AI Project phase-3
sentiment analysis for marketing

By
JAYALAKSHMI P (513421106021)
BE(ECE)-III YEAR
University College of Engineering Kanchipuram

What is data preprocessing ?

- What Is Data Preprocessing?
- Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

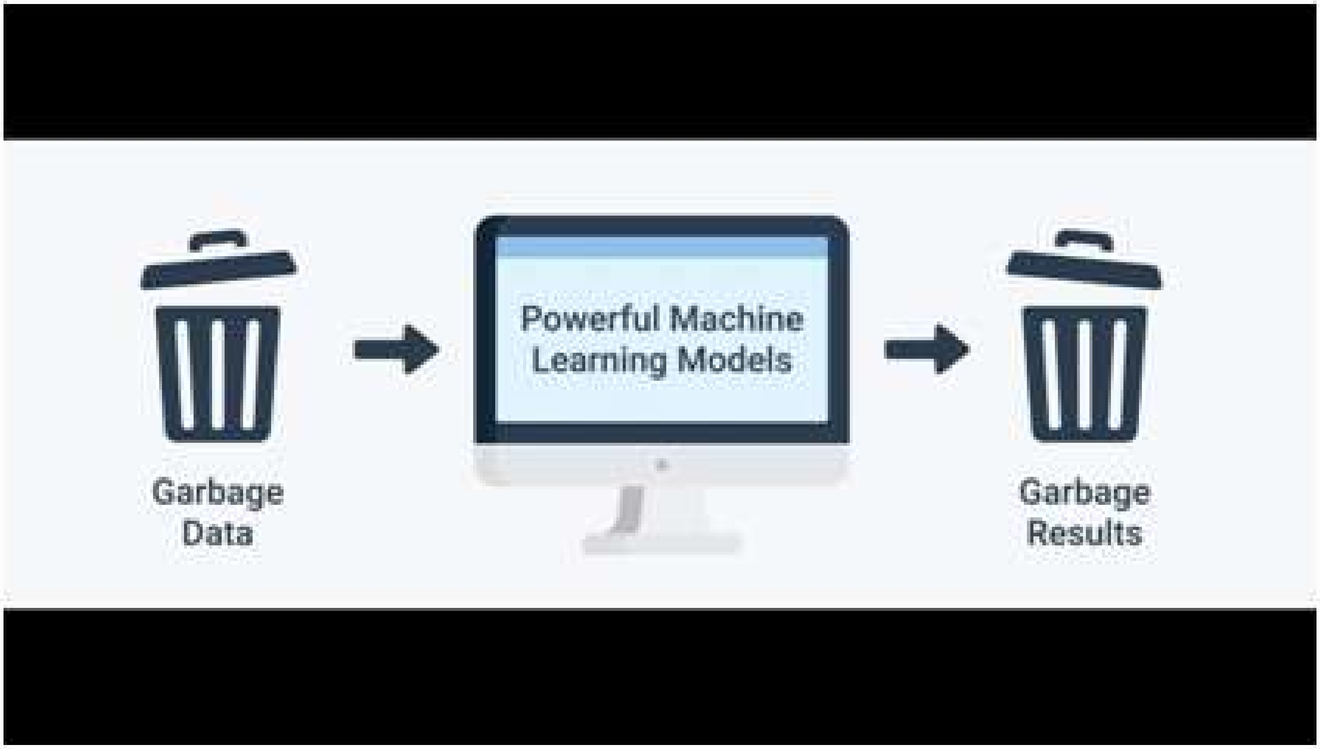
- Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn’t have a regular, uniform design.

- Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

Data preprocessing importance

- the phrase “garbage in, garbage out” This means that if you use bad or “dirty” data to train your model, you’ll end up with a bad, improperly trained model that won’t actually be relevant to your analysis.

- Good, preprocessed data is even more important than the most powerful algorithms, to the point that machine learning models trained with bad data could actually be harmful to the analysis you’re trying to do – giving you “garbage” results.data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.When using data sets to train machine learning models, you’ll often hear the phrase “garbage


Garbage
Data
Powerful Machine
Learning Models
Garbage
Results

Data Preprocessing Steps

- Data quality assessment
- Data cleaning
- Data transformation
- Data reduction

DATA QUALITY ASSESSMENT

- Take a good look at your data and get an idea of its overall quality, relevance to your project, and consistency.
- There are a number of data anomalies and inherent problems to look out for in almost any data set, for example:
- Data outliers: Outliers can have a huge impact on data analysis results. For example if you're averaging test scores for a class, and one student didn’t respond to any of the questions, their 0% could greatly skew the results.

- Missing data: Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you'll have to perform data cleaning.

Mixed data values:

Perhaps different sources use different descriptors for features – for example, man or male.

These value descriptors should all be made uniform.

2.Data cleaning

- Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set.
- Dating cleaning is the most important step of preprocessing because it will ensure that your data is ready to go for your

- Missing data
- There are a number of ways to correct for missing data, but the two most common are:

- Ignore the tuples: A tuple is an ordered list or sequence of numbers or entities.

- If multiple values are missing within tuples, you may simply discard the tuples with that missing information.

- This is only recommended for large data sets, when a few ignored tuples won’t harm further analysis.

- Manually fill in missing data: This can be tedious, but is definitely necessary when working with smaller data sets.

- If you're working with text data, for example, some things you should consider when cleaning your data are:

- Remove URLs, symbols, emojis, etc., that aren't relevant to your analysis
- Translate all text into the language you'll be working in
- Remove HTML tags
- Remove boilerplate email text
- Remove unnecessary blank text between words
- Remove duplicate data

- After data cleaning, you may realize you have insufficient data for the task at hand.
- At this point you can also perform data wrangling or data enrichment to add new data sets and
- run them through quality assessment and cleaning again before adding them to your original data.

3.Data transformation

- With data cleaning, we’ve already begun to modify our data, but data transformation will begin the process of turning the data into the proper format(s) you’ll need for analysis and other downstream processes.

- This generally happens in one or more of the below:
- Aggregation
- Normalization
- Feature selection
- Discreditization
- Concept hierarchy generation

4 .Data reduction

- The more data you're working with, the harder it will be to analyze, even after cleaning and transforming it.
- Depending on your task at hand, you may actually have more data than you need. Especially when working with text analysis, much of regular human speech is superfluous or irrelevant to the needs of the researcher.
- Data reduction not only makes the analysis easier and more accurate, but cuts down on data storage.
- It will also help identify the most important features to the process at hand.

- Attribute selection: Similar to discreditization, attribute selection can fit your data into smaller pools.
- It, essentially, combines tags or features, so that tags like male/female and professor could be combined into male professor/female professor.
- Numerosity reduction : This will help with data storage and transmission.
- You can use a regression model, for example, to use only the data and variables that are relevant to your analysis.
- Dimensionality reduction: This, again, reduces the amount of data used to help facilitate analysis and downstream processes.
- Algorithms like K-nearest neighbors use pattern recognition to combine similar data and make it more

The Wrap Up

- Good data-driven decision making requires good, prepared data.
- Once you’ve decided on the analysis you need to do and where to find the data you need,
- just follow the steps above and your data will be all set for any number of downstream processes.

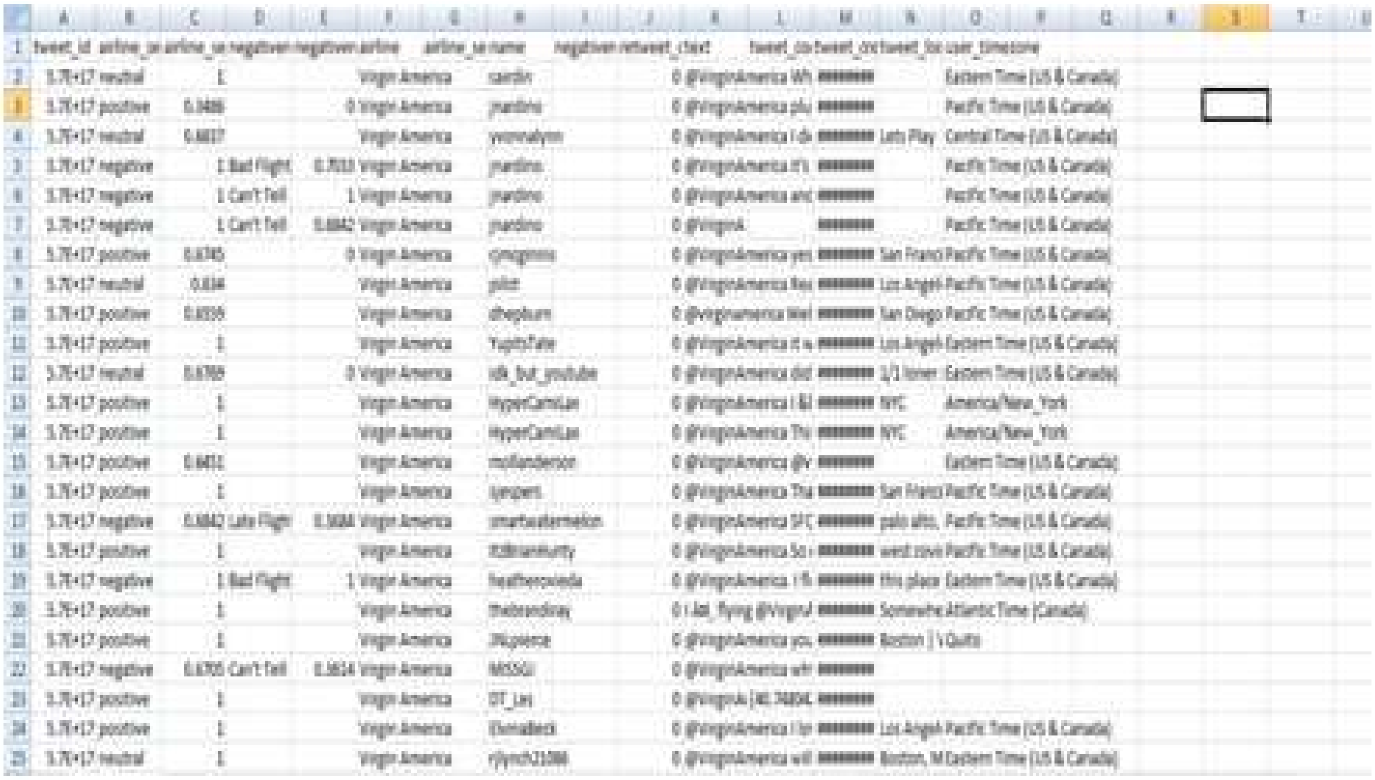
- Data preprocessing can be a tedious task, for sure,
- but once you have your methods and procedures set up, you’ll reap the benefits down the line.

Importing the libraries and loading the data

```
In[1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot  as plt
import os
print(os.listdir("../input"))
import re
import nltk
from nltk.corpus import stopwords
from sklearn.model_selection import trai _test_split
from mlxtend.plotting  import plot_confusion  matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

In[2]:
df= pd.read_csv("../input/Tweets.csv")
 df.head()
```

output

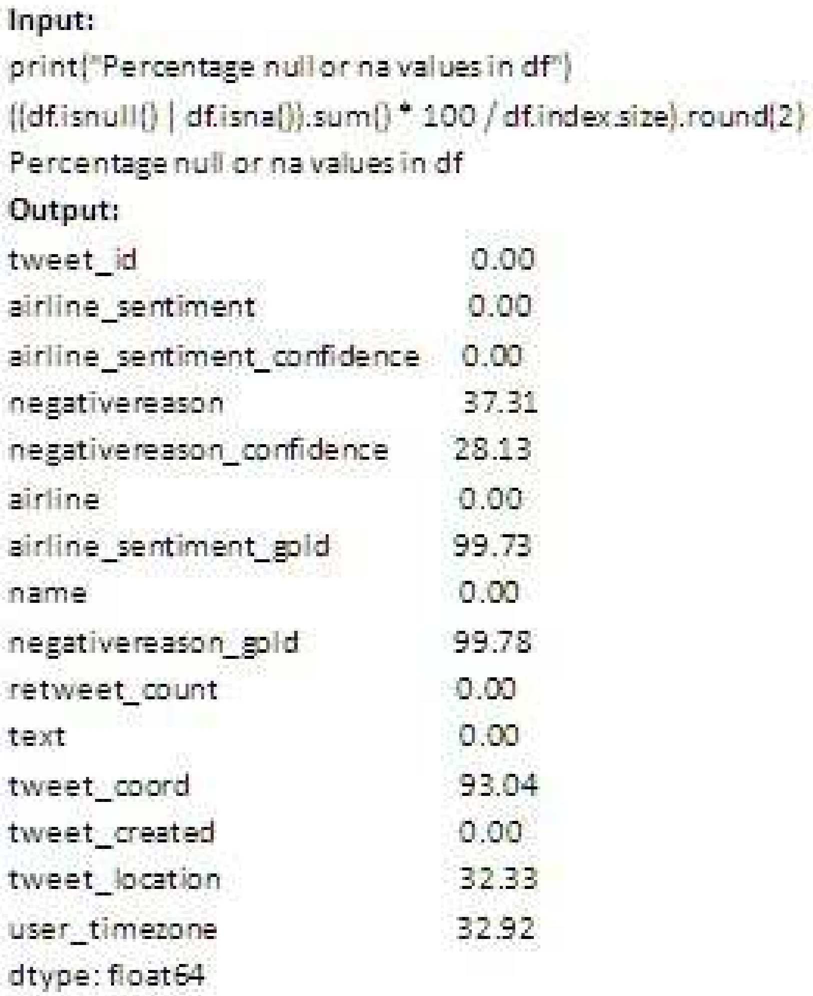


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	tweet_id	airline_se	airline_se	negativer	negativer	airline	airline_se	name	negativer	retweet_c	text	tweet_co	tweet_cre	tweet_loc	user_timezone
2	5.7E+17	neutral	1			Virgin America		cairdin		0	@VirginAmerica Wh		########		Eastern Time (US & Canada)
3	5.7E+17	positive	0.3486		0	Virgin America		jnardino		0	@VirginAmerica plu		########		Pacific Time (US & Canada)
4	5.7E+17	neutral	0.6837			Virgin America		yvonnalynn		0	@VirginAmerica I di		########	Lets Play	Central Time (US & Canada)
5	5.7E+17	negative	1	Bad Flight	0.7033	Virgin America		jnardino		0	@VirginAmerica it's		########		Pacific Time (US & Canada)
6	5.7E+17	negative	1	Can't Tell	1	Virgin America		jnardino		0	@VirginAmerica and		########		Pacific Time (US & Canada)
7	5.7E+17	negative	1	Can't Tell	0.6842	Virgin America		jnardino		0	@VirginA		########		Pacific Time (US & Canada)
8	5.7E+17	positive	0.6745		0	Virgin America		cjmcginnis		0	@VirginAmerica yes		########	San Franc	Pacific Time (US & Canada)
9	5.7E+17	neutral	0.634			Virgin America		pilot		0	@VirginAmerica Rea		########	Los Angel	Pacific Time (US & Canada)
10	5.7E+17	positive	0.6559			Virgin America		dhepburn		0	@virginamerica Wel		########	San Diego	Pacific Time (US & Canada)
11	5.7E+17	positive	1			Virgin America		YupitsTate		0	@VirginAmerica it w		########	Los Angel	Eastern Time (US & Canada)
12	5.7E+17	neutral	0.6769		0	Virgin America		idk_but_youtube		0	@VirginAmerica did		########	1/1 loner	Eastern Time (US & Canada)
13	5.7E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica I &I		########	NYC	America/New_York
14	5.7E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica Thi		########	NYC	America/New_York
15	5.7E+17	positive	0.6451			Virgin America		mollanderson		0	@VirginAmerica @v		########		Eastern Time (US & Canada)
16	5.7E+17	positive	1			Virgin America		sjespers		0	@VirginAmerica Tha		########	San Franc	Pacific Time (US & Canada)
17	5.7E+17	negative	0.6842	Late Fligh	0.3684	Virgin America		smartwatermelon		0	@VirginAmerica SFC		########	palo alto,	Pacific Time (US & Canada)
18	5.7E+17	positive	1			Virgin America		ItzBrianHunty		0	@VirginAmerica So		########	west cova	Pacific Time (US & Canada)
19	5.7E+17	negative	1	Bad Flight	1	Virgin America		heatherovieda		0	@VirginAmerica I fl		########	this place	Eastern Time (US & Canada)
20	5.7E+17	positive	1			Virgin America		thebrandiray		0	I â_flying @VirginA		########	Somewhe	Atlantic Time (Canada)
21	5.7E+17	positive	1			Virgin America		JNLpierce		0	@VirginAmerica you		########	Boston | \	Quito
22	5.7E+17	negative	0.6705	Can't Tell	0.3614	Virgin America		MISSGJ		0	@VirginAmerica wh		########		
23	5.7E+17	positive	1			Virgin America		DT_Les		0	@VirginA	[40.74804	########		
24	5.7E+17	positive	1			Virgin America		ElvinaBeck		0	@VirginAmerica I lo		########	Los Angel	Pacific Time (US & Canada)
25	5.7E+17	neutral	1			Virgin America		rjlynch21086		0	@VirginAmerica will		########	Boston, M	Eastern Time (US & Canada)

Data preprocessing

- The first step should be to check the shape of the dataframe and then check the number of null values in each column.

- In this way we can get an idea of the redundant columns in the data frame depending on which columns have the highest number of null values.



Input:

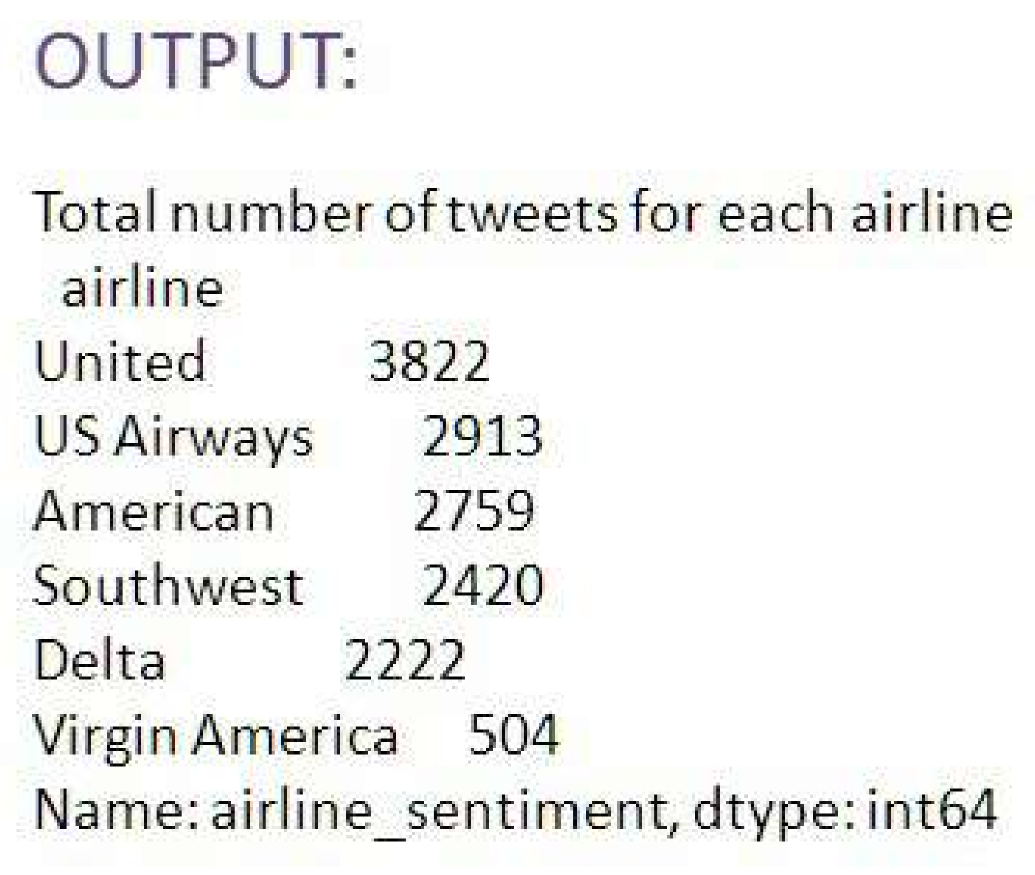
```
print("Percentage null or na values in df")
((df.isnull() | df.isna()).sum() * 100 / df.index.size).round(2)
Percentage null or na values in df
```

Output:

```
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
```

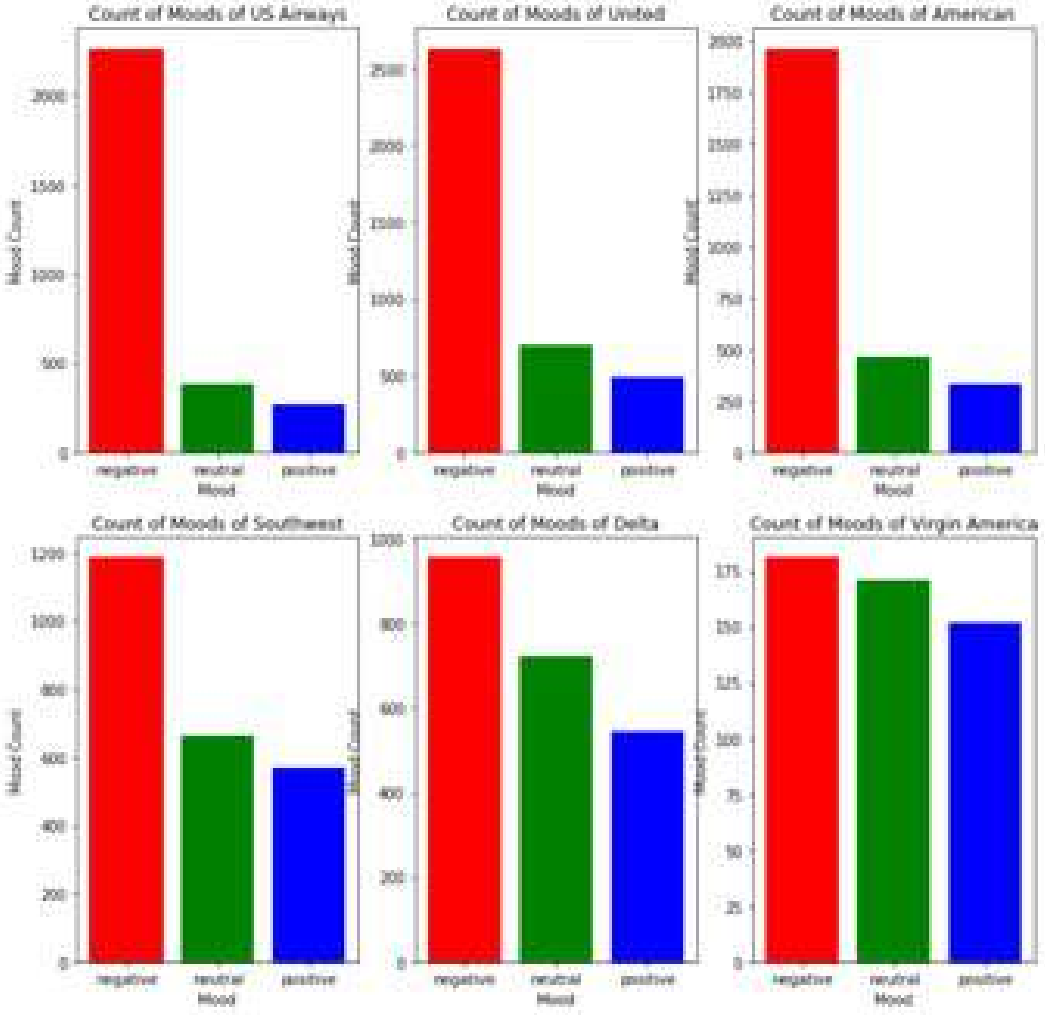
Airline sentiments for each airline

```
•   print("Total number of tweets for each airline \n
    ",df.groupby('airline')['airline_sentiment'].count().sort_values(ascending=False))
•   airlines= ['US Airways','United','American','Southwest','Delta','Virgin America']
•   plt.figure(1,figsize=(12, 12))
•   for i in airlines:
•       indices= airlines.index(i)
•       plt.subplot(2,3,indices+1)
•       new_df=df[df['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)
```



OUTPUT:

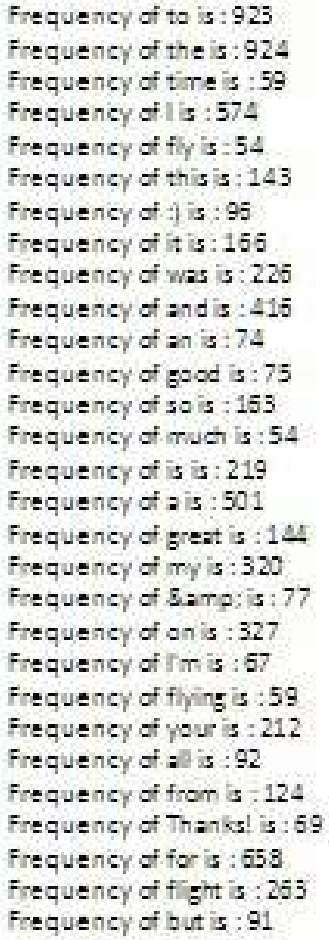
```
Total number of tweets for each airline
 airline
United          3822
US Airways       2913
American        2759
Southwest        2420
Delta          2222
Virgin America     504
Name: airline_sentiment, dtype: int64
```


Count of Moods of US Airways
Mood Count
0
500
1000
1500
2000
negative
neutral
positive
Mood
Count of Moods of United
Mood Count
0
500
1000
1500
2000
2500
negative
neutral
positive
Mood
Count of Moods of American
Mood Count
0
250
500
750
1000
1250
1500
1750
2000
negative
neutral
positive
Mood
Count of Moods of Southwest
Mood Count
0
200
400
600
800
1000
1200
negative
neutral
positive
Mood
Count of Moods of Delta
Mood Count
0
200
400
600
800
1000
negative
neutral
positive
Mood
Count of Moods of Virgin America
Mood Count
0
25
50
75
100
125
150
175
negative
neutral
positive
Mood

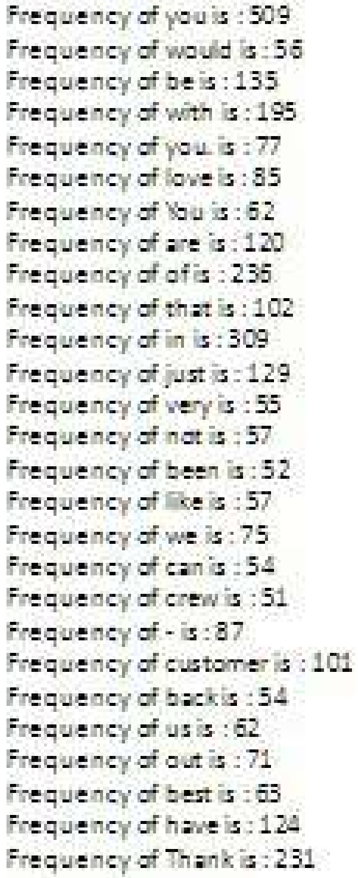
POSITIVE SENTIMENTAL TWEETS

```
•   def freq(str):
•     str = str.split()
•      str2 = []
•      for i in str:
•
•             if i not in str2:
•
•
•              str2.append(i)
•
•      for i in range(0, len(str2)):
•          if(str.count(str2[i])>50):
•              print('Frequency of', str2[i], 'is :', str.count(str2[i]))
•
•   print(freq(cleaned_word))
```

- **OUTPUT**



```
Frequency of to is : 923
Frequency of the is : 924
Frequency of time is : 59
Frequency of I is : 574
Frequency of fly is : 54
Frequency of this is : 143
Frequency of :) is : 96
Frequency of it is : 166
Frequency of was is : 226
Frequency of and is : 416
Frequency of an is : 74
Frequency of good is : 75
Frequency of so is : 163
Frequency of much is : 54
Frequency of is is : 219
Frequency of a is : 501
Frequency of great is : 144
Frequency of my is : 320
Frequency of &amp; is : 77
Frequency of on is : 327
Frequency of I'm is : 67
Frequency of flying is : 59
Frequency of your is : 212
Frequency of all is : 92
Frequency of from is : 124
Frequency of Thanks! is : 69
Frequency of for is : 658
Frequency of flight is : 263
Frequency of but is : 91
```



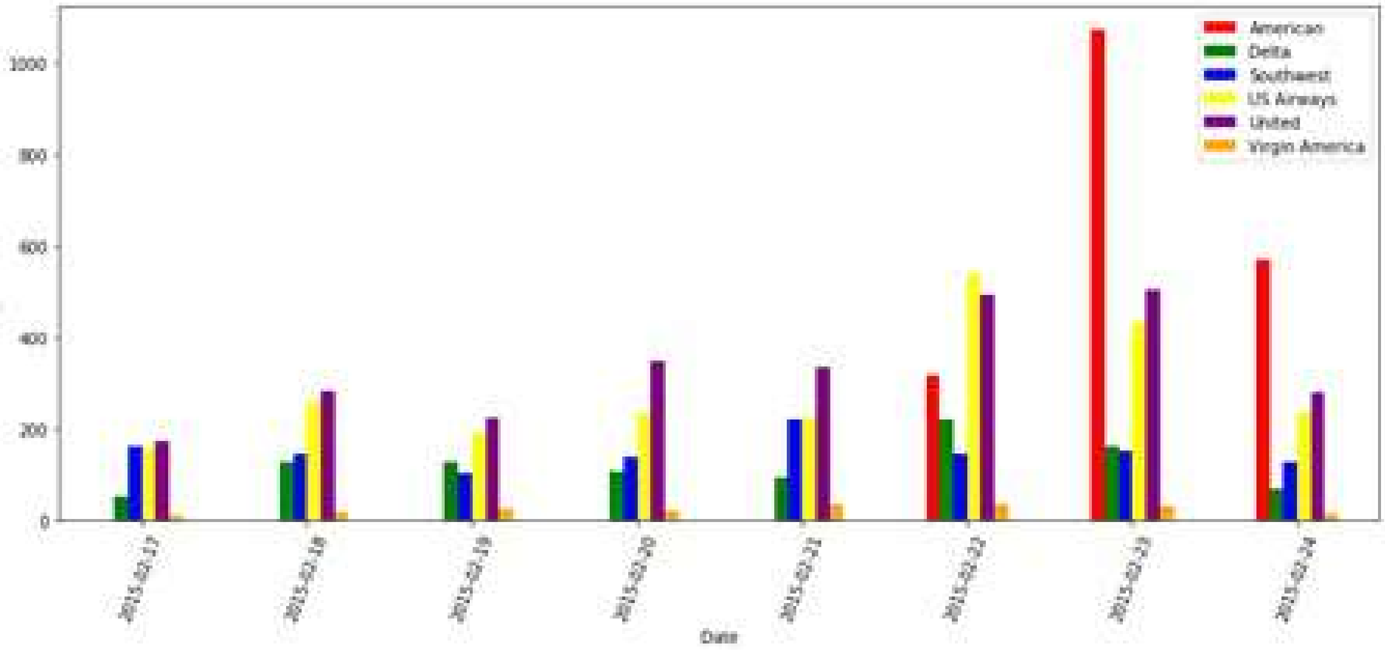
```
Frequency of you is : 509
Frequency of would is : 56
Frequency of be is : 135
Frequency of with is : 195
Frequency of you. is : 77
Frequency of love is : 85
Frequency of You is : 62
Frequency of are is : 120
Frequency of of is : 236
Frequency of that is : 102
Frequency of in is : 309
Frequency of just is : 129
Frequency of very is : 55
Frequency of not is : 57
Frequency of been is : 52
Frequency of like is : 57
Frequency of we is : 75
Frequency of can is : 54
Frequency of crew is : 51
Frequency of - is : 87
Frequency of customer is : 101
Frequency of back is : 54
Frequency of us is : 62
Frequency of out is : 71
Frequency of best is : 63
Frequency of have is : 124
Frequency of Thank is : 231
```

NEGATIVE SENTIMENTAL TWEETS

```
day_df = day_df.loc(axis=0)[:,:,'negative']

#groupby and plot data
ax2 =
day_df.groupby(['tweet_created','airline']).sum
().unstack().plot(kind = 'bar', color=['red',
'green', 'blue','yellow','purple','orange'], figsize
= (15,6), rot = 70)
labels = ['American','Delta','Southwest','US
Airways','United','Virgin America']
ax2.legend(labels = labels)
ax2.set_xlabel('Date')
ax2.set_ylabel('Negative Tweets')
plt.show()
```

OUTPUT


American
Delta
Southwest
US Airways
United
Virgin America
0
200
400
600
800
1000
2015-02-17
2015-02-18
2015-02-19
2015-02-20
2015-02-21
2015-02-22
2015-02-23
2015-02-24
Date

CONCLUSION

- This analysis can help businesses understand in which areas their products are working well and in which areas they are working poorly.
- This can help them track their brand reputation among customers.
- You can build your own Sentiment Analysis model or
- utilize various tools to implement in your organization.