```
import re
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report

# Download stopwords
nltk.download('stopwords')

# Read the dataset
tweets = pd.read_csv('Tweets.csv')

# Filter tweets with high confidence in sentiment
tweets_df = tweets[tweets['airline_sentiment_confidence'] >= 0.5]

# Define Navarasa emotions
navarasa_emotions = {
    'shringara': ['love', 'lovely', 'romantic'],
    'hasya': ['funny', 'hilarious', 'laugh'],
    'karuna': ['sad', 'sorrow', 'tear'],
    'raudra': ['angry', 'furious', 'rage'],
    'veera': ['brave', 'courage', 'hero'],
    'bhayanaka': ['fear', 'scared', 'terrified'],
    'bibhatsa': ['disgust', 'disgusting', 'gross'],
    'adbhuta': ['amazing', 'awe', 'wonder'],
    'shanta': ['calm', 'peaceful', 'serene']
}

# Preprocessing function
def preprocess_text(text):
    # Convert to lowercase and remove non-alphabetic characters
    text = re.sub('[^a-zA-Z]', ' ', text.lower())
    # Tokenize and remove stopwords
    stop_words = set(stopwords.words('english'))
    stemmer = PorterStemmer()
    text = [stemmer.stem(word) for word in text.split() if word not in stop_words]
    return ' '.join(text)

# Apply preprocessing
tweets_df['cleaned_text'] = tweets_df['text'].apply(preprocess_text)

# Define Navarasa sentiment mapping function
def map_navarasa_sentiment(text):
    for emotion, keywords in navarasa_emotions.items():
        if any(keyword in text for keyword in keywords):
            return emotion
    return 'other'

# Apply Navarasa sentiment mapping
```

```
tweets_df['navarasa_sentiment'] = tweets_df['cleaned_text'].apply(map_navarasa_sentiment

# Drop rows with 'other' in Navarasa sentiment column
tweets_df = tweets_df[tweets_df['navarasa_sentiment'] != 'other']

# Feature extraction using Bag of Words
cv = CountVectorizer(max_features=3000)
X = cv.fit_transform(tweets_df['cleaned_text']).toarray()

# Target variable
y = tweets_df['navarasa_sentiment']

# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42

# Building the Multinomial Naive Bayes model
model = MultinomialNB()
model.fit(X_train, y_train)

# Predicting the Test set results
y_pred = model.predict(X_test)

# Evaluating the model
print(classification_report(y_test, y_pred))

# Check the distribution of sentiments
print(tweets_df['navarasa_sentiment'].value_counts())
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
<ipython-input-4-303ad0e05c2d>:44: 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/u
  tweets_df['cleaned_text'] = tweets_df['text'].apply(preprocess_text)
<ipython-input-4-303ad0e05c2d>:54: 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/u
  tweets_df['navarasa_sentiment'] = tweets_df['cleaned_text'].apply(map_navarasa_sen
              precision    recall  f1-score   support

    adbhuta       0.77      0.91      0.83        58
  bhayanaka       0.00      0.00      0.00         2
   bibhatsa       1.00      0.30      0.46        10
      hasya       0.00      0.00      0.00         1
     karuna       0.90      0.56      0.69        16
     raudra       0.00      0.00      0.00         2
  shringara       0.90      0.98      0.94       100
      veera       1.00      0.33      0.50         3

   accuracy                           0.85       192
  macro avg       0.57      0.39      0.43       192
weighted avg      0.84      0.85      0.83       192
```

```
navarasa_sentiment
shringara    323
adbhuta      191
karuna        56
bibhatsa      33
veera         12
raudra        12
hasya          7
bhayanaka      5
shanta         1
Name: count, dtype: int64
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: Und
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: Und
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: Und
  _warn_prf(average, modifier, msg_start, len(result))
```

Start coding or generate with AI.