

SENTIMENT ANALYSIS FOR MARKETING

PROJECT TITLE	SENTIMENT ANALYSIS FOR MAKETING
SKILLS TAKEN AWAY	<ul style="list-style-type: none">❖ Python script❖ EDA❖ UI deployment
DOMAIN	FMCG[FAST MOVING CONSUMER GOODS]

PROBLEM STATEMENT :

Customer review for any product . Product review feedback from customer play apivotal role in enriching the product quality along site meet the market expectation . It is easy for any seller to get review to one to one conversation with custom . If the product is stored in offline store but it is difficult to retrive and analyse the same review , if the same product sold online .

“E- commerce” one of the booming industries and is the one stop destination for various sellers to market and sell their product online to attract a large market .

Given set of review of each category for product that is live on a e-commerce plat form like flipkart/amazon/myntra etc...

LIBRARY INSTALLATION :

1.INSTALLING LIBRARIES :

```
[ ] !pip install -U scikit-learn
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.2)
Requirement already satisfied: numpy<2.0,>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.3)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)

[ ] !pip install nltk
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)

[ ] # import the libraries
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
from sklearn.model_selection import train_test_split
import numpy as np
import seaborn as sns
import re
import nltk
from nltk.corpus import stopwords
from wordcloud import WordCloud,STOPWORDS
```

2.IMPORTING DATA SET :

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] # Load the dataset
df=pd.read_csv('/content/Tweets.csv')
```

```
[ ] df.shape

(14640, 15)
```

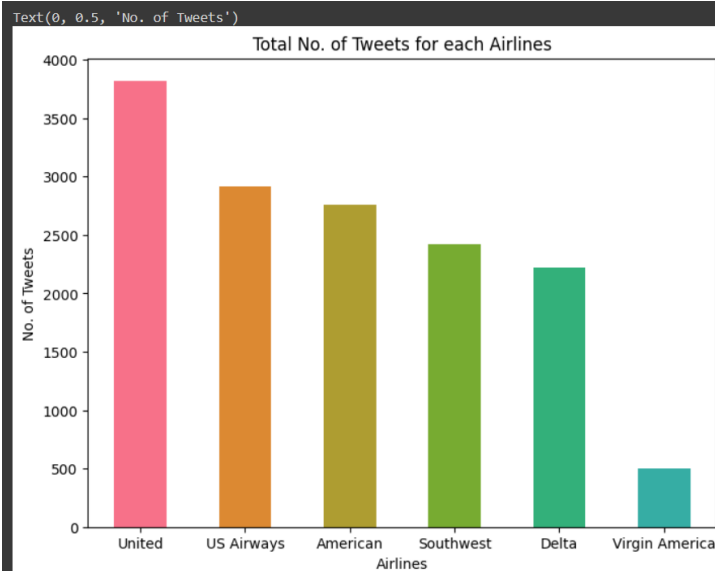
3.COUNTING NO. OF TWEETS EACH AIRLINES RECEIVED :

```
#counting the number of tweets each airlines has received  
df.airline.value_counts()
```

```
United      3822  
US Airways  2913  
American    2759  
Southwest   2420  
Delta       2222  
Virgin America  504  
Name: airline, dtype: int64
```

4.PLOTTING NO. OF TWEETS EACH AIRLINE RECEIVED :

```
#Plotting the number of tweets each airlines has received  
colors=sns.color_palette("husl", 10)  
pd.Series(df["airline"]).value_counts().plot(kind = "bar",  
                                              color=colors,figsize=(8,6),fontsize=10,rot = 0, title = "Total No. of Tweets for each Airlines")  
plt.xlabel('Airlines', fontsize=10)  
plt.ylabel('No. of Tweets', fontsize=10)
```



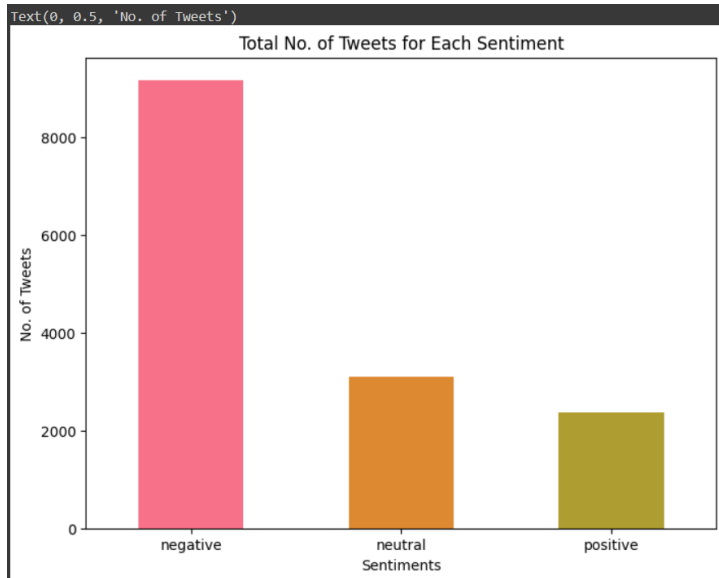
5.COUNTING THE NO. OF EACH SENTIMENT :

```
[ ] #counting the number of each type of sentiments  
df.airline_sentiment.value_counts()
```

```
negative  9178  
neutral   3099  
positive  2363  
Name: airline_sentiment, dtype: int64
```

6.PLOTTING THE NO. OF EACH SENTIMENT :

```
[ ] #Plotting the number of each type of sentiments  
colors=sns.color_palette("husl", 10)  
pd.Series(df["airline_sentiment"]).value_counts().plot(kind = "bar",  
                                                       color=colors,figsize=(8,6),rot=0, title = "Total No. of Tweets for Each Sentiment")  
plt.xlabel('Sentiments', fontsize=10)  
plt.ylabel('No. of Tweets', fontsize=10)
```

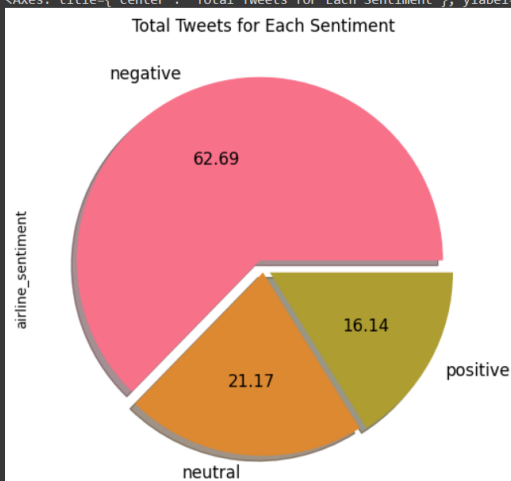


```

colors=sns.color_palette("husl", 10)
pd.Series(df["airline_sentiment"].value_counts()).plot(kind="pie",colors=colors,
labels=["negative", "neutral", "positive"],explode=[0.05,0.02,0.04],
shadow=True,autopct='%2f', fontsize=12,figsize=(6, 6),title = "Total Tweets for Each Sentiment")

```

<Axes: title=('center': 'Total Tweets for Each Sentiment'), ylabel='airline_sentiment'>



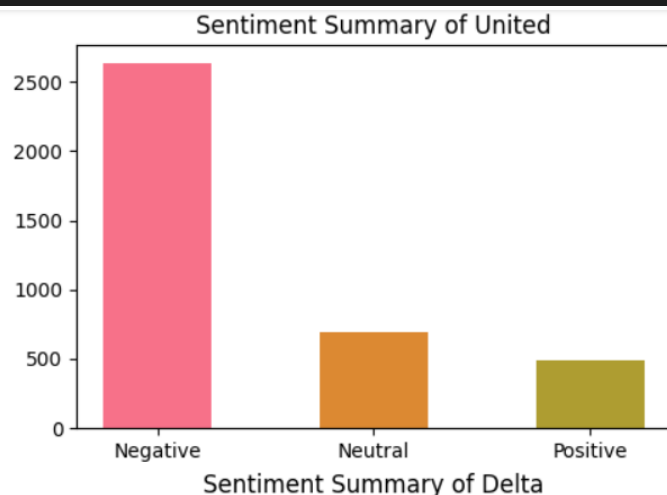
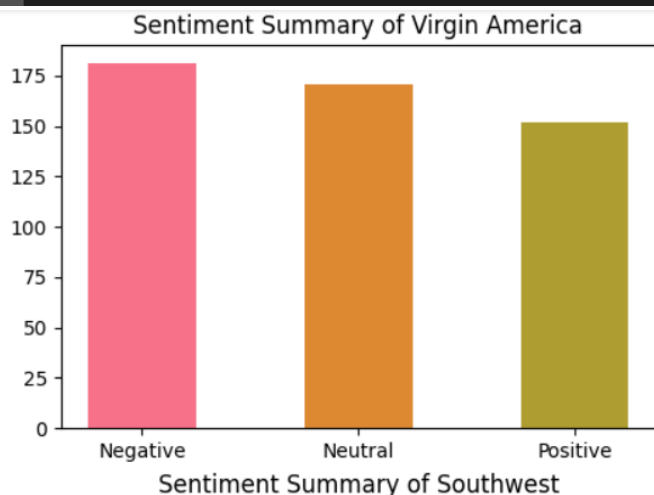
7.PLOTTING REVIEW FOR EACH COUNTRY AIRLINES :

```

def plot_sub_sentiment(Airline):
    pdf = df[df['airline']==Airline]
    count = pdf['airline_sentiment'].value_counts()
    Index = [1,2,3]
    color=sns.color_palette("husl", 10)
    plt.bar(Index,count,width=0.5,color=color)
    plt.xticks(Index,['Negative','Neutral','Positive'])
    plt.title('Sentiment Summary of ' + " " + Airline)

airline_name =df['airline'].unique()
plt.figure(1,figsize=(12,12))
for i in range(6):
    plt.subplot(3,2,i+1)
    plot_sub_sentiment(airline_name[i])

```





8.COUNTING TOTAL NO. OF NEGATIVE REASON :

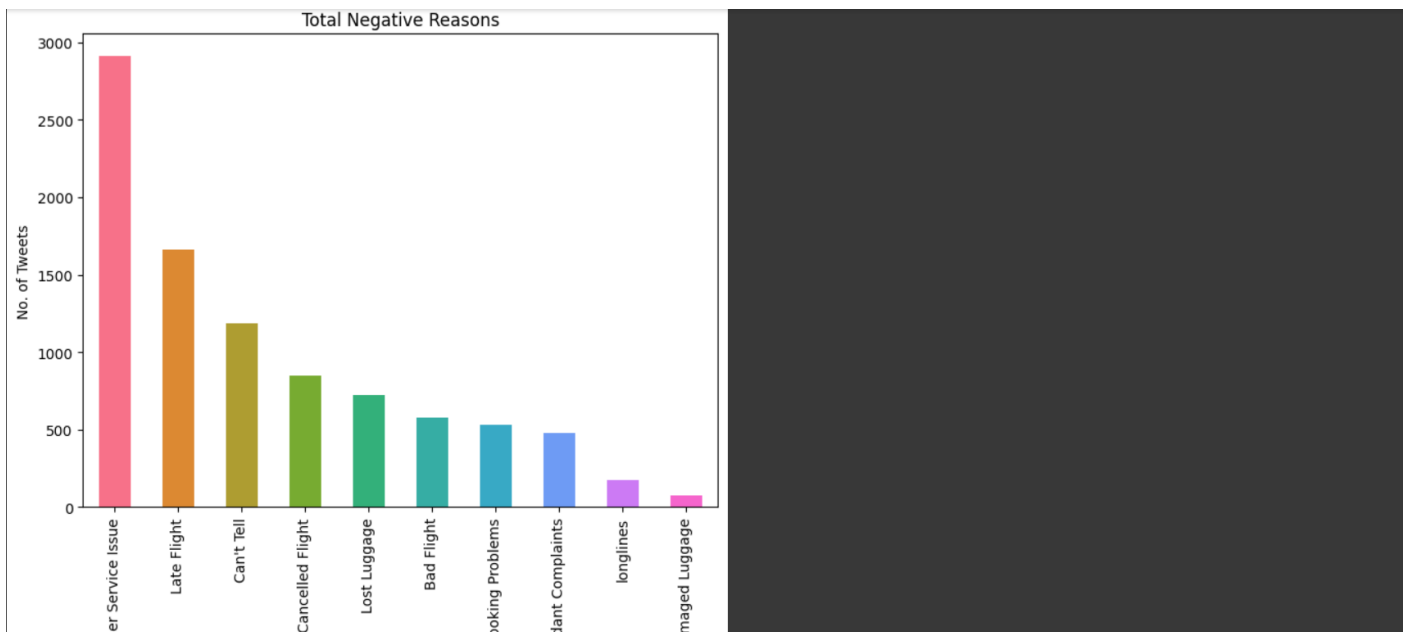
```
[ ] #counting the total number of negative reasons
df.negative_reason.value_counts()
```

Negative Reason	Count
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: negative_reason, dtype: int64

9.PLOTTING TOTAL NO. OF NEGATIVE REASON:

```
#Plotting all the negative reasons
color=sns.color_palette("husl", 10)
pd.Series(df["negative_reason"].value_counts().plot(kind = "bar",
color=color,figsize=(8,6),title = "Total Negative Reasons")
plt.xlabel('Negative Reasons', fontsize=10)
plt.ylabel('No. of Tweets', fontsize=10)
```

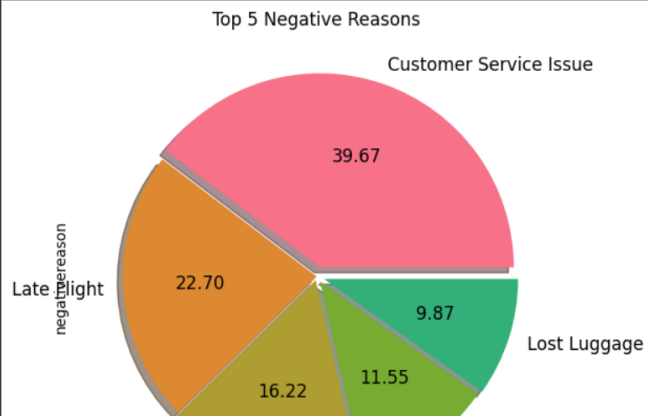


```

color=sns.color_palette("husl", 10)
pd.Series(df["negativereason"].value_counts().head(5).plot(kind="pie",
labels=["Customer Service Issue", "Late Flight", "Can't Tell", "Cancelled Flight", "Lost Luggage"],
colors=color, autopct='%2f', explode=[0.05, 0, 0.02, 0.03, 0.04], shadow=True,
fontsize=12, figsize=(6, 6), title="Top 5 Negative Reasons"))

```

<Axes: title={'center': 'Top 5 Negative Reasons'}, ylabel='negativereason'>



```

[ ] air_senti=pd.crosstab(df.airline,df.airline_sentiment)
air_senti

```

airline_sentiment	negative	neutral	positive
airline			
American	1960	463	336
Delta	955	723	544
Southwest	1186	664	570
US Airways	2263	381	269
United	2633	697	492
Virgin America	181	171	152

Lost Luggage 724
Name: negativereason, dtype: int64

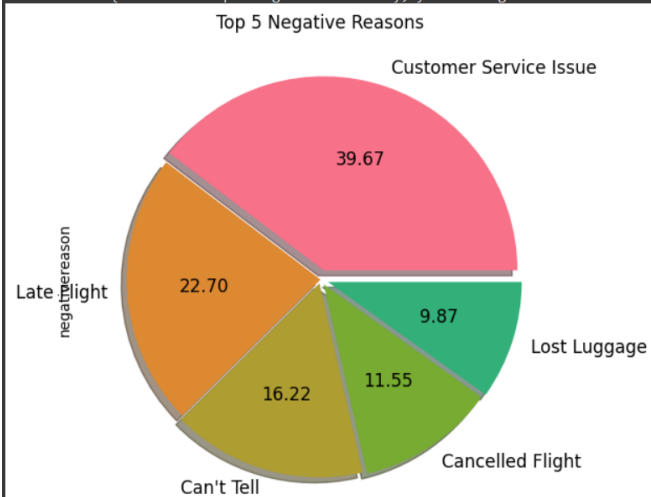
```

color=sns.color_palette("husl", 10)
pd.Series(df["negativereason"].value_counts().head(5).plot(kind="pie",
labels=["Customer Service Issue", "Late Flight", "Can't Tell", "Cancelled Flight", "Lost Luggage"],
colors=color, autopct='%2f', explode=[0.05, 0, 0.02, 0.03, 0.04], shadow=True,
fontsize=12, figsize=(6, 6), title="Top 5 Negative Reasons"))

```

10.COUNTING EACH AIRLINE'S NEGATIVE REVIEW :

<Axes: title={'center': 'Top 5 Negative Reasons'}, ylabel='negativereason'>



11.TABULATING THE REVIEWS :

```
[ ] air_senti=pd.crosstab(df.airline,df.airline_sentiment)
air_senti
```

airline_sentiment	negative	neutral	positive
airline			
American	1960	463	336
Delta	955	723	544
Southwest	1186	664	570
US Airways	2263	381	269
United	2633	697	492
Virgin America	181	171	152

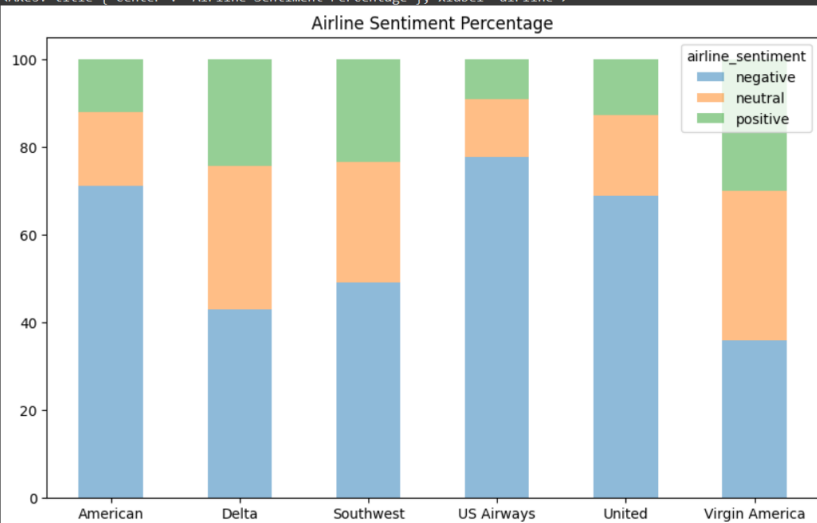
12.CALCULATING AND DISPLAYING PERCENTAGE OF EACH REVIEW :

```
percent=air_senti.apply(lambda a: a / a.sum() * 100, axis=1)
percent
```

airline_sentiment	negative	neutral	positive
airline			
American	71.040232	16.781443	12.178325
Delta	42.979298	32.538254	24.482448
Southwest	49.008264	27.438017	23.553719
US Airways	77.686234	13.079300	9.234466
United	68.890633	18.236525	12.872841
Virgin America	35.912698	33.928571	30.158730

```
percent.plot(kind='bar',figsize=(10, 6),alpha=0.5,
rot=0,stacked=True,title="Airline Sentiment Percentage")
```

<Axes: title={'center': 'Airline Sentiment Percentage'}, xlabel='airline'>



13.DATE OF CUSTOMER TWEETED :

```
[ ] df['tweet_created'] = pd.to_datetime(df['tweet_created'])
    df["date_created"] =df["tweet_created"].dt.date
```

```
df["date_created"]
```

0	2015-02-24
1	2015-02-24
2	2015-02-24
3	2015-02-24
4	2015-02-24

14635 2015-02-22

14636	2015-02-22
14637	2015-02-22

14638	2015-02-22
14639	2015-02-22

Name: date created,

14.VISULIZATION OF WORDS USED FOR NEGATIVE TWEETS :

```
print(df_filtered)
```

```
words = ' '.join(df['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))
```

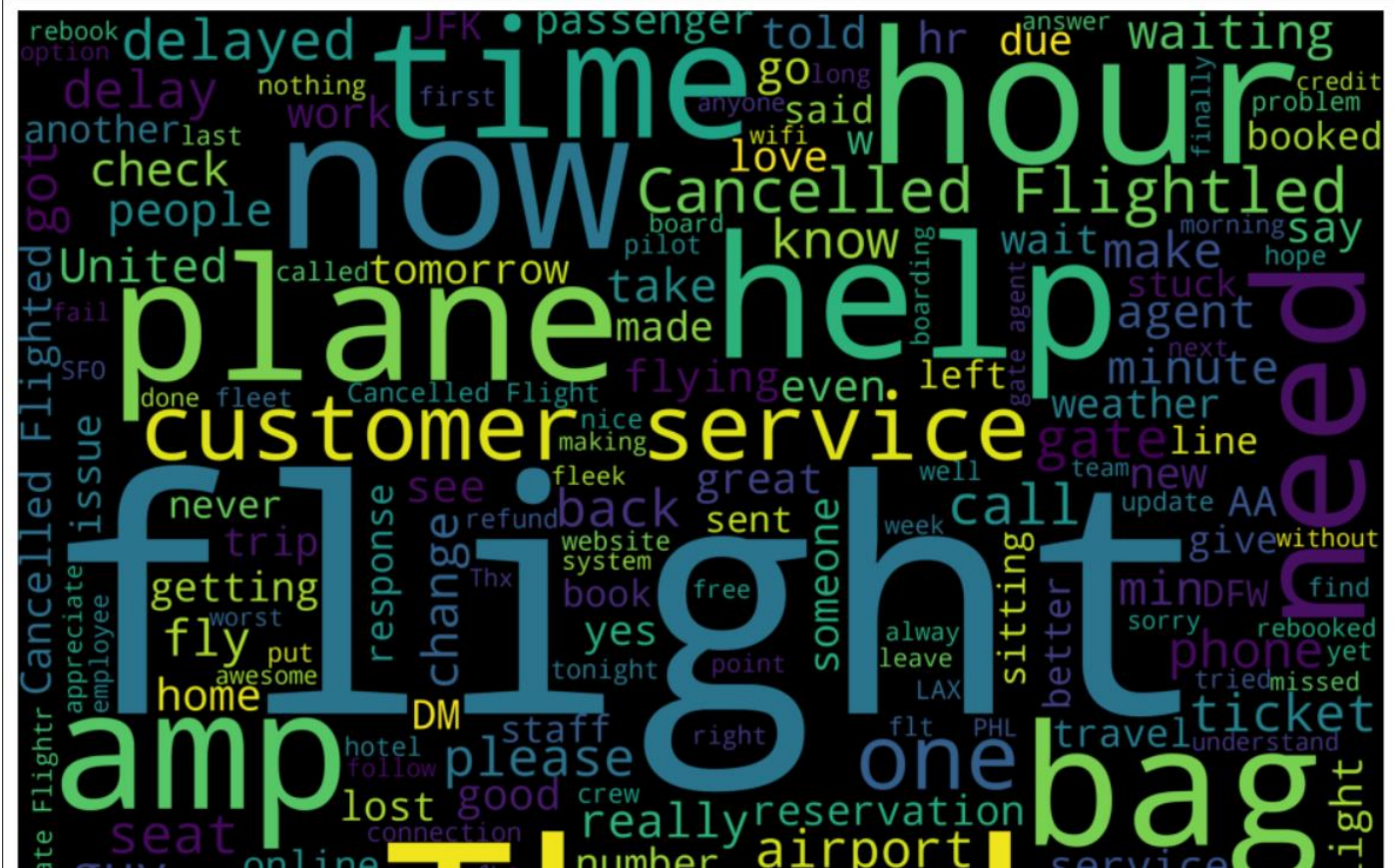
```
plt.imshow(wordcloud)
```

```
plt.axis('off')
```

```
plt.show()
```



15.VISULISATION OF WORDS USED FOR POSITIVE TWEETS:



```
[ ] def tweet_to_words(raw_tweet):
    letters_only = re.sub("[^a-zA-Z]", " ", raw_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 ))

[ ] def clean_tweet_length(raw_tweet):
    letters_only = re.sub("[^a-zA-Z]", " ", raw_tweet)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words))
```

```
> df['sentiment']=df['airline_sentiment'].apply(lambda x: 0 if x=='negative' else 1)
df.sentiment.head()
```

```
0    1
1    1
2    1
3    0
4    0
Name: sentiment, dtype: int64
```

```
[ ] nltk.download('stopwords')
```

```
[nltk data] Downloading package stopwords to /root/nltk_data...
[nltk data] Unzipping corpora/stopwords.zip.
True
```