**SENTIMENT ANALYSIS FOR MARKETING**

**PROJECT PHASE 3**

**INTRODUCTION:**

* The field of [sentiment analysis](https://indatalabs.com/blog/sentiment-analysis) involves looking at text to determine what opinions, emotions and attitudes it reflects. Sentiment analysis AI automates much of this process by [employing natural language processing](https://indatalabs.com/nlp-services) (NLP).
* Sentiment analysis itself predates AI and doesn’t necessarily need it to work. However, bringing NLP into the picture makes the process far easier and more reliable. As [machine learning consulting and development](https://indatalabs.com/services/machine-learning-consulting) grows, companies that don’t capitalize on these tools may quickly fall behind their more tech-centric competition.

**PROBLEM STATEMENT:**

**OBJECTIVE:**

* The given problem statement is a data of US Airline tweets and their sentiment. The task is to do sentiment analysis about the problems of each major U.S. airline. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service").

**LOADING THE DATA :**

IN [1]:

*#importing req. Lib.*

import pandas aspd

import numpy asnp

import seaborn assns

import matplotlib.pyplot asplt

importre

importnltk

from nltk.corpus importstopwords

from sklearn.model\_selection importtrain\_test\_split

from mlxtend.plotting importplot\_confusion\_matrix

from sklearn.tree importDecisionTreeClassifier

from sklearn.ensemble importRandomForestClassifier

from sklearn.metrics importaccuracy\_score,confusion\_matrix,classification\_report

IN [2]:

*#load our data set*

data = pd.read\_csv('../input/twitter-airline-sentiment/Tweets.csv')

IN [3]:

data.shape

OUT [3]:

(14640, 15)

IN [4]:

*#looking into our data*

data.head()

IN [5]:

*#checking last 5 entries*

data.tail()

IN[6]:

*#checking columns in our data*

data.columns

OUT[6]:

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')

IN[7]:

*#checking info our data*

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 14640 entries, 0 to 14639

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 tweet\_id 14640 non-null int64

1 airline\_sentiment 14640 non-null object

2 airline\_sentiment\_confidence 14640 non-null float64

3 negativereason 9178 non-null object

4 negativereason\_confidence 10522 non-null float64

5 airline 14640 non-null object

6 airline\_sentiment\_gold 40 non-null object

7 name 14640 non-null object

8 negativereason\_gold 32 non-null object

9 retweet\_count 14640 non-null int64

10 text 14640 non-null object

11 tweet\_coord 1019 non-null object

12 tweet\_created 14640 non-null object

13 tweet\_location 9907 non-null object

14 user\_timezone 9820 non-null object

dtypes: float64(2), int64(2), object(11)

memory usage: 1.7+ MB

IN[8]:

*#checking unique values*

data.nunique()

OUT[8]:

tweet\_id 14485

airline\_sentiment 3

airline\_sentiment\_confidence 1023

negativereason 10

negativereason\_confidence 1410

airline 6

airline\_sentiment\_gold 3

name 7701

negativereason\_gold 13

retweet\_count 18

text 14427

tweet\_coord 832

tweet\_created 14247

tweet\_location 3081

user\_timezone 85

dtype: int64

IN[9]:

*#checking null values in our data*

data.isnull().sum()

OUT[9]:

tweet\_id 0

airline\_sentiment 0

airline\_sentiment\_confidence 0

negativereason 5462

negativereason\_confidence 4118

airline 0

airline\_sentiment\_gold 14600

name 0

negativereason\_gold 14608

retweet\_count 0

text 0

tweet\_coord 13621

tweet\_created 0

tweet\_location 4733

user\_timezone 4820

dtype: int64

**PREPROCESSING THE DATA:**

* Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

IN[10]:

data['tweet\_created'] = pd.to\_datetime(data['tweet\_created']).dt.date

IN[11]:

data['tweet\_created'] = pd.to\_datetime(data['tweet\_created'])

IN[12]:

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 14640 entries, 0 to 14639

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 tweet\_id 14640 non-null int64

1 airline\_sentiment 14640 non-null object

2 airline\_sentiment\_confidence 14640 non-null float64

3 negativereason 9178 non-null object

4 negativereason\_confidence 10522 non-null float64

5 airline 14640 non-null object

6 airline\_sentiment\_gold 40 non-null object

7 name 14640 non-null object

8 negativereason\_gold 32 non-null object

9 retweet\_count 14640 non-null int64

10 text 14640 non-null object

11 tweet\_coord 1019 non-null object

12 tweet\_created 14640 non-null datetime64[ns]

13 tweet\_location 9907 non-null object

14 user\_timezone 9820 non-null object

dtypes: datetime64[ns](1), float64(2), int64(2), object(10)

memory usage: 1.7+ MB

In [13]:

data.head()

IN[14]:

data['tweet\_created'].min()

OUT[14]:

Timestamp('2015-02-16 00:00:00')

IN[15]:

data['tweet\_created'].max()

OUT[15]:

Timestamp('2015-02-24 00:00:00')

we have data from 16th feb 2015 to 25 feb 2015 mins we have data of 9 days.

IN[16]:

*#checking uniques values in tweet\_created columns*

data['tweet\_created'].nunique()

OUT[16]:

9

IN[17]:

numberoftweets = data.groupby('tweet\_created').size()

IN[18]:

numberoftweets.dtype

OUT[18]:

dtype('int64')

IN[19]:

numberoftweets

OUT[19]:

tweet\_created

2015-02-16 4

2015-02-17 1408

2015-02-18 1344

2015-02-19 1376

2015-02-20 1500

2015-02-21 1557

2015-02-22 3079

2015-02-23 3028

2015-02-24 1344

dtype: int64

**TREATING WITH NULL VALUES:**

IN[20]:

data.isna().sum()

OUT[20]:

weet\_id 0

airline\_sentiment 0

airline\_sentiment\_confidence 0

negativereason 5462

negativereason\_confidence 4118

airline 0

airline\_sentiment\_gold 14600

name 0

negativereason\_gold 14608

retweet\_count 0

text 0

tweet\_coord 13621

tweet\_created 0

tweet\_location 4733

user\_timezone 4820

dtype: int64

IN[21]:

print("Percentage null or na values in df")

((data.isnull() | data.isna()).sum() \*100/ data.index.size).round(2)

Percentage null or na values in df

OUT[21]:

tweet\_id 0.00

airline\_sentiment 0.00

airline\_sentiment\_confidence 0.00

negativereason 37.31

negativereason\_confidence 28.13

airline 0.00

airline\_sentiment\_gold 99.73

name 0.00

negativereason\_gold 99.78

retweet\_count 0.00

text 0.00

tweet\_coord 93.04

tweet\_created 0.00

tweet\_location 32.33

user\_timezone 32.92

dtype: float64

IN[22]:

deldata['tweet\_coord']

deldata['airline\_sentiment\_gold']

deldata['negativereason\_gold']

data.head()

IN[23]:

freq = data.groupby('negativereason').size()

IN[24]:

freq

OUT[24]:

negativereason

Bad Flight 580

Can't Tell 1190

Cancelled Flight 847

Customer Service Issue 2910

Damaged Luggage 74

Flight Attendant Complaints 481

Flight Booking Problems 529

Late Flight 1665

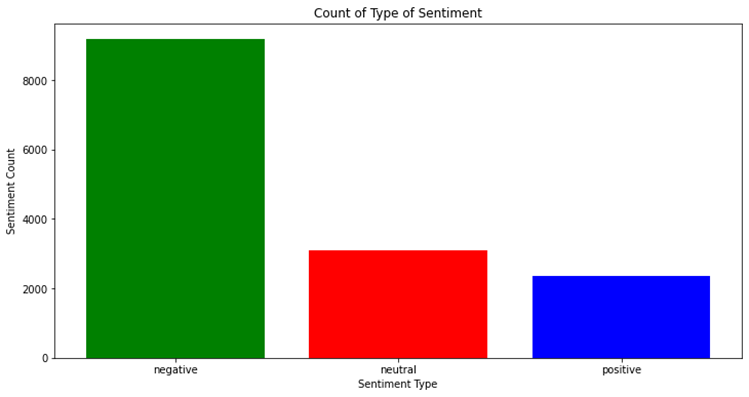
Lost Luggage 724

longlines 178

**EDA:**

* Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

**COUNT OF TYPE OF SENTIMENT:**

****

IN[25]:

counter = data.airline\_sentiment.value\_counts()

index = [1,2,3]

plt.figure(1,figsize=(12,6))

plt.bar(index,counter,color=['green','red','blue'])

plt.xticks(index,['negative','neutral','positive'],rotation=0)

plt.xlabel('Sentiment Type')

plt.ylabel('Sentiment Count')

plt.title('Count of Type of Sentiment')

OUT[25]:

Text(0.5, 1.0, 'Count of Type of Sentiment')

IN[26]:

*#checking differtent airlines we have*

data['airline'].unique()

OUT[26]:

array(['Virgin America', 'United', 'Southwest', 'Delta', 'US Airways',

'American'], dtype=object)

IN[27]:

print("Total number of tweets for each airline **\n** ",data.groupby('airline')['airline\_sentiment'].count().sort\_values(ascending=False))

airlines= ['US Airways','United','American','Southwest','Delta','Virgin America']

plt.figure(1,figsize=(12, 12))

fori **in** airlines:

indices= airlines.index(i)

plt.subplot(2,3,indices+1)

new\_df=data[data['airline']==i]

count=new\_df['airline\_sentiment'].value\_counts()

Index = [1,2,3]

plt.bar(Index,count, color=['red', 'green', 'blue'])

plt.xticks(Index,['negative','neutral','positive'])

plt.ylabel('Mood Count')

plt.xlabel('Mood')

plt.title('Count of Moods of '+i)

Total number of tweets for each airline

airline

United 3822

US Airways 2913

American 2759

Southwest 2420

Delta 2222

Virgin America 504

Name: airline\_sentiment, dtype: int64

IN[28]:

neg\_tweets = data.groupby(['airline','airline\_sentiment']).count().iloc[:,0]

total\_tweets = data.groupby(['airline'])['airline\_sentiment'].count()

my\_dict = {'American':neg\_tweets[0] / total\_tweets[0],'Delta':neg\_tweets[3] / total\_tweets[1],'Southwest': neg\_tweets[6] / total\_tweets[2],

'US Airways': neg\_tweets[9] / total\_tweets[3],'United': neg\_tweets[12] / total\_tweets[4],'Virgin': neg\_tweets[15] / total\_tweets[5]}

perc = pd.DataFrame.from\_dict(my\_dict, orient ='index')

perc.columns = ['Percent Negative']

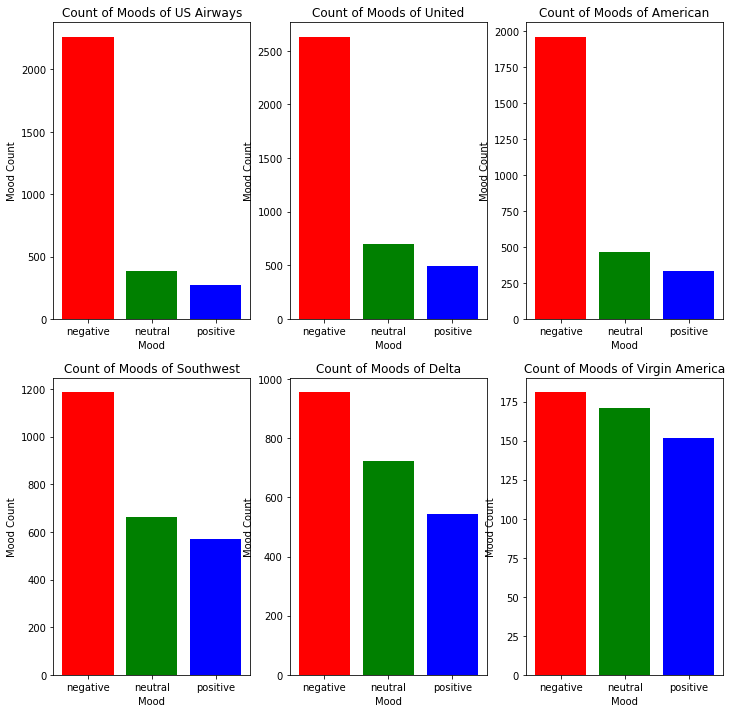
print(perc)

ax = perc.plot(kind ='bar', rot=0, colormap ='Greens\_r', figsize = (15,6))

ax.set\_xlabel('Airlines')

ax.set\_ylabel('Percentage of negative tweets')

plt.show()



Percent Negative

American 0.710402

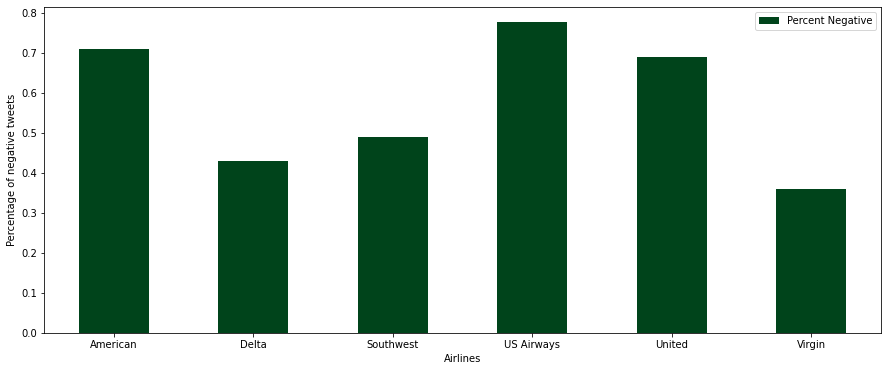
Delta 0.429793

Southwest 0.490083

US Airways 0.776862

United 0.688906

Virgin 0.359127

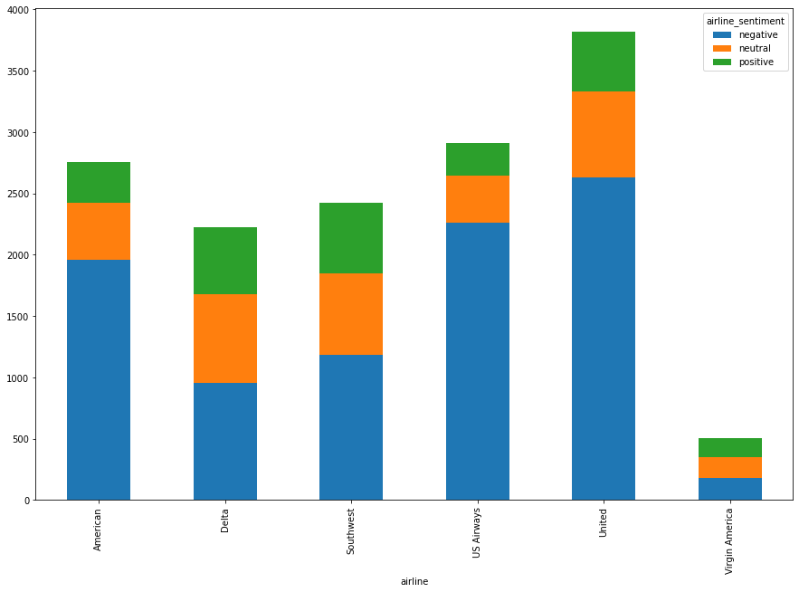


* United, US Airways, American substantially get negative reactions.
* Tweets for Virgin America are the most balanced.

IN[29]:

figure\_2 = data.groupby(['airline', 'airline\_sentiment']).size()

figure\_2.unstack().plot(kind='bar', stacked=True, figsize=(15,10))

OUT[29]:

<AxesSubplot:xlabel='airline'>

IN[30]:

print(figure\_2)

airline airline\_sentiment

American negative 1960

neutral 463

positive 336

Delta negative 955

neutral 723

positive 544

Southwest negative 1186

neutral 664

positive 570

US Airways negative 2263

neutral 381

positive 269

United negative 2633

neutral 697

positive 492

Virgin America negative 181

neutral 171

positive 152

dtype: int64

IN[31]:

negative\_reasons = data.groupby('airline')['negativereason'].value\_counts(ascending=True)

negative\_reasons.groupby(['airline','negativereason']).sum().unstack().plot(kind='bar',figsize=(22,12))

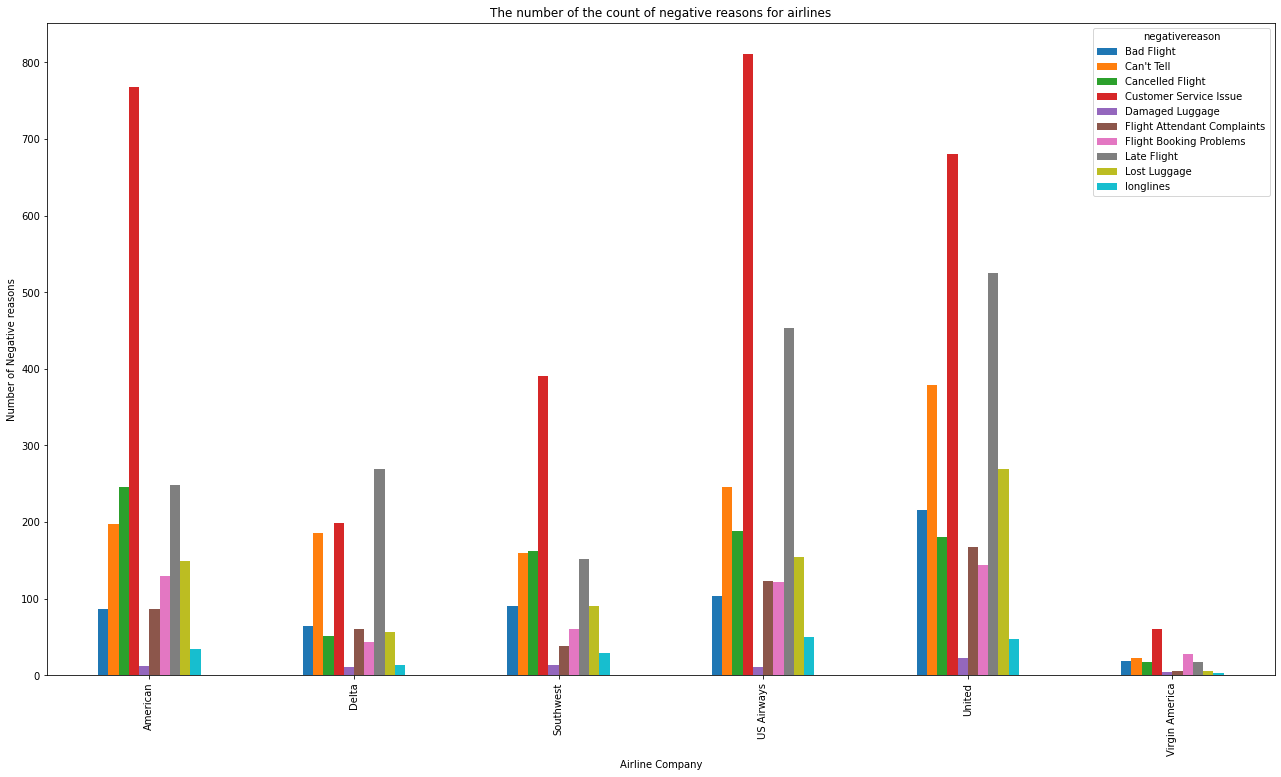
plt.xlabel('Airline Company')

plt.ylabel('Number of Negative reasons')

plt.title("The number of the count of negative reasons for airlines")

plt.show()

**Last but not least, people complain for many reasons about their flights; 10 reasons to be specific**



IN[32]:

*#get the number of negative reasons*

data['negativereason'].nunique()

NR\_Count=dict(data['negativereason'].value\_counts(sort=False))

defNR\_Count(Airline):

ifAirline=='All':

a=data

else:

a=data[data['airline']==Airline]

count=dict(a['negativereason'].value\_counts())

Unique\_reason=list(data['negativereason'].unique())

Unique\_reason=[x for x **in** Unique\_reason ifstr(x) !='nan']

Reason\_frame=pd.DataFrame({'Reasons':Unique\_reason})

Reason\_frame['count']=Reason\_frame['Reasons'].apply(lambda x: count[x])

returnReason\_frame

defplot\_reason(Airline):

a=NR\_Count(Airline)

count=a['count']

Index =range(1,(len(a)+1))

plt.bar(Index,count, color=['red','yellow','blue','green','black','brown','gray','cyan','purple','orange'])

plt.xticks(Index,a['Reasons'],rotation=90)

plt.ylabel('Count')

plt.xlabel('Reason')

plt.title('Count of Reasons for '+Airline)

plot\_reason('All')

plt.figure(2,figsize=(13, 13))

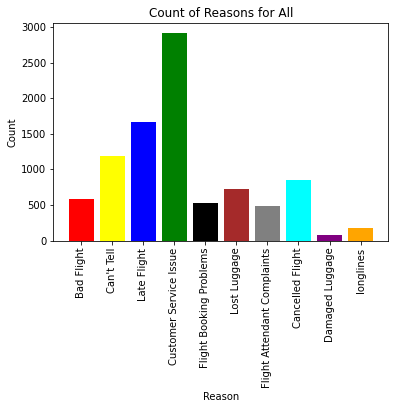
fori **in** airlines:

indices= airlines.index(i)

plt.subplot(2,3,indices+1)

plt.subplots\_adjust(hspace=0.9)

plot\_reason(i)



**Is there a relationship between negative sentiments and date?**

date = data.reset\_index()

*#convert the Date column to pandas datetime*

date.tweet\_created = pd.to\_datetime(date.tweet\_created)

*#Reduce the dates in the date column to only the date and no time stamp using the 'dt.date' method*

date.tweet\_created = date.tweet\_created.dt.date

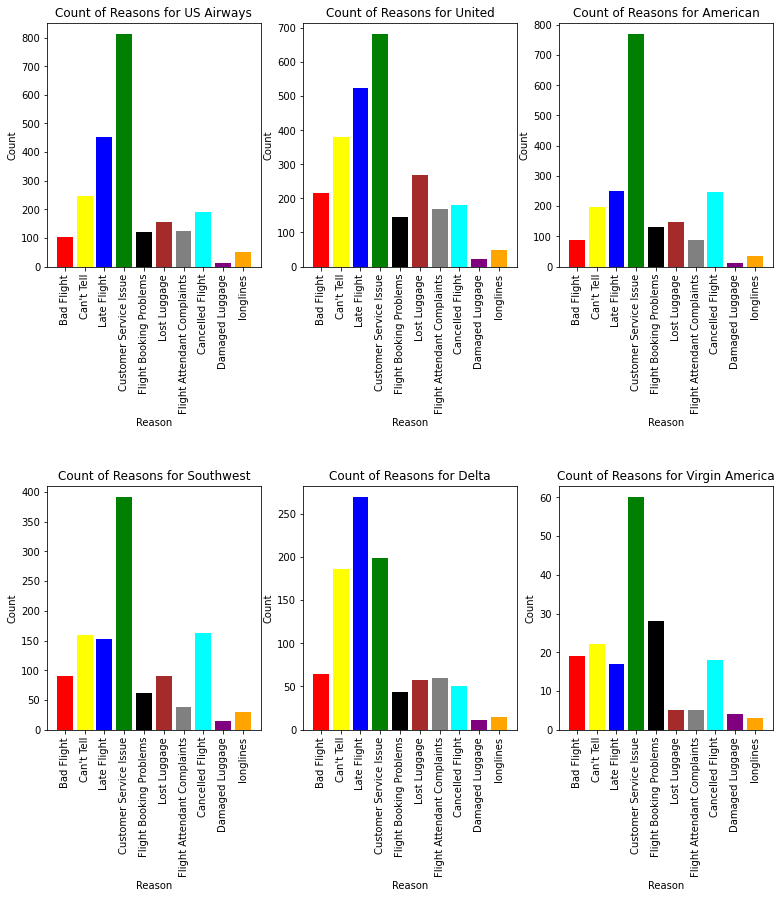
date.tweet\_created.head()

df = date

day\_df = df.groupby(['tweet\_created','airline','airline\_sentiment']).size()

*# day\_df = day\_df.reset\_index()*

day\_df



OUT[33]:

tweet\_created airline airline\_sentiment

2015-02-16 Delta negative 1

neutral 1

United negative 2

2015-02-17 Delta negative 108

neutral 86

...

2015-02-24 United neutral 49

positive 25

Virgin America negative 10

neutral 6

positive 13

Length: 136, dtype: int64

**WORKCLOUD FOR POSITIVE REASONS:**

IN[34]:

from wordcloud importWordCloud,STOPWORDS

IN[35]:

new\_df=data[data['airline\_sentiment']=='positive']

words =' '.join(new\_df['text'])

cleaned\_word =" ".join([word for word **in** words.split()

if'http'**notin**word

**andnot**word.startswith('@')

**and**word !='RT'

])

wordcloud = WordCloud(stopwords=STOPWORDS,

background\_color='black',

width=3000,

height=2500

).generate(cleaned\_word)

plt.figure(1,figsize=(12, 12))

plt.imshow(wordcloud)

plt.axis('off')

plt.show()



IN[36]:

new\_df=data[data['airline\_sentiment']=='negative']

words =' '.join(new\_df['text'])

cleaned\_word =" ".join([word for word **in** words.split()

if'http'**notin**word

**andnot**word.startswith('@')

**and**word !='RT'

])

wordcloud = WordCloud(stopwords=STOPWORDS,

background\_color='black',

width=3000,

height=2500

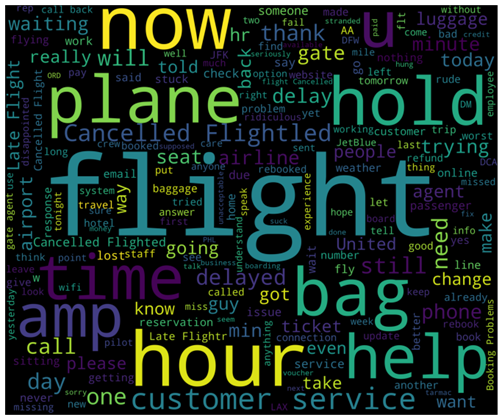
).generate(cleaned\_word)

plt.figure(1,figsize=(12, 12))

plt.imshow(wordcloud)

plt.axis('off')

plt.show()



**DROPING THE ROWS WITH SENTIMENTAL ANALYSIS:**

IN[37]:

data.drop(data.loc[data['airline\_sentiment']=='neutral'].index, inplace=True)

**LABEL ENCODING ON AIRLINE SENTIMENTS:**

IN[39]:

from sklearn.preprocessing importLabelEncoder

le = LabelEncoder()

le.fit(data['airline\_sentiment'])

data['airline\_sentiment\_encoded'] = le.transform(data['airline\_sentiment'])

data.head()

**PREPROCESSING THE TWEET TEXT DATA :**

IN[39]:

deftweet\_to\_words(tweet):

letters\_only = re.sub("[^a-zA-Z]", " ",tweet)

words = letters\_only.lower().split()

stops =set(stopwords.words("english"))

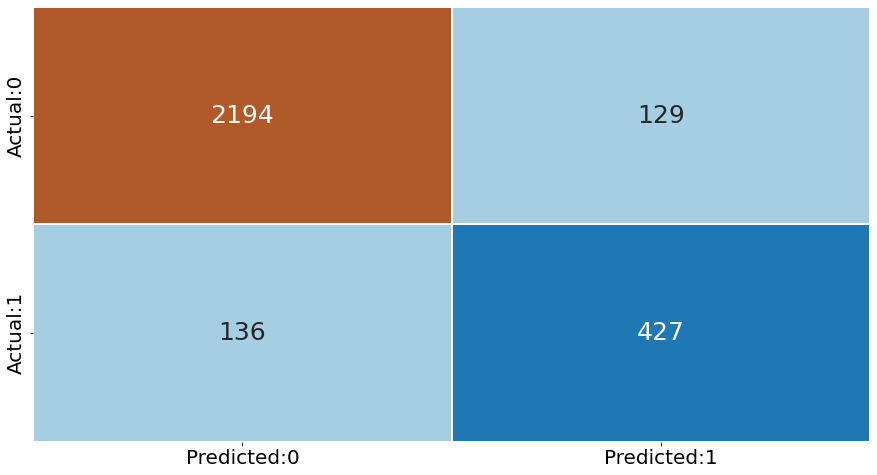
meaningful\_words = [w for w **in** words if**not** w **in** stops]

return( " ".join( meaningful\_words ))

IN[40]:

nltk.download('stopwords')

data['clean\_tweet']=data['text'].apply(lambda x: tweet\_to\_words(x))



**CONCLUSIONS:**

Twitter sentiment analysis tools are awesome because they help you to understand how people feel about a particular topic, event, or brand. Twitter is one of the most popular social media platforms and is a great place to get feedback about your product or service.