Twitter US Airline Sentiment

Objective: Analyze how travelers in February 2015 expressed their feelings on Twitter

In current data set we have tweets for 6 US airlines and we need to predict whether the tweets are positive, negative or neutral

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import datetime

import re
import nltk
from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn import metrics

from sklearn.metrics import accuracy_score, classification_report

from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression


import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Saved successfully!

```
tweets_df = pd.read_csv("/content/drive/MyDrive/varun/Tweets.csv")
```

```
tweets_df.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativerea
0	570306133677760513	neutral	1.0000	N
1	570301130888122368	positive	0.3486	N
2	570301083672813571	neutral	0.6837	N

```
tweets_df.shape
```

```
(14640, 15)
```

```
tweets_df.columns
```

```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline',
       'airline_sentiment_gold', 'name', 'negativereason_gold',
       'retweet_count', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

```
# Check if any of the columns have unique values
nonunique_cols = [featr for featr in tweets_df.columns if len(tweets_df[featr].unique()) <
nonunique_cols
```

```
[]
```

missing value analysis:

```
#Check for missing values
100*tweets_df.isna().sum()/len(tweets_df)
```

```
tweet_id                         0.000000
airline_sentiment                0.000000
airline_sentiment_confidence     0.000000
negativereason                  37.308743
negativereason_confidence       28.128415
                                 0.000000
                                99.726776
```

Saved successfully!

```
name                              0.000000
negativereason_gold              99.781421
retweet_count                     0.000000
text                              0.000000
tweet_coord                      93.039617
tweet_created                     0.000000
tweet_location                   32.329235
user_timezone                    32.923497
dtype: float64
```

we observe that airline_sentiment_gold, negativereason_gold and tweet_coord have more tha 90% of missing values, let us drop them as they don't provide any constructive feedback

```
tweets_df.drop(['airline_sentiment_gold', 'negativereason_gold', 'tweet_coord'], axis=1, i
```

```
100*tweets_df.isna().sum()/len(tweets_df)
```

```
tweet_id                          0.000000
airline_sentiment                 0.000000
airline_sentiment_confidence      0.000000
negativereason                   37.308743
negativereason_confidence        28.128415
airline                           0.000000
name                              0.000000
retweet_count                     0.000000
text                              0.000000
tweet_created                     0.000000
tweet_location                   32.329235
user_timezone                    32.923497
dtype: float64
```

```
tweets_df[['negativereason', 'negativereason_confidence', 'tweet_location', 'user_timezone
```

	negativereason	negativereason_confidence	tweet_location	user_timezone
0	NaN	NaN	NaN	Eastern Time (US & Canada)
1	NaN	0.0000	NaN	Pacific Time (US & Canada)
2	NaN	NaN	Lets Play	Central Time (US & Canada)
				Pacific Time (US &

EDA

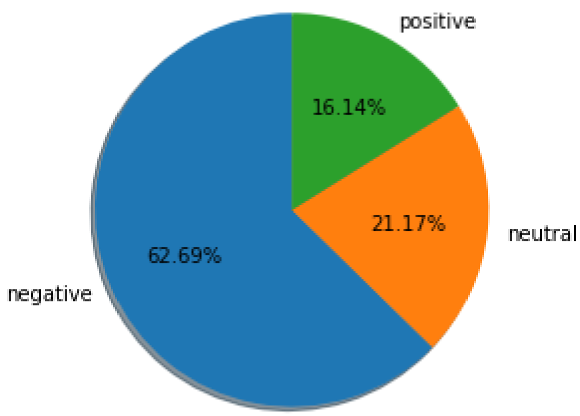
```
# Data balance
def createPieChartFor(t_df):
    Lst = 100*t_df.value_counts()/len(t_df)
```

Saved successfully!

```
dex.values
```

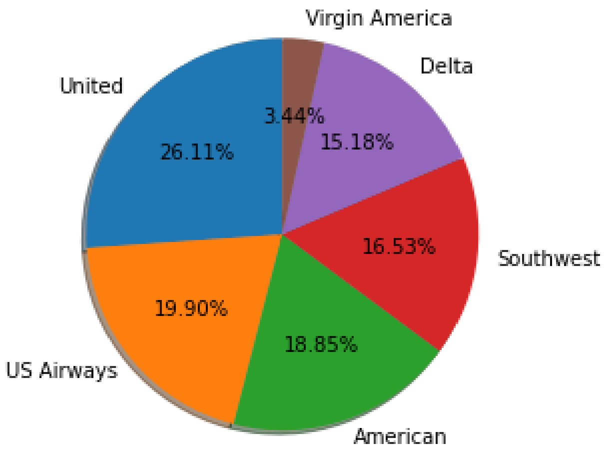
```
    sizes =  Lst

    # set labels
    fig1, ax1 = plt.subplots()
    ax1.pie(sizes, labels=labels, autopct='%1.2f%%', shadow=True, startangle=90)
    ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
    plt.show()
```

```
createPieChartFor(tweets_df.airline_sentiment)
```



from above we can see that we have majority of negative comments (63%) followed by neutral (21%) and positive (16%)

```
createPieChartFor(tweets_df.airline)
```

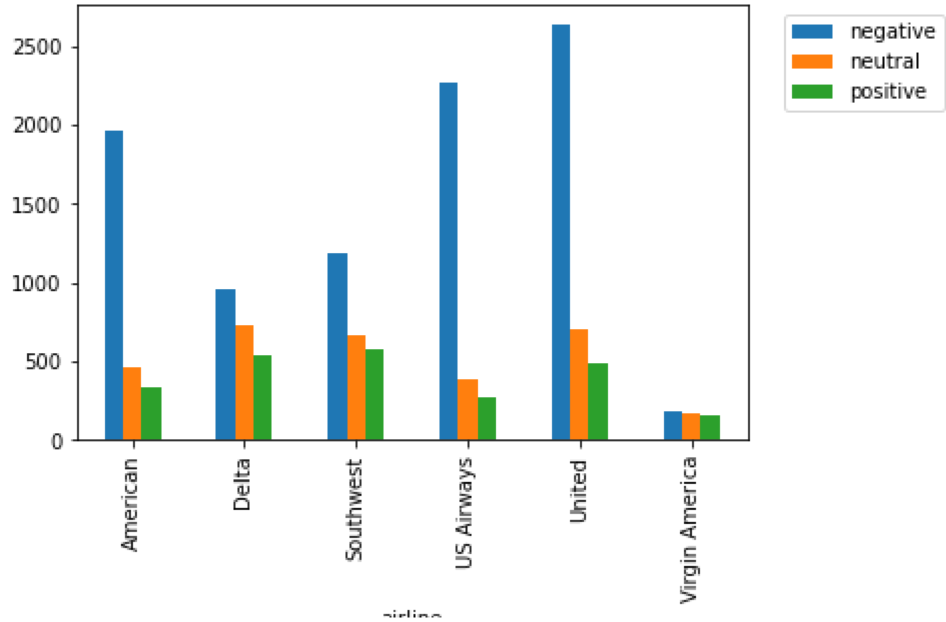


1. now check total tweets for each of the airlines and
2. how many of these tweets per airline are negative, positive and neutral

```
airline_sentiment_df = tweets_df.groupby(['airline','airline_sentiment']).airline_sentimer
airline_sentiment_df.plot(kind='bar')
plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
```

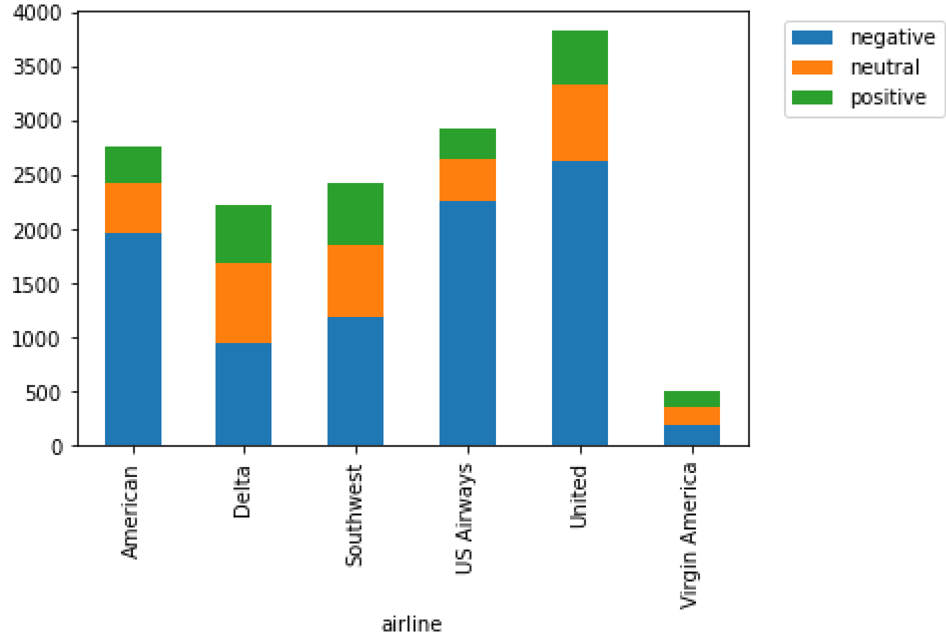
Saved successfully!

```
<matplotlib.legend.Legend at 0x7f8952603c50>
```




```
airline_sentiment_df.plot(kind='bar', stacked=True)
plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
```

```
<matplotlib.legend.Legend at 0x7f8951249250>
```



From above graph we can see that

1. United, US Airways and American have substatially negative tweets, these also have got over all more tweets
2. Virgin America, Delta and Southwest have fairly balanced tweets

Let's convert tweet_created to datetime check if we can get any insights

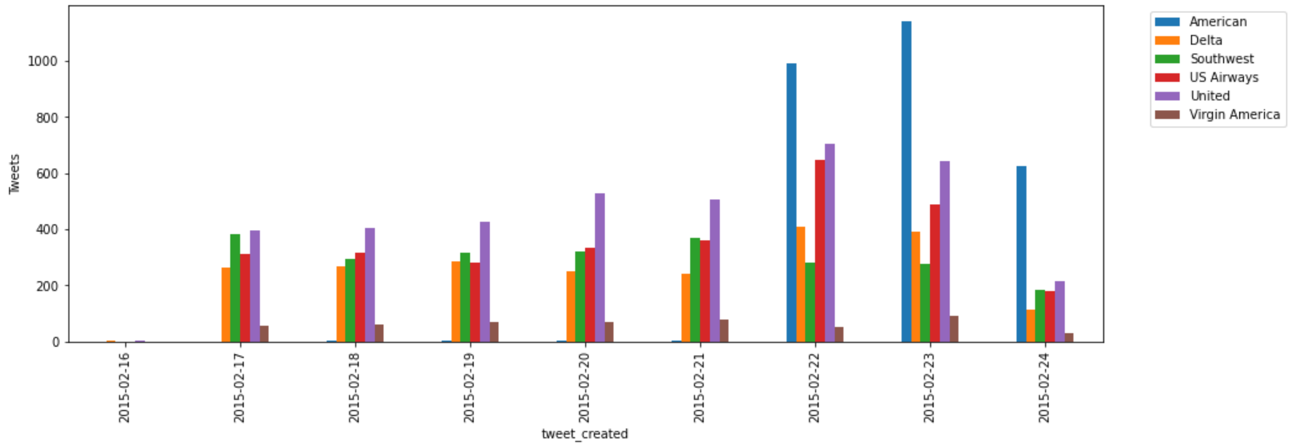
```
tweets_df.tweet_created= tweets_df.tweet_created.apply(pd.to_datetime).dt.date
```

Saved successfully!

```
created', 'airline']).airline_sentiment.count().unstack
```

```
temp_df = tweets_df.groupby(['tweet_created','airline']).airline_sentiment.count().unstack
ax1 = temp_df.plot(kind='bar', figsize = (15,5))
ax1.set_ylabel('Tweets')
plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
```

```
<matplotlib.legend.Legend at 0x7f8951181510>
```



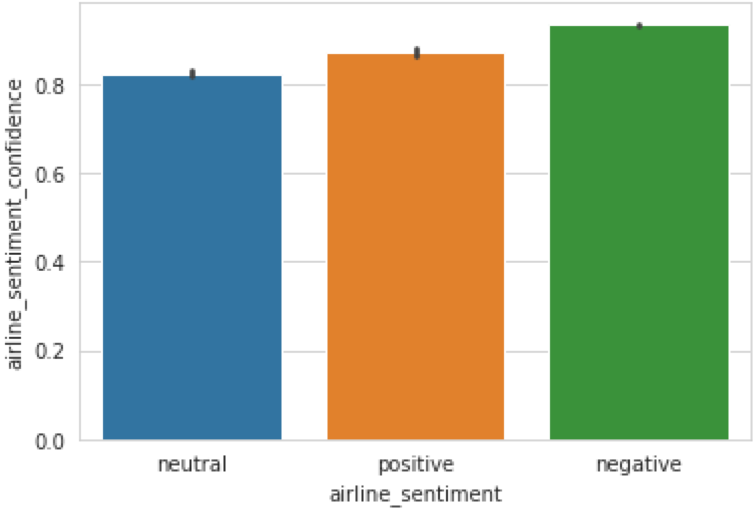
For American we have the tweets coming in from 22-02-2015 onwards

```
neg_tweet_df = tweets_df.groupby(['tweet_created','airline','airline_sentiment']).size()
neg_tweet_df = neg_tweet_df.loc(axis=0)[:,:,'negative']
ax2 = neg_tweet_df.groupby(['tweet_created','airline']).sum().unstack().plot(kind='bar', f
ax2.set_ylabel('Negative Tweets')
plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")
```

Saved successfully!

```
<matplotlib.legend.Legend at 0x7f894faa6890>
```

```
sns.set_style("whitegrid")
ax = sns.barplot(x="airline_sentiment", y="airline_sentiment_confidence", data=tweets_df)
```



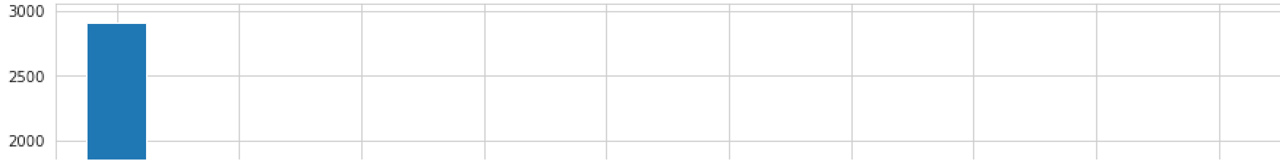

```
tweets_df.negativereason.value_counts()
```

```
Customer Service Issue         2910
Late Flight                    1665
Can't Tell                     1190
Cancelled Flight                847
Lost Luggage                    724
Bad Flight                      580
Flight Booking Problems         529
Flight Attendant Complaints     481
longlines                       178
Damaged Luggage                  74
Name: negativereason, dtype: int64
```

```
tweets_df.negativereason.value_counts().plot(kind='bar', figsize=(15,5))
```

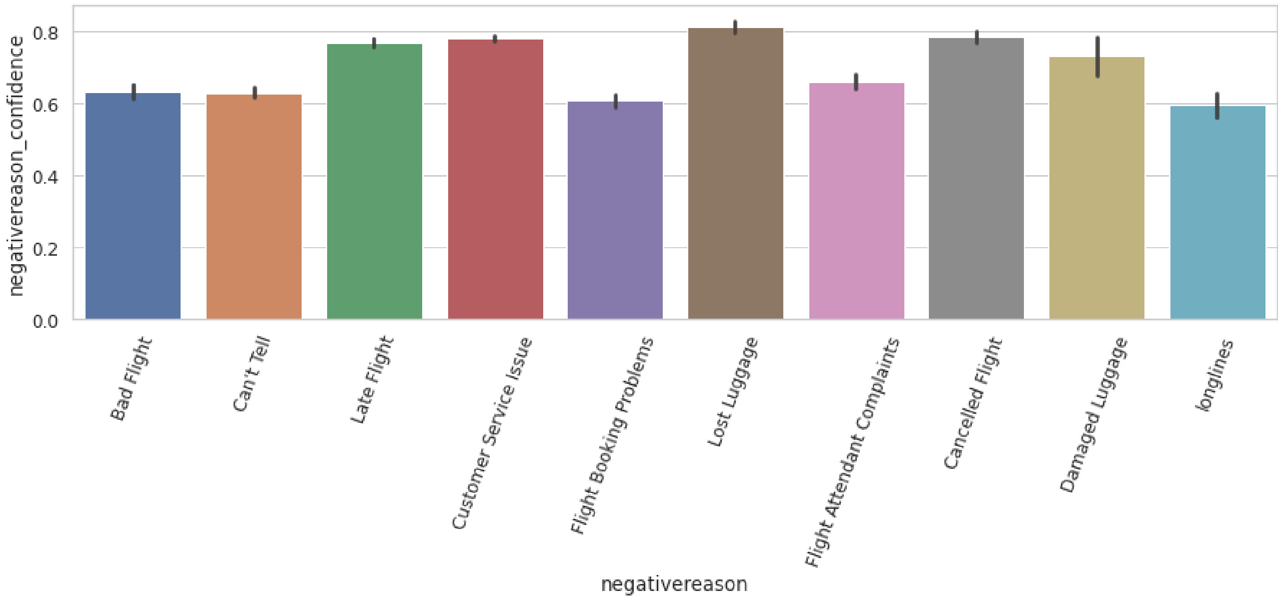
Saved successfully!

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f894f84f850>
```




```
plt.figure(figsize=(15, 4))
sns.set(font_scale = 1.1)
sns.set_style("whitegrid")
ax = sns.barplot(x="negativereason", y="negativereason_confidence", data=tweets_df)
plt.xticks(rotation=70)
```

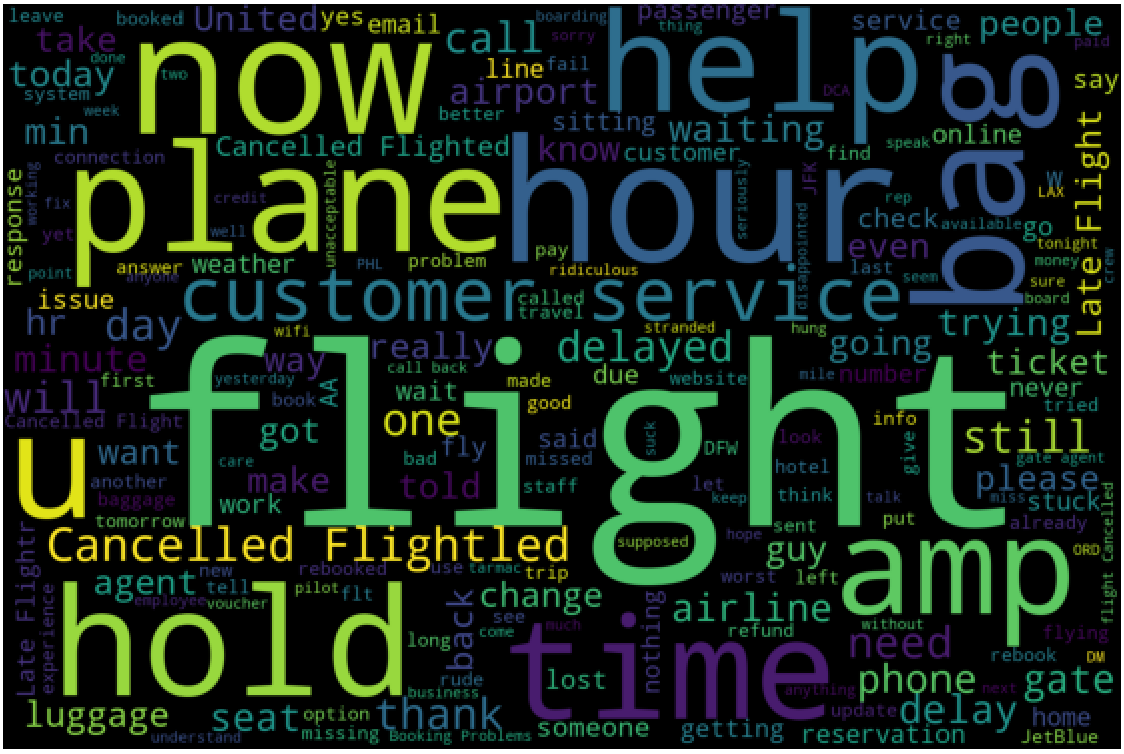
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 <a list of 10 Text major ticklabel objects>)
```




```
from wordcloud import WordCloud,STOPWORDS
def createWrdCloudForSentiment(sentiment):
    temp_df = tweets_df[tweets_df.airline_sentiment==sentiment]
    words = " ".join(temp_df.text)
    cleaned_words = " ".join([w for w in words.split()
                              if 'http' not in w
                                and not w.startswith('@')
                                and w!='RT'])

    wrdcld = WordCloud(stopwords=STOPWORDS,
                       background_color='black',
                       width=1500,
                       height=1000).generate(cleaned_words)
    plt.figure(figsize=(10,10))
    plt.imshow(wrdcld)
```

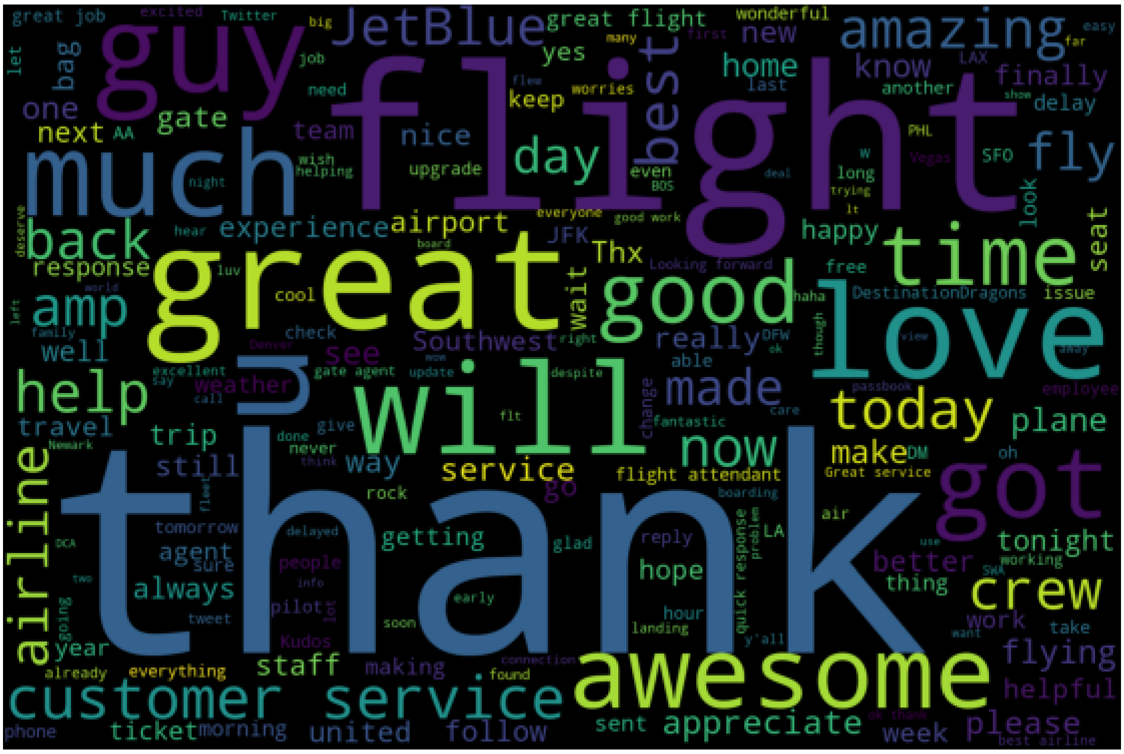
Saved successfully!

```
createWrdCloudForSentiment('negative')
```



we observe that 'flight', 'hour', 'hrlp', 'time' 'hold', 'bag', 'plane' are present more frequently in negative statements.

```
createWrdCloudForSentiment('positive')
```



Saved successfully!

we observe that 'thank', 'flight', 'great', 'will', 'awesome' 'love' are present more frequently in positve statements.

```
tweets_df.columns
```

```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline', 'name',
       'retweet_count', 'text', 'tweet_created', 'tweet_location',
       'user_timezone'],
      dtype='object')
```

1. Remove all the special characters
2. convert all letters to lower case
3. filter out english stop words
4. stemmer (optional)

```
tweets_df.text
```

```
0                      @VirginAmerica What @dhepburn said.
1        @VirginAmerica plus you've added commercials t...
2        @VirginAmerica I didn't today... Must mean I n...
3        @VirginAmerica it's really aggressive to blast...
4        @VirginAmerica and it's a really big bad thing...
                               ...
14635    @AmericanAir thank you we got on a different f...
14636    @AmericanAir leaving over 20 minutes Late Flig...
14637    @AmericanAir Please bring American Airlines to...
14638    @AmericanAir you have my money, you change my ...
14639    @AmericanAir we have 8 ppl so we need 2 know h...
Name: text, Length: 14640, dtype: object
```

```
nltk.download('stopwords')
eng_stops = set(stopwords.words("english"))
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

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

```
## We'll check latter if stemmer will make any difference
#from nltk.stem.porter import PorterStemmer
#stemmer = PorterStemmer()
#
#from nltk.stem import WordNetLemmatizer
#lemmatizer = WordNetLemmatizer()
```

```
def process_message(tweet):
    # remove all the special characters
                                             " ",tweet)
                                            case
                                             ()
```

Saved successfully!

```
    words = new_tweet.lower().split()
    # remove stop words
    words = [w for w in words if not w in eng_stops]
    # stemming
    #words = [stemmer.stem(word) for word in words]
    # lemmatizer
    #words = [lemmatizer.lemmatize(word) for word in words]
    # join all words back to text
    return (" ".join(words))
```

```
tweets_df['clean_tweet']=tweets_df['text'].apply(lambda x: process_message(x))
```

```
tweets_df['clean_tweet'].to_list()
```

```
     'virginamerica hi booked flight need add baggage',
     'virginamerica airline awesome lax loft needs step game dirty tables floors http
     'virginamerica worried great ride new plane great crew airlines like',
     'virginamerica awesome flew yall sat morning way correct bill',
     'virginamerica watch best student films country feet cmfat feet http co kek pdmg
     'virginamerica first time flying different rate policy media bags thanks',
     'virginamerica going customer service anyway speak human asap thank',
     'virginamerica happened doom',
     'virginamerica supp biz traveler like southwestair customer service like jetblue
     'virginamerica applied member inflight crew team im interested flightattendant c
     'virginamerica best whenever begrudgingly use airline delayed late flight',
     'virginamerica interesting flying cancelled flight next four flights planned nev
     'virginamerica disappointing experience shared every business traveler meet neve
     'virginamerica trouble adding flight wife booked elevate account help http co px
     'virginamerica bring reservation online using flight booking problems code',
     'virginamerica random q distribution elevate avatars bet kitty disproportionate
     'virginamerica lt flying va life happens trying change trip jperhi help va home
     'virginamerica site back',
     'virginamerica rnp yeah know',
     'virginamerica hi get points elevate account recent flight add flight points acc
     'virginamerica like tv interesting video disappointed cancelled flightled flight
     'virginamerica landed lax hour late flight bag check business travel friendly nc
     'virginamerica flight redirected',
     'virginamerica website btw new website great user experience time another redesi
     'virginamerica check add bag website working tried desktop mobile http co avyqdm
     'virginamerica let scanned passengers leave plane told someone remove bag st cla
     'virginamerica phone number find call flight reservation',
     'virginamerica anyone anything today website useless one answering phone',
     'virginamerica trying add boy prince ressie sf thursday virginamerica lax http c
     'virginamerica must traveler miss flight late flight check bag missed morning ap
     'virginamerica check new music http co marcnocwzn',
     'virginamerica direct flight fll gt sfo unexpected layover vegas fuel yet peeps
     'virginamerica late flight bag check lost business missed flight apt three peopl
     'virginamerica amazing customer service raeann sf best customerservice virginame

     'virginamerica called service line hung awesome sarcasm',
     'virginamerica site tripping trying check getting plain text version reluctant e
     'virginamerica scheduled sfo dal flight today changed th due weather looks like
     'virginamerica getaway deals may one way lots cool cities http co tzzjhuibch che
     'virginamerica getaway deals may one way lots cool cities http co rpdbpx wnd che
     'virginamerica getaway deals may one way lots cool cities http co b xi yg cheapf
     'virginamerica getaway deals may one way lots cool cities http co qdljhsloi chea
     'virginamerica great week',
                                already need take us horrible cold pleasecomeback h
                                plane needs delayed due tech stop',
                                down easy change reservation helpful representative
```

Saved successfully!

```
'virginamerica another rep kicked butt naelah represents team beautifully thank'
'virginamerica beautiful front end design right cool still book ticket b c back
'virginamerica love team running gate e las tonight waited delayed flight kept t
'virginamerica use another browser amp brand reputation built tech response cros
'virginamerica flight flight booking problems site totally folks problem',
'virginamerica like customer service min delay connecting passengers seems long
'virginamerica thanks outstanding nyc jfk crew moved mountains get home san frar
'virginamerica absolute best team customer service ever every time fly delightec
'virginamerica provide complimentary upgrades first class available seats',
'virginamerica need change flight thats scheduled hours min wait time phone im c
'virginamerica completely awesome experience last month bos las nonstop thanks a
'virginamerica watch oscars jfk gt sfo flight',
'virginamerica flight cancelled flightled'
```

Make test-train split

```
train_df, test_df = train_test_split(tweets_df, test_size=0.3, random_state=42)
```

```
train_tweets =[]
for tweet in train_df.clean_tweet:
    train_tweets.append(tweet)

test_tweets =[]
for tweet in test_df.clean_tweet:
    test_tweets.append(tweet)
```

TF-IDF

```
# bag of words model
vectorizer = TfidfVectorizer()
train_tfidf_model = vectorizer.fit_transform(train_tweets)
test_tfidf_model = vectorizer.transform(test_tweets)
```

```
# let's look at the dataframe
train_tfidf = pd.DataFrame(train_tfidf_model.toarray(), columns=vectorizer.get_feature_nam
train_tfidf
```

Saved successfully!

	aa	aaaand	aaadvantage	aaalwayslate	aadavantage	aadelay	aadv	aadvantag
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.(
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.(
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.(
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.(
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.(

```
print(vectorizer.get_feature_names())
```

```
['aa', 'aaaand', 'aaadvantage', 'aaalwayslate', 'aadavantage', 'aadelay', 'aadv', 'a
```

Now we"ll apply model to predicit sentiments from tweet text data

```
cls = [LogisticRegression(),
       MultinomialNB(),
       DecisionTreeClassifier(),
       RandomForestClassifier(n_estimators=200),
       KNeighborsClassifier(n_neighbors = 5)]

cls_name = []


lbl_actual = test_df.airline_sentiment
i = 0
accuracy = []
for cl in cls:
    model = cl.fit(train_tfidf_model,train_df.airline_sentiment)
    lbl_pred = model.predict(test_tfidf_model)
    a = (100*accuracy_score(lbl_pred, lbl_actual))
    a = round(a,2)
    accuracy.append(a)
    cls_name.append(cl.__class__.__name__)
    print ("{}  Accuracy Score : {}%".format(cls_name[i],a))
    print ( classification_report(lbl_pred, lbl_actual))
    i +=1
```

```
LogisticRegression  Accuracy Score : 79.1%
              precision    recall  f1-score   support

    negative       0.93      0.81      0.87      3232
     neutral       0.48      0.66      0.56       648
    positive       0.60      0.81      0.69       512

    accuracy                           0.79      4392
                             .76      0.71      4392
                             .79      0.80      4392
```

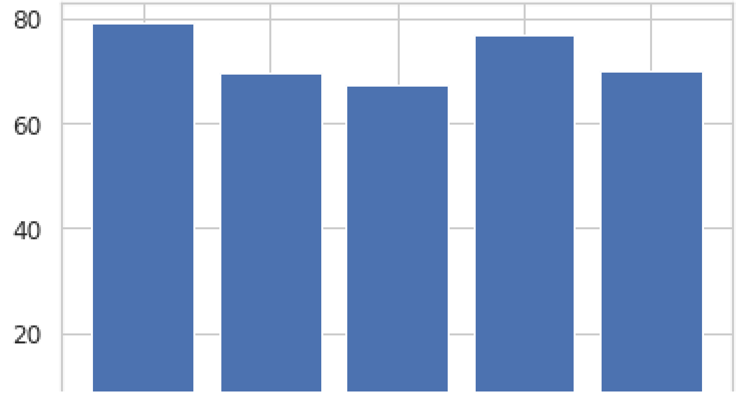
Saved successfully!

```
MultinomialNB  Accuracy Score : 69.69%
              precision    recall  f1-score   support

    negative       0.99      0.69      0.81      4081
     neutral       0.15      0.78      0.26       174
    positive       0.18      0.93      0.31       137

    accuracy                           0.70      4392
   macro avg       0.44      0.80      0.46      4392
weighted avg       0.94      0.70      0.77      4392

DecisionTreeClassifier  Accuracy Score : 67.42%
              precision    recall  f1-score   support

    negative       0.79      0.78      0.79      2841
     neutral       0.40      0.41      0.40       879
    positive       0.55      0.57      0.56       672

    accuracy                           0.67      4392
   macro avg       0.58      0.58      0.58      4392
weighted avg       0.68      0.67      0.67      4392

RandomForestClassifier  Accuracy Score : 76.78%
              precision    recall  f1-score   support

    negative       0.94      0.79      0.86      3378
     neutral       0.38      0.63      0.48       535
    positive       0.54      0.79      0.64       479

    accuracy                           0.77      4392
   macro avg       0.62      0.74      0.66      4392
weighted avg       0.83      0.77      0.79      4392

KNeighborsClassifier  Accuracy Score : 69.83%
              precision    recall  f1-score   support

    negative       0.82      0.80      0.81      2866
     neutral       0.46      0.42      0.44       960
    positive       0.53      0.65      0.58       566

    accuracy                           0.70      4392
   macro avg       0.60      0.62      0.61      4392
weighted avg       0.70      0.70      0.70      4392
```

```
plt.bar(cls_name, accuracy)
plt.xticks(rotation=70)
```

Saved successfully!

```
([0, 1, 2, 3, 4], <a list of 5 Text major ticklabel objects>)
```



Output

```
lg_model = LogisticRegression().fit(train_tfidf_model,train_df.airline_sentiment)
lg_lbl_pred = model.predict(test_tfidf_model)
```

```
lg_lbl_pred_df = pd.DataFrame({'tweet_id': test_df.tweet_id,
                               'text' : test_df.text,
                               'lg_reg' : lg_lbl_pred})
lg_lbl_pred_df.head()
```

	tweet_id	text	lg_reg
4794	569731104070115329	@SouthwestAir you're my early frontrunner for ...	positive
10480	569263373092823040	@USAirways how is it that my flt to EWR was Ca...	negative
8067	568818669024907264	@JetBlue what is going on with your BDL to DCA...	negative
8880	567775864679456768	@JetBlue do they have to depart from Washingto...	negative
8292	568526521910079488	@JetBlue I can probably find some of them. Are...	neutral

```
lg_lbl_pred_df.to_csv('sentiments.csv', index=False)
```

Saved successfully!

0s completed at 12:54 PM

Saved successfully!