In [1]:
```
!pip install scikit-learn
!pip install imbalanced-learn
!pip install imblearn
```

```
Requirement already satisfied: scikit-learn in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silicon/
lib/python3.9/site-packages (1.2.0)
Requirement already satisfied: scipy>=1.3.2 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silicon/
lib/python3.9/site-packages (from scikit-learn) (1.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_
silicon/lib/python3.9/site-packages (from scikit-learn) (3.1.0)
Requirement already satisfied: numpy>=1.17.3 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from scikit-learn) (1.23.2)
Requirement already satisfied: joblib>=1.1.1 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: imbalanced-learn in /Users/pravinanandpawar/miniforge3/envs/tensorflow_sili
con/lib/python3.9/site-packages (0.10.1)
Requirement already satisfied: joblib>=1.1.1 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from imbalanced-learn) (1.2.0)
Requirement already satisfied: scipy>=1.3.2 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silicon/
lib/python3.9/site-packages (from imbalanced-learn) (1.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_
silicon/lib/python3.9/site-packages (from imbalanced-learn) (3.1.0)
Requirement already satisfied: numpy>=1.17.3 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from imbalanced-learn) (1.23.2)
Requirement already satisfied: scikit-learn>=1.0.2 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_s
ilicon/lib/python3.9/site-packages (from imbalanced-learn) (1.2.0)
Requirement already satisfied: imblearn in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silicon/lib/
python3.9/site-packages (0.0)
Requirement already satisfied: imbalanced-learn in /Users/pravinanandpawar/miniforge3/envs/tensorflow_sili
con/lib/python3.9/site-packages (from imblearn) (0.10.1)
Requirement already satisfied: numpy>=1.17.3 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from imbalanced-learn->imblearn) (1.23.2)
Requirement already satisfied: joblib>=1.1.1 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silico
n/lib/python3.9/site-packages (from imbalanced-learn->imblearn) (1.2.0)
Requirement already satisfied: scipy>=1.3.2 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_silicon/
lib/python3.9/site-packages (from imbalanced-learn->imblearn) (1.10.0)
Requirement already satisfied: scikit-learn>=1.0.2 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_s
ilicon/lib/python3.9/site-packages (from imbalanced-learn->imblearn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/pravinanandpawar/miniforge3/envs/tensorflow_
silicon/lib/python3.9/site-packages (from imbalanced-learn->imblearn) (3.1.0)
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,f1_score
from wordcloud import WordCloud, STOPWORDS
from sklearn.feature_extraction.text import TfidfVectorizer
from imblearn.over_sampling import SMOTE
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

In [3]:
```
df = pd.read_csv('Tweets.csv')
df.head()
```

Out[3]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airl
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	

In [4]:
```
listAirline = df['airline'].unique()
listAirline
```

Out[4]:
```
array(['Virgin America', 'United', 'Southwest', 'Delta', 'US Airways',
       'American'], dtype=object)
```

In [5]:
```
listSentiment = df['airline_sentiment'].unique()
listSentiment
```

Out[5]:
```
array(['neutral', 'positive', 'negative'], dtype=object)
```

In [6]:
```
sentiment_details = df['airline_sentiment'].value_counts()
```

In [7]:
```
idx = sentiment_details.index
idx[0]
```

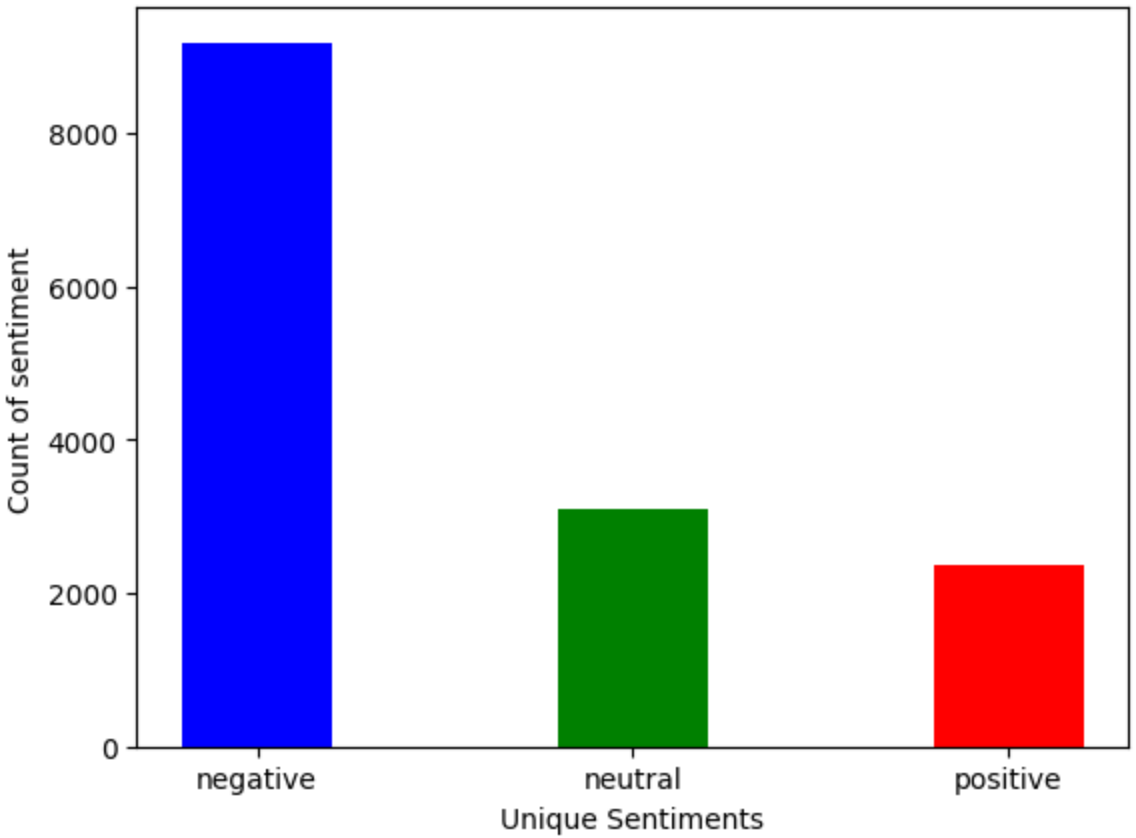
Out[7]:
```
'negative'
```

In [8]:
```
sentiment_details.values
```

Out[8]:
```
array([9178, 3099, 2363])
```

In [9]:
```
def plot(x, y, x_lable, y_label):
    plt.bar( x, y, color=['blue','Green','red', 'brown', 'olive'],
            width = 0.4)
    plt.xlabel(x_lable)
    plt.ylabel(y_label)
    plt.show()
```

In [10]:
```
plot(sentiment_details.index, sentiment_details.values, "Unique Sentiments", "Count of sentiment")
```



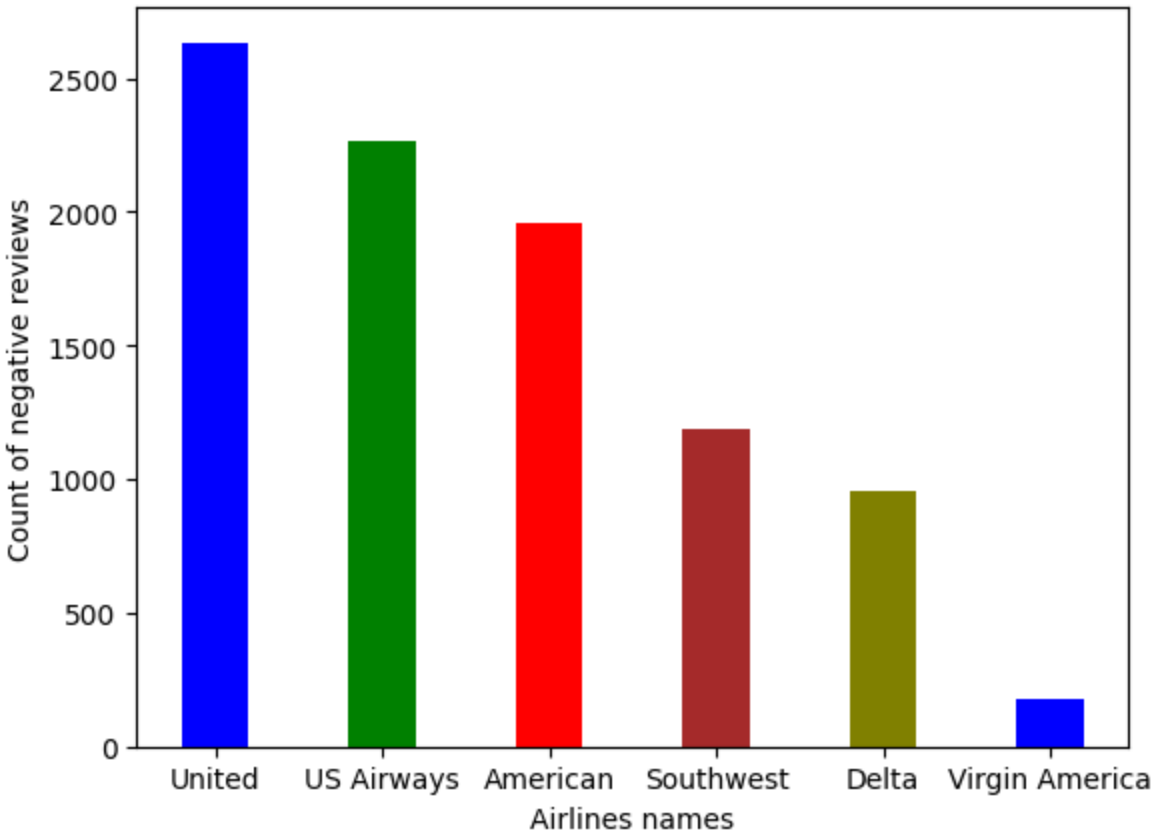
In [11]:
```
df_companies = df[['airline_sentiment', 'airline']]
df_companies
```

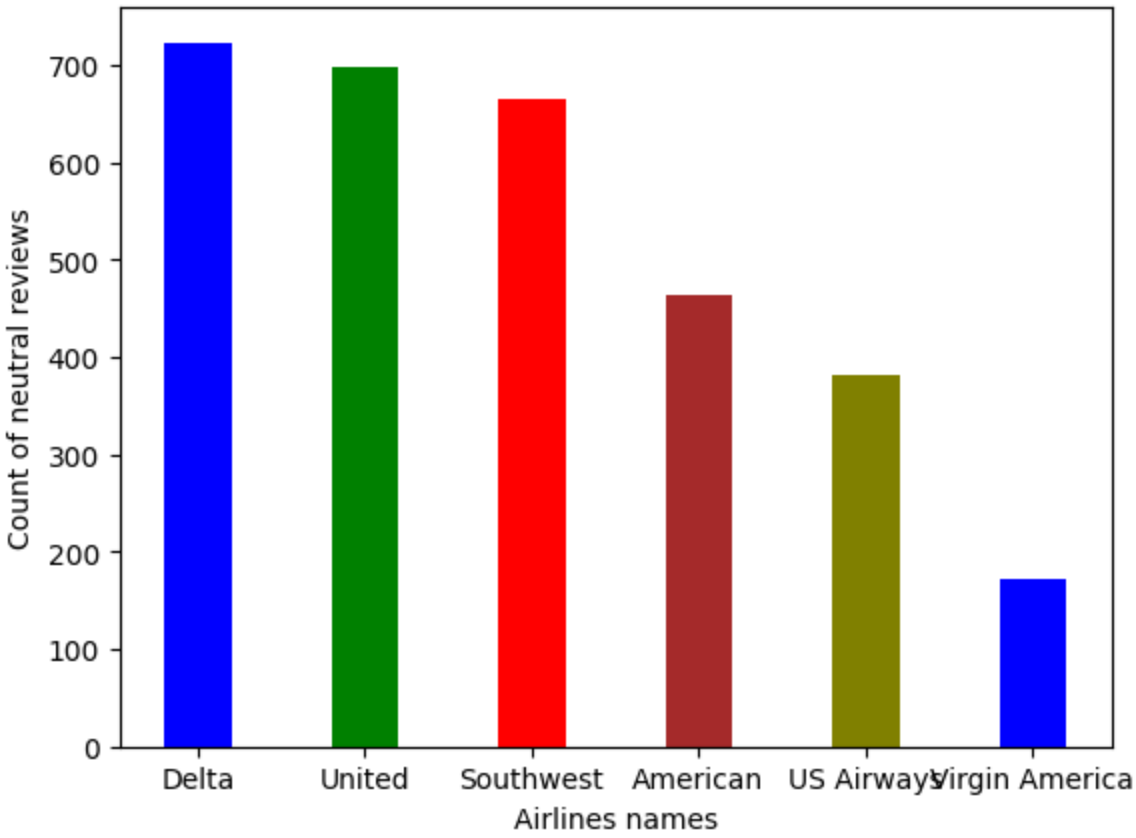
Out[11]:

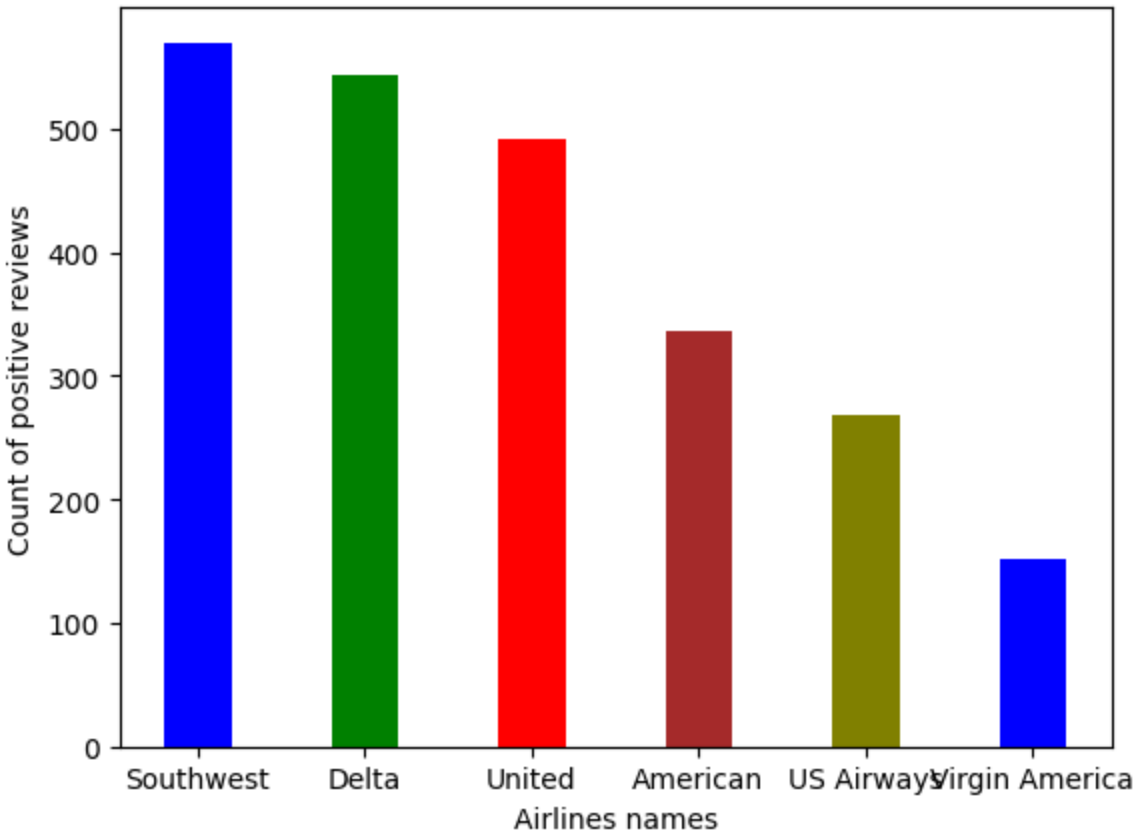
	airline_sentiment	airline
0	neutral	Virgin America
1	positive	Virgin America
2	neutral	Virgin America
3	negative	Virgin America
4	negative	Virgin America
...	...	...
14635	positive	American
14636	negative	American
14637	neutral	American
14638	negative	American
14639	neutral	American

14640 rows × 2 columns

In [12]:
```
for x in sentiment_details.index:
    airline_review = df_companies[df_companies.airline_sentiment == x]['airline'].value_counts()
    plot(airline_review.index, airline_review.values, "Airlines names", "Count of "+x+" reviews")
```


2500
2000
1500
1000
500
0
Count of negative reviews
United
US Airways
American
Southwest
Delta
Virgin America
Airlines names


Count of neutral reviews
0
100
200
300
400
500
600
700
Delta
United
Southwest
American
US Airways
Virgin America
Airlines names



In [13]:
```
# Preprocessing
```

In [14]:
```
df = df[df.airline_sentiment_confidence > 0.7]
airline_df = df[['airline_sentiment', 'text']]
airline_df
```

Out[14]:

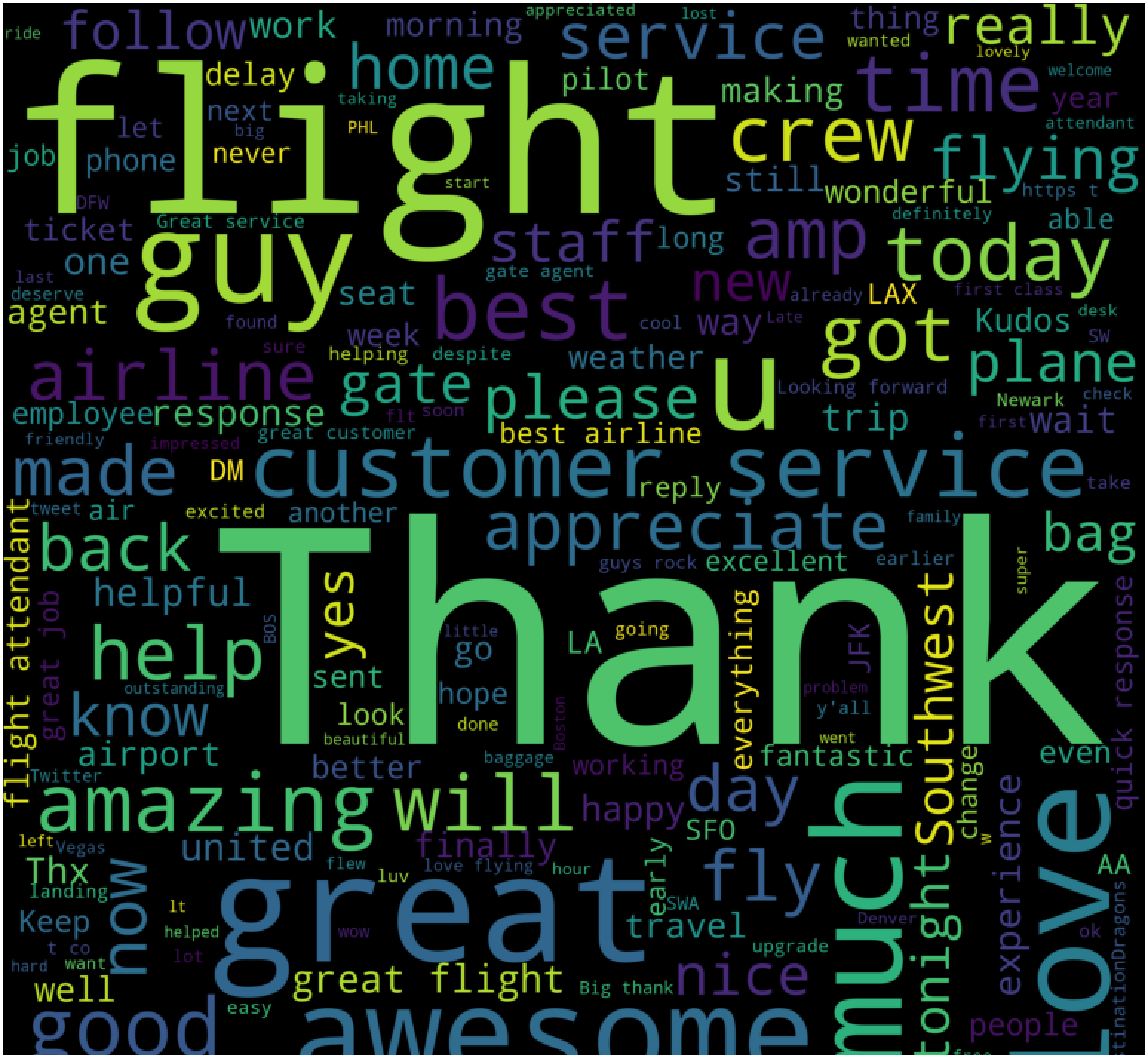
	airline_sentiment	text
0	neutral	@VirginAmerica What @dhepburn said.
3	negative	@VirginAmerica it's really aggressive to blast...
4	negative	@VirginAmerica and it's a really big bad thing...
5	negative	@VirginAmerica seriously would pay $30 a fligh...
9	positive	@VirginAmerica it was amazing, and arrived an ...
...	...	...
14631	negative	@AmericanAir thx for nothing on getting us out...
14633	negative	@AmericanAir my flight was Cancelled Flightled...
14636	negative	@AmericanAir leaving over 20 minutes Late Flig...
14637	neutral	@AmericanAir Please bring American Airlines to...
14638	negative	@AmericanAir you have my money, you change my ...

10760 rows × 2 columns

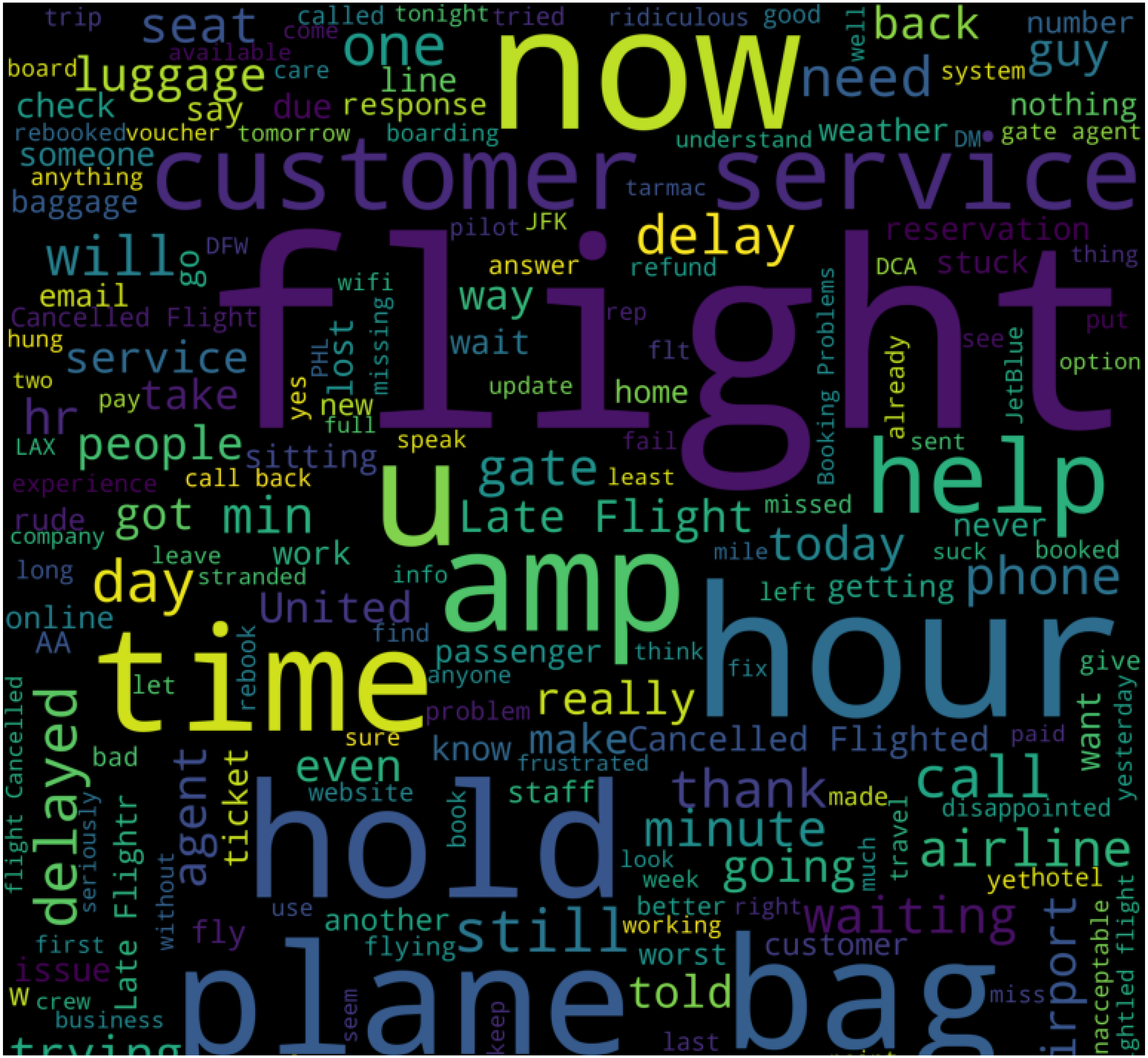
In [15]:
```
# Word Cloud
```

In [16]:
```
def plotWordCloud(t):
    words = ' '.join(l for l in  df[df['airline_sentiment']==t].text)
    filtered_words = " ".join([word for word in words.split()
                               if not word.startswith('@') and not word.startswith('http://')
                               ])
    wordcloud = WordCloud(stopwords=STOPWORDS,
                          background_color='black',
                          width=2500,
                          height=2500
                         ).generate(filtered_words)
    plt.figure(1,figsize=(10, 10))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()
```

In [17]:
```
# Positive
plotWordCloud('positive')
```


flight
Thank
great
awesome
guy
customer service
much
love
u
best
airline
time
service
crew
today
got
plane
flying
really
home
follow
work
morning
appreciated
lost
thing
wanted
lovely
welcome
year
attendant
ride
delay
next
let
big
PHL
taking
pilot
making
job
phone
never
start
still
wonderful
https t
definitely
DFW
Great service
ticket
staff
long
amp
able
one
last
gate agent
deserve
new
already
LAX
first class
agent
seat
found
cool
way
Late
desk
Kudos
SW
week
sure
helping
despite
weather
gate
please
Looking forward
check
Newark
employee
response
flt
soon
trip
wait
first
friendly
impressed
great customer
best airline
made
DM
reply
take
tweet
air
excited
another
appreciate
family
bag
back
guys rock
excellent
earlier
super
helpful
yes
flight attendant
great job
BOS
help
little
going
LA
go
JFK
Southwest
everything
quick response
sent
outstanding
know
hope
problem
look
y'all
Boston
done
beautiful
went
airport
better
baggage
working
fantastic
even
Twitter
change
amazing
will
day
happy
experience
SFO
finally
united
left
Vegas
w
flew
love flying
hour
AA
Thx
luv
landing
early
fly
now
lt
SWA
tonight
Keep
Denver
helped
wow
ok
travel
t co
upgrade
lot
hard
want
nice
great flight
well
Big thank
easy
people
good

In [18]:
```
# Negative
plotWordCloud('negative')
```


trip
seat
called
tonight
tried
ridiculous
good
back
number
come
available
now
one
guy
well
board
luggage
care
need
system
line
check
say
due
response
nothing
rebooked
voucher
tomorrow
boarding
weather
gate agent
understand
DM
someone
customer
service
anything
tarmac
baggage
pilot
JFK
reservation
will
go
DFW
delay
thing
answer
refund
DCA
stuck
email
wifi
way
Cancelled Flight
rep
put
hung
missing
wait
flt
see
service
lost
PHL
Booking Problems
already
JetBlue
option
two
update
pay
take
yes
new
home
hr
full
flight
fail
sent
LAX
people
speak
sitting
experience
call back
help
least
gate
missed
rude
got min
u
Late Flight
never
company
today
leave
work
mile
suck
booked
long
stranded
info
day
getting
phone
left
amp
United
online
find
hour
AA
passenger
think
fix
give
rebook
let
anyone
time
really
problem
want
flight Cancelled
sure
paid
yesterday
delayed
bad
know
make
Cancelled Flighted
frustrated
agent
ticket
even
thank
call
website
staff
made
seriously
book
disappointed
Late Flightr
minute
travel
airline
without
look
much
hold
going
week
yet
hotel
use
better
right
fly
another
still
waiting
first
working
flying
airport
unacceptable
Cancelled Flightled flight
worst
customer
issue
w
told
crew
miss
seem
keep
business
plane
bag
last
trying

In [19]:
```
# Encoding
y = pd.factorize(airline_df['airline_sentiment'])
y
```

Out[19]:
```
(array([0, 1, 1, ..., 1, 0, 1]),
 Index(['neutral', 'negative', 'positive'], dtype='object'))
```

In [20]:
```
# load data
X = airline_df['text']
tf_idf_features = TfidfVectorizer()
X = tf_idf_features.fit_transform(X)
print(X[0])
```

```
  (0, 9665)     0.4326189761894123
  (0, 3954)     0.7682167910584247
  (0, 12094)    0.31014240570373325
  (0, 11872)    0.3556620187214576
```

In [21]:
```
smote = SMOTE()
X, y = smote.fit_resample(X, y[0])
```

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=31)
```

In [23]:
```
# SVC
```

In [24]:
```
model = SVC(class_weight='balanced', C=1,gamma='scale', kernel='rbf')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [25]:
```
accuracy_score = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test,y_pred,average='weighted')
```

In [26]:
```
accuracy_score
```

Out[26]:
```
0.9670936395759717
```

In [27]:
```
f1
```

Out[27]:
```
0.9671904410391737
```

In [28]:
```
# Encoding
test_case =  tf_idf_features.transform(['flying with @united is always a great experience'])
```

In [29]:
```
model.predict(test_case)
```

Out[29]:
```
array([2])
```

In [ ]:

In [ ]: