Step1: Install pandas and numpy

```
import pandas as pd
import numpy as np
```

step2: CountVectorizer is a feature extraction technique used in natural language processing (NLP) and text mining to convert a collection of text documents into a numerical feature matrix. It is a part of the scikit-learn library in Python and is used to transform a set of text data to numerical feature vectors. Each feature represents the frequency of a word in the given text.

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
```

step3: WordNetLemmatizer is a part of the NLTK (Natural Language Toolkit) library in Python. It is used for lemmatizing words, which means reducing a word to its base or root form.

```
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import nltk
```

step4: Upload dataset file(twitter airlines US)

```
from google.colab import files
data=files.upload()
```

Choose Files Tweets.csv

- **Tweets.csv**(text/csv) - 3421431 bytes, last modified: 10/16/2019 - 100% done

```
Saving Tweets.csv to Tweets.csv
```

step5: In NLTK (Natural Language Toolkit) and other NLP libraries in Python, stopwords can be easily accessed and used. Here's how you can use stopwords using NLTK:

```
nltk.download('stopwords')
nltk.download('wordnet')
```

step6: Load the dataset

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

step7: Display the frames

```
# Display the first 5 rows of the dataframe
df.head()
```

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

step8: Install nltk (natural processing language)

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

step9: To drop unnecessary columns from a DataFrame in Python, you can use the drop() method provided by pandas, a powerful data manipulation library. Here's how you can do it

```
# Drop unnecessary columns
df = df[['airline_sentiment', 'text']]

# Display the first 5 rows of the dataframe after dropping unnecessary columns
df.head(125)
```

	airline_sentiment	text
0	neutral	virginamerica what dhepburn said
1	positive	virginamerica plus you ve added commercials t...
2	neutral	virginamerica didn today must mean need to ta...
3	negative	virginamerica it really aggressive to blast o...
4	negative	virginamerica and it a really big bad thing a...
...	...	...
120	negative	virginamerica use another browser amp brand w...
121	negative	virginamerica and now the flight flight booki...
122	negative	virginamerica like the customer service but m...
123	positive	virginamerica thanks to your outstanding nyc ...
124	positive	virginamerica you have the absolute best team...

125 rows × 2 columns

```
step10:
The preprocess_tweet function performs
various preprocessing steps on each
tweet.
df['text'] refers to the column
containing the original tweets, and df
['preprocessed_text'] is the column
where preprocessed tweets will be
stored.
The apply function applies the
preprocess_tweet function to each row
of the 'text' column, creating a new
'preprocessed_text' column in the
DataFrame.
```

step10: The preprocess_tweet function performs various preprocessing steps on each tweet. df['text'] refers to the column containing the original tweets, and df['preprocessed_text'] is the column where preprocessed tweets will be stored. The apply function applies the preprocess_tweet function to each row of the 'text' column, creating a new 'preprocessed_text' column in the DataFrame.

```
# function to preprocess the text
def preprocess_text(text):
    # Remove punctuations and numbers
```

```
    text = re.sub('[^a-zA-Z]', ' ', text)

    # Single character removal
    text = re.sub(r'\s+[a-zA-Z]\s+', ' ', text)

    # Removing multiple spaces
    text = re.sub(r'\s+', ' ', text)

    # Converting to Lowercase
    text = text.lower()

    # Lemmatization
    #text = text.split()
    #lemmatizer = WordNetLemmatizer()
    #text = [lemmatizer.lemmatize(word) for word in text if not word in set(stopw
    #text = ' '.join(text)

    return text
```

finally, after preprocessing the dataset,

```
# Apply the preprocessing to the 'text' column
df['text'] = df['text'].apply(preprocess_text)

# Display the first 5 rows of the dataframe after preprocessing
df.head()
```

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