In [1]:
```python
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt

from wordcloud import WordCloud
from matplotlib import style
style.use('ggplot')
from textblob import TextBlob

from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import nltk

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, ConfusionMatrixDisplay
```

In [2]:
```python
df = pd.read_csv('vaccination_tweets.csv')
```

In [3]:
```python
df.head()
```

Out[3]:

	id	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date	text	hashtags	sour
0	1340539111971516416	Rachel Roh	La Crescenta-Montrose, CA	Aggregator of Asian American news; scanning di...	2009-04-08 17:52:46	405	1692	3247	False	2020-12-20 06:06:44	Same folks said daikon paste could treat a cyt...	['PfizerBioNTech']	Twit Andr
1	1338158543359250433	Albert Fong	San Francisco, CA	Marketing dude, tech geek, heavy metal & '80s ...	2009-09-21 15:27:30	834	666	178	False	2020-12-13 16:27:13	While the world has been on the wrong side of ...	NaN	Twit W A
2	1337858199140118533	eli🇱🇹🇪🇺👌	Your Bed	heil, hydra 🖐☺	2020-06-25 23:30:28	10	88	155	False	2020-12-12 20:33:45	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf...	Twit Andr
3	1337855739918835717	Charles Adler	Vancouver, BC - Canada	Hosting "CharlesAdlerTonight" Global News Radi...	2008-09-10 11:28:53	49165	3933	21853	True	2020-12-12 20:23:59	Facts are immutable, Senator, even when you're...	NaN	Twit W A
4	1337854064604966912	Citizen News Channel	NaN	Citizen News Channel bringing you an alternati...	2020-04-23 17:58:42	152	580	1473	False	2020-12-12 20:17:19	Explain to me again why we need a vaccine @Bor...	['whereareallthesickpeople', 'PfizerBioNTech']	Twit iPho

In [4]:
```python
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  10340 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 [5]:
```python
df.columns
```

Out[5]:
```
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 [6]:
```python
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[6]:

	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 [7]:
```python
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 [8]:
```python
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 [9]:
```python
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 = nltk.word_tokenize(text)
    filtered_text = [w for w in text_tokens if not w in stopwords.words('english')]
    return " ".join(filtered_text)

text_df.text = text_df['text'].apply(data_processing)
text_df = text_df.drop_duplicates('text')

stemmer = PorterStemmer()
def stemming(data):
    text = [stemmer.stem(word) for word in data.split()]
    return " ".join(text)

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

In [10]:
```python
text_df.head()
```

Out[10]:

	text
0	folk said daikon past could treat cytokin stor...
1	world wrong side histori year hope biggest vac...
2	coronaviru sputnikv astrazeneca pfizerbiontech...
3	fact immut senat even your ethic sturdi enough...
4	explain need vaccin borisjohnson matthancock w...

In [11]:
```python
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")
```

```
folk said daikon past could treat cytokin storm pfizerbiontech

world wrong side histori year hope biggest vaccin effort weve ev

coronaviru sputnikv astrazeneca pfizerbiontech moderna covid_19 russian vaccin creat last 24 year

fact immut senat even your ethic sturdi enough acknowledg 1 born

explain need vaccin borisjohnson matthancock whereareallthesickpeopl pfizerbiontech
```

In [12]:
```python
text_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10543 entries, 0 to 11019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   text    10543 non-null  object
dtypes: object(1)
memory usage: 164.7+ KB
```

In [13]:
```python
def polarity(text):
    return TextBlob(text).sentiment.polarity
```

In [14]:
```python
text_df['polarity'] = text_df['text'].apply(polarity)
```

In [15]:
```python
text_df.head(10)
```

Out[15]:

	text	polarity
0	folk said daikon past could treat cytokin stor...	-0.250000
1	world wrong side histori year hope biggest vac...	-0.500000
2	coronaviru sputnikv astrazeneca pfizerbiontech...	0.000000
3	fact immut senat even your ethic sturdi enough...	0.000000
4	explain need vaccin borisjohnson matthancock w...	0.000000
5	anyon use adviceguid whether covid vaccin safe...	0.500000
6	bit sad claim fame success vaccin patriot comp...	-0.100000
7	mani bright day 2020 best 1 bidenharri win ele...	0.833333
8	covid vaccin get covidvaccin covid19 pfizerbio...	0.000000
9	covidvaccin state start get covid19vaccin mond...	0.000000

In [16]:
```python
def sentiment(label):
    if label <0:
        return "Negative"
    elif label ==0:
        return "Neutral"
    elif label>0:
        return "Positive"
```

In [17]:
```python
text_df['sentiment'] = text_df['polarity'].apply(sentiment)
```

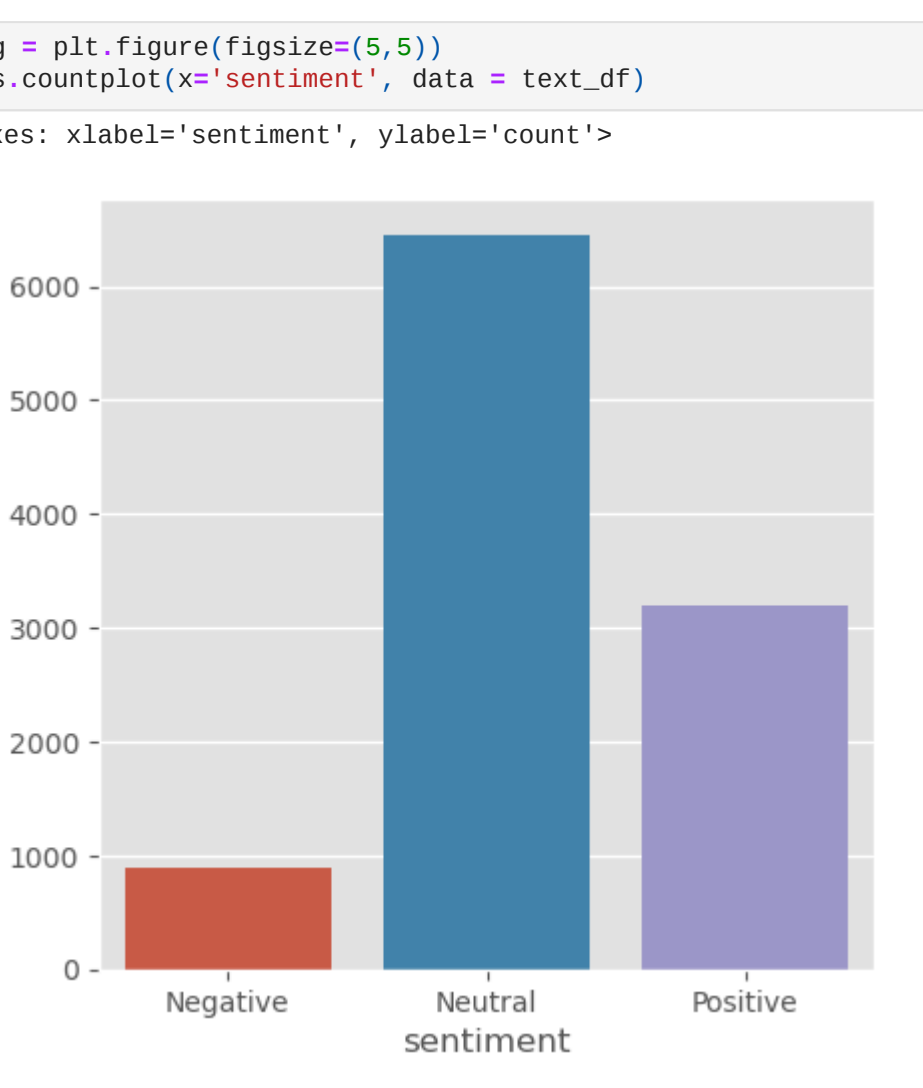
In [18]:
```python
text_df.head()
```

Out[18]:

	text	polarity	sentiment
0	folk said daikon past could treat cytokin stor...	-0.25	Negative
1	world wrong side histori year hope biggest vac...	-0.50	Negative
2	coronaviru sputnikv astrazeneca pfizerbiontech...	0.00	Neutral
3	fact immut senat even your ethic sturdi enough...	0.00	Neutral
4	explain need vaccin borisjohnson matthancock w...	0.00	Neutral

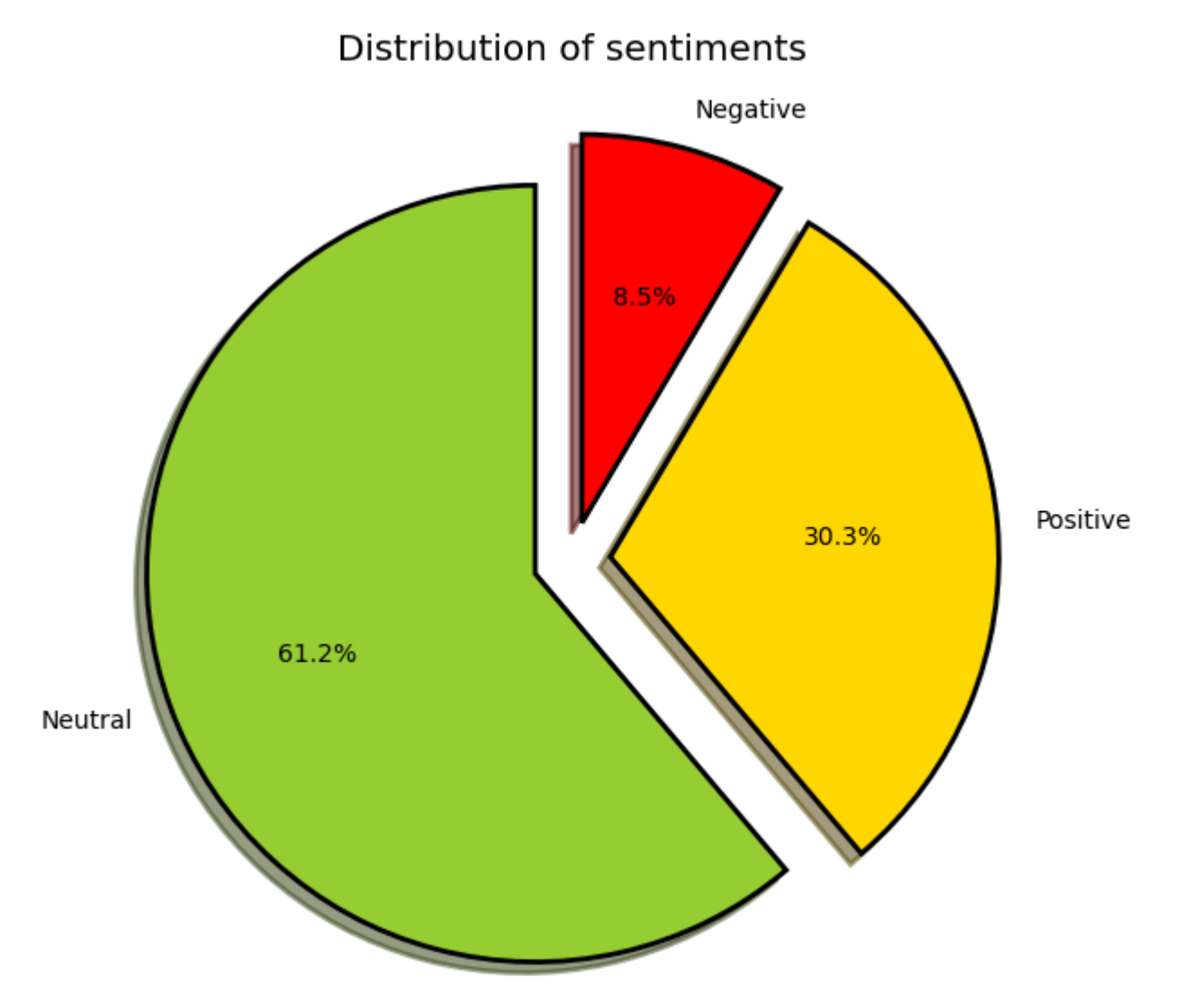
In [20]:
```python
fig = plt.figure(figsize=(5,5))
sns.countplot(x='sentiment', data = text_df)
```

Out[20]: <Axes: xlabel='sentiment', ylabel='count'>



In [21]:
```python
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, wedgeprops = wp, explode = explode, label='')
plt.title('Distribution of sentiments')
```

Out[21]: Text(0.5, 1.0, 'Distribution of sentiments')

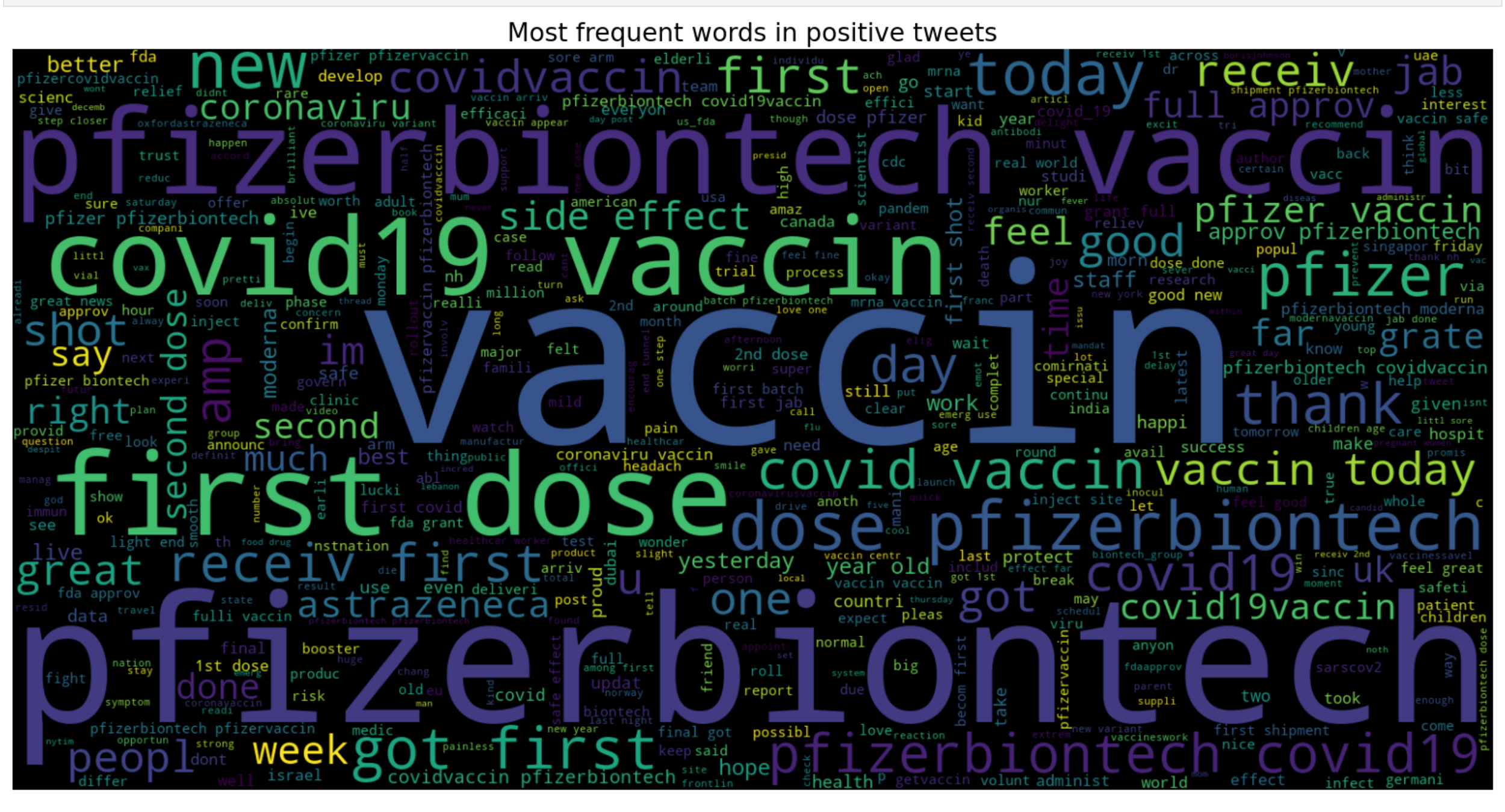


In [22]:
```python
pos_tweets = text_df[text_df.sentiment == 'Positive']
pos_tweets = pos_tweets.sort_values(['polarity'], ascending= False)
pos_tweets.head()
```

Out[22]:

	text	polarity	sentiment
2004	fulli vaccin covid 19 best gift 2021 thank hel...	1.0	Positive
5889	prguy17 scottythequeuejump resignaustralian de...	1.0	Positive
9216	despit israel pfizerbiontech vaccin consid bes...	1.0	Positive
9317	best way get merrygoround pfizer pfizerbiontec...	1.0	Positive
5636	thank europ develop best vaccin avail let us s...	1.0	Positive

In [23]:
```python
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 [24]:
```python
neg_tweets = text_df[text_df.sentiment == 'Negative']
neg_tweets = neg_tweets.sort_values(['polarity'], ascending= False)
neg_tweets.head()
```

Out[24]:

	text	polarity	sentiment
7715	got first dose less wait time airport vaccin c...	-0.005556	Negative
7157	nas_k27 second dose due end next month well fa...	-0.006250	Negative
575	u think bill gate evil becoim rich nice person...	-0.008333	Negative
3738	due experi norway suggest covid19 vaccin may r...	-0.012500	Negative
8138	long american british isra feed final got shot...	-0.012500	Negative

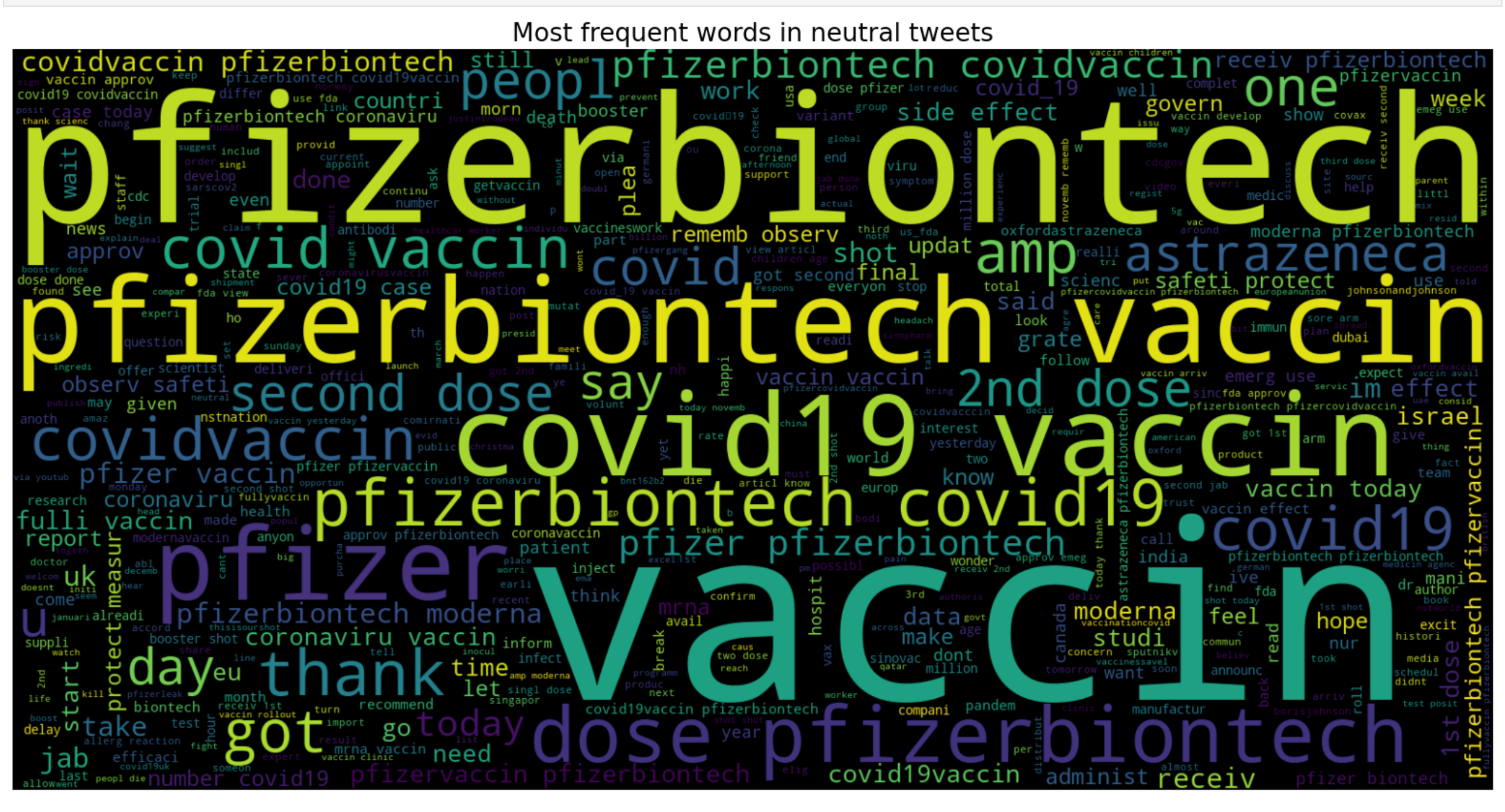
In [25]:
```python
text = ' '.join([word for word in neg_tweets['text']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Most frequent words in negative tweets', fontsize=19)
plt.show()
```

In [26]:
```python
neutral_tweets = text_df[text_df.sentiment == 'Neutral']
neutral_tweets = neutral_tweets.sort_values(['polarity'], ascending= False)
neutral_tweets.head()
```

Out[26]:

	text	polarity	sentiment
2	coronaviru sputnikv astrazeneca pfizerbiontech...	0.0	Neutral
7364	test pfizerbiontech vaccin hospit worker cambr...	0.0	Neutral
7377	1 1 go pfizerbiontech pr	0.0	Neutral
7375	break moderna pfizerbiontech vaccin highli eff...	0.0	Neutral
7374	eu mam vaccinat covid19 covidwar pfizerbiontech	0.0	Neutral

In [27]:
```python
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 [29]:
```python
vect = CountVectorizer(ngram_range=(1,2)).fit(text_df['text'])
X = text_df['text']
Y = text_df['sentiment']
X = vect.transform(X)
```

In [30]:
```python
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

In [31]:
```python
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: (8434, 71682)
Size of y_train: (8434,)
Size of x_test: (2109, 71682)
Size of y_test: (2109,)
```

In [32]:
```python
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: 90.14%
```

In [ ]: