```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px

from sklearn.naive_bayes import ComplementNB
from sklearn.metrics import confusion_matrix
from wordcloud import WordCloud
import nltk
import re
import string
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer
from sklearn.feature_extraction.text import TfidfVectorizer
from collections import Counter

from nltk.stem import PorterStemmer


stop_words = stopwords.words()
```

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

```
with open('/content/drive/MyDrive/ColabNotebooks/airline.csv', encoding='utf-8', errors='ignore') as file:
    data = pd.read_csv(file)
data.head(20)
```

	_unit_id	_golden	_unit_state	_trusted_judgments	_last_judgment_at	airline_sentiment	airline_sentiment:confidence	negativereason	neg
0	681448150	False	finalized	3	2/25/15 5:24	neutral	1.0000	NaN	
1	681448153	False	finalized	3	2/25/15 1:53	positive	0.3486	NaN	
2	681448156	False	finalized	3	2/25/15 10:01	neutral	0.6837	NaN	
3	681448158	False	finalized	3	2/25/15 3:05	negative	1.0000	Bad Flight	
4	681448159	False	finalized	3	2/25/15 5:50	negative	1.0000	Can't Tell	
5	681448162	False	finalized	3	2/25/15 9:10	negative	1.0000	Can't Tell	
6	681448165	False	finalized	3	2/25/15 8:11	positive	0.6745	NaN	
7	681448167	False	finalized	3	2/25/15 2:11	neutral	0.6340	NaN	
8	681448169	False	finalized	3	2/25/15 9:01	positive	0.6559	NaN	
9	681448171	False	finalized	3	2/25/15 4:15	positive	1.0000	NaN	
10	681448174	False	finalized	3	2/25/15 8:34	neutral	0.6769	NaN	
11	681448176	False	finalized	3	2/25/15 2:03	positive	1.0000	NaN	
12	681448178	False	finalized	3	2/25/15 2:13	positive	1.0000	NaN	
13	681448181	False	finalized	3	2/25/15 5:39	positive	0.6451	NaN	

```
stemmer = PorterStemmer()
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    # Remove URLs and non-alphanumeric characters
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = text.lower()
    tokens = word_tokenize(text)
    # Remove stop words and stem the words
    tokens = [stemmer.stem(word) for word in tokens if word not in stop_words]
    return ' '.join(tokens)

# Apply preprocessing
data['text'] = data['text'].apply(preprocess_text)
```

```
data.rename(columns={'airline_sentiment':'sentiment'}, inplace = True)
data
```

	_unit_id	_golden	_unit_state	_trusted_judgments	_last_judgment_at	sentiment	airline_sentiment:confidence	negativereason	negativerea
0	681448150	False	finalized	3	2/25/15 5:24	neutral	1.0000	NaN	
1	681448153	False	finalized	3	2/25/15 1:53	positive	0.3486	NaN	
2	681448156	False	finalized	3	2/25/15 10:01	neutral	0.6837	NaN	
3	681448158	False	finalized	3	2/25/15 3:05	negative	1.0000	Bad Flight	
4	681448159	False	finalized	3	2/25/15 5:50	negative	1.0000	Can't Tell	
...	...	...	...	...	...	...	...	...	
14635	681679794	False	finalized	3	2/25/15 19:46	positive	0.3487	NaN	
14636	681679795	False	finalized	3	2/25/15 19:14	negative	1.0000	Customer Service Issue	
14637	681679796	False	finalized	3	2/25/15 19:04	neutral	1.0000	NaN	
14638	681679797	False	finalized	3	2/25/15 18:59	negative	1.0000	Customer Service Issue	
14639	681679798	False	finalized	3	2/25/15 19:06	neutral	0.6771	NaN	

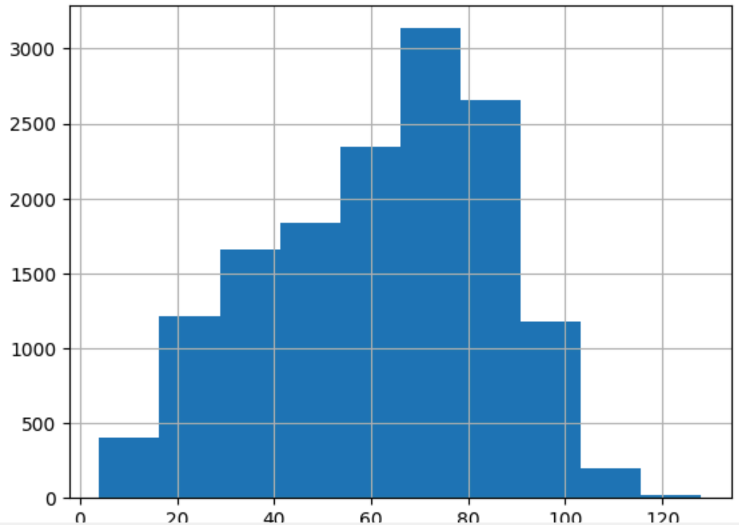
14640 rows × 20 columns

```
data['sentiment'].value_counts()
```

	count
sentiment	
negative	9178
neutral	3099
positive	2363

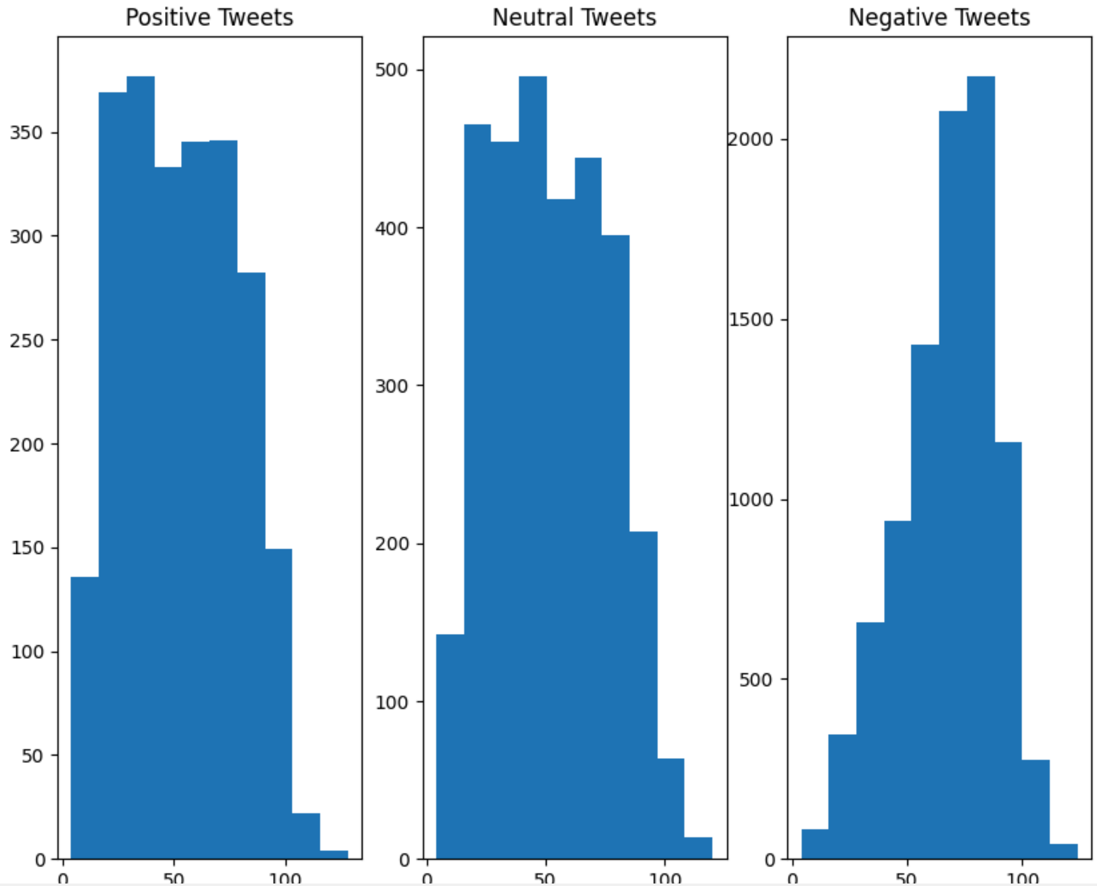
```
data['text'].str.len().hist()
```

```
<Axes: >
```



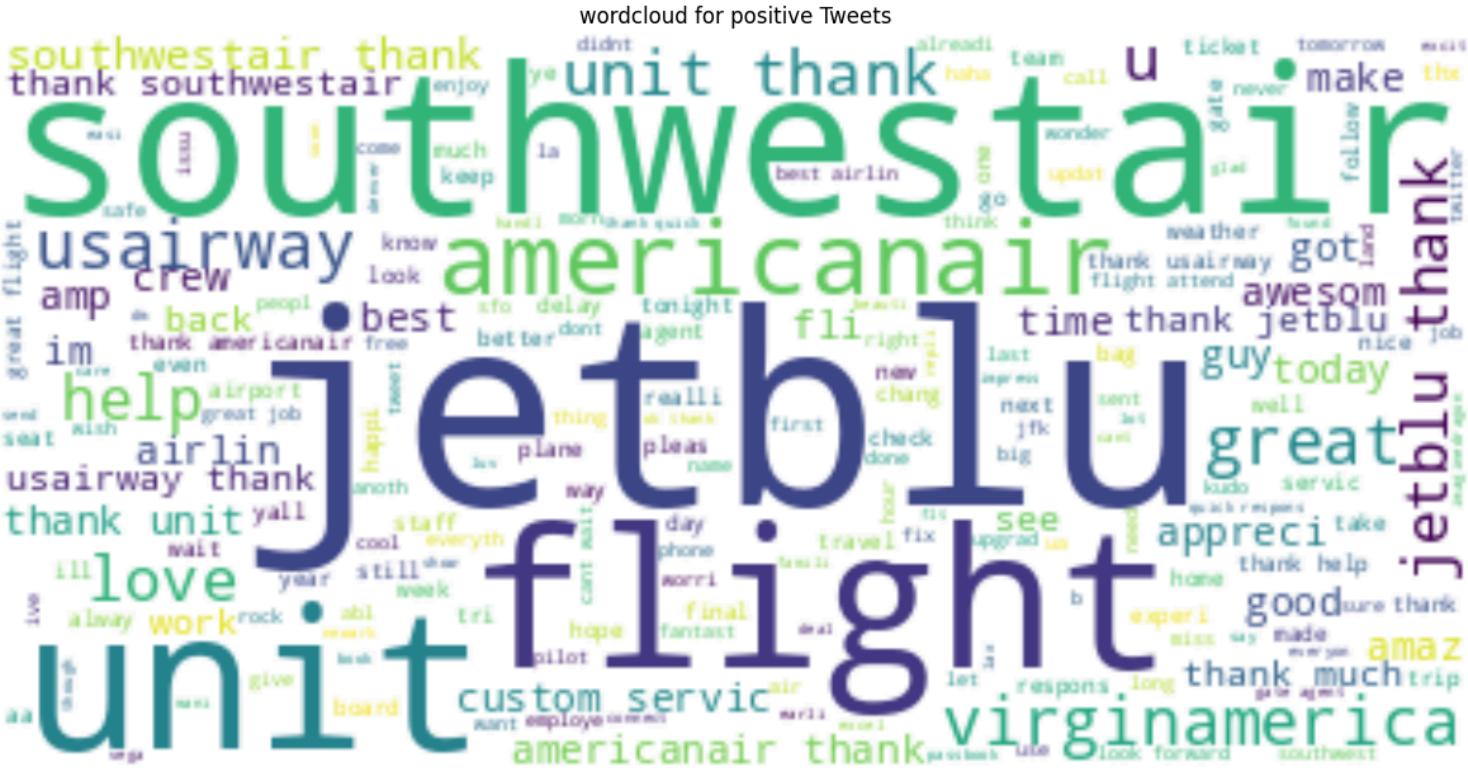
```
fig,(ax1,ax2,ax3)=plt.subplots(1,3,figsize=(10,8))
ax1.hist(data[data['sentiment']=='positive']['text'].str.len())
ax1.set_title( 'Positive Tweets')
ax2.hist(data[data['sentiment']=='neutral']['text'].str.len())
ax2.set_title( 'Neutral Tweets')
ax3.hist(data[data['sentiment']=='negative']['text'].str.len())
ax3.set_title( 'Negative Tweets')
```

```
Text(0.5, 1.0, 'Negative Tweets')
```

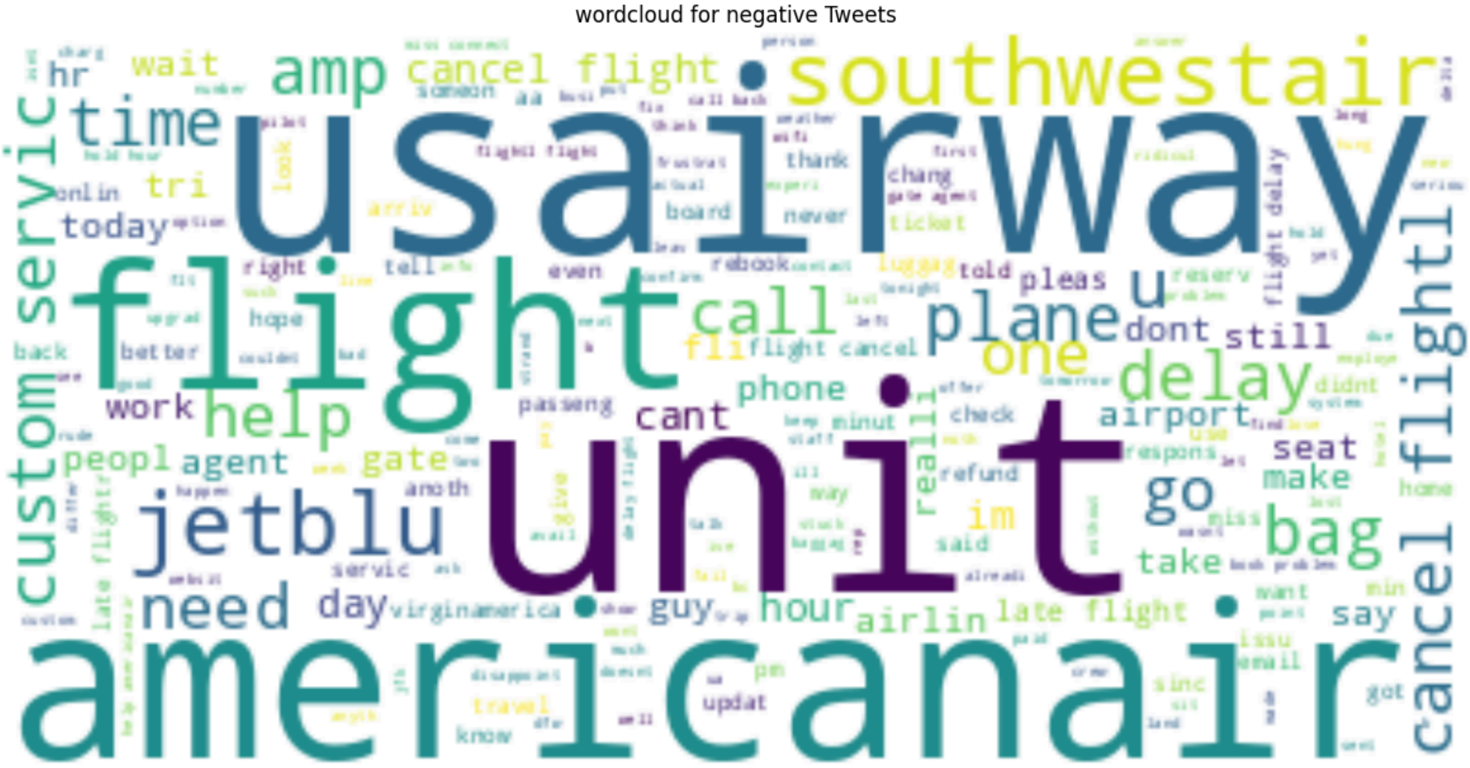


```
text = " ".join(i for i in data[data['sentiment']=='positive']['text'])
wordcloud = WordCloud( background_color="white").generate(text)

plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('wordcloud for positive Tweets')
plt.show()
```

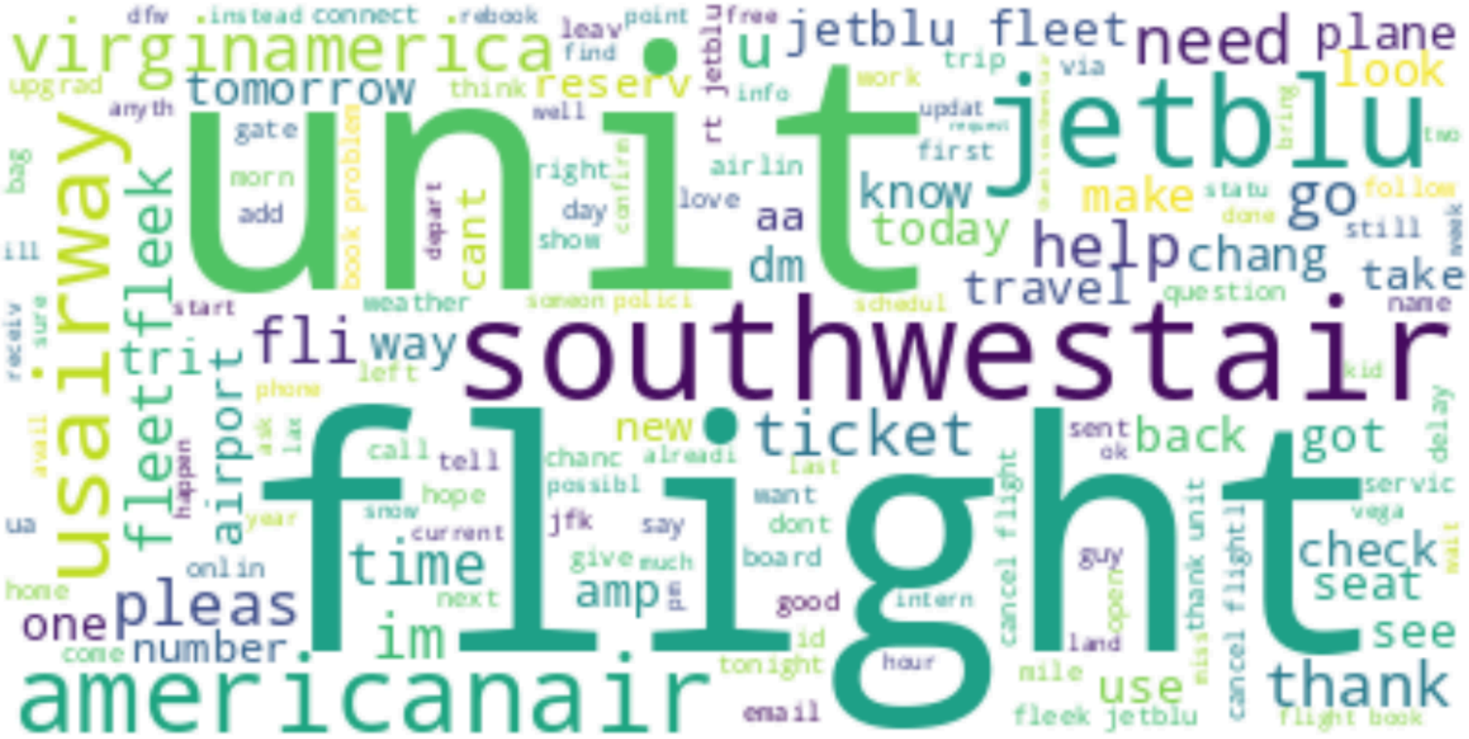


```
text = " ".join(i for i in data[data['sentiment']=='negative']['text'])
#stopwords = set(STOPWORDS)
wordcloud = WordCloud( background_color="white").generate(text)
#wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('wordcloud for negative Tweets')
plt.show()
```



```
text = " ".join(i for i in data[data['sentiment']=='neutral']['text'])
#stopwords = set(STOPWORDS)about:blank#blocked
wordcloud = WordCloud( background_color="white").generate(text)
#wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('wordcloud for neutral Tweets')
plt.show()
```

wordcloud for neutral Tweets



```
dt = data['text']
dt = pd.DataFrame(dt)
dt['sentiment']=data['sentiment']
dt
```

	text	sentiment
0	virginamerica dhepburn said	neutral
1	virginamerica plu youv ad commerci experi tacki	positive
2	virginamerica didnt today must mean need take ...	neutral
3	virginamerica realli aggress blast obnoxi ente...	negative
4	virginamerica realli big bad thing	negative
...	...	...
14635	americanair thank got differ flight chicago	positive
14636	americanair leav minut late flight warn commun...	negative
14637	americanair pleas bring american airlin blackb...	neutral
14638	americanair money chang flight dont answer pho...	negative
14639	americanair ppl need know mani seat next fligh...	neutral

14640 rows × 2 columns

```
dt['no_sw'] = dt['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop_words)]))
```

```
dt
```

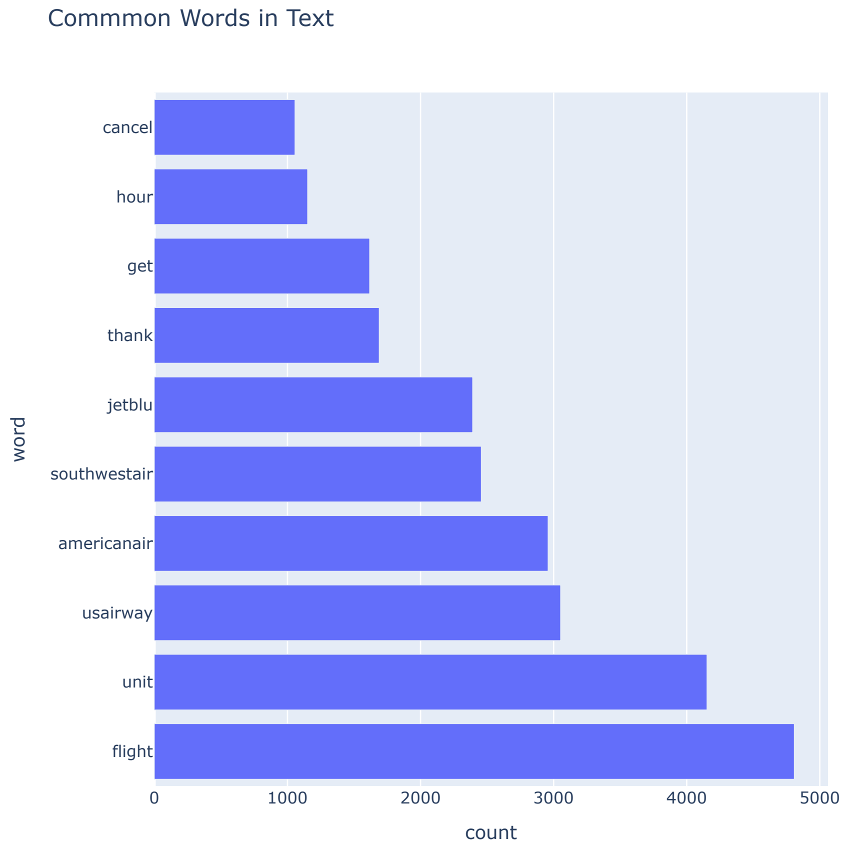
	text	sentiment	no_sw
0	virginamerica dhepburn said	neutral	virginamerica dhepburn said
1	virginamerica plu youv ad commerci experi tacki	positive	virginamerica plu youv ad commerci experi tacki
2	virginamerica didnt today must mean need take ...	neutral	virginamerica didnt today must mean need take ...
3	virginamerica realli aggress blast obnoxi ente...	negative	virginamerica realli aggress blast obnoxi ente...
4	virginamerica realli big bad thing	negative	virginamerica realli big bad thing
...	...	...	...
14635	americanair thank got differ flight chicago	positive	americanair thank got differ flight chicago
14636	americanair leav minut late flight warn commun...	negative	americanair leav minut late flight warn commun...
14637	americanair pleas bring american airlin blackb...	neutral	americanair pleas bring american airlin blackb...
14638	americanair money chang flight dont answer pho...	negative	americanair money chang flight dont answer pho...
14639	americanair ppl need know mani seat next fligh...	neutral	americanair ppl need know mani seat next fligh...

14640 rows × 3 columns

```
cnt = Counter()
for text in dt["no_sw"].values:
    for word in text.split():
        cnt[word] += 1
cnt.most_common(10)
temp = pd.DataFrame(cnt.most_common(10))
temp.columns=['word', 'count']
temp
```

	word	count
0	flight	4808
1	unit	4152
2	usairway	3052
3	americanair	2958
4	southwestair	2456
5	jetblu	2391
6	thank	1689
7	get	1617
8	hour	1151
9	cancel	1056

```
px.bar(temp, x="count", y="word", title='Commmon Words in Text', orientation='h',
             width=700, height=700)
```



```
FREQWORDS = set([w for (w, wc) in cnt.most_common(10)])
def remove_freqwords(text):
    """custom function to remove the frequent words"""
    return " ".join([word for word in str(text).split() if word not in FREQWORDS])
dt["wo_stopfreq"] = dt["no_sw"].apply(lambda text: remove_freqwords(text))
dt.head()
```

	text	sentiment	no_sw	wo_stopfreq
0	virginamerica dhepburn said	neutral	virginamerica dhepburn said	virginamerica dhepburn said
1	virginamerica plu youv ad commerci experi tacki	positive	virginamerica plu youv ad commerci experi tacki	virginamerica plu youv ad commerci experi tacki
2	virginamerica didnt today must mean need take ...	neutral	virginamerica didnt today must mean need take ...	virginamerica didnt today must mean need take ...
3	virginamerica realli aggress blast obnoxi ente...	negative	virginamerica realli aggress blast obnoxi ente...	virginamerica realli aggress blast obnoxi ente...
4	virginamerica realli big bad thing	negative	virginamerica realli big bad thing	virginamerica realli big bad thing

```
#TF-IDF vectors
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(data['text'])
y = data['sentiment']


import nltk
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')

wordnet_lem = WordNetLemmatizer()

dt['wo_stopfreq_lem'] = dt['wo_stopfreq'].apply(wordnet_lem.lemmatize)
dt
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

	text	sentiment	no_sw	wo_stopfreq	wo_stopfreq_lem
0	virginamerica dhepburn said	neutral	virginamerica dhepburn said	virginamerica dhepburn said	virginamerica dhepburn said
1	virginamerica plu youv ad commerci experi tacki	positive	virginamerica plu youv ad commerci experi tacki	virginamerica plu youv ad commerci experi tacki	virginamerica plu youv ad commerci experi tacki
2	virginamerica didnt today must mean need take ...	neutral	virginamerica didnt today must mean need take ...	virginamerica didnt today must mean need take ...	virginamerica didnt today must mean need take ...
3	virginamerica realli aggress blast obnoxi ente...	negative	virginamerica realli aggress blast obnoxi ente...	virginamerica realli aggress blast obnoxi ente...	virginamerica realli aggress blast obnoxi ente...
4	virginamerica realli big bad thing	negative	virginamerica realli big bad thing	virginamerica realli big bad thing	virginamerica realli big bad thing
...	...	...	...	...	...
14635	americanair thank got differ flight chicago	positive	americanair thank got differ flight chicago	got differ chicago	got differ chicago
14636	americanair leav minut late flight warn commun...	negative	americanair leav minut late flight warn commun...	leav minut late warn commun minut late call sh...	leav minut late warn commun minut late call sh...
14637	americanair pleas bring american airlin blackb...	neutral	americanair pleas bring american airlin blackb...	pleas bring american airlin blackberri	pleas bring american airlin blackberri
14638	americanair money chang flight dont answer pho...	negative	americanair money chang flight dont answer pho...	money chang dont answer phone suggest make commit	money chang dont answer phone suggest make commit

```
mapping = {
    'negative': 0,
    'neutral': 1,
    'positive': 2
}
```

```
nb=dt.drop(columns=['text','no_sw', 'wo_stopfreq'])
nb.columns=['sentiment','text']

# Apply the mapping to the 'sentiment' column
nb['sentiment'] = nb['sentiment'].map(mapping)
```

```
nb
```

	sentiment	text
0	1	virginamerica dhepburn said
1	2	virginamerica plu youv ad commerci experi tacki
2	1	virginamerica didnt today must mean need take ...
3	0	virginamerica realli aggress blast obnoxi ente...
4	0	virginamerica realli big bad thing
...	...	...
14635	2	got differ chicago
14636	0	leav minut late warn commun minut late call sh...
14637	1	pleas bring american airlin blackberri
14638	0	money chang dont answer phone suggest make commit
14639	1	ppl need know mani seat next plz put us standb...

14640 rows × 2 columns

```
tokenized_review=nb['text'].apply(lambda x: x.split())
tokenized_review.head(5)
```

	text
0	[virginamerica, dhepburn, said]