Pune Institute of Computer Technology

Department of Computer Engineering

(2022- 2023)

"Tweet Analysis System"

Submitted to the

Savitribai Phule Pune University

In partial fulfilment for the award of the Degree of

Bachelor of Engineering

in

Computer Engineering

By

1) **Sushant Awathare** 41107

2) **Vaishnavi Bhujbal** 41114

3) **Udayan Chavan** 41117

Under the guidance of

Prof. Anjali Deshpande

Problem Statement

Develop a Tweet Analysis System.

Objective

To build a classification system that classifies tweets by twitter user as having a positive or negative sentiment using information retrieval and machine learning.

Theory

Sentiment Analysis:

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.

Machine Learning:

Machine learning is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. Here, we use it to analyse what makes a tweet positive or negative and thus predict future tweets as positive or negative.

Logistic Regression Model:

It is a statistical model often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn’t vote, based on a given dataset of independent variables. Here we use it to classify tweets as positive or negative.

Libraries used:

- **Pandas:** Used for performing data manipulation and analytics with Python.
- **Seaborn:** Used for data visualization and plotting graphics using Python.
- **Sklearn:** A machine learning library that has models and evaluation tools.

CODE:

LP4 - IR Mini-Project

Develop a Tweet Analysis System.

Group members:

- **41107** - Sushant Awathare
- **41114** - Vaishnavi Bhujbal
- **41117** - Udayan Chavan

Import all dependencies

In [ ]:

```
import re
import numpy as np
import pandas as pd
import string

# seaborn and matplotlib for visualization
import seaborn as sns
import matplotlib.pyplot as plt

# NLTK for NLP
import nltk
nltk.download('wordnet')

# sklearn for Machine Learning
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
```

Load Dataset and create column names

In [2]:

```
cols=['target','ids','date','flag','user','text']
enc = "ISO-8859-1"

df = pd.read_csv('tweets.csv', encoding=enc, names=cols)
df = df.iloc[799000:801000,]
df.head()
```

	target	ids	date	flag	user	text
799000	0	2328818606	Thu Jun 25 10:01:34 PDT 2009	NO_QUERY	redstar72	CAN'T BEAT LIVE MUSIC, WISH I COULD SING BUT I...
799001	0	2328818761	Thu Jun 25 10:01:35 PDT 2009	NO_QUERY	LIDA360	Charlie lost an angel today Very sad http:...
799002	0	2328818847	Thu Jun 25 10:01:35 PDT 2009	NO_QUERY	rnspires21	at work, and very bored
799003	0	2328819310	Thu Jun 25 10:01:37 PDT 2009	NO_QUERY	lasthonestlook	It's weird how celebrities go in threes. Carra...
799004	0	2328819347	Thu Jun 25 10:01:37 PDT 2009	NO_QUERY	deiann	@adelate Farrah's dead? Had no idea.. RIP Farr...

General information

In [3]:

```
len(df)
```

```
2000
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 799000 to 800999
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   target  2000 non-null   int64 
 1   ids     2000 non-null   int64 
 2   date    2000 non-null   object
 3   flag    2000 non-null   object
 4   user    2000 non-null   object
 5   text    2000 non-null   object
dtypes: int64(2), object(4)
memory usage: 93.9+ KB
```

In [5]:
```
np.sum(df.isnull().any(axis=1))
```

```
0
```

Check the number of unique values in target

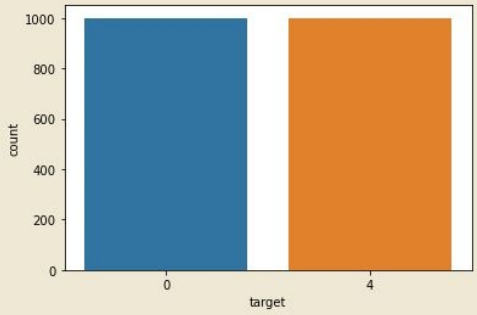
In [6]:
```
df['target'].unique()

```

```
array([0, 4], dtype=int64)
```

According to dataset, 0: negative sentiment and 1: positive sentiment

In [7]:
```
sns.countplot(x='target', data=df)
```

```
<AxesSubplot:xlabel='target', ylabel='count'>
```



Select text and target column for analysis

In [8]:
```
data=df[['text','target']].copy()
```

In [9]:
```
data.head()
```

	text	target
799000	CAN'T BEAT LIVE MUSIC, WISH I COULD SING BUT I...	0
799001	Charlie lost an angel today Very sad http:...	0
799002	at work, and very bored	0
799003	It's weird how celebrities go in threes. Carra...	0
799004	@adelate Farrah's dead? Had no idea.. RIP Farr...	0

In [10]:
```
data['target'] = data['target'].replace([4],1) # replace 4 with 1 for better understanding
```

In [11]:
```
data.tail()
```

	text	target
800995	I have this strange desire to go to confession...	1
800996	@i_reporter answer sent in dm. try it	1
800997	@brooklynunion cuz ur 3pm is my 9am and Id be ...	1
800998	@littrellfans Its all good. Just figured you w...	1
800999	@nicolerichie Yea I remember it	1

Separate positive and negative tweets

In [12]:
```
data_pos = data[data['target'] == 1].copy()
data_neg = data[data['target'] == 0].copy()
```

In [13]:
```
data_pos.head()
```

	text	target
800000	I LOVE @Health4UandPets u guys r the best!!	1
800001	im meeting up with one of my besties tonight! ...	1
800002	@DaRealSunisaKim Thanks for the Twitter add, S...	1
800003	Being sick can be really cheap when it hurts t...	1
800004	@LovesBrooklyn2 he has that effect on everyone	1

In [14]:
```
data_neg.head()
```

	text	target
799000	CAN'T BEAT LIVE MUSIC, WISH I COULD SING BUT I...	0
799001	Charlie lost an angel today Very sad http:...	0
799002	at work, and very bored	0
799003	It's weird how celebrities go in threes. Carra...	0
799004	@adelate Farrah's dead? Had no idea.. RIP Farr...	0

In [15]:
```
dataset = pd.concat([data_pos, data_neg]) # Combine positive and negative tweets
```

In [16]:
```
dataset.head()
```

	text	target
800000	I LOVE @Health4UandPets u guys r the best!!	1
800001	im meeting up with one of my besties tonight! ...	1
800002	@DaRealSunisaKim Thanks for the Twitter add, S...	1
800003	Being sick can be really cheap when it hurts t...	1
800004	@LovesBrooklyn2 he has that effect on everyone	1

In [17]:
```
dataset.tail()
```

	text	target
799995	Sick Spending my day laying in bed listening ...	0
799996	Gmail is down?	0
799997	rest in peace Farrah! So sad	0
799998	@Eric_Urbane Sounds like a rival is flagging y...	0
799999	has to resit exams over summer... wishes he w...	0

Change text to lowercase

In [18]:
```
dataset['text']=dataset['text'].str.lower()
dataset.head()
```

	text	target
800000	i love @health4uandpets u guys r the best!!	1
800001	im meeting up with one of my besties tonight! ...	1
800002	@darealsunisakim thanks for the twitter add, s...	1
800003	being sick can be really cheap when it hurts t...	1
800004	@lovesbrooklyn2 he has that effect on everyone	1

Create a list containing stop words

In [19]:
```
stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
             'and','any','are', 'as', 'at', 'be', 'because', 'been', 'before',
             'being', 'below', 'between','both', 'by', 'can', 'd', 'did', 'do',
             'does', 'doing', 'down', 'during', 'each','few', 'for', 'from',
             'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
             'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
             'into','is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
             'me', 'more', 'most','my', 'myself', 'now', 'o', 'of', 'on', 'once',
             'only', 'or', 'other', 'our', 'ours','ourselves', 'out', 'own', 're','s', 'same', 'she', "
             't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
             'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
             'through', 'to', 'too','under', 'until', 'up', 've', 'very', 'was',
             'we', 'were', 'what', 'when', 'where','which','while', 'who', 'whom',
             'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
             "youve", 'your', 'yours', 'yourself', 'yourselves']
```

Remove stop words from the text

In [20]:
```
STOPWORDS = set(stopwordlist)
def cleaning_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])

dataset['text'] = dataset['text'].apply(lambda text: cleaning_stopwords(text))
dataset['text'].head()
```

```
800000              love @health4uandpets u guys r best!!
800001    im meeting one besties tonight! cant wait!! - ...
800002    @darealsunisakim thanks twitter add, sunisa! g...
800003    sick really cheap hurts much eat real food plu...
800004                      @lovesbrooklyn2 effect everyone
Name: text, dtype: object
```

Remove punctuation marks from text

In [21]:
```
english_punctuations = string.punctuation
punctuations_list = english_punctuations

def cleaning_punctuations(text):
    translator = str.maketrans('', '', punctuations_list)
    return text.translate(translator)

dataset['text']= dataset['text'].apply(lambda x: cleaning_punctuations(x))
dataset['text'].head()
```

```
800000                 love health4uandpets u guys r best
800001    im meeting one besties tonight cant wait  girl...
800002    darealsunisakim thanks twitter add sunisa got ...
800003    sick really cheap hurts much eat real food plu...
800004                      lovesbrooklyn2 effect everyone
Name: text, dtype: object
```

Remove URL data from tweets

In [22]:
```
def cleaning_URLs(data):
    return re.sub('((www.[^s]+)|(https?://[^s]+))',' ',data)

dataset['text'] = dataset['text'].apply(lambda x: cleaning_URLs(x))
dataset['text'].head()
```

```
800000                love health4uandpets u guys r best
800001    im meeting one besties tonight cant wait  girl...
800002    darealsunisakim thanks twitter add sunisa got ...
800003    sick really cheap hurts much eat real food plu...
800004                     lovesbrooklyn2 effect everyone
Name: text, dtype: object
```

Remove numeric data from tweets

In [23]:
```
def cleaning_numbers(data):
    return re.sub('[0-9]+', '', data)

dataset['text'] = dataset['text'].apply(lambda x: cleaning_numbers(x))
dataset['text'].head()
```

```
800000                  love healthuandpets u guys r best
800001    im meeting one besties tonight cant wait  girl...
800002    darealsunisakim thanks twitter add sunisa got ...
800003    sick really cheap hurts much eat real food plu...
800004                      lovesbrooklyn effect everyone
Name: text, dtype: object
```

Tokenize tweet text

In [24]:
```
from nltk.tokenize import RegexpTokenizer
tokenizer = RegexpTokenizer(r'\w+')

dataset['text'] = dataset['text'].apply(tokenizer.tokenize)
dataset['text'].head()
```

```
800000           [love, healthuandpets, u, guys, r, best]
800001    [im, meeting, one, besties, tonight, cant, wai...
800002    [darealsunisakim, thanks, twitter, add, sunisa...
800003    [sick, really, cheap, hurts, much, eat, real, ...
800004                  [lovesbrooklyn, effect, everyone]
Name: text, dtype: object
```

Apply lemmatization

In [25]:
```
lm = nltk.WordNetLemmatizer()
def lemmatizer_on_text(data):
    text = [lm.lemmatize(word) for word in data]
    return data

dataset['text'] = dataset['text'].apply(lambda x: lemmatizer_on_text(x))
dataset['text'].head()
```

```
800000           [love, healthuandpets, u, guys, r, best]
800001    [im, meeting, one, besties, tonight, cant, wai...
800002    [darealsunisakim, thanks, twitter, add, sunisa...
800003    [sick, really, cheap, hurts, much, eat, real, ...
800004                  [lovesbrooklyn, effect, everyone]
Name: text, dtype: object
```

Separate input feature and label

In [26]:
```
X=data.text
Y=data.target
```

In [27]:
```
print(X)
print("\n------------------\n")
print(Y)
```

```
799000    CAN'T BEAT LIVE MUSIC, WISH I COULD SING BUT I...
799001    Charlie lost an angel today   Very sad   http:...
799002                                 at work, and very bored
799003    It's weird how celebrities go in threes. Carra...
799004    @adelate Farrah's dead? Had no idea.. RIP Farr...
                                ...
800995    I have this strange desire to go to confession...
800996               @i_reporter answer sent in dm. try it
800997    @brooklynunion cuz ur 3pm is my 9am and Id be ...
800998    @littrellfans Its all good. Just figured you w...
800999                       @nicolerichie Yea I remember it
Name: text, Length: 2000, dtype: object

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

799000    0
799001    0
799002    0
799003    0
799004    0
         ..
800995    1
800996    1
```

Split dataset for training and testing

In [28]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size = 0.8, random_state=10)
```

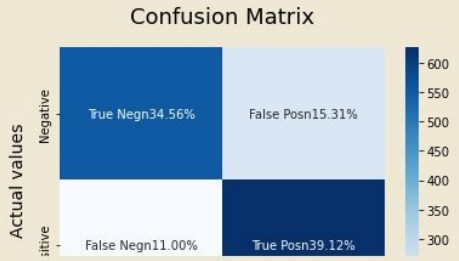
Function to evaluate the model

In [32]:
```
def model_Evaluate(model):
# Predict values for Test dataset
    Y_pred = model.predict(X_test)

    # Print the evaluation statistics
    print(classification_report(Y_test, Y_pred))

    # Plot confusion matrix as a heatmap
    cf_matrix = confusion_matrix(Y_test, Y_pred)
    categories = ['Negative','Positive']
    group_names = ['True Neg','False Pos', 'False Neg','True Pos']
    group_percentages = ['{0:.2%}'.format(value) for value in cf_matrix.flatten() / np.sum(cf_matrix)]
    labels = [f'{v1}n{v2}' for v1, v2 in zip(group_names,group_percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(cf_matrix, annot = labels, cmap = 'Blues',fmt = '',
    xticklabels = categories, yticklabels = categories)
    plt.xlabel("Predicted values", fontdict = {'size':14}, labelpad = 10)
    plt.ylabel("Actual values" , fontdict = {'size':14}, labelpad = 10)
    plt.title ("Confusion Matrix", fontdict = {'size':18}, pad = 20)
```

Creating a Logisitic Regression Model

In [33]:
```
LRmodel = LogisticRegression(C = 2, max_iter = 1000)

LRmodel.fit(X_train, Y_train)
model_Evaluate(LRmodel)

Y_pred = LRmodel.predict(X_test)
```

	precision	recall	f1-score	support
0	0.76	0.69	0.72	798
1	0.72	0.78	0.75	802
accuracy			0.74	1600
macro avg	0.74	0.74	0.74	1600
weighted avg	0.74	0.74	0.74	1600



In [34]:
```
test_data_accuracy = accuracy_score(Y_test,Y_pred)
print('Accuracy score of test data : ', test_data_accuracy)
```

```
Accuracy score of test data :  0.736875
```

Conclusion

Hence we successfully implemented a Tweet Analysis System and classified tweets into positive and negative sentiment holding tweets using Information Retrieval and Machine Learning.