Airline Twitter Sentimental Analysis using Random Forest Classification

ISDS 558 – ADVANCED SOFTWARE DEVELOPMENT WITH WEB APPLICATIONS

Cherish K Reddy

Kinjal Parikh

Gayathri Pujari

Contents

Background/Motivation

In this Generation Social media receives highest attention. Opinions about a wide variety of subjects are expressed and spread continually via numerous social Media. Twitter is one of the social Media that is gaining popularity. There are 250+ million active Twitter users tweeting about 500 million of tweets daily. There are wide variety of Twitter users. Tweets are short in length and users always share their personal experiences. Twitter offers organizations a fast and effective way to analyze customers' perspectives toward the critical to success in the market place. Developing a program for sentiment analysis is an approach to be used to computationally measure customers' perceptions. This report briefs on the design of a sentiment analysis, extracting a vast number of tweets. Results classify customers perspective via tweets into positive and negative, which is represented in a pie chart and bar graphs. This Model helps Airline companies on how they can improve Airline Services for customers and avoid negative reviews and make more Profits.

Sentiment Analysis & Its Importance

Sentiment analysis is also known as “opinion mining” or “emotion Artificial Intelligence” and alludes to the utilization of natural language processing (NLP), text mining, computational linguistics, and bio measurements to methodically recognize, extricate, evaluate, and examine emotional states and subjective information. Sentiment analysis is generally concerned with the voice in client materials; for example, surveys and reviews on the Web and web-based social networks. It’s the Process of determining whether a piece of writing is positive, negative or neutral. Identifying opinions on review if positive or negative/ satisfied or dissatisfied.

Problem statement

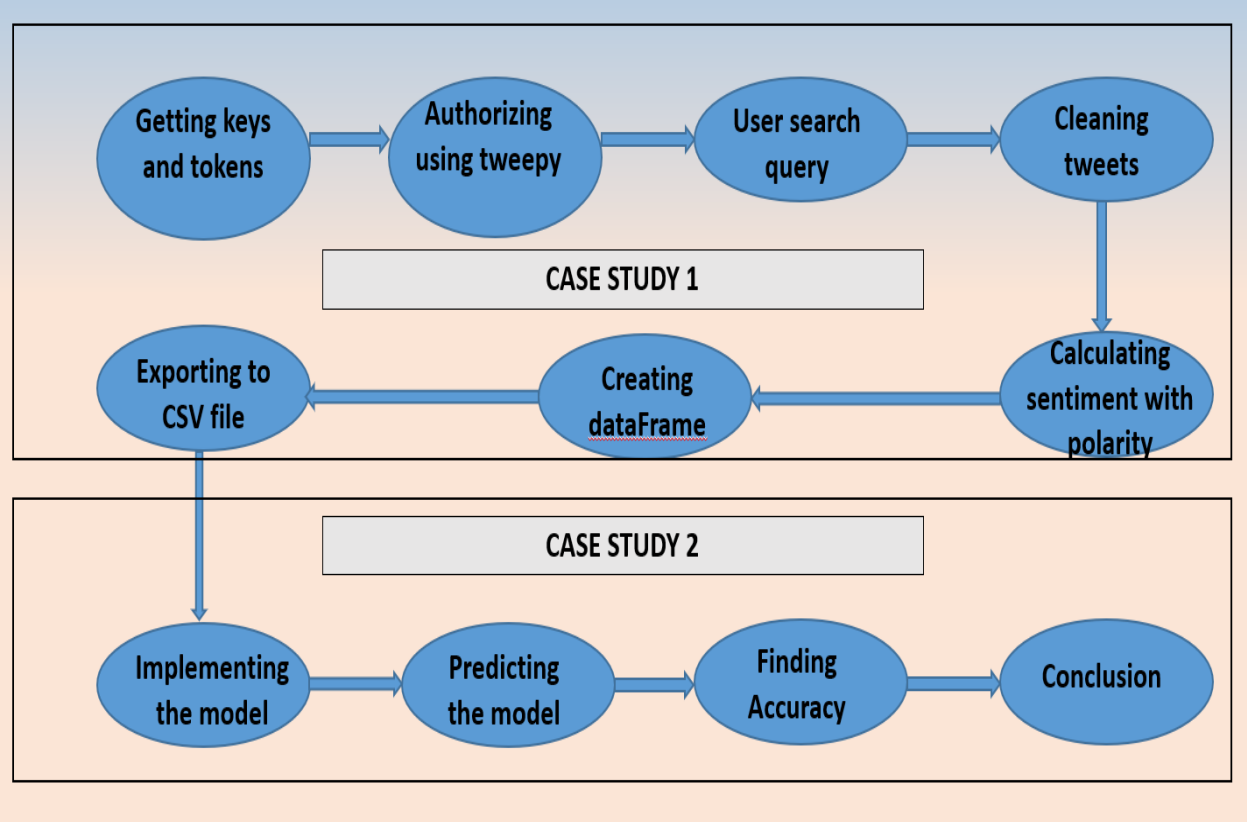
In twitter sentimental analysis, our main aim of the project is to extract features of tweets and analyses the tweets opinion for Several famous US Airlines. We are analyzing the sentiments based on polarity of a given text in a sentence or document or feature/aspect level. When a given

text is inputted, it searches for the tweet related to that input text and checks whether the expressed opinion in extracted text is positive or negative or neutral. To Identify the reasons for negative Tweets for different Airlines.

Challenges faced in this Project using Twitter data

- People expression opinions in Twitter in complex ways.
- Twitter Data is Unstructured and also non-grammatical.
- Twitter data contains Lexical variations and out of Vocabulary words.
- Twitter data contains Rhetorical devices/moods such as sarcasm, irony, implication etc.
- Twitter data contains Extensive usage of acronyms like asap, lol etc.

Overall Process Flow



Case Study 1: Fetch the Twitter Data

1. **Connect to Twitter API by requesting for the access tokens consumer_key, consumer_secret, access_token_key, access_token_secret keys.**
2. **Take the input of the user search query and gather tweets matching a particular keyword/hashtag/mention**.

```
Enter the topic name: Virgin America
IDEOU LeadingForCreativity c2 3 Inside The Creative Office Cultures At Facebook IDEO And Virgin America
Sentiment(polarity=0.5, subjectivity=1.0)
Ok i can understand that she was supposed to be the america s sweethheart the beautiful a
Sentiment(polarity=0.675, subjectivity=0.75)
LOUIE GOHME All I got to say is if you love America mamas don t let your babies grow up to go to Harvard or Stanfor
Sentiment(polarity=0.5, subjectivity=0.6)
```

3. **Clean the fetched tweets.**
4. **Creating a data frame having the column names as Polarity, Sentiment, Tweet.**
5. **Calculating the polarity of tweets having the sentiments as positive, negative and neutral**.

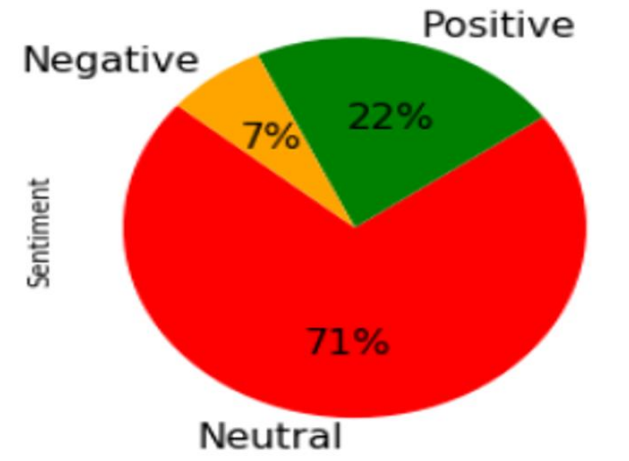
```
New = [];
for x in df['Polarity']:
    if x <0 :
        value = "Negative"
    elif (x>=0 and x<0.2):
        value = "Neutral"
    else :
        value = "Positive"

    New.append(value)


New
df['Sentiment_compare'] = New
df.head(10)
```

	Polarity	Sentiment	Tweet	Sentiment_compare
0	0.500000	Positive	IDEOU LeadingForCreativity c2 3 Inside The Cre...	Positive
1	0.675000	Positive	Ok i can understand that she was supposed to b...	Positive
2	0.500000	Positive	LOUIE GOHME All I got to say is if you love Am...	Positive
3	1.000000	Positive	We have THE BEST economy in the history of the...	Positive
4	0.216667	Positive	Virgin America Letting the loser of the popula...	Positive
5	0.216667	Positive	Virgin America Letting the loser of the popula...	Positive
6	0.000000	Neutral	America s easternmost point is in the US Virgi...	Neutral

6. **Plotting the percentage count for the sentiments as positive, negative and neutral.**



7. Generating a csv file for tweets and its sentiments

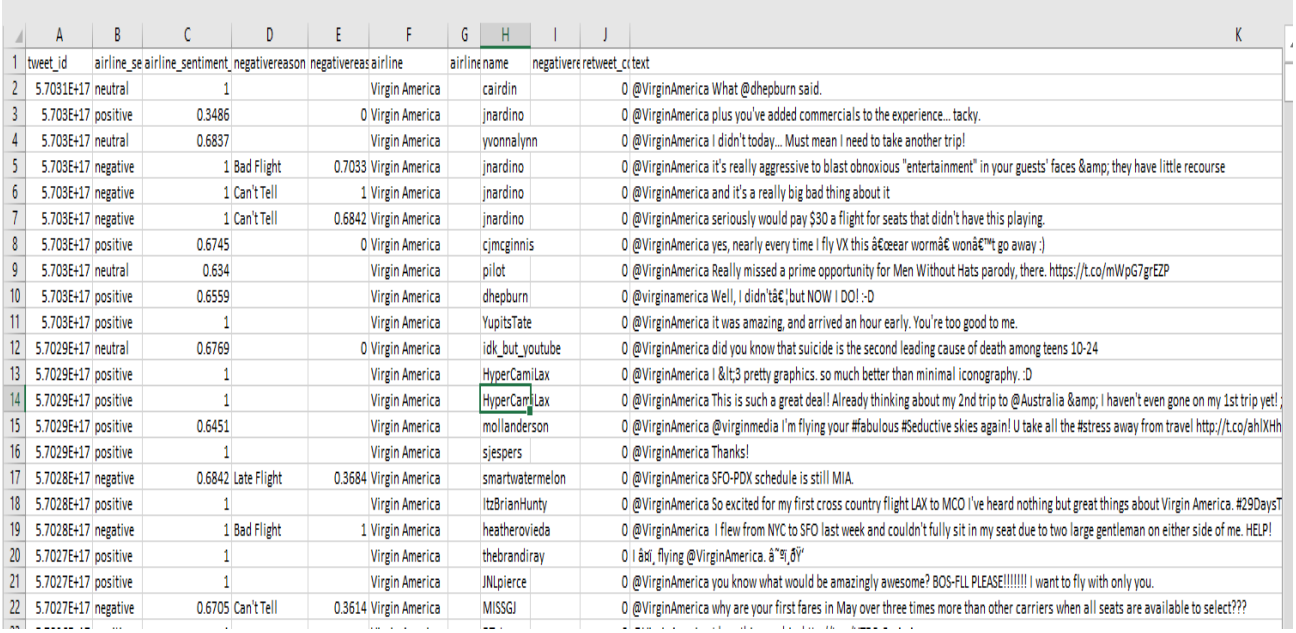
	Polarity	Sentiment	Tweet
0	0.166666667	Neutral	Jerry Jones and America in the same position the only way anything gets better is if Jerry dies
1	0.5	Positive	IDEOU LeadingForCreativity c2 3 Inside The Creative Office Cultures At Facebook IDEO And Virgin America
2	0.675	Positive	Ok i can understand that she was supposed to be the america s sweethheart the beautiful a
3	0.5	Positive	LOUIE GOHME All I got to say is if you love America mamas don t let your babies grow up to go to Harvard or Stanfor
4	1	Positive	We have THE BEST economy in the history of the United States of America This is what Democrats want to impeach
5	0.216666667	Positive	Virgin America Letting the loser of the popular vote become President Chad UK Giving the guy who won less than 35 of
6	0.216666667	Positive	Virgin America Letting the loser of the popular vote become President Chad UK Giving the guy who won less than 3
7	0	Neutral	America s easternmost point is in the US Virgin Islands St Croix is a tiny island that packs a big punch with pee
8	0	Neutral	I miss Virgin America
9	0	Neutral	I miss Virgin America
10	0	Neutral	I miss Virgin America
11	0.1	Neutral	Adulterous Love Is Common In America
12	0	Neutral	Outsider Pictures Acquires North America on Jonas Trueba s The August Virgin EXCLUSIVE
13	0.053571429	Neutral	The other man was like isn t sex different for yall Like you were a virgin before you came to America right S
14	0.16	Neutral	S2019 mcgarry S2019 Who are you The Virgin Mary Get off your High Horse amp GET O

Case Study 2: Analyzing the Sentiment data, Implementing the model, Prediction, Retrieving Confusion Matrix and Finding the accuracy of the data

1. **Fetch the tweets data from the csv file**
2. **Creating a Data Frame**
3. **Counting the number of tweets percentage for the six US Airlines**
4. **Calculating the Percentage of count for the twitter sentiments.**
5. **Calculating the Total mood counts.**
6. **Cleaning the twitter text.**
7. **Segmenting the twitter count for all the six airlines.**
8. **Implementing the model**
9. **Exploratory Data Analysis**
10. **Making Predictions on the Model**
11. **Retrieving Confusion Matrix, Classification Report and Finding the accuracy of the data**
12. **Accuracy Rate and Conclusion**

Data

The data has been made available on Local Python Repository for analyzing the Twitter sentimental analysis case study. The Tweets.csv contains 14640 rows and 15 columns.



	A	B	C	D	E	F	G	H	I	J	K
1	tweet_id	airline_se	airline_sentiment_	negativereason	negativereas	airline	airline	name	negativere	retweet_c	text
2	5.7031E+17	neutral	1			Virgin America		cairdin		0	@VirginAmerica What @dhepburn said.
3	5.703E+17	positive	0.3486		0	Virgin America		jnardino		0	@VirginAmerica plus you've added commercials to the experience... tacky.
4	5.703E+17	neutral	0.6837			Virgin America		yvonnalynn		0	@VirginAmerica I didn't today... Must mean I need to take another trip!
5	5.703E+17	negative	1	Bad Flight	0.7033	Virgin America		jnardino		0	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse
6	5.703E+17	negative	1	Can't Tell	1	Virgin America		jnardino		0	@VirginAmerica and it's a really big bad thing about it
7	5.703E+17	negative	1	Can't Tell	0.6842	Virgin America		jnardino		0	@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing.
8	5.703E+17	positive	0.6745		0	Virgin America		cjmcginnis		0	@VirginAmerica yes, nearly every time I fly VX this â€œear wormâ€™t go away :)
9	5.703E+17	neutral	0.634			Virgin America		pilot		0	@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https://t.co/mWpG7grEZP
10	5.703E+17	positive	0.6559			Virgin America		dhepburn		0	@virginamerica Well, I didn't…but NOW I DO! :-D
11	5.703E+17	positive	1			Virgin America		YupitsTate		0	@VirginAmerica it was amazing, and arrived an hour early. You're too good to me.
12	5.7029E+17	neutral	0.6769		0	Virgin America		idk_but_youtube		0	@VirginAmerica did you know that suicide is the second leading cause of death among teens 10-24
13	5.7029E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica I <3 pretty graphics. so much better than minimal iconography. :D
14	5.7029E+17	positive	1			Virgin America		HyperCamiLax		0	@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet!
15	5.7029E+17	positive	0.6451			Virgin America		mollanderson		0	@VirginAmerica @virginmedia I'm flying your #fabulous #Seductive skies again! U take all the #stress away from travel http://t.co/ahlXHh
16	5.7029E+17	positive	1			Virgin America		sjespers		0	@VirginAmerica Thanks!
17	5.7028E+17	negative	0.6842	Late Flight	0.3684	Virgin America		smartwatermelon		0	@VirginAmerica SFO-PDX schedule is still MIA.
18	5.7028E+17	positive	1			Virgin America		ItzBrianHunty		0	@VirginAmerica So excited for my first cross country flight LAX to MCO I've heard nothing but great things about Virgin America. #29DaysT
19	5.7028E+17	negative	1	Bad Flight	1	Virgin America		heatherovieda		0	@VirginAmerica I flew from NYC to SFO last week and couldn't fully sit in my seat due to two large gentleman on either side of me. HELP!
20	5.7027E+17	positive	1			Virgin America		thebrandiray		0	I âx¦, flying @VirginAmerica. â˜¤¡ðŸ'
21	5.7027E+17	positive	1			Virgin America		JNLpierce		0	@VirginAmerica you know what would be amazingly awesome? BOS-FLL PLEASE!!!!!!! I want to fly with only you.
22	5.7027E+17	negative	0.6705	Can't Tell	0.3614	Virgin America		MISSGJ		0	@VirginAmerica why are your first fares in May over three times more than other carriers when all seats are available to select???

The columns in the file are Twitter_id, airline_sentiment, airline_sentiment_confidence, negativereason , negativereason_confidence, , airline, airline_sentiment_gold, name, negativereason_gold, text , retweet_count,tweet_coord, tweet_created, tweet_location, user_timezone.

Data Transformation

All the data we used for our project is from a single CSV file. We extracted the data from the file and loaded it into a Pandas data frame using the function read_csv().

```
Twitter_Tweet = pandas.read_csv("C:/Users/CSUFTitan/Downloads/Tweets.csv")
Twitter_Tweet.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	NaN	cairdin
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	NaN	jnardino
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino

Data Quality Assessment and Data Cleaning:

Data cleaning an important process in implementing the model on any data set. According to IBM Data Analytics, we can expect to spend up to 80% of our time in cleaning data alone. We performed most of the data cleaning tasks using Pandas library. For data cleaning, our focus will be revolving around checking for inconsistent formatting, extreme outliers, missing/null values and inconsistent data types.

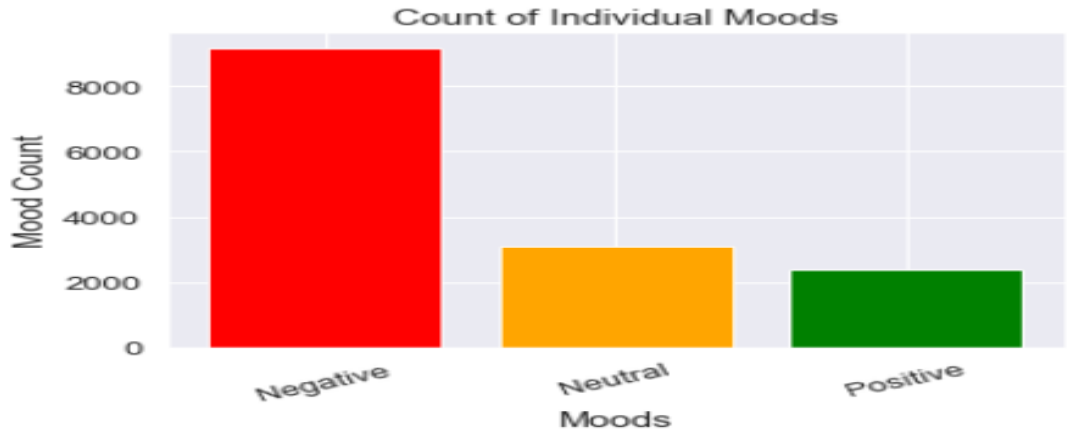
Data cleaning and Repair

During the process of cleaning the data we dealt with Missing values and Inconsistent Data types and Formats. we performed the below activities in cleaning the data. The first step in our approach towards dealing with the missing values is to find the number of missing values in each column by using the Pandas function isnull() and sum(). Then we calculated the proportion of missing values out of the all the data points available, to gain a clear idea of the severity of the amount of missing values.We used the Pandas function dtypes() and nunique() to know the data type and the unique values of all the features. we did not find any inconsistent data types in our data set.

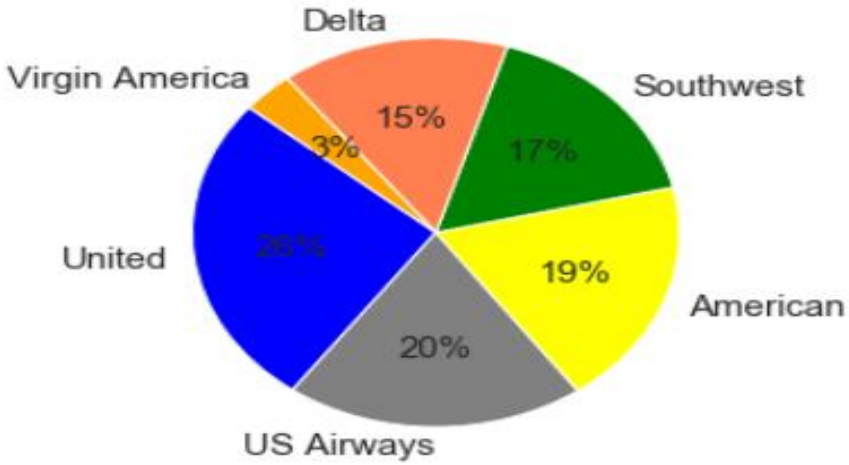
- Deleting specific columns.
- Removing all the special characters.
- Deleting Duplicates
- Remove single characters from the start
- Substituting multiple spaces with single space
- Eliminating words prefixed with 'b'
- Converting specific words to Lowercase

Analyzing our data

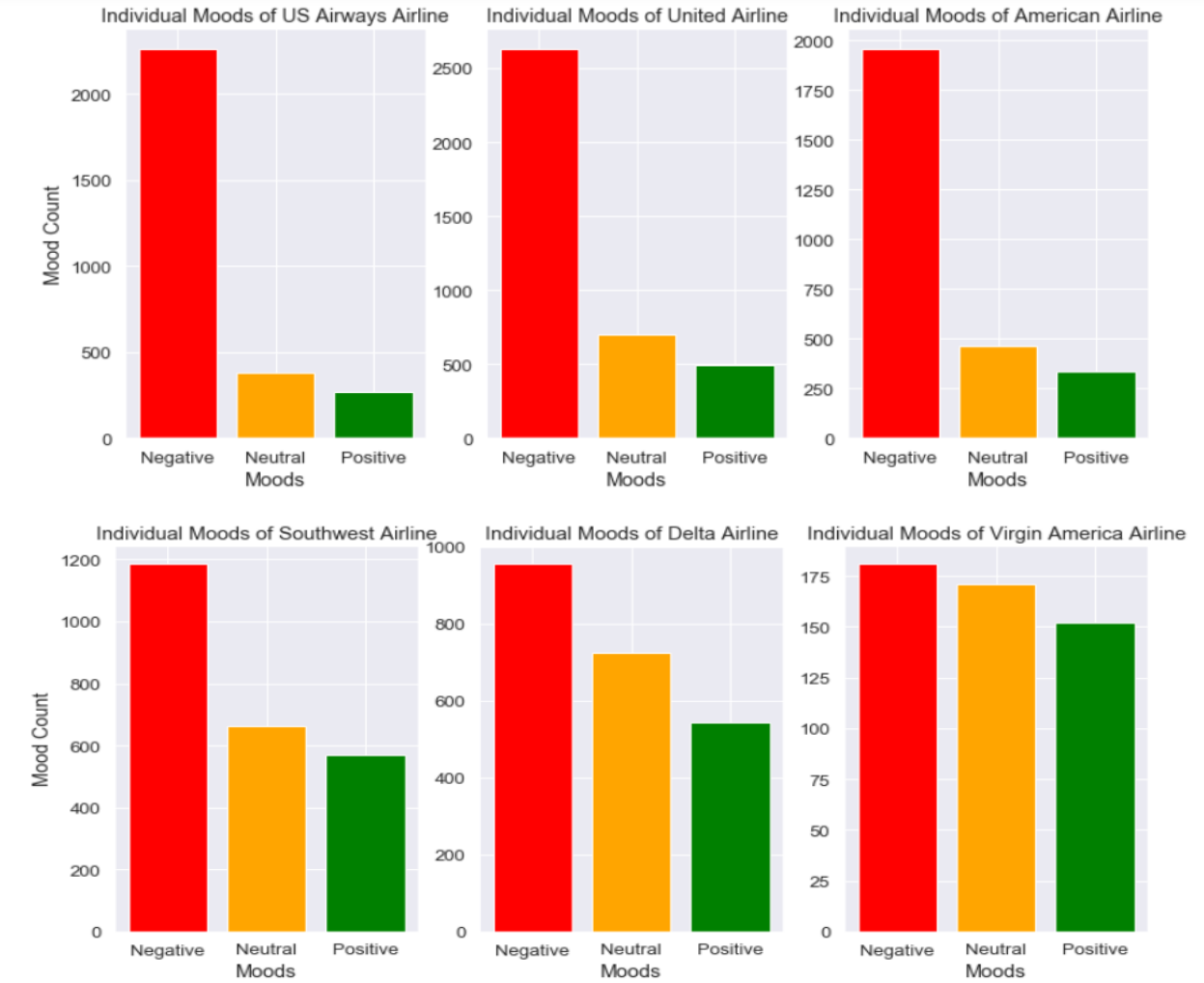
Calculating the mood counts



Percentage of Tweets for the six US_Airlines



Analyzing the Sentiment data

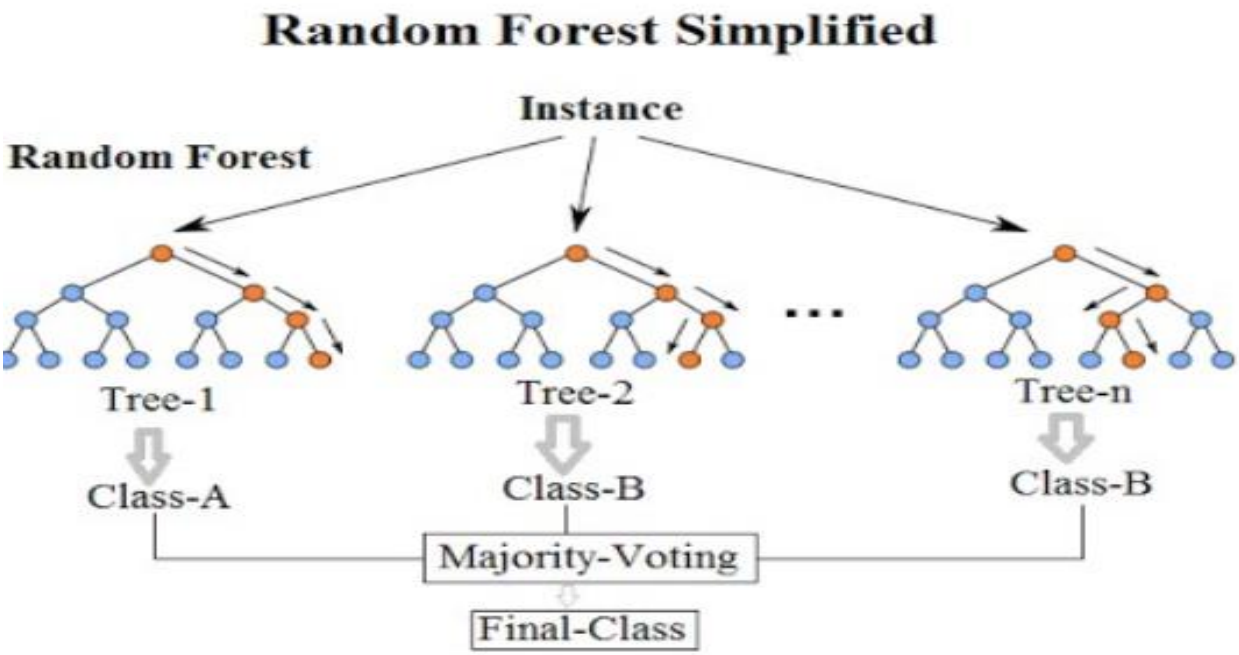


- From the plots one can find that the distribution of moods for the first three airlines- US, United and American are always skewed toward negative moods.
- On contrary, the moods are distributed more balanced with the later three airline companies – Southwest, Delta, Virgin America.

Featurization Algorithms

Understanding Random Forest Algorithm

The random forest is a classification algorithm consisting of many decisions' trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

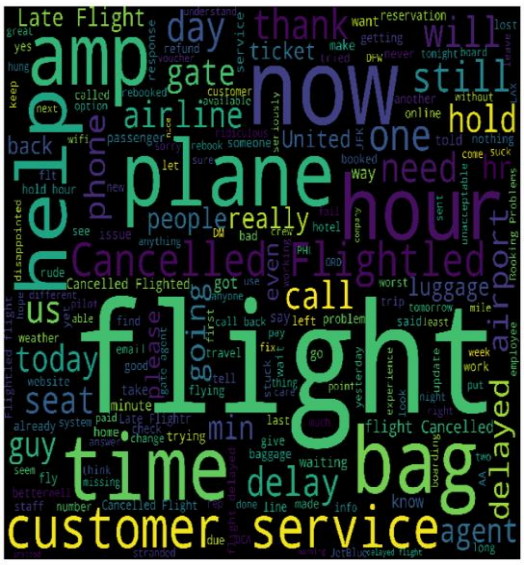


A decision tree is a Machine Learning algorithm capable of fitting complex datasets and performing both classification and regression tasks. The idea behind a tree is to search for a pair of variable-value within the training set and split it in such a way that will generate the "best" two child subsets. The goal is to create branches and leaves based on an optimal splitting criterion, a process called tree growing. Specifically, at every branch or node, a conditional statement classifies the data point based on a fixed threshold in a specific variable, therefore splitting the data. To make predictions, every new instance starts in the root node (top of the tree) and moves along the branches until it reaches a leaf node where no further branching is possible.

Advantages of Random Forest Classifier

- The same random forest algorithm or the random forest classifier can use for both classification and the regression task.
- Random forest classifier will handle the missing values.
- When we have more trees in the forest, random forest classifier won't overfit the model.
- Can model the random forest classifier for categorical values also.

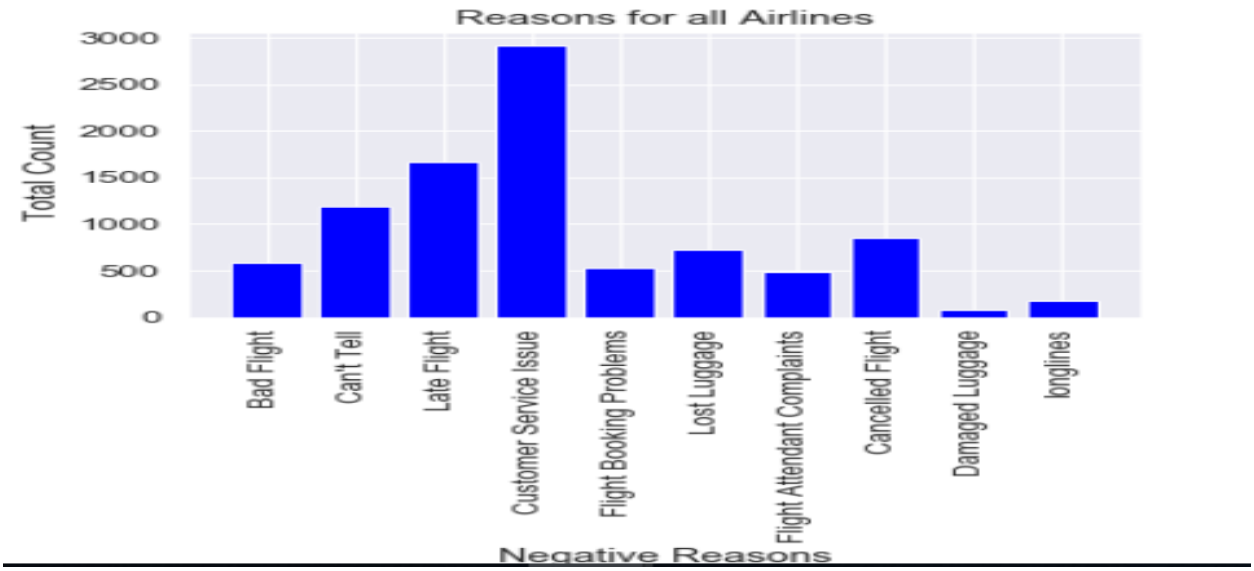
Implementing the model



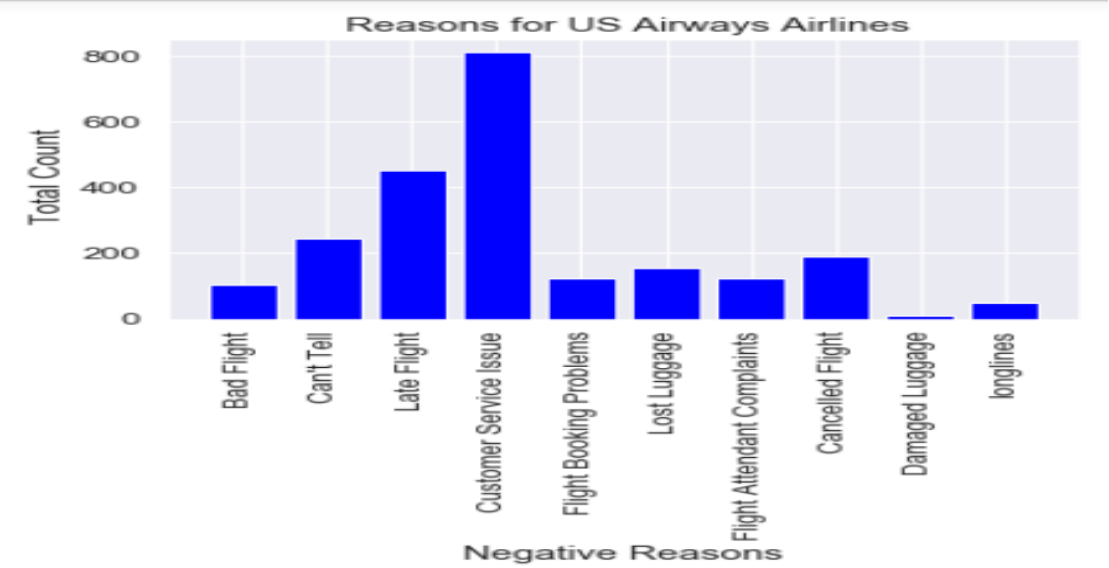
Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites. We can find that the Tweets with negative moods are frequently involved some words like cancelled, late flight, customer or service. People might guess that customer tends to complain when they are waiting for the delayed flights.

Exploratory Data Analysis

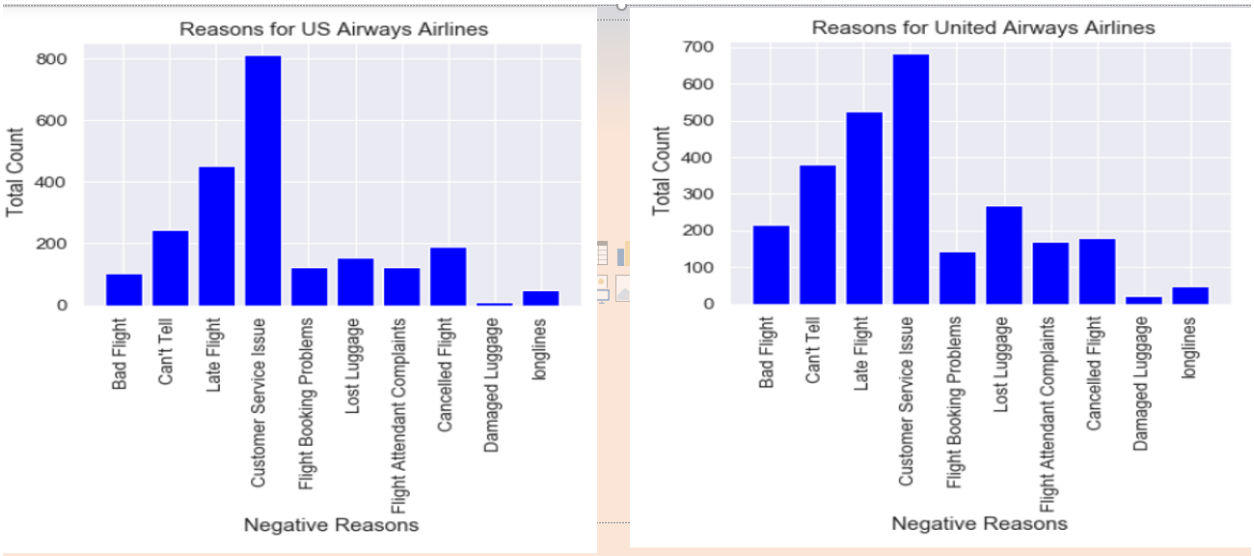
Negative Reason Count for all the airlines



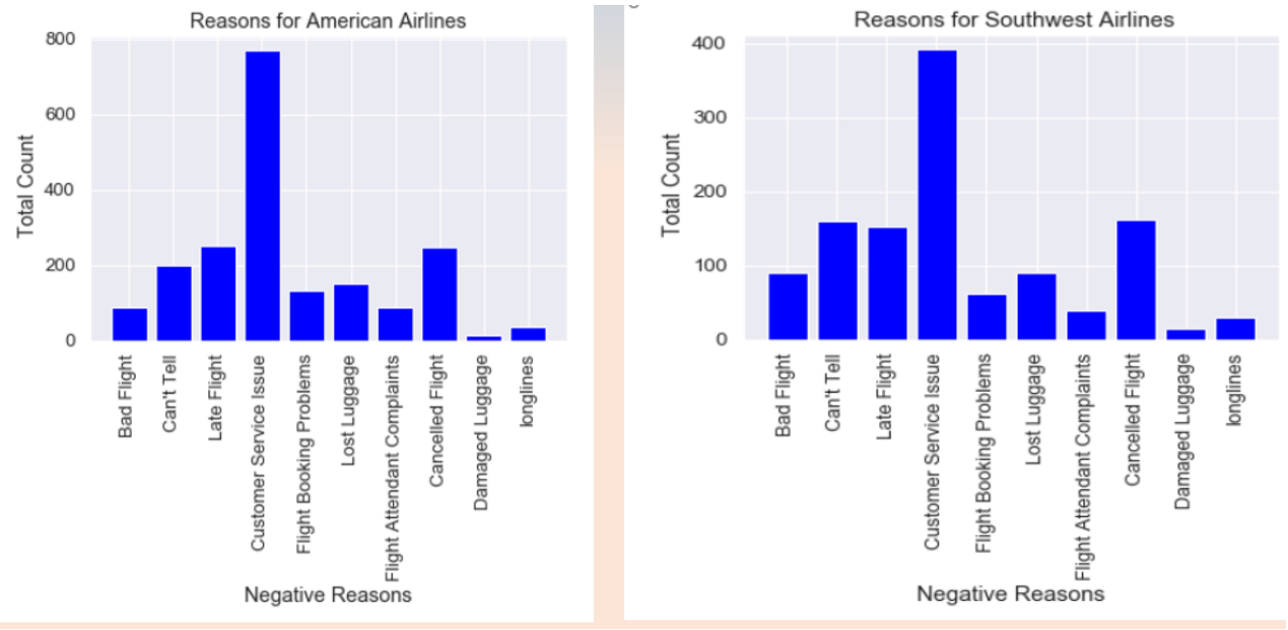
Negative Reason Count for Individual Airlines



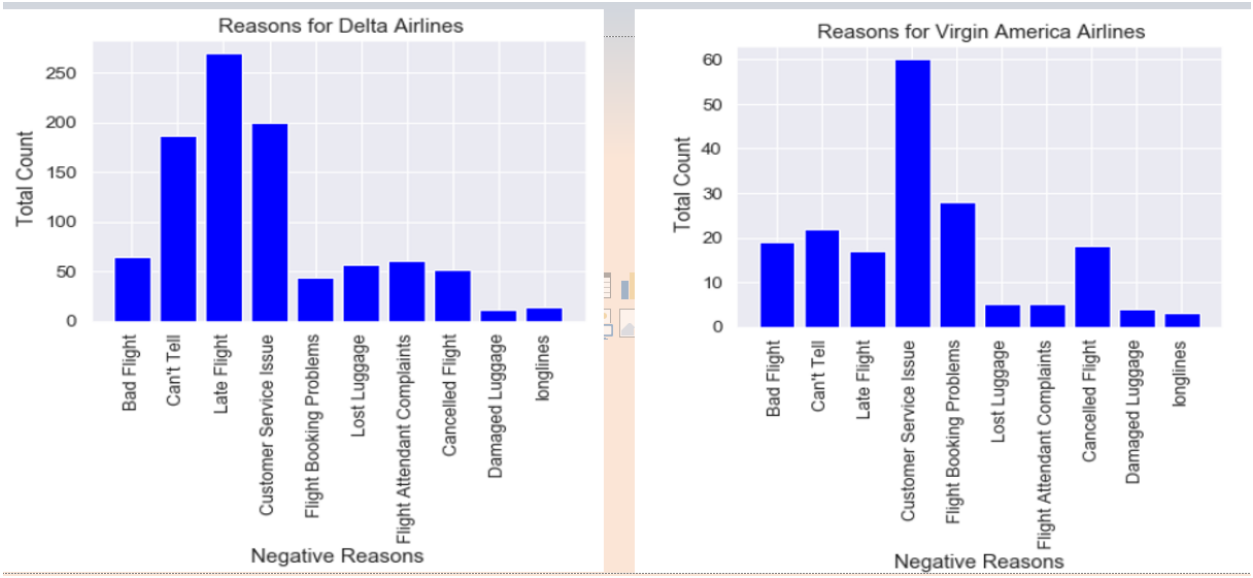
Negative Reason Count for US and United Airway Airlines



Negative Reason Count for America and Southwest Airlines



Negative Reason Count for Delta and Virgin America Airlines



Implementation of Model

We Divided our data into training and testing sets. We used the 80% dataset for training and 20% dataset for testing. We used Random Forest algorithm, owing to its ability to act upon non-normalized data and good accuracy rate results.

We trained our data to random forest algorithm to solve the classification problem. After training we evaluated the performance of the algorithm.

```
from sklearn.ensemble import RandomForestClassifier

text_classifier = RandomForestClassifier(n_estimators=200, random_state=0)
text_classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

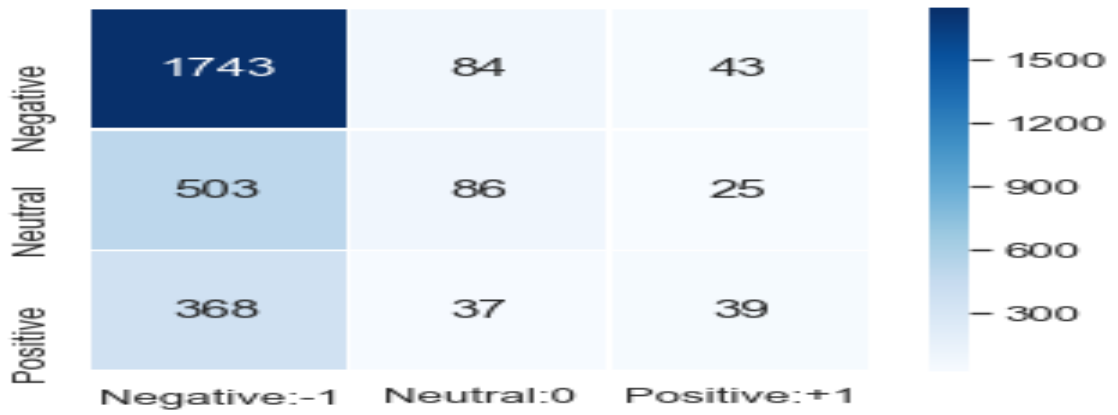
```
predictions = text_classifier.predict(X_test)
```

Confusion Matrix and Finding the accuracy of the data

Confusion Matrix

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
confusion_matrix(y_test,predictions)
```

```
array([[1743,   84,   43],
       [ 503,   86,   25],
       [ 368,   37,   39]], dtype=int64)
```



Confusion Matrix

Classification Report

The classification report for our model is as follows

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

    negative       0.67      0.93      0.78      1870
     neutral       0.42      0.14      0.21       614
    positive       0.36      0.09      0.14       444

    accuracy                           0.64      2928
   macro avg       0.48      0.39      0.38      2928
weighted avg       0.57      0.64      0.56      2928
```

Accuracy and Results

```
print(accuracy_score(y_test, predictions))
```

```
0.6379781420765027
```

From the output, you can see that our algorithm achieved an accuracy of 63.79%. We performed an analysis of public tweets regarding 6 US airlines and achieved an accuracy of around 64%.

Accuracy of the model is 64%

Some Methods That Didn't Work or Perform Optimally

We did apply KNN The k-nearest neighbors (KNN) algorithm and SVM Support vector machine algorithms and tested the accuracy but we got the highest accuracy rate with Random Forest Classifier.

Recommendation

Several Ways to improve Airline services

- Steps to prevent delay
    - Fly on point to point carriers, flying non-stop routes.
    - Scheduling flights in morning where time is not clashing with another airlines schedule.
- To improve customer service
    - Improve quality of food served.
    - Hire Potential and Polite staff willing to assist customers.
    - Ensure better communication well in advance.
    - Help families with small children

Conclusion

Features

- Twitter is a social media tool that many would like to express their view over use of any product or services, be it good or bad, sarcasm or serious topic discussion.
- It is one of the best places to know how your product or services are being received by the consumers.
- We can predict from the tweets the performance of your product or services and find different ways to improve and provide a overall customer satisfaction.

Data and Model

- Our algorithm achieved an accuracy of 63.79%. We performed an analysis of public tweets regarding six US airlines and achieved an accuracy of around 64%.
- From the tweet's visualization, we see that most of the airlines are affected by people complaining more on late/Cancelled flights and customer service issues which airlines need to work upon.

Tools

Jupyter Notebooks: The Jupyter notebook is an open source web application that is accessed by Anaconda Navigator. This web application allows us to create and share documents that include live codes, explanatory narratives, visualizations and texts. Jupyter notebooks are generally used

for data cleaning, transformations, machine learning and many more. Jupyter Notebooks supports over 40 languages, including R, Python, Julia, Scala etc. In this data analysis, we are using Jupyter Notebooks through Python.

Python: Our aim is to analyze the data using four major libraries i.e. Pandas, Numpy, Matplotlib, Sklearn and seaborn. Pandas stand for Python Data Analysis Library and it is used to analyzing, wrangling of data. Numpy is a package which we used for the scientific computing of data. Numpy provides high performance multidimensional array and basic functions to manipulate these arrays. Matplotlib is a library which is used to create 2D graphs and plots to provide visualization of the data. Seaborn is an advanced version of matplotlib and is used for the high-level interface of drawing attractive and informative statistical graphics. Lastly, Sklearn is a library which is used for machine learning and statistical modeling including clustering, regression. This library is used to build models

Anaconda Navigator: Anaconda Navigator is a desktop graphical user interface (GUI) that allows you to launch applications and allows you to access conda packages, environments and channels without using the command-line commands. Anaconda Navigator allows us to launch many applications like JupyterLab, Jupyter Notebook, Spyder, VSCode, Orange 3 App, Rodeo and many more.

Contributions by team members

As a team of three, all the members hold the responsibility of completing the work which was assigned in equal proportion. Searching for the topic, finding data online, coming up with ideas, writing the code, making the report and PPT was done by all the team members with mutual consent. Cherish: While writing the code, cherish completed writing the explanatory analysis of the data and performed some data transformations by replacing the missing values. Gayathri wrote the data explanatory analysis part in the report and completed the same while making of the PPT. Kinjal took care of writing the code for performing Random Forest and wrote about the same in the report and the PPT as well. While checking the accuracy of the model, all of us implemented the data on our individual laptops which helped us understanding the code in a much better way. After implementing we found out the same answer and there, we finalized the code. Overall, all the team member cross verified, corrected, supported and helped each other at every point. It was a joint effort to make this project a success.

Code

Case Study 1

```
import tweepy
from textblob import TextBlob
import csv
```

```
import re
import sys
import pandas as pd

consumer_key='wBFd523O3wvF4aAhFf9PaM0Kh'
consumer_secret='9DLi9J1vQg0QNvg23GvfAmMe74CVSbqMZpXrq7mBMDwMWs0rAY'

access_token_key='1192370641787219968-e3MqqNokpR2SI8g1aLyiH2sMqeC7jT'
access_token_secret='Qm4vzPQasKSdM1lhkvqI8ty5vUXxiJCpRjz095kc95FnM'

auth=tweepy.OAuthHandler(consumer_key,consumer_secret)
auth.set_access_token(access_token_key,access_token_secret)

api=tweepy.API(auth)
topic_name=input("Enter the topic name: ")
pubic_tweets=api.search(topic_name,count=1000)
unwanted_words=['@','RT',':','https','http']
symbols=['@','#']
data=[]
for tweet in pubic_tweets:
    text=tweet.text
    textWords=text.split()
    #print (textWords)
    cleaning_Tweet=' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|(RT)", " ", text).split())
    print (cleaning_Tweet)
    #print (TextBlob(cleanedTweet).tags)
    analysis= TextBlob(cleaning_Tweet)
    print (analysis.sentiment)
    polarity = 'Positive'
    if(analysis.sentiment.polarity < 0):
        polarity = 'Negative'
    if(0<=analysis.sentiment.polarity <=0.2):
        polarity = 'Neutral'
    #print (polarity)
    dic={}
    dic['Sentiment']=polarity
    dic['Tweet']=cleaning_Tweet
    dic['Polarity']=analysis.sentiment.polarity
    data.append(dic)
df=pd.DataFrame(data)
df
#df.to_csv('Virgin America.csv')

New = [];
for x in df['Polarity']:
    if x <0 :
        value = "Negative"
    elif (x>=0 and x<0.2):
        value = "Neutral"
    else :
        value = "Positive"

    New.append(value)
```

```
New
df['Sentiment_compare'] = New
df.head(10)
df.Sentiment.value_counts().plot(kind='pie', autopct='%1.0f%%', colors=["orange", "green", "gold"])

#V0 = df[(df['Polarity']>0) & (df['Polarity']<0.2)]
V0 =
Neutral = V0.groupby('Polarity', as_index=False).agg({"Sentiment":"count"})
Neutral

V1 = df[(df['Polarity']>=0.2)]
Positive = V1.groupby('Polarity', as_index=False).agg({"Tweet":"count"})
Positive

V2 = df[(df['Polarity']<0)]
Negative = V2.groupby('Polarity', as_index=False).agg({"Tweet":"count"})
Negative
```

Case Study 2

```
import sklearn  # for applyling  maching leraning algorithms
import matplotlib.pyplot as plt # for plotting
%matplotlib inline
import pandas as pd       # for data handling
from sklearn.model_selection import train_test_split
import numpy as np       # for numerical methods and data structures
Twitter_Tweet = pandas.read_csv("C:/Users/CSUFTitan/Downloads/Tweets.csv")
Twitter_Tweet.head()
del Twitter_Tweet['tweet_coord']
del Twitter_Tweet['airline_sentiment_gold']
del Twitter_Tweet['negativereason_gold']
Twitter_Tweet.head()
#Calculating airline_sentiment Tweet Count TotalMood_count=Twitter_Tweet['airline_sentiment'].value_counts()
Index = [1,2,3]
plt.bar(Index,TotalMood_count,color=['Red', 'Orange', 'Green'])
plt.xticks(Index,['Negative','Neutral','Positive'],rotation=20)
plt.ylabel('Mood Count')
plt.xlabel('Moods')
plt.title('Count of Individual Moods')
plt.figure(0.5,figsize=(6,10))
plt.show()
Index = [1,2,3,4,5,6]
plt.bar(Index,Individual_Airline_count,color=['lightblue'])
plt.xticks(Index,['United','US Airways','American','Southwest','Delta','Virgin America'],rotation=20)
plt.ylabel('Airline Tweet Count')
plt.xlabel('Airlines')
plt.title('Individual Airline Tweet Count')
plt.show()
#Individual Airline Mood Graphs
def plot_airline_sentiment(Airline):
    dataFrame=Twitter_Tweet[Twitter_Tweet['airline']==Airline]
```

```
    TotalCount=dataFrame['airline_sentiment'].value_counts()
    Index = [1,2,3]
    plt.bar(Index,TotalCount,color=['Red', 'Orange', 'Green'])
    plt.xticks(Index,['Negative','Neutral','Positive'])
    plt.xlabel('Moods')
    plt.title('Individual Moods of '+Airline+' Airline')
plt.figure(1,figsize=(13,13))
plt.subplot(231)
plot_airline_sentiment('US Airways')
plt.ylabel('Mood Count')
plt.subplot(232)
plot_airline_sentiment('United')
plt.subplot(233)
plot_airline_sentiment('American')
plt.subplot(234)
plot_airline_sentiment('Southwest')
plt.ylabel('Mood Count')
plt.subplot(235)
plot_airline_sentiment('Delta')
plt.subplot(236)
plot_airline_sentiment('Virgin America')
Negative_Tweets=dict(Twitter_Tweet['negativereason'].value_counts(sort=False))
def Negative_Tweets(Airline):
    if Airline=='All':
        dataFrame=Twitter_Tweet
    else:
        dataFrame=Twitter_Tweet[Twitter_Tweet['airline']==Airline]
    Total_count=dict(dataFrame['negativereason'].value_counts())
    Negative_reason=list(Twitter_Tweet['negativereason'].unique())
    Negative_reason=[x for x in Negative_reason if str(x) != 'nan']
    dframe_Reasons=pandas.DataFrame({'Reasons':Negative_reason})

    dframe_Reasons['Total_count']=dframe_Reasons['Reasons'].apply(lambda x: Total_count[x])
    return dframe_Reasons
def Airline_Plot(Airline):
    dataFrame=Negative_Tweets(Airline)
    Total_count=dataFrame['Total_count']
    Index = range(1,(len(dataFrame)+1))
    plt.bar(Index,Total_count,color =['lightblue'])
    plt.xticks(Index,dataFrame['Reasons'],rotation=90)
    plt.ylabel(' Total Count')
    plt.xlabel('Negative Reasons')
plt.title('Reasons for all Airlines')
Airline_Plot('All')
from wordcloud import WordCloud,STOPWORDS
dataFrame=Twitter_Tweet[Twitter_Tweet['airline_sentiment']=='negative']
words = ' '.join(dataFrame['text'])
cleaned_word = " ".join([word for word in words.split()
                         if 'http' not in word
                            and not 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()
import numpy as np
import pandas as pd
import re    # regular expressions
import nltk  # natural language Toolkit
import matplotlib.pyplot as plt
%matplotlib inline

Attributes = Twitter_Tweet.iloc[:, 10].values
Columns = Twitter_Tweet.iloc[:, 1].values
Processed_Attributes = []

for line in range(0, len(Attributes)):
    # Remove all the special characters
    Processed_Attribute = re.sub(r'\W', ' ', str(Attributes[line]))

    # remove all single characters
    Processed_Attribute= re.sub(r'\s+[a-zA-Z]\s+', ' ', Processed_Attribute)

    # Remove single characters from the start
    Processed_Attribute = re.sub(r'\^[a-zA-Z]\s+', ' ', Processed_Attribute)

    # Substituting multiple spaces with single space
    Processed_Attribute = re.sub(r'\s+', ' ', Processed_Attribute, flags=re.I)

    # Removing prefixed 'b'
    Processed_Attribute = re.sub(r'^b\s+', '', Processed_Attribute)

    # Converting to Lowercase
    Processed_Attribute = Processed_Attribute.lower()

    Processed_Attributes.append(Processed_Attribute)
from nltk.corpus import stopwords #removing stop words
from sklearn.feature_extraction.text import TfidfVectorizer #vectors for different tasks

vectorizer = TfidfVectorizer (max_features=2500, min_df=7, max_df=0.8)
processed_features = vectorizer.fit_transform(processed_features).toarray()
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(processed_features, labels, test_size=0.2, random_state=0)
from sklearn.ensemble import RandomForestClassifier
#A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the
dataset

text_classifier = RandomForestClassifier(n_estimators=200, random_state=0)
text_classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)

predictions = text_classifier.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
confusion_matrix(y_test,predictions)
confusion_matrix = confusion_matrix(y_test,predictions)
from sklearn.metrics import confusion_matrix
import seaborn as sn
array = [[1743 ,  84 ,  43],[ 503 ,  86 ,  25], [ 368 ,  37 ,  39]]
confusion_matrix= pd.DataFrame(array,columns=['Negative:-
1','Neutral:0','Positive:+1'],index=['Negative','Neutral','Positive'],)
sn.set(font_scale=1.2)
ax = sn.heatmap(confusion_matrix,square=1,annot=True,fmt='d',cmap="Blues",annot_kws={"size":
15},linewidths=.5)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
print(classification_report(y_test,predictions))
print(accuracy_score(y_test, predictions))
```