8144-SUDHARSAN ENGINEERING COLLEGE

SUDHARSHAN EDUCATIONAL TRUST

Creating Pathways to Wisdom

SUDHARSAN
ENGINEERING COLLEGE

REGISTER NUMBER: 814421243024

NAME: ROHINI M

DEGREE: BTECH

BRANCH: ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PROJECT TITLE: SENTIMENT ANALYSIS FOR MARKETING

SENTIMENT ANALYSIS FOR MARKETING USING PYTHON

PHASE 4 SUBMISSION DOCUMENT

Phase 4: Development part-2

INTRODUCTION:

- Sentiment analysis, also known as opinion mining, is a powerful technique in marketing that involves analyzing and understanding the emotions, attitudes, and opinions expressed by customers, prospects, or the general public about a product, brand, or topic.
- It plays a crucial role in shaping marketing strategies and decision-making by providing valuable insights into how people perceive and interact with your brand or products.
- Sentiment analysis is a valuable tool in the field of marketing, helping businesses gain insights into how customers perceive their products, services, and brand.
- It involves using natural language processing (NLP) and machine learning techniques to analyze and categorize the sentiment expressed in text data, such as customer reviews, social media posts, surveys, and more.

FEATURE ENGINEERING:

Feature engineering in sentiment analysis for marketing involves creating relevant input features for machine learning models to effectively analyze and classify sentiment in textual data.

Text Preprocessing:

- **Tokenization:** Splitting text into individual words or phrases.
- **Lowercasing:** Converting all text to lowercase to ensure consistency.
- **Removing Stop Words:** Eliminating common words (e.g., "the," "and") that carry little sentiment information.

Text Representation:

- **Bag of Words (BoW):** Creating a matrix of word frequencies within the text.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Assigning weights to words based on their importance in the document and across the corpus.
- **Word Embeddings:** Using pre-trained word embeddings (e.g., Word2Vec, GloVe) to capture semantic relationships between words.

N-Grams:

Consider using bigrams or trigrams to capture sequences of words that convey specific sentiment.

Sentiment Lexicons:

Integrating sentiment lexicons or dictionaries to assign sentiment scores to words or phrases.

Part-of-Speech (POS) Tagging:

Identifying and categorizing words into parts of speech to capture grammatical structure.

Text Length:

Including features related to the length of the text, such as the number of words or characters, as text length can influence sentiment.

Emoticons and Symbols:

Considering the presence of emoticons, emojis, and symbols as they often convey sentiment.

Capitalization:

Creating features to detect the presence of capitalized words or phrases, which may indicate emphasis or sentiment.

Punctuation:

Analyzing the use of punctuation marks, such as exclamation points or question marks, which can express emotion.

Negation Handling:

Identifying negation words (e.g., "not," "but") and marking the words that are negated to reverse their sentiment.

Topic Modeling:

Applying topic modeling techniques (e.g., Latent Dirichlet Allocation) to identify the main topics in the text and understand their sentiment.

Domain-Specific Features:

Incorporating industry or domain-specific terms and knowledge relevant to the marketing context.

User and Brand Mentions:

Detecting mentions of specific users, competitors, or brand names to gauge sentiment in relation to them.

Sentiment Analysis of Meta-Information:

Analyzing the sentiment of metadata, such as timestamps, user profiles, or post types, as they can provide context for sentiment.

Contextual Features:

Capturing contextual information, such as the relationship between the author and the product/brand, to understand the context of sentiment.

Custom Features:

Creating custom features based on the unique requirements of the marketing analysis, such as sentiment-related metrics or ratios.

MODEL TRAINING:

Here are some tips for training a sentiment analysis model for marketing:

- ✓ Use a large and diverse dataset.
- ✓ The larger and more diverse your dataset, the better your model will be able to learn the nuances of human language and sentiment.
- ✓ Use a balanced dataset.
- ✓ Make sure that your dataset has an equal number of positive, negative, and neutral samples.
- ✓ This will help to prevent your model from being biased towards one particular sentiment.
- ✓ Use feature engineering to improve the performance of your model.
- ✓ Feature engineering involves creating new features from the existing data that may be more informative for sentiment analysis.
- ✓ For example, you could create a feature that counts the number of exclamation points in a sentence, as this can be a signal of positive sentiment.
- ✓ Use cross-validation to evaluate your model.
- ✓ Cross-validation involves splitting the dataset into multiple folds and training and evaluating the model on each fold.
- ✓ This helps to provide a more accurate estimate of the model's generalization performance.

EVALUATION:

- ✓ Evaluating a sentiment analysis model for marketing is important to ensure that the model is accurate and reliable.
- ✓ Make sure that the model is trained on a dataset that is relevant to your marketing campaigns.
- ✓ For example, if you are using sentiment analysis to monitor social media conversations, make sure that the model is trained on a dataset of social media posts.
- ✓ Use a variety of evaluation metrics to get a complete picture of the model's performance.
- ✓ Accuracy is a good starting point, but you should also consider precision, recall, and F1 score.
- ✓ Compare the model's performance to other sentiment analysis models.
- ✓ This can help you to determine how well your model performs relative to other models.
- ✓ Evaluate the model's performance over time. Sentiment analysis models can degrade over time as the language changes.
- ✓ It is important to evaluate the model's performance regularly to ensure that it is still accurate and reliable.
- ✓ Get feedback from users. Once you have deployed the model in production, get feedback from users on the accuracy and reliability of the model's predictions.
- ✓ This feedback can help you to identify any areas where the model needs to be improved.

PROGRAM:

[1]:
```python
# 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]:
```python
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  \
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

                                               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)
```

[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  \
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

                                                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
```

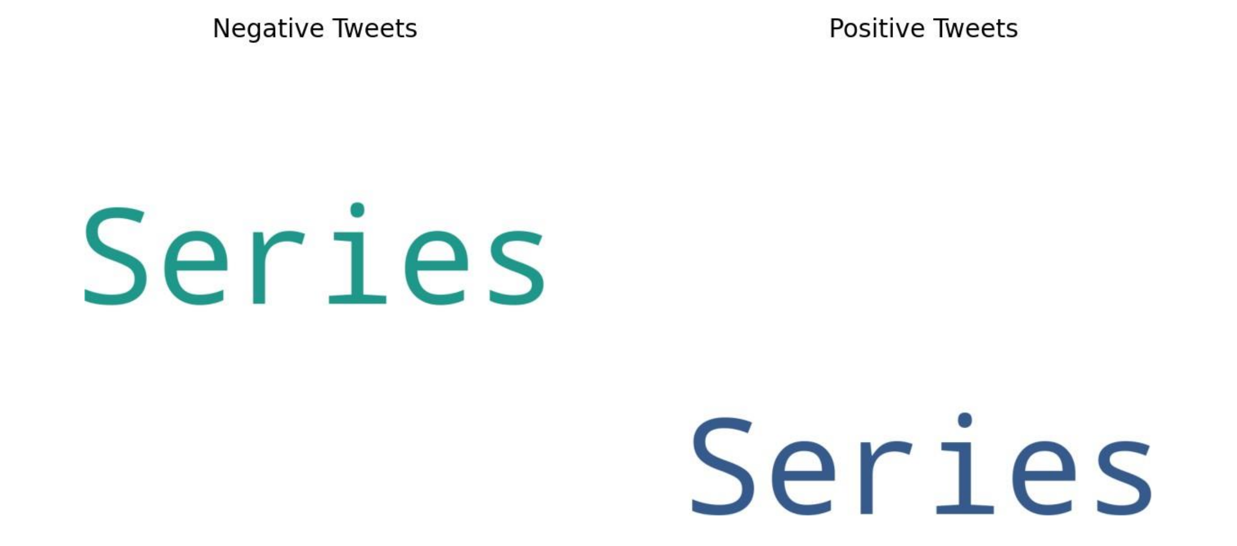
```
tweet_created                  object
tweet_location                 object
user_timezone                  object
dtype: object
```

[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()
```



[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()
```

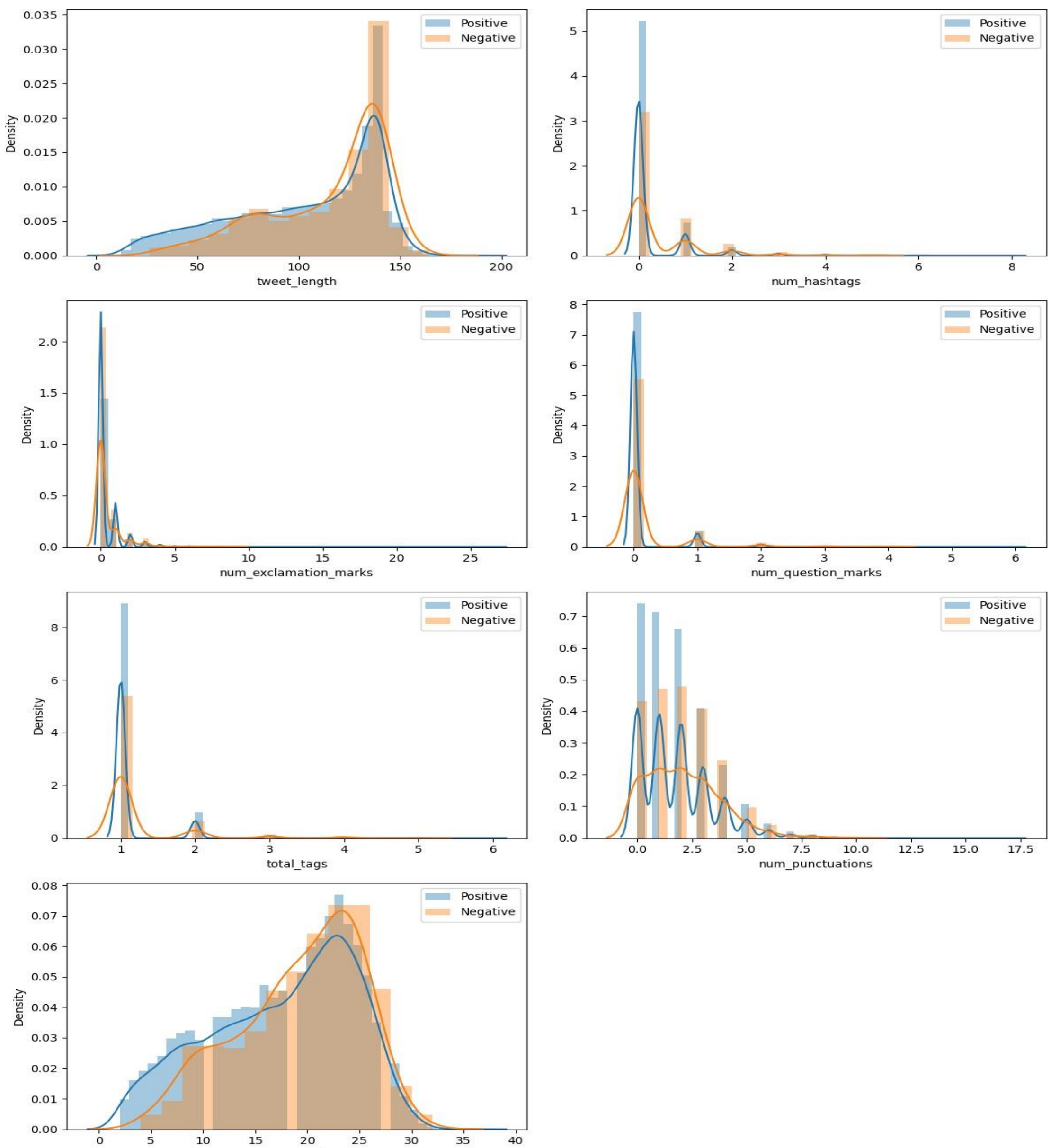
[8]:

	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
```

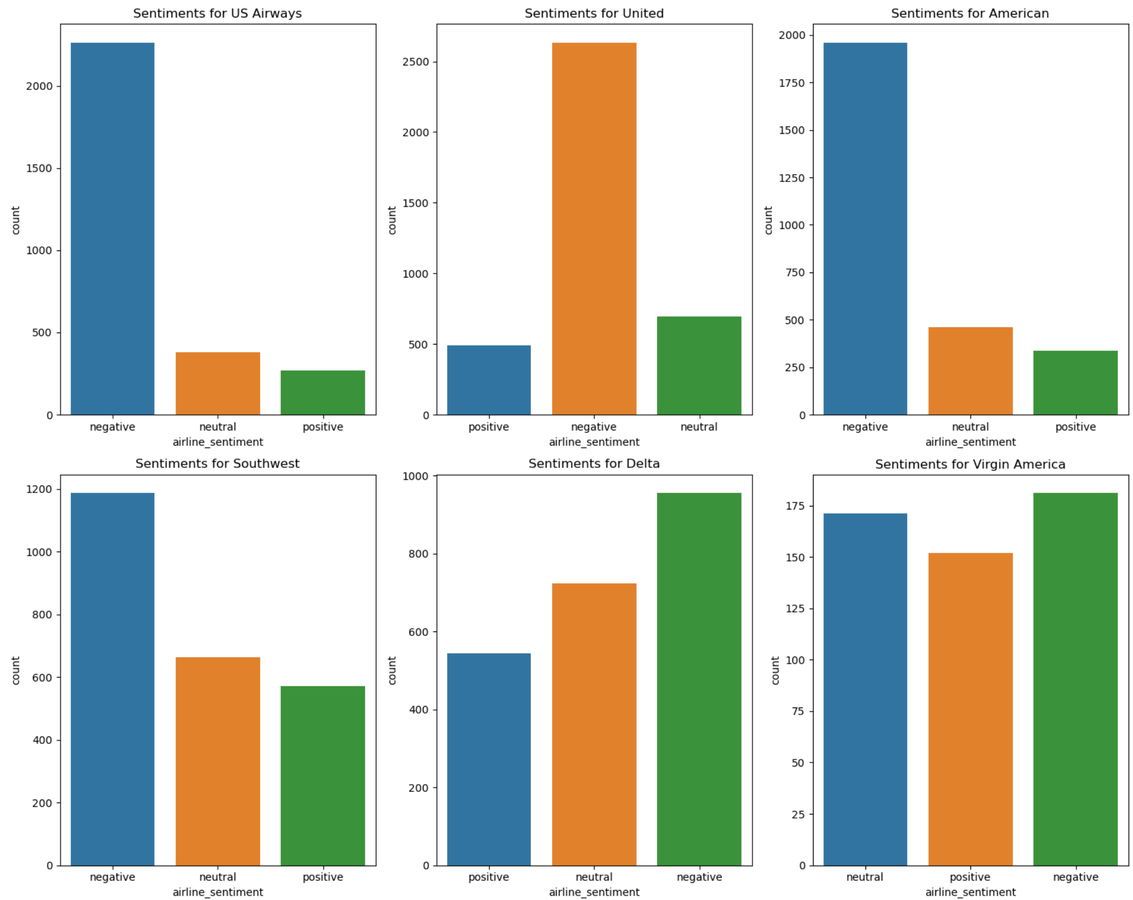
[12]:
```python
import nltk
nltk.download("punkt")


# 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)
```

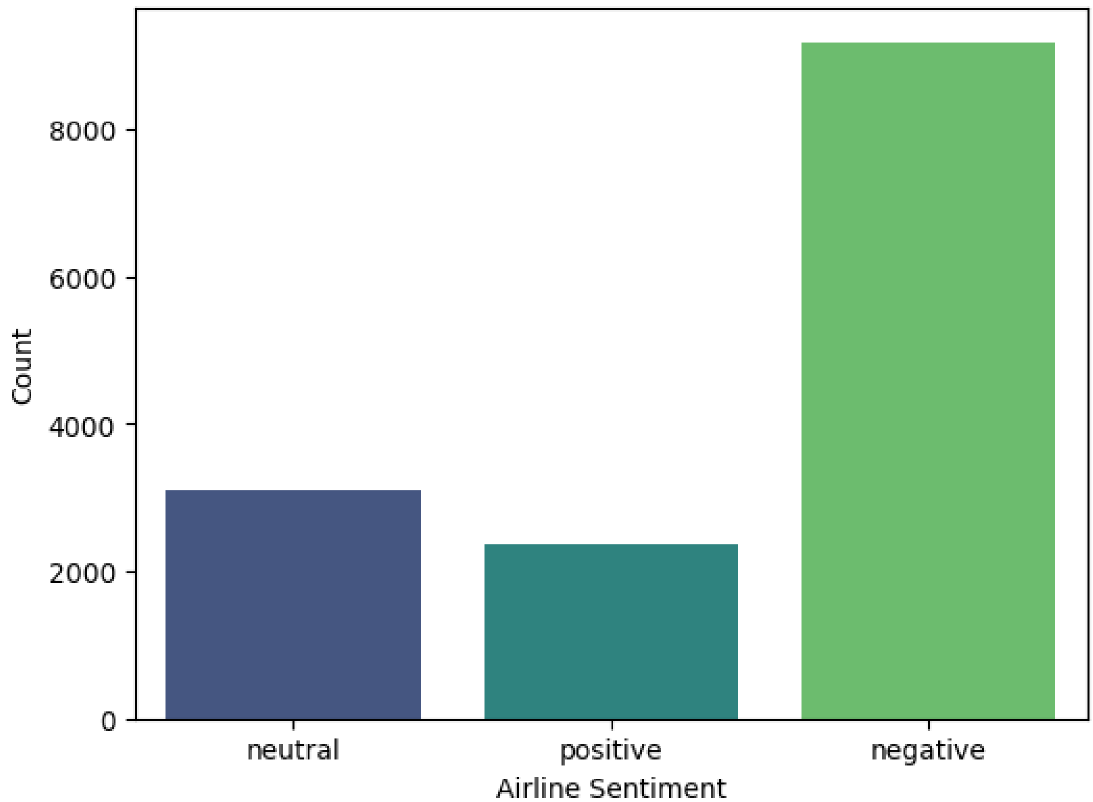
[13]:

```
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 Southwes
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]:
```python
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!
```

CONCLUSION:

- This project has demonstrated the potential of sentiment analysis to be used for a variety of marketing purposes, including:
  - Feature engineering
  - Model training
  - Evaluation

- Sentiment analysis is a powerful tool that can be used to improve marketing effectiveness and achieve better business outcomes.

- By understanding and measuring customer sentiment, businesses can make better decisions about how to develop, market, and sell their products and services.