

```
In [1]: pip install textblob
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: textblob in c:\users\vinay\appdata\roaming\python\python39\site-packages (0.18.0.post0)
Requirement already satisfied: nltk>=3.8 in c:\users\vinay\appdata\roaming\python\python39\site-packages (from textblob) (3.9.1)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (8.0.4)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\programdata\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (2022.7.9)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (from nltk>=3.8->textblob) (4.64.1)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from click->nltk>=3.8->textblob) (0.4.5)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.0 -> 24.3.1
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [2]: pip install wordcloud
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: wordcloud in c:\users\vinay\appdata\roaming\python\python39\site-packages (1.9.4)
Requirement already satisfied: numpy>=1.6.1 in c:\users\vinay\appdata\roaming\python\python39\site-packages (from wordcloud) (1.24.4)
Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-packages (from wordcloud) (9.2.0)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-packages (from wordcloud) (3.5.2)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.2)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.0 -> 24.3.1
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [3]: import pandas as pd
```

```
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
from textblob import TextBlob
```

```
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
stop_words=set(stopwords.words('english'))
from wordcloud import WordCloud
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, C
```

In [4]: df=pd.read_csv('C:/Users/VINAY/Desktop/vaccine/vaccination_tweets.csv')

In [5]: df.head()

0	1340539111971516416	Rachel Roh	La Crescenta-Montrose, CA	Aggregator of Asian American news; scanning di...	2009-04-08 17:52:46	405	
1	1338158543359250433	Albert Fong	San Francisco, CA	Marketing dude, tech geek, heavy metal & '80s ...	2009-09-21 15:27:30	834	
2	1337858199140118533	eliLTEU 🌻	Your Bed	heil, hydra 🙌 ☺	2020-06-25 23:30:28	10	
3	1337855739918835717	Charles Adler	Vancouver, BC - Canada	Hosting "CharlesAdlerTonight" Global News Radi...	2008-09-10 11:28:53	49165	
4	1337854064604966912	Citizen News Channel	NaN	Citizen News Channel bringing you an alternati...	2020-04-23 17:58:42	152	

In [6]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11020 entries, 0 to 11019
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               11020 non-null   int64  
 1   user_name        11020 non-null   object  
 2   user_location    8750 non-null   object  
 3   user_description 10341 non-null   object  
 4   user_created     11020 non-null   object  
 5   user_followers   11020 non-null   int64  
 6   user_friends     11020 non-null   int64  
 7   user_favourites  11020 non-null   int64  
 8   user_verified    11020 non-null   bool   
 9   date             11020 non-null   object  
 10  text              11020 non-null   object  
 11  hashtags          8438 non-null   object  
 12  source            11019 non-null   object  
 13  retweets          11020 non-null   int64  
 14  favorites         11020 non-null   int64  
 15  is_retweet       11020 non-null   bool  
dtypes: bool(2), int64(6), object(8)
memory usage: 1.2+ MB
```

```
In [7]: df.isnull().sum()
```

```
Out[7]: id                0
user_name        0
user_location    2270
user_description 679
user_created     0
user_followers   0
user_friends     0
user_favourites  0
user_verified    0
date              0
text              0
hashtags         2582
source            1
retweets          0
favorites         0
is_retweet       0
dtype: int64
```

```
In [8]: df.columns
```

```
Out[8]: Index(['id', 'user_name', 'user_location', 'user_description', 'user_created',
               'user_followers', 'user_friends', 'user_favourites', 'user_verified',
               'date', 'text', 'hashtags', 'source', 'retweets', 'favorites',
               'is_retweet'],
              dtype='object')
```

```
In [9]: text_df=df.drop(['id', 'user_name', 'user_location', 'user_description', 'user_created',
                       'user_followers', 'user_friends', 'user_favourites', 'user_verified',
                       'date', 'hashtags', 'source', 'retweets', 'favorites',
                       'is_retweet'],axis=1)
```

```
text_df.head()
```

Out[9]:

	text
0	Same folks said daikon paste could treat a cyt...
1	While the world has been on the wrong side of ...
2	#coronavirus #SputnikV #AstraZeneca #PfizerBio...
3	Facts are immutable, Senator, even when you're...
4	Explain to me again why we need a vaccine @Bor...

In [10]:

```
print(text_df['text'].iloc[0], "\n")
print(text_df['text'].iloc[1], "\n")
print(text_df['text'].iloc[2], "\n")
print(text_df['text'].iloc[3], "\n")
print(text_df['text'].iloc[4], "\n")
```

Same folks said daikon paste could treat a cytokine storm #PfizerBioNTech https://t.co/xeHhIMg1kF

While the world has been on the wrong side of history this year, hopefully, the biggest vaccination effort we've ev... https://t.co/dlCHrZjkhm

#coronavirus #SputnikV #AstraZeneca #PfizerBioNTech #Moderna #Covid_19 Russian vaccine is created to last 2-4 years... https://t.co/ieYlCKBr8P

Facts are immutable, Senator, even when you're not ethically sturdy enough to acknowledge them. (1) You were born i... https://t.co/jqgV18kch4

Explain to me again why we need a vaccine @BorisJohnson @MattHancock #whereareallthesickpeople #PfizerBioNTech... https://t.co/KxbSRoBEHQ

In [11]:

```
text_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11020 entries, 0 to 11019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  -- 
 0   text    11020 non-null  object 
dtypes: object(1)
memory usage: 86.2+ KB
```

In [12]:

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\VINAY\AppData\Roaming\nltk_data...
[nltk_data]     Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\VINAY\AppData\Roaming\nltk_data...
[nltk_data]     Package stopwords is already up-to-date!
```

Out[12]:

True

In [13]:

```
import nltk
print(nltk.data.path)
```

```
[ 'C:\\\\Users\\\\VINAY\\\\nltk_data', 'C:\\\\ProgramData\\\\Anaconda3\\\\nltk_data', 'C:\\\\ProgramD  
ata\\\\Anaconda3\\\\share\\\\nltk_data', 'C:\\\\ProgramData\\\\Anaconda3\\\\lib\\\\nltk_data',  
'C:\\\\Users\\\\VINAY\\\\AppData\\\\Roaming\\\\nltk_data', 'C:\\\\nltk_data', 'D:\\\\nltk_data',  
'E:\\\\nltk_data' ]
```

```
In [14]: nltk.data.path.append('C:\\\\nltk_data')  
nltk.data.path.append('D:\\\\nltk_data')
```

```
In [15]: import re  
from nltk.tokenize import word_tokenize  
from nltk.corpus import stopwords  
import nltk  
  
# Download necessary NLTK resources  
nltk.download('punkt_tab')  
nltk.download('stopwords')  
  
# Define stopwords  
stop_words = set(stopwords.words('english'))  
  
# Define the function  
def data_processing(text):  
    text = text.lower()  
    text = re.sub(r"https\S+|www\S+https\S+", '', text, flags=re.MULTILINE)  
    text = re.sub(r'@\w+|\#', '', text)  
    text = re.sub(r'[^\\w\\s]', '', text)  
    text_tokens = word_tokenize(text)  
    filtered_text = [w for w in text_tokens if not w in stop_words]  
    return " ".join(filtered_text)  
  
# Apply the function to your DataFrame  
text_df['text'] = text_df['text'].apply(data_processing)
```

```
[nltk_data] Downloading package punkt_tab to  
[nltk_data]      C:\\Users\\VINAY\\AppData\\Roaming\\nltk_data...  
[nltk_data]      Package punkt_tab is already up-to-date!  
[nltk_data] Downloading package stopwords to  
[nltk_data]      C:\\Users\\VINAY\\AppData\\Roaming\\nltk_data...  
[nltk_data]      Package stopwords is already up-to-date!
```

```
In [16]: stemmer=PorterStemmer()  
def stemming(data):  
    text=[stemmer.stem(word) for word in data]  
    return data
```

```
In [17]: text_df['text']=text_df['text'].apply(lambda x: stemming(x))
```

```
In [18]: text_df.head()
```

```
Out[18]:
```

	text
0	folks said daikon paste could treat cytokine s...
1	world wrong side history year hopefully bigges...
2	coronavirus sputnikv astrazeneca pfizerbiontec...
3	facts immutable senator even youre ethically s...
4	explain need vaccine whereareallthesickpeople ...

```
In [19]:
```

```
print(text_df['text'].iloc[0], "\n")
print(text_df['text'].iloc[1], "\n")
print(text_df['text'].iloc[2], "\n")
print(text_df['text'].iloc[3], "\n")
print(text_df['text'].iloc[4], "\n")
```

```
folks said daikon paste could treat cytokine storm pfizerbiontech
world wrong side history year hopefully biggest vaccination effort weve ev
coronavirus sputnikv astrazeneca pfizerbiontech moderna covid_19 russian vaccine crea
ted last 24 years
facts immutable senator even youre ethically sturdy enough acknowledge 1 born
explain need vaccine whereareallthesickpeople pfizerbiontech
```

```
In [20]:
```

```
text_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11020 entries, 0 to 11019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  -- 
 0   text    11020 non-null   object 
dtypes: object(1)
memory usage: 86.2+ KB
```

```
In [21]:
```

```
def polarity(text):
    return TextBlob(text).sentiment.polarity
```

```
In [22]:
```

```
text_df['polarity']=text_df['text'].apply(polarity)
```

```
In [23]:
```

```
text_df.head(10)
```

Out[23]:

		text	polarity
0		folks said daikon paste could treat cytokine s...	0.000
1		world wrong side history year hopefully bigges...	-0.500
2		coronavirus sputnikv astrazeneca pfizerbiontec...	0.000
3		facts immutable senator even youre ethically s...	0.100
4		explain need vaccine whereareallthesickpeople ...	0.000
5		anyone useful adviceguidance whether covid vac...	0.400
6		bit sad claim fame success vaccination patriot...	-0.100
7		many bright days 2020 best 1 bidenharris winni...	0.675
8		covid vaccine getting covidvaccine covid19 pfi...	0.000
9		covidvaccine states start getting covid19vacci...	0.000

In [24]:

```
def sentiment(label):
    if label <0:
        return "Negative"
    elif label ==0:
        return "Neutral"
    elif label>0:
        return "Positive"
```

In [25]:

```
text_df['sentiment']=text_df['polarity'].apply(sentiment)
```

In [26]:

```
text_df.head()
```

Out[26]:

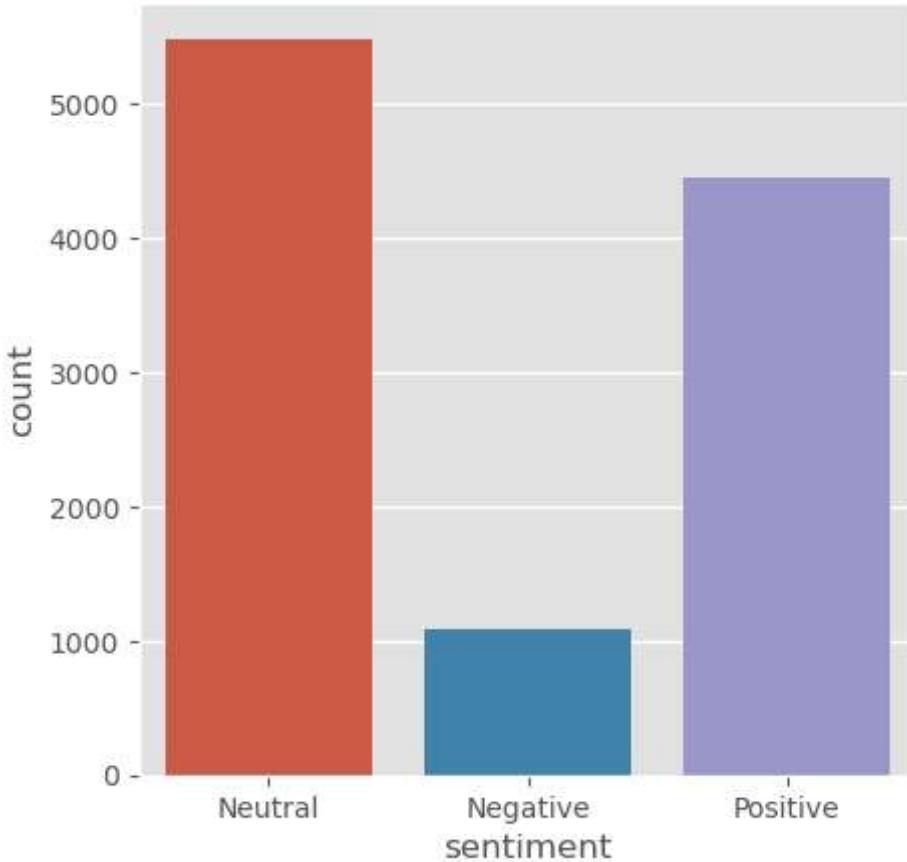
		text	polarity	sentiment
0		folks said daikon paste could treat cytokine s...	0.0	Neutral
1		world wrong side history year hopefully bigges...	-0.5	Negative
2		coronavirus sputnikv astrazeneca pfizerbiontec...	0.0	Neutral
3		facts immutable senator even youre ethically s...	0.1	Positive
4		explain need vaccine whereareallthesickpeople ...	0.0	Neutral

In [27]:

```
fig = plt.figure(figsize=(5,5))
sns.countplot(x='sentiment', data=text_df)
```

Out[27]:

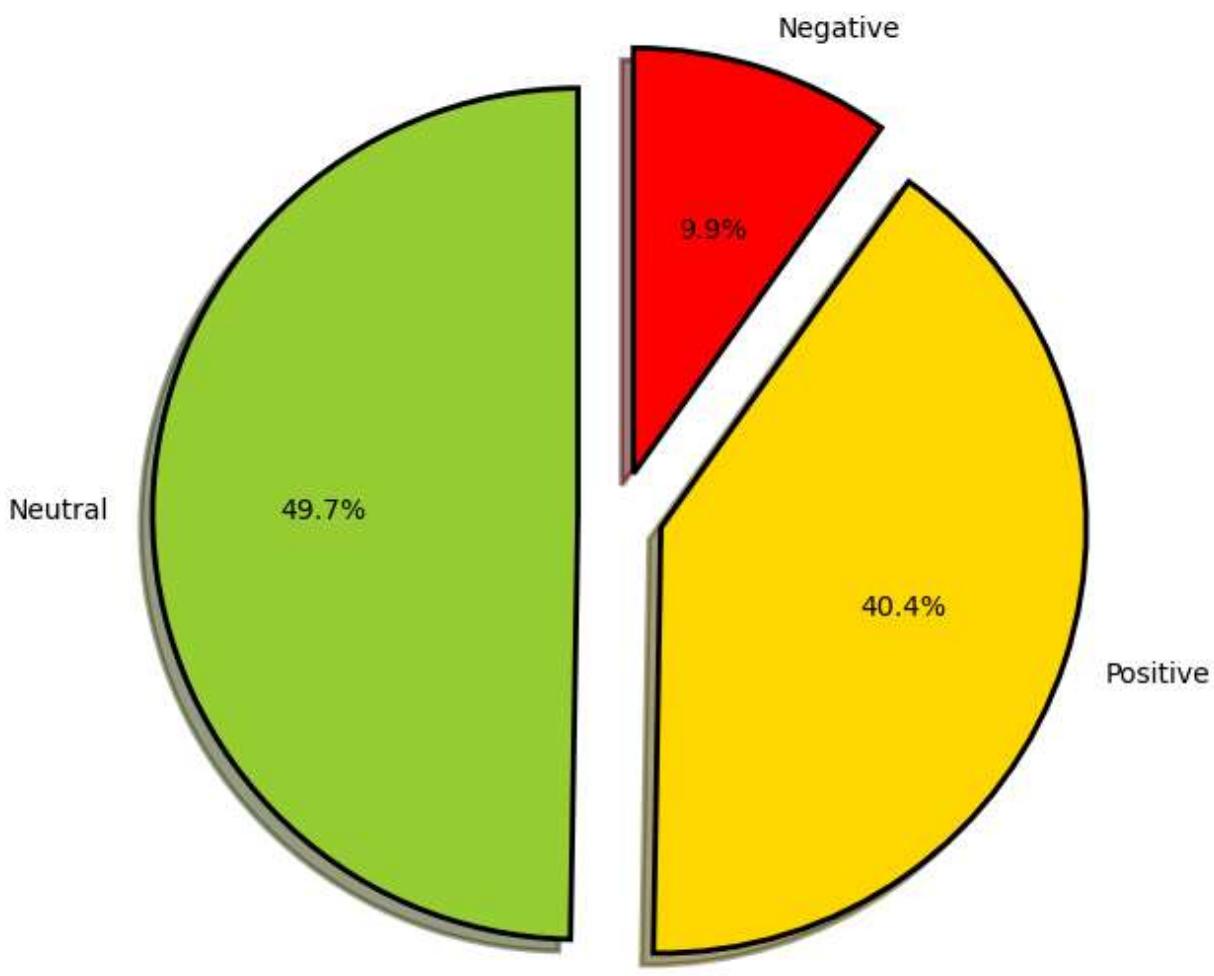
```
<AxesSubplot:xlabel='sentiment', ylabel='count'>
```



```
In [28]: fig=plt.figure(figsize=(7,7))
colors=("yellowgreen", "gold", "red")
wp= {'linewidth':2, 'edgecolor':'black'}
tags=text_df['sentiment'].value_counts()
explode= (0.1,0.1,0.1)
tags.plot(kind='pie', autopct='%1.1f%%', shadow=True, colors = colors, startangle=90,
plt.title('distribution of sentiments')
```

```
Out[28]: Text(0.5, 1.0, 'distribution of sentiments')
```

distribution of sentiments

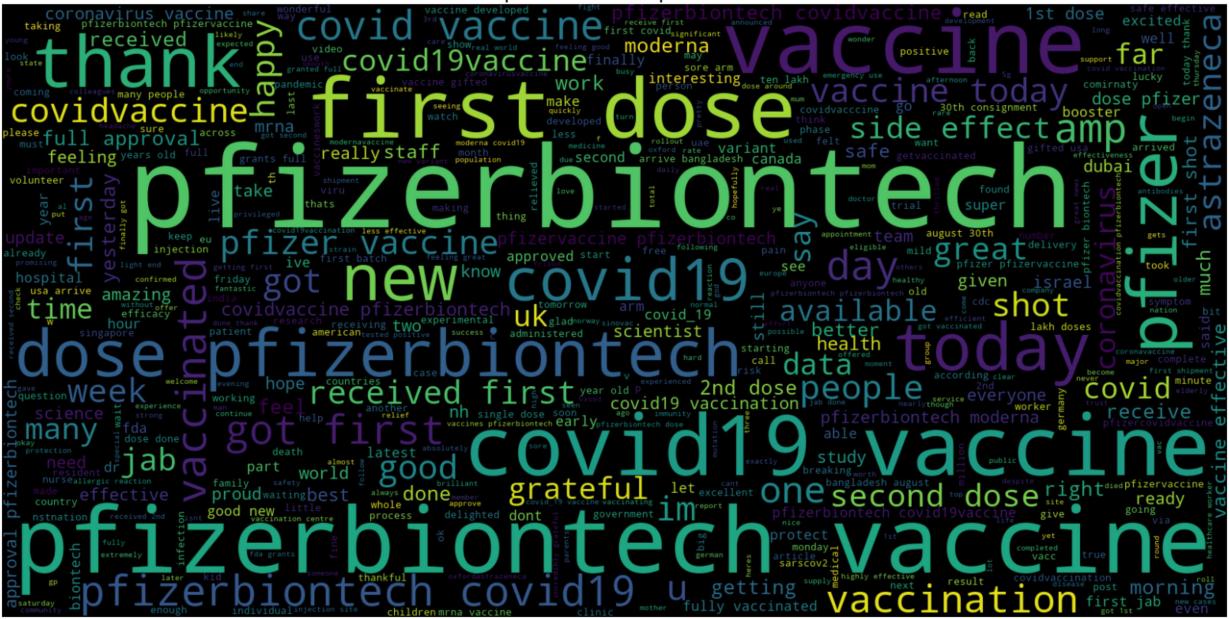


```
In [29]: pos_tweets=text_df[text_df.sentiment=='Positive']
pos_tweets=pos_tweets.sort_values(['polarity'], ascending=False)
pos_tweets.head()
```

Out[29]:

		text	polarity	sentiment
420		one downone go excellent vaccination service t...	1.0	Positive
9317		best way get merrygoround pfizer pfizerbiontec...	1.0	Positive
2217		dr marc siegel covid vaccinations increase rap...	1.0	Positive
5942		vaccines work excellent news study published 4...	1.0	Positive
347		md dr susannah thompson pulling stops leading ...	1.0	Positive

```
In [30]: text=' '.join([word for word in pos_tweets['text']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud=WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in positive tweets', fontsize=19)
plt.show()
```

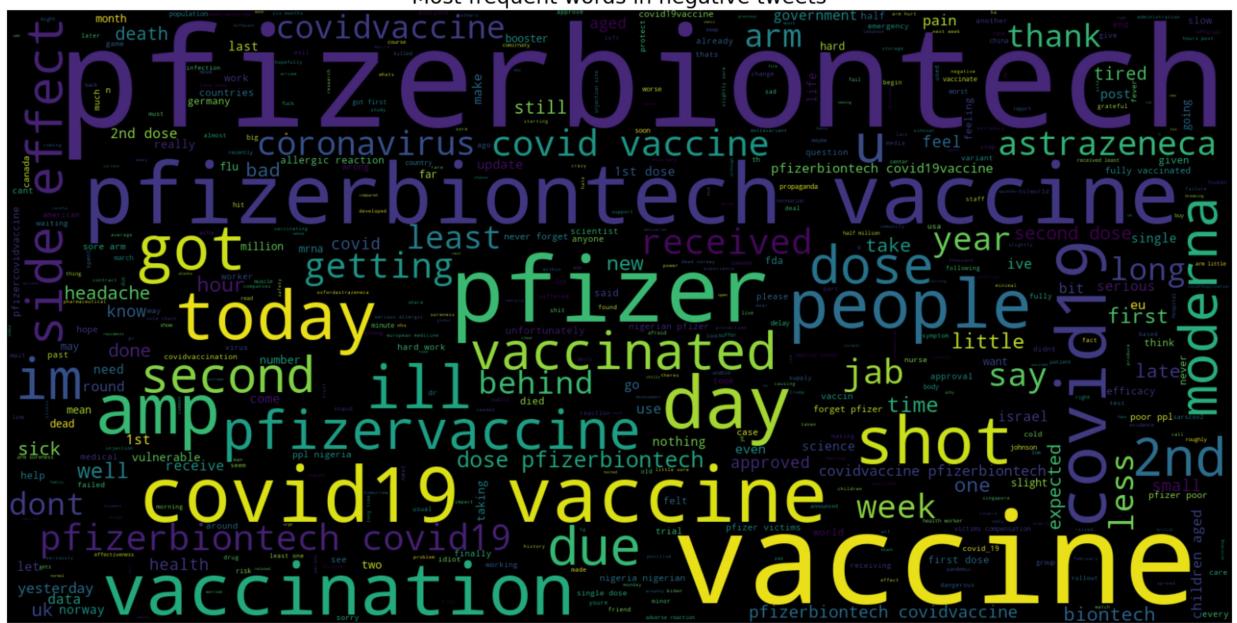


```
In [31]: neg_tweets=text_df[text_df.sentiment=='Negative']
neg_tweets=neg_tweets.sort_values(['polarity'], ascending=False)
neg_tweets.head()
```

Out[31]:

			text	polarity	sentiment
2912			work skilled nursing facility got first vaccin...	-0.003333	Negative
7256	200321 752308 vaccinations new daily record da...			-0.003409	Negative
2073	ukgovernment cant even vaccinate properly ethi...			-0.004762	Negative
7715	got first dose less waiting time airport vacci...			-0.005556	Negative
7157	second dose due end next month well far pfizer...			-0.006250	Negative

```
In [32]: text=' '.join([word for word in neg_tweets['text']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud=WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in negative tweets', fontsize=19)
plt.show()
```

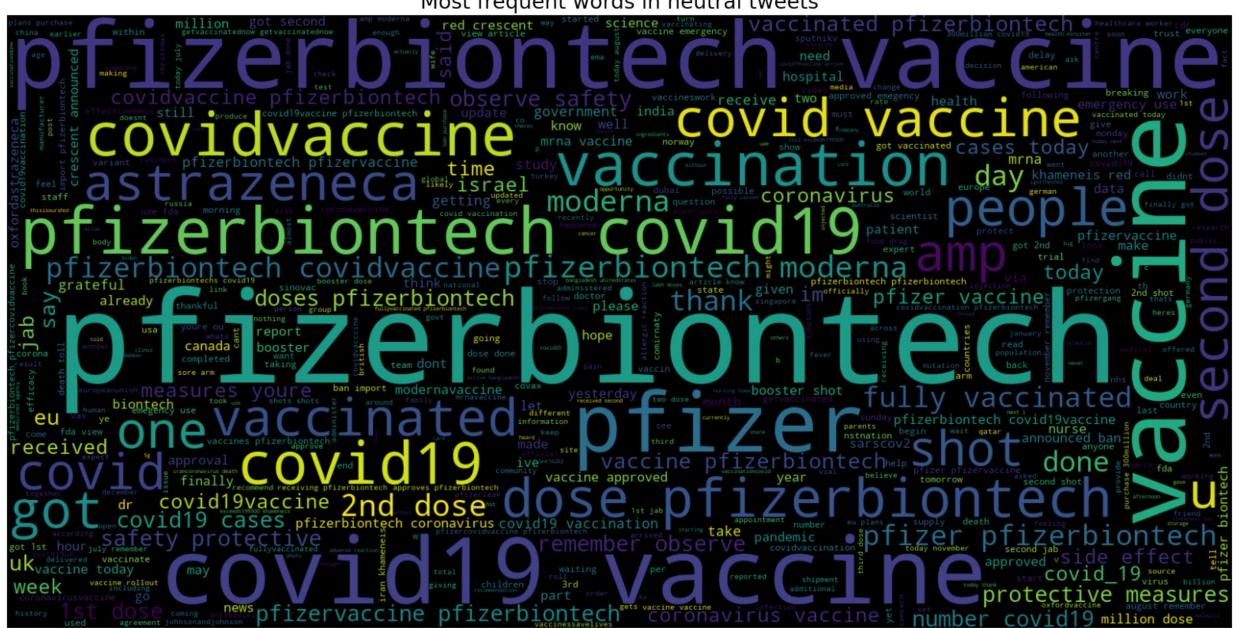


```
In [33]: neutral_tweets=text_df[text_df.sentiment=='Neutral']
neutral_tweets=neutral_tweets.sort_values(['polarity'], ascending=False)
neutral_tweets.head()
```

Out[33]:

		text	polarity	sentiment
0		folks said daikon paste could treat cytokine s...	0.0	Neutral
7539		additional measures discussions potential part...	0.0	Neutral
7558		2nd booster shot pfizerbiontech choosetovaccin...	0.0	Neutral
7557		2nd dose pfizerbiontech covid19vaccination	0.0	Neutral
7555		got mine two weeks ago pfizerbiontech side eff...	0.0	Neutral

```
In [34]: text=' '.join([word for word in neutral_tweets['text']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud=WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in neutral tweets', fontsize=19)
plt.show()
```



```
In [35]: vect=CountVectorizer(ngram_range=(1,2)).fit(text_df['text'])
```

```
In [37]: feature_names=vect.get_feature_names()
print("No of features: {}".format(len(feature_names)))
print("first 20 features: {}".format(feature_names[:20]))
```

No of features: 71637

```
first 20 features:  
['000', '000 doses', '000 initial', '000 people', '000 vaccines', '0000001', '000000  
1 covid19', '0011', '0011 abt', '004', '004 covid', '004 israelis', '01', '01 gettin  
g', '01 june', '01 november', '01aug2021', '01aug2021 doublevaxxed', '02', '02 175']
```

```
In [38]: X=text_df['text']
          Y=text_df['sentiment']
          X=vect.transform(X)
```

```
In [41]: x_train, x_test, y_train, y_test=train_test_split(X, Y, test_size=0.2, random_state=42)
```

```
In [42]: print("size of x_train: ", (x_train.shape))
        print("size of y_train: ", (y_train.shape))
        print("size of x_test: ", (x_test.shape))
        print("size of y test: ", (y_test.shape))
```

```
size of x_train: (8816, 71637)
size of y_train: (8816,)
size of x_test: (2204, 71637)
size of y test: (2204,)
```

```
In [44]: logreg=LogisticRegression()
logreg.fit(x_train, y_train)
logreg_pred=logreg.predict(x_test)
logreg_acc=accuracy_score(logreg_pred, y_test)
print("test accuracy: {:.2f}%" .format(logreg_acc*100))
```

test accuracy: 85.39%

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.  
  
Increase the number of iterations (max_iter) or scale the data as shown in:  
    https://scikit-learn.org/stable/modules/preprocessing.html  
Please also refer to the documentation for alternative solver options:  
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression  
n_iter_i = _check_optimize_result()
```

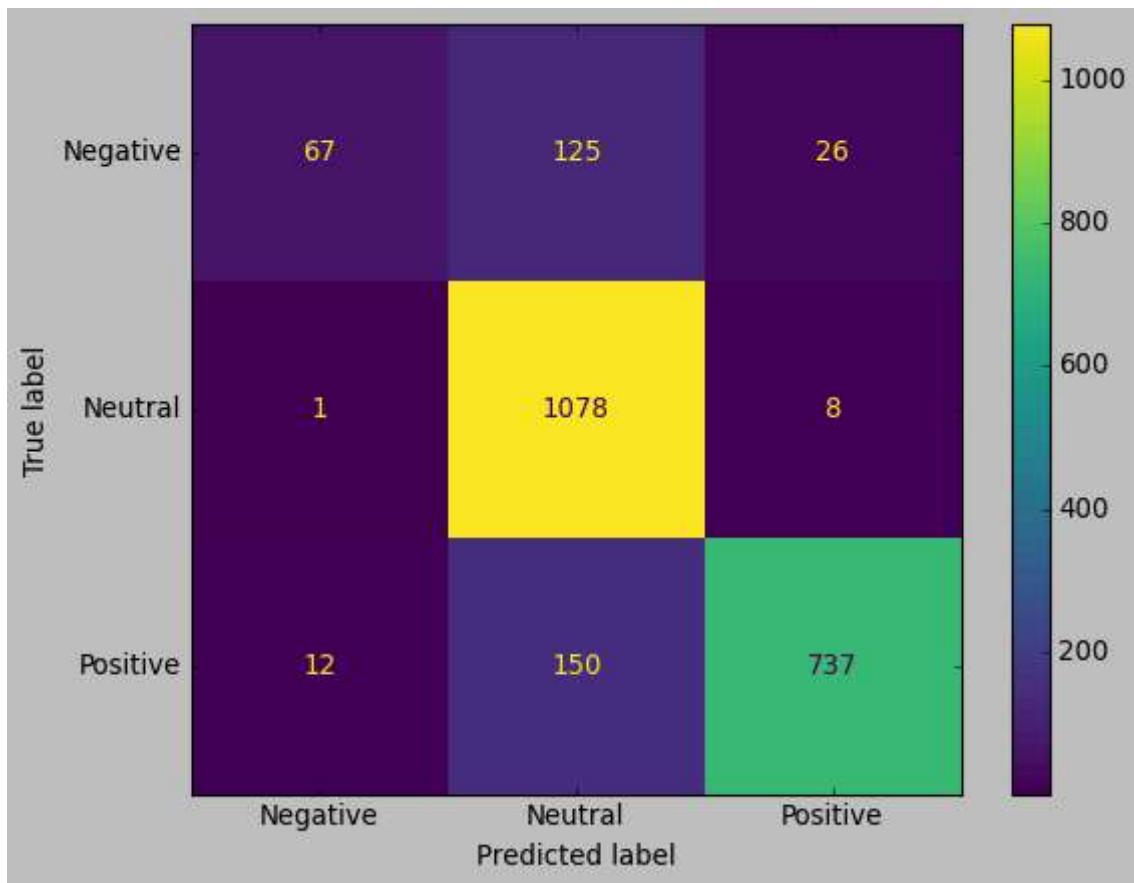
```
In [45]: print(confusion_matrix(y_test, logreg_pred))  
print("\n")  
print(classification_report(y_test, logreg_pred))
```

[[67 125 26]			
[1 1078 8]			
[12 150 737]]			

	precision	recall	f1-score	support
Negative	0.84	0.31	0.45	218
Neutral	0.80	0.99	0.88	1087
Positive	0.96	0.82	0.88	899
accuracy			0.85	2204
macro avg	0.86	0.71	0.74	2204
weighted avg	0.87	0.85	0.84	2204

```
In [46]: style.use('classic')  
cm=confusion_matrix(y_test, logreg_pred, labels=logreg.classes_)  
disp=ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=logreg.classes_ )  
disp.plot()
```

```
Out[46]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2190276d6a0>
```



```
In [48]: from sklearn.model_selection import GridSearchCV
```

```
In [49]: param_grid={'C':[0.001, 0.01, 0.1, 1, 10]}
grid=GridSearchCV(LogisticRegression(), param_grid)
grid.fit(x_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfsgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
```

```

Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result()
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result()

```

Out[49]:

```
GridSearchCV(estimator=LogisticRegression(),
             param_grid={'C': [0.001, 0.01, 0.1, 1, 10]})
```

In [50]:

```
print("best parameters: ", grid.best_params_)
```

```
best parameters: {'C': 10}
```

In [51]:

```
y_pred=grid.predict(x_test)
```

In [52]:

```
logreg_acc=accuracy_score(y_pred, y_test)
print("test accuracy: {:.2f}%".format(logreg_acc*100))
```

```
test accuracy: 86.71%
```

In [53]:

```
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

[[80	111	27]
[2	1078	7]
[14	132	753]]

	precision	recall	f1-score	support
Negative	0.83	0.37	0.51	218
Neutral	0.82	0.99	0.90	1087
Positive	0.96	0.84	0.89	899
accuracy			0.87	2204
macro avg	0.87	0.73	0.77	2204
weighted avg	0.88	0.87	0.86	2204

In [54]:

```
from sklearn.svm import LinearSVC
```

In [55]:

```
SVCmodel=LinearSVC()
SVCmodel.fit(x_train, y_train)
```

Out[55]:

```
LinearSVC()
```

In [56]:

```
svc_pred=SVCmodel.predict(x_test)
svc_acc=accuracy_score(svc_pred, y_test)
print("test accuracy: {:.2f}%".format(svc_acc*100))
```

```
test accuracy: 87.89%
```

```
In [57]: print(confusion_matrix(y_test, svc_pred))
print("\n")
print(classification_report(y_test, svc_pred))

[[ 96  98  24]
 [  2 1076   9]
 [ 16 118 765]]
```

	precision	recall	f1-score	support
Negative	0.84	0.44	0.58	218
Neutral	0.83	0.99	0.90	1087
Positive	0.96	0.85	0.90	899
accuracy			0.88	2204
macro avg	0.88	0.76	0.79	2204
weighted avg	0.89	0.88	0.87	2204

```
In [59]: grid={
    'C':[0.01, 0.1, 1, 10],
    'kernel':["linear","poly","rbf","sigmoid"],
    'degree':[1,3,5,7],
    'gamma':[0.01,1]
}

grid=GridSearchCV(SVCmodel, param_grid)
grid.fit(x_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
```

```
Out[59]: GridSearchCV(estimator=LinearSVC(), param_grid={'C': [0.001, 0.01, 0.1, 1, 10]})
```

```
In [60]: print("best parameter: ", grid.best_params_)

best parameter: {'C': 10}
```

```
In [61]: y_pred=grid.predict(x_test)
```

```
In [62]: logreg_acc=accuracy_score(y_pred, y_test)
print("test accuracy: {:.2f}%".format(logreg_acc*100))
```

test accuracy: 88.29%

```
In [63]: print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))

[[ 99  95  24]
 [  3 1075   9]
 [ 16  111 772]]
```

	precision	recall	f1-score	support
Negative	0.84	0.45	0.59	218
Neutral	0.84	0.99	0.91	1087
Positive	0.96	0.86	0.91	899
accuracy			0.88	2204
macro avg	0.88	0.77	0.80	2204
weighted avg	0.89	0.88	0.88	2204

```
In [ ]:
```