In [1]:

```
import pandas as pd
import numpy as np
import pickle
import warnings
warnings.filterwarnings("ignore")
from sklearn.metrics import ConfusionMatrixDisplay
```

In [2]:

```
data=pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\sentiment analysis for
```

In [3]:

```
data.head(10)
```

Out[3]:

	tweet_id	sentiment	sentiment_confidence	name	retweet_count	text	tweet_cre
0	5.700000e+17	neutral	1.0000	cairdin	0	What @dhepburn said.	24-02- 1
1	5.700000e+17	positive	0.3486	jnardino	0	plus you've added commercials to the experien...	24-02- 1
2	5.700000e+17	neutral	0.6837	yvonnalynn	0	I didn't today... Must mean I need to take an...	24-02- 1
3	5.700000e+17	negative	1.0000	jnardino	0	it's really aggressive to blast obnoxious "en...	24-02- 1
4	5.700000e+17	negative	1.0000	jnardino	0	and it's a really big bad thing about it	24-02- 1
5	5.700000e+17	negative	1.0000	jnardino	0	seriously would pay $30 a flight for seats th...	24-02- 1
6	5.700000e+17	positive	0.6745	cjmcginnis	0	yes, nearly every time I fly VX this “ear wor...	24-02- 1
7	5.700000e+17	neutral	0.6340	pilot	0	Really missed a prime opportunity for Men Wit...	24-02- 1
8	5.700000e+17	positive	0.6559	dhepburn	0	Well, I didn't...but NOW I DO! :-D	24-02- 1

	tweet_id	sentiment	sentiment_confidence	name	retweet_count	text	tweet_cre
9	5.700000e+17	positive	1.0000	YupitsTate	0	it was amazing, and arrived an hour early. Yo...	24-02- 1

In [4]:

```
data.tail(10)
```

Out[4]:

	tweet_id	sentiment	sentiment_confidence	name	retweet_count	text
14630	5.700000e+17	positive	1.0000	Laurelinesblog	0	Thanks! He is.
14631	5.700000e+17	negative	1.0000	MDDavis7	0	thx for nothing on getting us out of the coun...
14632	5.700000e+17	neutral	0.6760	Chad_SMFYM	0	“: @TilleyMonsta George, that doesn't look goo...
14633	5.700000e+17	negative	1.0000	RussellsWriting	0	my flight was Cancelled Flightled, leaving to...
14634	5.700000e+17	negative	0.6684	GolfWithWoody	0	right on cue with the delays 👌
14635	5.700000e+17	positive	0.3487	KristenReenders	0	thank you we got on a different flight to Chi...
14636	5.700000e+17	negative	1.0000	itsropes	0	leaving over 20 minutes Late Flight. No warni...
14637	5.700000e+17	neutral	1.0000	sanyabun	0	Please bring American Airlines to #BlackBerry10
14638	5.700000e+17	negative	1.0000	SraJackson	0	you have my money, you change my flight, and ...
14639	5.700000e+17	neutral	0.6771	daviddtwu	0	we have 8 ppl so we need 2 know how many seat...

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   tweet_id              14640 non-null  float64
 1   sentiment             14640 non-null  object 
 2   sentiment_confidence  14640 non-null  float64
 3   name                  14640 non-null  object 
 4   retweet_count         14640 non-null  int64  
 5   text                  14640 non-null  object 
 6   tweet_created         14640 non-null  object 
 7   latitude              14640 non-null  float64
 8   longitude             14640 non-null  float64
 9   airline               14640 non-null  object 
dtypes: float64(4), int64(1), object(5)
memory usage: 1.1+ MB
```

In [6]:

```
data.describe()
```

Out[6]:

	tweet_id	sentiment_confidence	retweet_count	latitude	longitude
count	1.464000e+04	14640.000000	14640.000000	14640.000000	14640.000000
mean	5.692602e+17	0.900169	0.082650	39.139901	-98.687615
std	8.100563e+14	0.162830	0.745778	3.599528	18.731383
min	5.680000e+17	0.335000	0.000000	24.295539	-121.892659
25%	5.690000e+17	0.692300	0.000000	36.664544	-119.106167
50%	5.690000e+17	1.000000	0.000000	40.060817	-101.278813
75%	5.700000e+17	1.000000	0.000000	40.759700	-74.002575
max	5.700000e+17	1.000000	44.000000	59.658972	-73.388100

In [7]:

```
data.shape
```

Out[7]:

```
(14640, 10)
```

In [8]:

```
data.isna().sum()
```

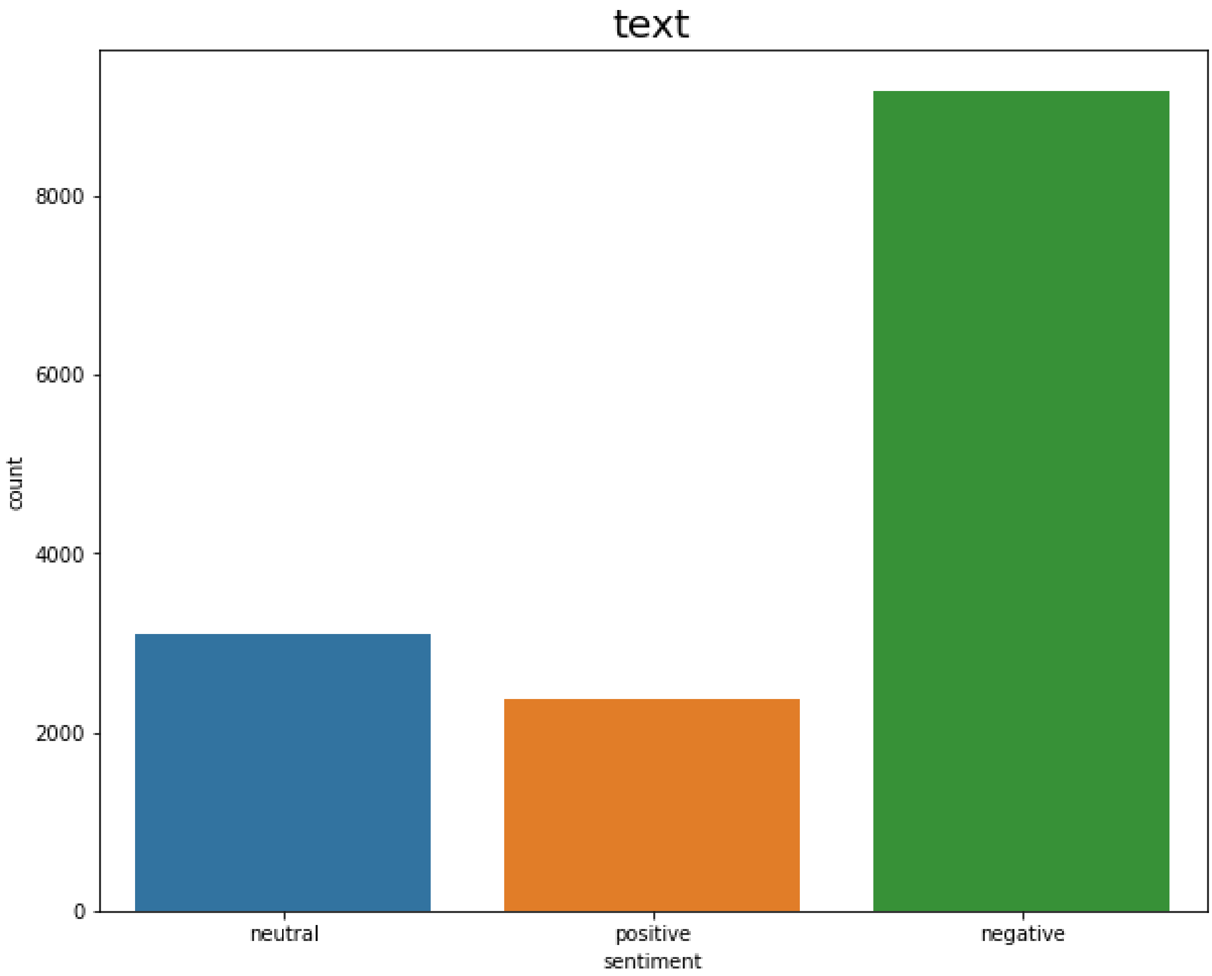
Out[8]:

```
tweet_id                0
sentiment               0
sentiment_confidence    0
name                    0
retweet_count           0
text                    0
tweet_created           0
latitude                0
longitude               0
airline                 0
dtype: int64
```

Visualization

In [9]:

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize = (10, 8))
ax = sns.countplot(x = 'sentiment', data = data)
ax.set_title(label = 'text', fontsize = 20)
plt.show()
```

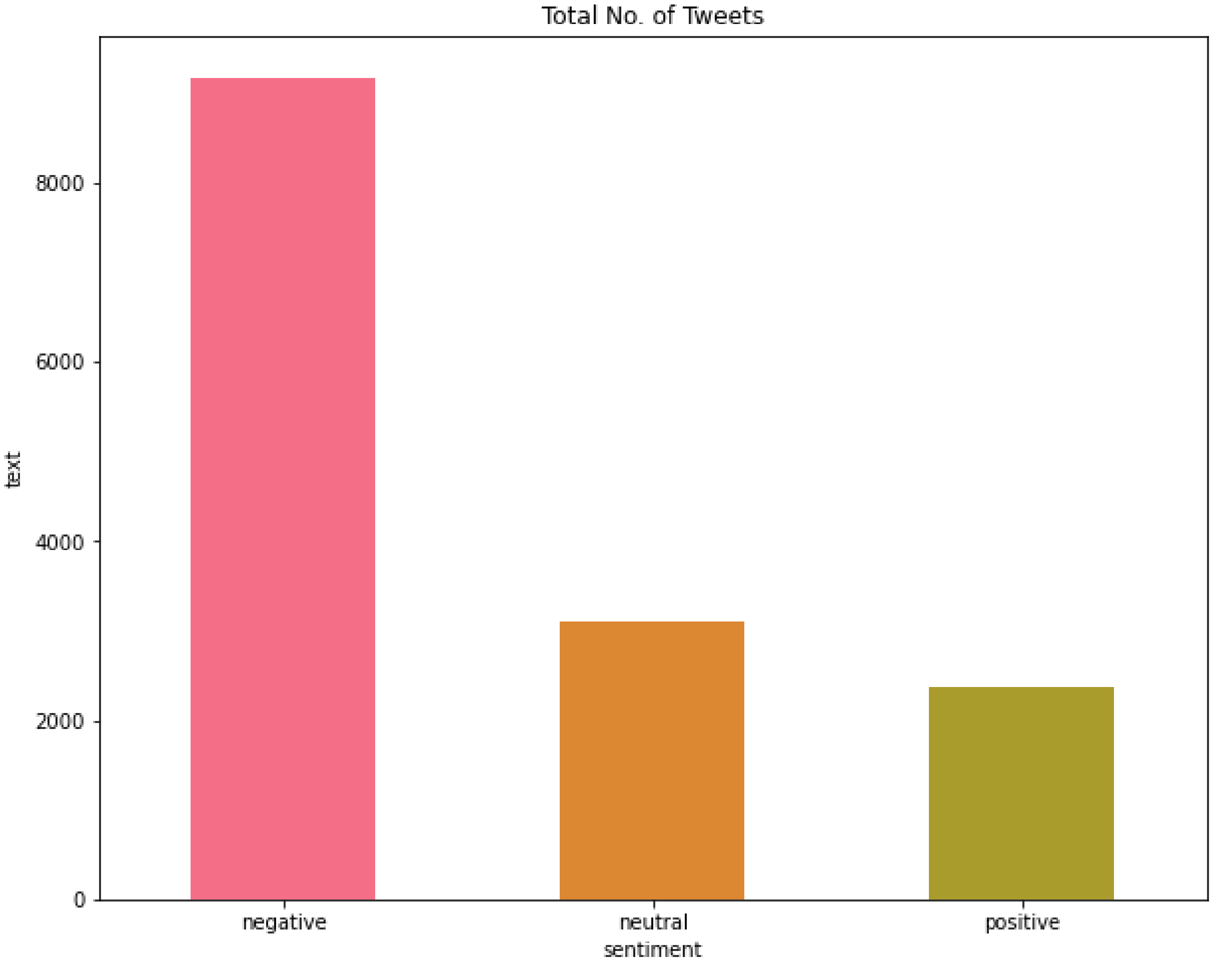


In [10]:

```
colors=sns.color_palette('husl',10)
pd.Series(data['sentiment']).value_counts().plot(kind="bar",color=colors,figsize=(10
plt.xlabel('sentiment',fontsize=10)
plt.ylabel('text',fontsize=10)
```

Out[10]:

```
Text(0, 0.5, 'text')
```

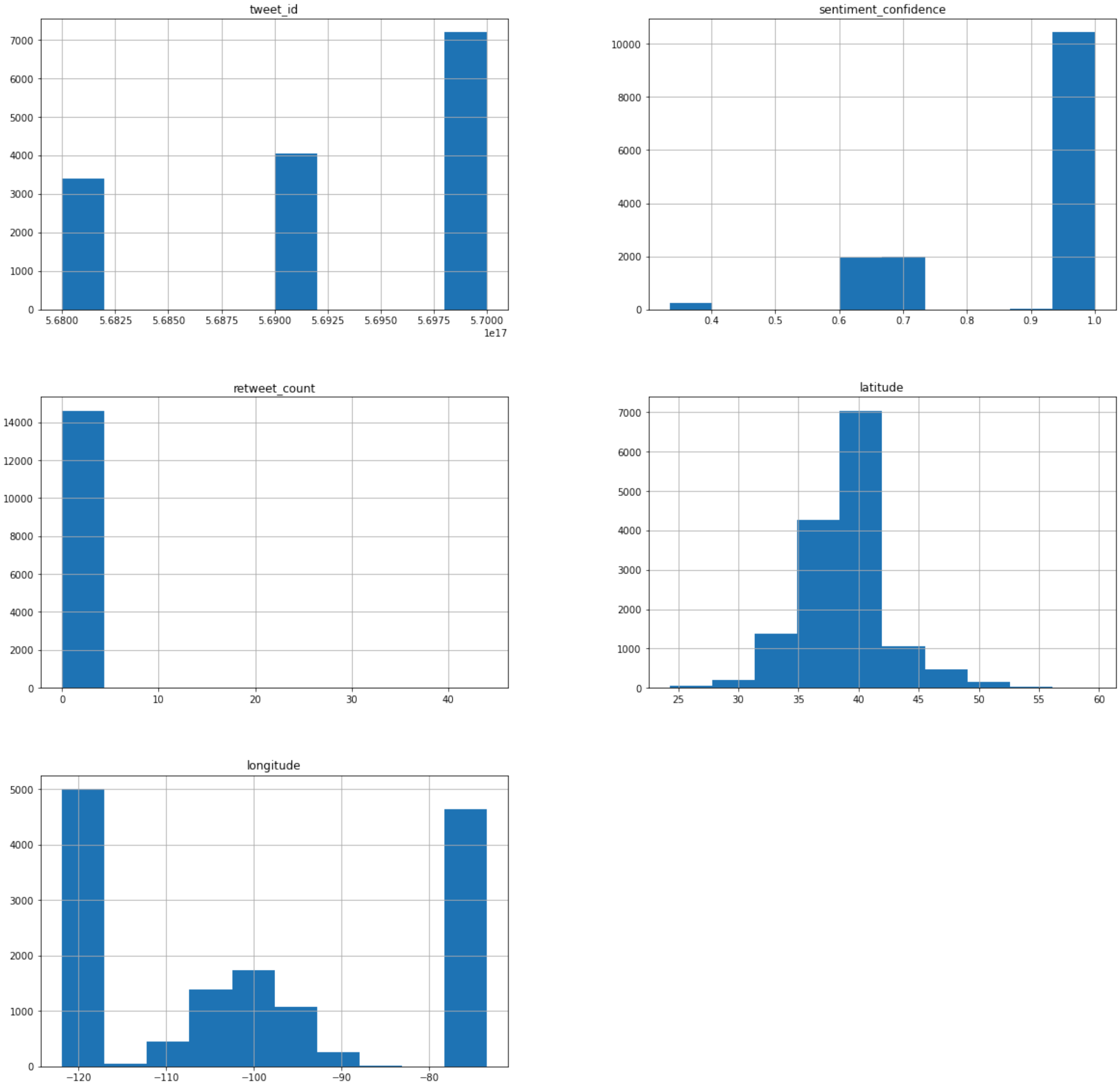


In [11]:

```
data.hist(figsize=(20,20))
```

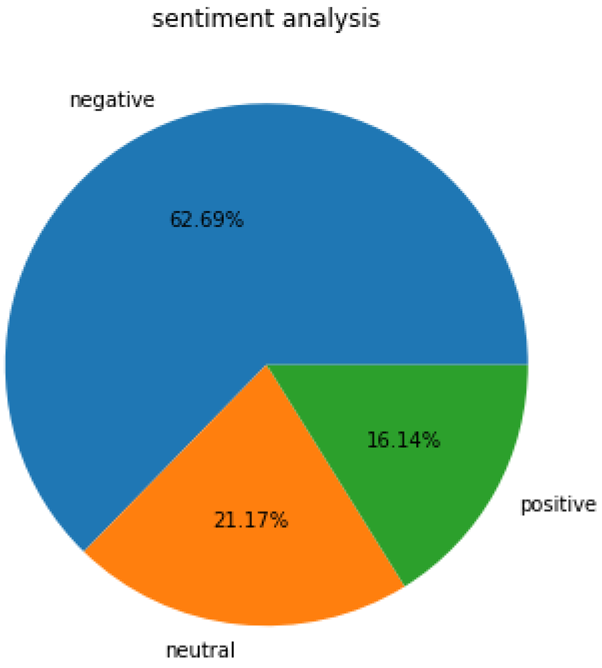
Out[11]:

```
array([[<AxesSubplot:title={'center':'tweet_id'}>,
        <AxesSubplot:title={'center':'sentiment_confidence'}>],
       [<AxesSubplot:title={'center':'retweet_count'}>,
        <AxesSubplot:title={'center':'latitude'}>],
       [<AxesSubplot:title={'center':'longitude'}>, <AxesSubplot:>]],
      dtype=object)
```



In [12]:

```
plt.figure(figsize=(8, 6))
data['sentiment'].value_counts().plot.pie(autopct='%2.2f%%')
plt.title('sentiment analysis')
plt.ylabel('')
plt.show()
```



In [13]:

```
cor=data.corr()
cor
```

Out[13]:

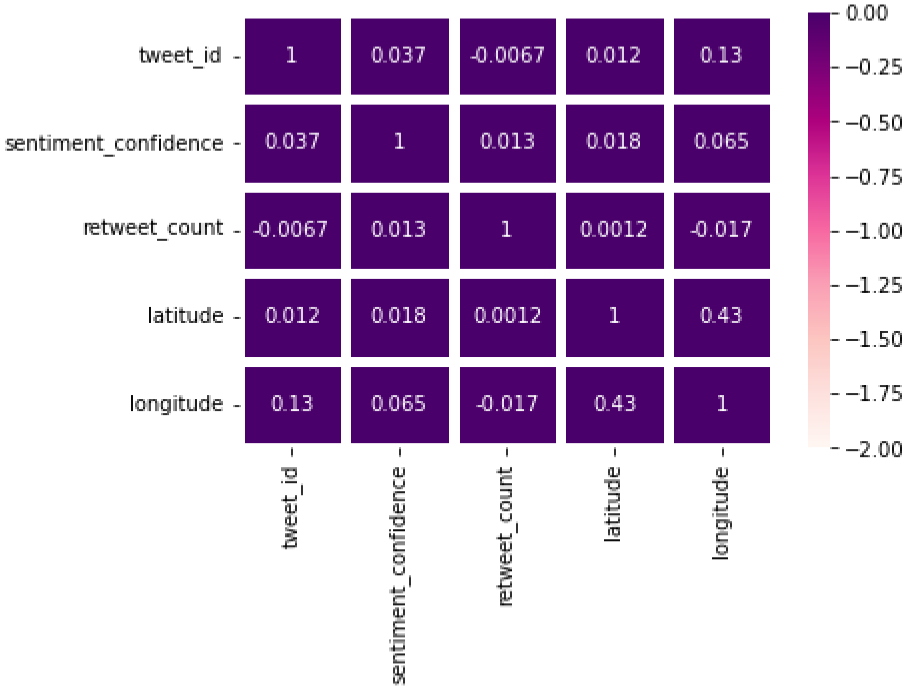
	tweet_id	sentiment_confidence	retweet_count	latitude	longitude
tweet_id	1.000000	0.037452	-0.006650	0.011716	0.126882
sentiment_confidence	0.037452	1.000000	0.012581	0.018142	0.064819
retweet_count	-0.006650	0.012581	1.000000	0.001165	-0.016931
latitude	0.011716	0.018142	0.001165	1.000000	0.425175
longitude	0.126882	0.064819	-0.016931	0.425175	1.000000

In [14]:

```
import seaborn as sb
sb.heatmap(cor,vmax=0,vmin=-2,annot=True,linewidth=-5,cmap="RdPu")
```

Out[14]:

```
<AxesSubplot:>
```



In [15]:

```
#data['negativereason'] = data['negativereason'].fillna('') + ' ' + data['negativere
#data['negativereason_confidence'] = data['negativereason_confidence'].fillna('') +
#data.fillna(1,inplace=True)
```

In [16]:

```
data['sentiment']=data['sentiment'].map({'neutral':0,'positive':1,'negative':-1})
data
```

Out[16]:

	tweet_id	sentiment	sentiment_confidence	name	retweet_count	text
0	5.700000e+17	0	1.0000	cairdin	0	What @dhepburn said.
1	5.700000e+17	1	0.3486	jnardino	0	plus you've added commercials to the experien...
2	5.700000e+17	0	0.6837	yvonnalynn	0	I didn't today... Must mean I need to take an...
3	5.700000e+17	-1	1.0000	jnardino	0	it's really aggressive to blast obnoxious "en...
4	5.700000e+17	-1	1.0000	jnardino	0	and it's a really big bad thing about it
...	...	...	...	...	...	...
14635	5.700000e+17	1	0.3487	KristenReenders	0	thank you we got on a different flight to Chi...

	tweet_id	sentiment	sentiment_confidence	name	retweet_count	text
14636	5.700000e+17	-1	1.0000	itsropes	0	leaving over 20 minutes Late Flight. No warni...
14637	5.700000e+17	0	1.0000	sanyabun	0	Please bring American Airlines to #BlackBerry10
14638	5.700000e+17	-1	1.0000	SraJackson	0	you have my money, you change my flight, and ...
14639	5.700000e+17	0	0.6771	daviddtwu	0	we have 8 ppl so we need 2 know how many seat...

14640 rows × 10 columns

In [17]:

```
data['text']
```

Out[17]:

```
0                                  What @dhepburn said.
1         plus you've added commercials to the experien...
2         I didn't today... Must mean I need to take an...
3         it's really aggressive to blast obnoxious "en...
4                 and it's a really big bad thing about it
                               ...
14635     thank you we got on a different flight to Chi...
14636     leaving over 20 minutes Late Flight. No warni...
14637      Please bring American Airlines to #BlackBerry10
14638     you have my money, you change my flight, and ...
14639     we have 8 ppl so we need 2 know how many seat...
Name: text, Length: 14640, dtype: object
```

In [18]:

```
x=data['text']
y=data['sentiment']
```

In [19]:

```
x
```

Out[19]:

```
0                                  What @dhepburn said.
1         plus you've added commercials to the experien...
2         I didn't today... Must mean I need to take an...
3         it's really aggressive to blast obnoxious "en...
4                 and it's a really big bad thing about it
                               ...
14635     thank you we got on a different flight to Chi...
14636     leaving over 20 minutes Late Flight. No warni...
14637      Please bring American Airlines to #BlackBerry10
14638     you have my money, you change my flight, and ...
14639     we have 8 ppl so we need 2 know how many seat...
Name: text, Length: 14640, dtype: object
```

In [20]:

```
y
```

Out[20]:

```
0         0
1         1
2         0
3        -1
4        -1
         ..
14635     1
14636    -1
14637     0
14638    -1
14639     0
Name: sentiment, Length: 14640, dtype: int64
```

In [21]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfid = TfidfVectorizer()
```

In [22]:

```
x_final=tfid.fit_transform(x)
```

In [23]:

```
pip install threadpoolctl==3.1.0
```

```
Requirement already satisfied: threadpoolctl==3.1.0 in c:\users\reshma_koduri\anacond
a3\lib\site-packages (3.1.0)
Note: you may need to restart the kernel to use updated packages.
DEPRECATION: pyodbc 4.0.0-unsupported has a non-standard version number. pip 24.1 wil
l enforce this behaviour change. A possible replacement is to upgrade to a newer vers
ion of pyodbc or contact the author to suggest that they release a version with a con
forming version number. Discussion can be found at https://github.com/pypa/pip/issue
s/12063
```

In [24]:

```
from imblearn.over_sampling import SMOTE
smote=SMOTE()
x_sm,y_sm=smote.fit_resample(x_final,y)
```

Data splitting

In [25]:

```
from sklearn.model_selection import train_test_split
```

In [26]:

```
x_train,x_test,y_train,y_test=train_test_split(x_sm,y_sm,test_size=0.33,random_state
```

In [27]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(18447, 15045)
(9087, 15045)
(18447,)
(9087,)
```

Logistic Regression

In [28]:

```
from sklearn.linear_model import LogisticRegression
cls=LogisticRegression()
cls=cls.fit(x_train,y_train)
```

In [29]:

```
y_pred1=cls.predict(x_test)
y_pred1
```

Out[29]:

```
array([ 0,  1,  0, ..., -1,  1,  1], dtype=int64)
```

In [30]:

```
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix(y_pred1,y_test)
```

Out[30]:

```
array([[2533,  289,   98],
       [ 437, 2548,  192],
       [ 122,  138, 2730]], dtype=int64)
```

In [31]:

```
accuracy_score(y_pred1,y_test)
```

Out[31]:

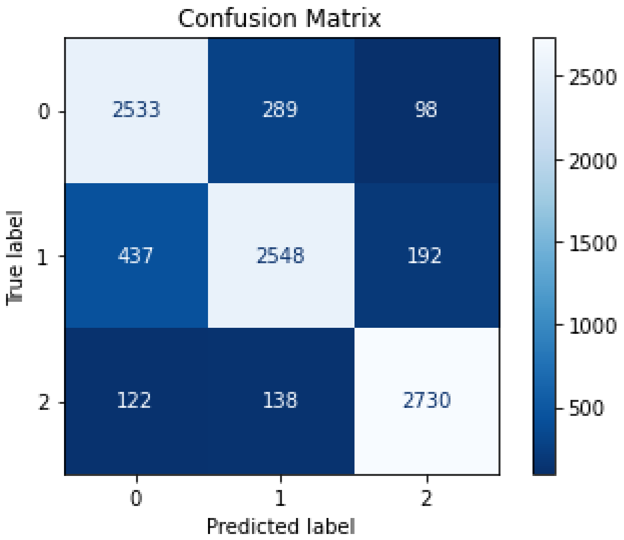
```
0.8595796192362716
```

In [32]:

```
cm1=confusion_matrix(y_pred1,y_test)
ConfusionMatrixDisplay(cm1).plot(cmap='Blues_r')
plt.title('Confusion Matrix')
```

Out[32]:

```
Text(0.5, 1.0, 'Confusion Matrix')
```



In [33]:

```
from sklearn.metrics import classification_report
report1=classification_report(y_pred1,y_test)
print("Classification Report:\n", report1)
```

```
Classification Report:
               precision    recall  f1-score   support

          -1       0.82      0.87      0.84      2920
           0       0.86      0.80      0.83      3177
           1       0.90      0.91      0.91      2990

    accuracy                           0.86      9087
   macro avg       0.86      0.86      0.86      9087
```

```
weighted avg       0.86      0.86      0.86      9087
```

RandomForest Classifier

In [34]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [35]:
```
cls=RandomForestClassifier()
cls.fit(x_train,y_train)
```

Out[35]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [36]:
```
y_pred2=cls.predict(x_test)
y_pred2
```

Out[36]:
```
array([ 0,  1,  0, ..., -1,  1,  1], dtype=int64)
```

In [37]:
```
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix(y_pred2,y_test)
```

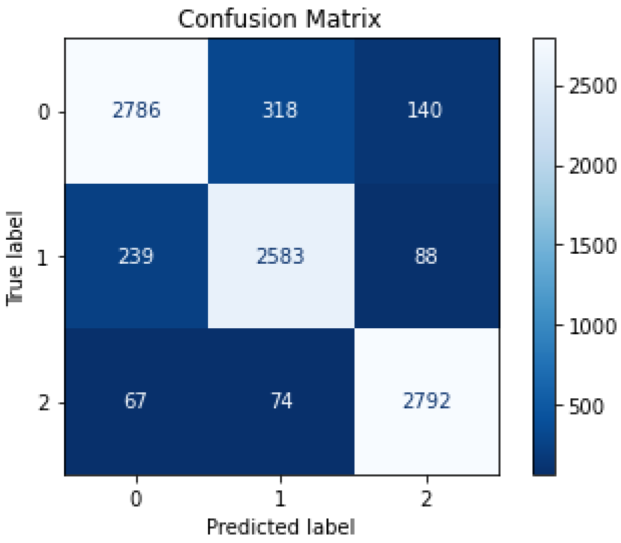
Out[37]:
```
array([[2786,  318,  140],
       [ 239, 2583,   88],
       [  67,   74, 2792]], dtype=int64)
```

In [38]:
```
accuracy_score(y_pred2,y_test)
```

Out[38]:
```
0.898096181357984
```

In [39]:
```
cm2=confusion_matrix(y_pred2,y_test)
ConfusionMatrixDisplay(cm2).plot(cmap='Blues_r')
plt.title('Confusion Matrix')
```

Out[39]:
```
Text(0.5, 1.0, 'Confusion Matrix')
```



In [40]:

```
from sklearn.metrics import classification_report
report2=classification_report(y_pred2,y_test)
print("Classification Report:\n", report2)
```

```
Classification Report:
               precision    recall  f1-score   support

          -1       0.90      0.86      0.88      3244
           0       0.87      0.89      0.88      2910
           1       0.92      0.95      0.94      2933

    accuracy                           0.90      9087
   macro avg       0.90      0.90      0.90      9087
weighted avg       0.90      0.90      0.90      9087
```

DecisionTree Classifier

In [41]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [42]:

```
tree=DecisionTreeClassifier()
tree.fit(x_train,y_train)
```

Out[42]:

```
▾ DecisionTreeClassifier

DecisionTreeClassifier()
```

In [43]:

```
y_pred3=tree.predict(x_test)
y_pred3
```

Out[43]:

```
array([1, 1, 0, ..., 1, 1, 1], dtype=int64)
```

In [44]:

```
confusion_matrix(y_pred3,y_test)
```

Out[44]:

```
array([[2283,  473,  194],
       [ 559, 2293,  263],
       [ 250,  209, 2563]], dtype=int64)
```

In [45]:

```
accuracy_score(y_pred3,y_test)
```

Out[45]:

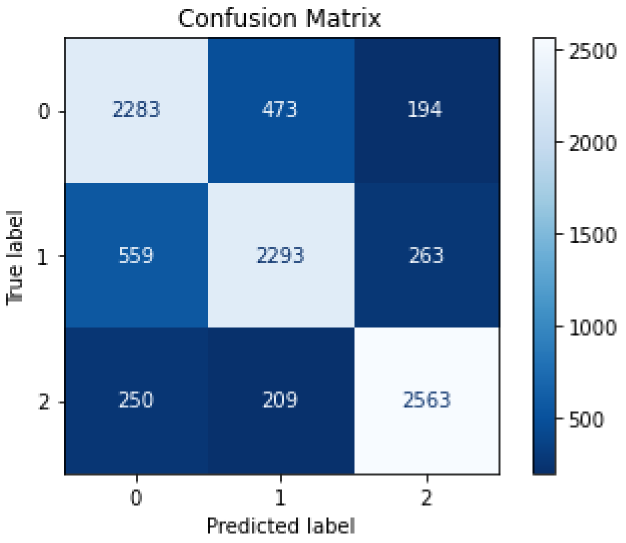
```
0.7856278199625839
```

In [46]:

```
cm3=confusion_matrix(y_pred3,y_test)
ConfusionMatrixDisplay(cm3).plot(cmap='Blues_r')
plt.title('Confusion Matrix')
```

Out[46]:

```
Text(0.5, 1.0, 'Confusion Matrix')
```



In [47]:

```
from sklearn.metrics import classification_report
report3=classification_report(y_pred3,y_test)
print("Classification Report:\n", report3)
```

```
Classification Report:
               precision    recall  f1-score   support

          -1       0.74      0.77      0.76      2950
           0       0.77      0.74      0.75      3115
           1       0.85      0.85      0.85      3022

    accuracy                           0.79      9087
   macro avg       0.79      0.79      0.79      9087
weighted avg       0.79      0.79      0.79      9087
```

Support Vector Classifier

In [48]:

```
from sklearn.svm import SVC
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(x_train, y_train)
```

Out[48]:

```
▾                     SVC
SVC(kernel='linear', random_state=42)
```

In [49]:

```
y_pred4 = svm_classifier.predict(x_test)
confusion_matrix(y_pred4,y_test)
```

Out[49]:

```
array([[2557,  236,   72],
       [ 416, 2618,  124],
       [ 119,  121, 2824]], dtype=int64)
```

In [50]:

```
accuracy_score(y_test,y_pred4)
```

Out[50]:

```
0.8802685154616485
```

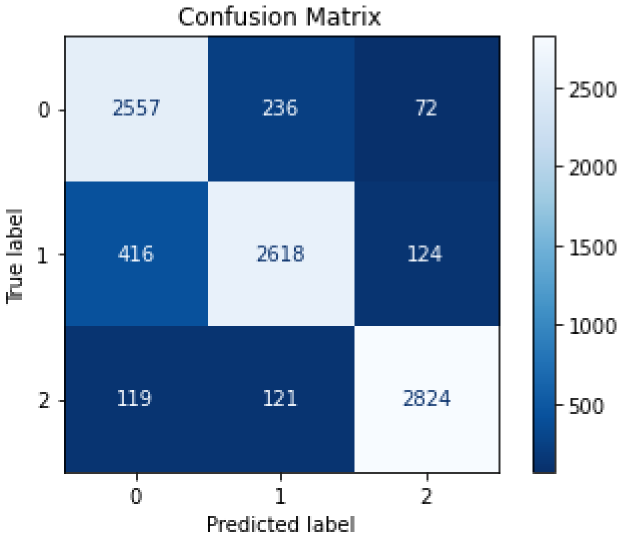
In [51]:

```
cm4=confusion_matrix(y_pred4,y_test)
ConfusionMatrixDisplay(cm4).plot(cmap='Blues_r')
```

```
plt.title('Confusion Matrix')
```

Out[51]:

```
Text(0.5, 1.0, 'Confusion Matrix')
```



In [52]:

```
from sklearn.metrics import classification_report
report4=classification_report(y_pred4,y_test)
print("Classification Report:\n", report4)
```

```
Classification Report:
               precision    recall  f1-score   support

          -1       0.83      0.89      0.86      2865
           0       0.88      0.83      0.85      3158
           1       0.94      0.92      0.93      3064

    accuracy                           0.88      9087
   macro avg       0.88      0.88      0.88      9087
weighted avg       0.88      0.88      0.88      9087
```

In [53]:

```
from prettytable import PrettyTable
table=PrettyTable()
table.field_names = ["TEST_SIZE","0.33"]
table.add_row(["Logistic Regression",0.8595796192362716])
table.add_row(["Random Forest Classifier",0.898096181357984])
table.add_row(["DecisionTree Classifier",0.7856278199625839])
table.add_row(["Support Vector Classifier",0.8802685154616485])
print(table)
```

```
+---------------------------+--------------------+
|         TEST_SIZE         |        0.33        |
+---------------------------+--------------------+
|    Logistic Regression    | 0.8595796192362716 |
|  Random Forest Classifier | 0.898096181357984  |
|  DecisionTree Classifier  | 0.7856278199625839 |
| Support Vector Classifier | 0.8802685154616485 |
+---------------------------+--------------------+
```

In [54]:

```
import tkinter as tk
from tkinter import ttk
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import pandas as pd

def analyze_sentiment():
    text = text_entry.get("1.0", "end-1c").strip()  # Get text from the Text widget
    if text:
        analyzer = SentimentIntensityAnalyzer()
        sentiment_scores = analyzer.polarity_scores(text)
        sentiment = sentiment_scores['compound']
        if sentiment >= 0.05:
            result_label.config(text="Positive", foreground="green")
        elif sentiment <= -0.05:
            result_label.config(text="Negative", foreground="red")
        else:
            result_label.config(text="Neutral", foreground="blue")
    else:
        result_label.config(text="Please enter text!", foreground="orange")

data = pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\sentiment analysis fo
window = tk.Tk()
window.title('Sentiment Analysis')

text_label = ttk.Label(window, text='Enter text:')
text_label.pack()

text_entry = tk.Text(window, height=5, width=50)
text_entry.pack()

analyze_button = ttk.Button(window, text='Analyze Sentiment', command=analyze_sentim
analyze_button.pack()

result_label = ttk.Label(window, text='')
result_label.pack()
window.mainloop()
```

In [ ]: