02-24 11:35:52 -0800             NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800             NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800       Lets Play  Central Time (US & Canada)
3  2015-02-24 11:15:36 -0800             NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800             NaN  Pacific Time (US & Canada)

  tweet_length  num_hashtags  num_exclamation_marks  num_question_marks  \
0           35             0                      0                   0
1           72             0                      0                   0
2           71             0                      1                   0
3          126             0                      0                   1
4           55             0                      0                   0

   total_tags  num_punctuations  num_words
0           2                 1          4
1           1                 4          9
2           1                 3         12
3           1                 1         17
4           1                 0         10

[5 rows x 22 columns]
```

[9]:
```
# Visualizing relationship of newly created features with the tweet sentiments
plt.figure(figsize=(12, 16))
features = ['tweet_length', 'num_hashtags', 'num_exclamation_marks',␣
 ↪'num_question_marks',
            'total_tags', 'num_punctuations', 'num_words']
for i in range(len(features)):
```

```
    plt.subplot(4, 2, i+1)
    sns.distplot(train_df_fe[train_df_fe.retweet_count ==0][features[i]], label
 ↪= "Positive")
    sns.distplot(train_df_fe[train_df_fe.retweet_count ==1][features[i]], label
 ↪= "Negative")
    plt.legend()
plt.tight_layout()
plt.show()
```



[10]:
```
test = test_df
#Data Preprocessing
```

```
# Train-Test Splitting
X = train_df.drop(columns=['tweet_id'])
y = train_df['tweet_id']

print(X.shape, test.shape, y.shape)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=8)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(14640, 14) (14640, 15) (14640,)
(11712, 14) (2928, 14) (11712,) (2928,)
```

[11]:
```
# Function to tokenize and clean the text
def tokenize_and_clean(text):
    # Changing case of the text to lower case
    lowered = text.lower()

    # Cleaning the text
    cleaned = re.sub('@user', '', lowered)

    # Tokenization
    tokens = word_tokenize(cleaned)
    filtered_tokens = [token for token in tokens if re.match(r'\w{1,}', token)]

    # Stemming
    stemmer = PorterStemmer()
    stems = [stemmer.stem(token) for token in filtered_tokens]
    return stems
```

[12]:
```
import nltk
nltk.download('punkt')


# BOW Vectorization
# bow_vectorizer = CountVectorizer(tokenizer=tokenize_and_clean,␣
↪stop_words='english')
# X_train_tweets_bow = bow_vectorizer.fit_transform(X_train['tweet'])
# X_test_tweets_bow = bow_vectorizer.transform(X_test['tweet'])
# print(X_train_tweets_bow.shape, X_test_tweets_bow.shape)

# TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(tokenizer=tokenize_and_clean,␣
↪stop_words='english')
X_train_tweets_tfidf = tfidf_vectorizer.fit_transform(X_train['name'])
X_test_tweets_tfidf = tfidf_vectorizer.transform(X_test['name'])
```

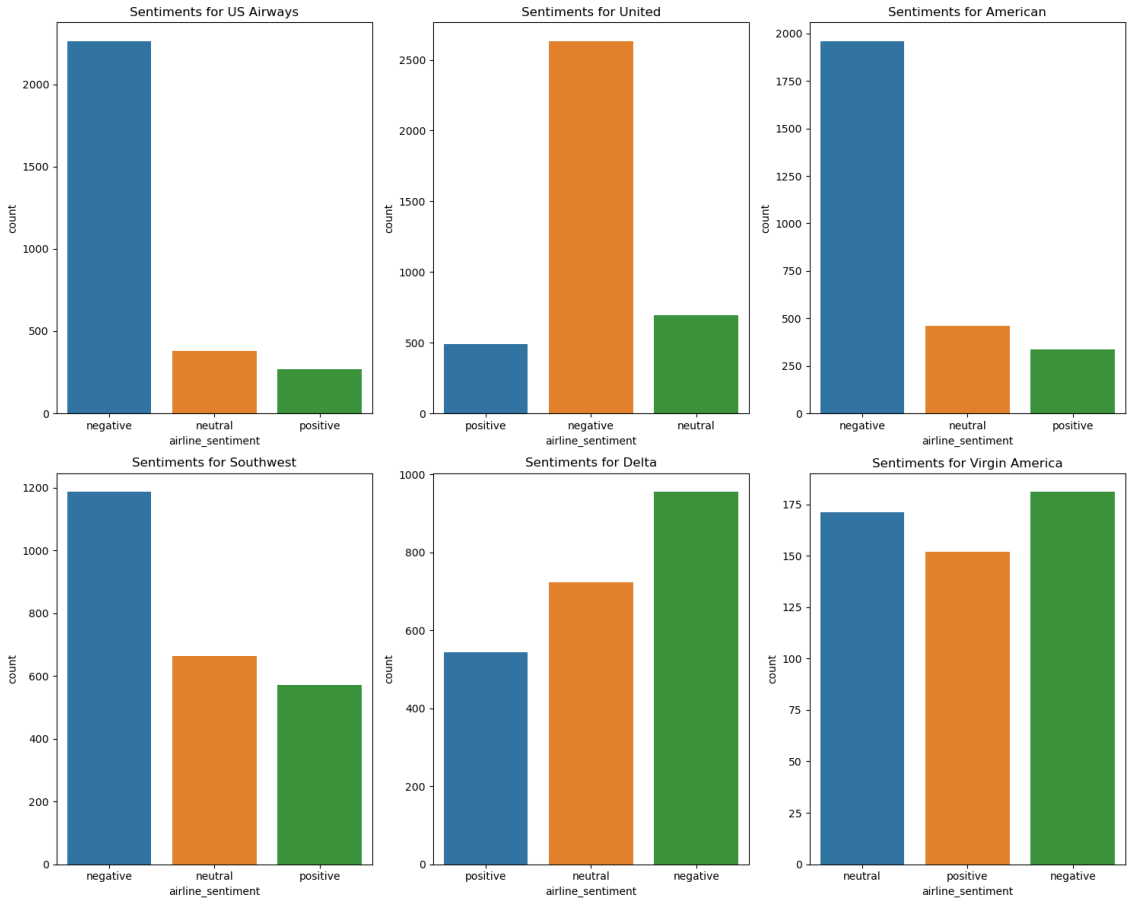
```
print(X_train_tweets_tfidf.shape,  X_test_tweets_tfidf.shape)

# TF-IDF Vectorization on full training data
tfidf_vectorizer = TfidfVectorizer(tokenizer=tokenize_and_clean,␣
↪stop_words='english')
X_tweets_tfidf = tfidf_vectorizer.fit_transform(X['name'])
test_tweets_tfidf = tfidf_vectorizer.transform(test['name'])
print(X_tweets_tfidf.shape,  test_tweets_tfidf.shape)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Ragu\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
(11712, 6730) (2928, 6730)
(14640, 7704) (14640, 7704)
```

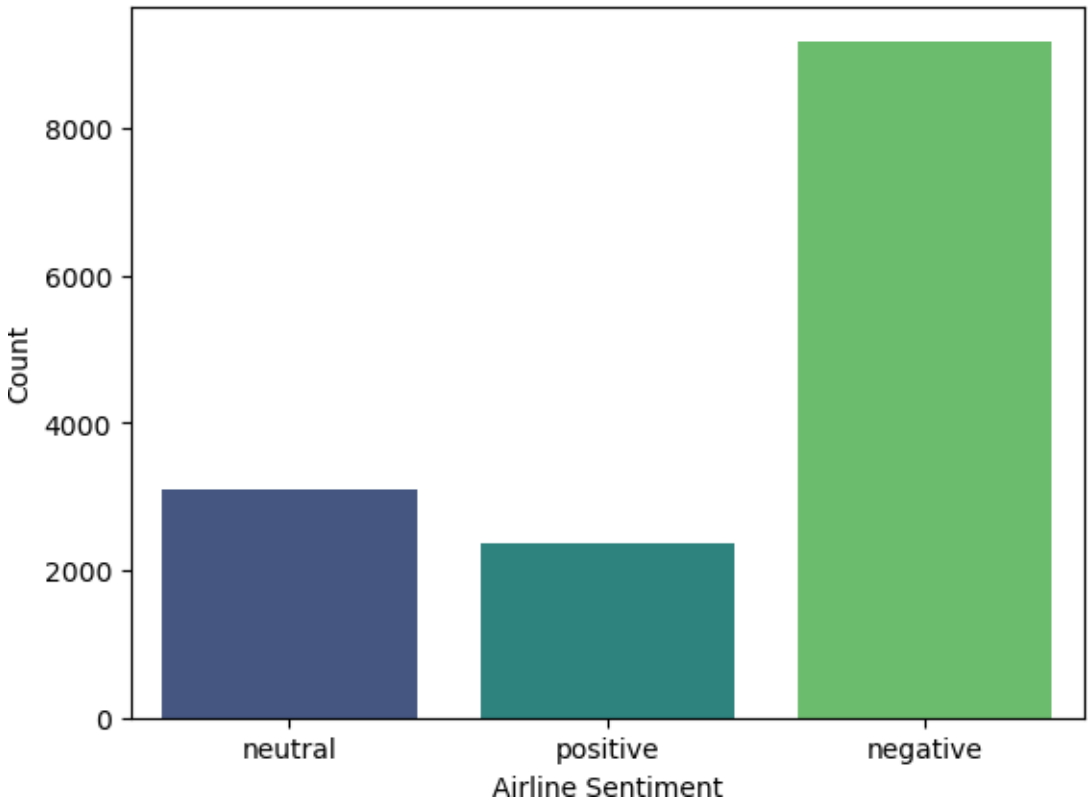
[13]:
```
plt.figure(1, figsize=(15, 12))  # Adjust the figsize as needed
airlines = ['US Airways', 'United', 'American', 'Southwest', 'Delta', 'Virgin␣
↪America']

for i, airline in enumerate(airlines, 1):
    plt.subplot(2, 3, i)
    new_value = train_df[train_df['airline'] == airline]

    print(new_value['airline_sentiment'].value_counts(), airline)

    sns.countplot(data=new_value, x='airline_sentiment')
    plt.title(f'Sentiments for {airline}')

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

```
negative    2263
neutral      381
positive     269
Name: airline_sentiment, dtype: int64 US Airways
negative    2633
neutral      697
positive     492
Name: airline_sentiment, dtype: int64 United
negative    1960
neutral      463
positive     336
Name: airline_sentiment, dtype: int64 American
negative    1186
neutral      664
positive     570
Name: airline_sentiment, dtype: int64 Southwest
```

```
negative     955
neutral      723
positive     544
Name: airline_sentiment, dtype: int64 Delta
negative     181
neutral      171
positive     152
Name: airline_sentiment, dtype: int64 Virgin America
```



[14]:
```
sns.countplot(train_df, x = 'airline_sentiment', palette= 'viridis');
plt.xlabel("Airline Sentiment")
plt.ylabel("Count")
plt.show()
```



[15]:
```
from transformers import pipeline
classifier = pipeline("sentiment-analysis")
texts = train_df['text'].tolist()
predictions = classifier(texts)
predictions[:5]
```

No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.

Downloading (…)lve/main/config.json: 0%| | 0.00/629 [00:00<?, ?B/s]

Downloading model.safetensors: 0%| | 0.00/268M [00:00<?, ?B/s]

Downloading (…)okenizer_config.json: 0%| | 0.00/48.0 [00:00<?, ?B/s]

Downloading (…)solve/main/vocab.txt: 0%| | 0.00/232k [00:00<?, ?B/s]

[15]:
```
[{'label': 'POSITIVE', 'score': 0.8633624911308289},
 {'label': 'POSITIVE', 'score': 0.6070874333381653},
 {'label': 'NEGATIVE', 'score': 0.9973426461219788},
```

```
{'label': 'NEGATIVE', 'score': 0.9973449110984802},
{'label': 'NEGATIVE', 'score': 0.9995823502540588}]
```

[19]:
```
submission = pd.DataFrame({"tweet_id":test_df.tweet_id, "label":predictions})
submission.head()

submission.to_csv("Submission.csv", index=False)
print("Submission is successful!")
```

```
Submission is successful!
```

[ ]: