Web Scraping and Sentiment Analysis

by Honey bhardwaj

Submission date: 22-Jul-2020 02:58PM (UTC+0530)
Submission ID: 1360457837
File name: raping_and_Sentiment_Analysis_PS2_Honey_Bhardwaj_1800221C203.pdf (855.47K)
Word count: 1878
Character count: 9711

A REPORT

ON

WEB SCRAPING AND SENTIMENT ANALYSIS FOR HOTELS REVIEWS OF WWW.YELP.COM

By

Name of the Student	**Enrolment/Registration No.**
Honey Bhardwaj	1800221C203

Prepared in the partial fulfillment of the

Practice School II Course

AT

BML MUNJAL UNIVERSITY

National Highway 8, 67, KM Milestone, Gurugram, Haryana 122413

A Practice School II Station of

BML MUNJAL UNIVERSITY
FROM HERE TO THE WORLD

BML MUNJAL UNIVERSITY

(May 2020 -July 2020)

BML MUNJAL UNIVERSITY™

Table of Contents

BML MUNJAL UNIVERSITY™

Table of Figures

Certificate of authenticity

CERTIFICATE

This is to certify that Practice School Project of _____Honey Bhardwaj_____ titled _____Web Scraping and Sentiment Analysis For Hotels Reviews of www.yelp.com_____is an original work and that this work has not been submitted anywhere in any form. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried during ___9 july 2020_____ to _____23 july 2020_____ in BML MUNJAL UNIVERSITY.

Signature of PS-II faculty	**Signature of industry mentor/Supervisor**
Name:	**Name:**
Designation:	**Designation:**
(Seal of the organization with Date)	***(Seal of the organization with Date)***

ACKNOWLEDGEMENT

The Project was made successful by guidance and help from a lot of people. I would like to thank Prof. (Dr) Maneek Kumar and Prof. Manoj K. Arora . I would like to thank Dr. Maheshwar Dwivedi for providing the assistance and solving all the problems faced Moreover like to thank Mr. Alipta Dutta for contacting me to solve the problems that I was facing. I would like to thank Ms. Jyoti pruthi and Dr. kiran khattar for supporting me at every stage of the project and guiding me at each and every point of the project. Lastly, I would like to thank all the people whom i contacted for help and assistance from linkedin and forums and other communities. The people mentioned above had helped me a lot and without them this project won't be possible. Also i would like to thank Bml munjal university and all teams that were involved in helping us during PS2.

Objective

The main objective is to learn , explore and create a project implementing whatever learned during a course of time.

The project is Web Scraping and Sentiment Analysis For Hotels Reviews of www.yelp.com. This project have 2 sub parts in which work was carried out and learning was done accordingly:

1. WEB SCRAPING:

   This part deals with the creation of an autonomous system that can scrap the large amount of data that is Reviews in this case. The system uses some automation tools for automating the tasks that are required without the intervention of humans. Without this program it would take days to save the data collected by humans and doing this process.

2. SENTIMENT ANALYSIS

   This part deals with the analysis of the reviews collected by the user and classify them as positive or a negative sentiment. This is generally done to check the sentiment of a product by the terms of feedback provided by the user. In this system a model is created that uses some classification algorithm and the accuracy of prediction is done. This Analysis helps the companies to test the long set of data about their product and can then check whether its going good or bad.

BML MUNJAL UNIVERSITY™

Problem Statement

By some estimations it is calculated that 80% of the data is unstructured and present in the form of text, surveys, emails etc. These Data are generated each day and are very difficult to analyse , understand and to process such a huge data it is very expensive and time consuming. Understanding customer opinions and feedback is a very crucial part to grow any business since by analysing their feedback it is very easy to get back to customers with a proper service and needs that are required by them. So sentiment analysis is the process or a machine learning technique used to detect the polarity from the text data into positive or negative. it can be a best way to analyse huge data of any document or set of documents. it helps businesses to have a meaning out of that textual feedback which could help them in knowing more about the product and services. The title of the project is Web Scraping and Sentiment analysis for hotels of www.yelp.com. yelp.com is a website which publishes reviews about different businesses such as spa, restaurants etc. The project deals with creating a machine learning model which predicts the user reviews as positive or negative since we know that in any business it is a very crucial part to track their feedback of the product they serve to users by the user's review and by this product they will be able to check the sentiment of user towards their product by means of reviews in form of text by using some text analysis techniques.

Workflow of the project:

- choosing of training dataset from kaggle
- scraping data from yelp.com
- importing datasets
- data preparation and cleaning
- bag of words and count vectorization (stemming and stopwords removal)
- calculation of tf-idf values
- datasets splitting
- training classification models using training set
- checking accuracy for different model

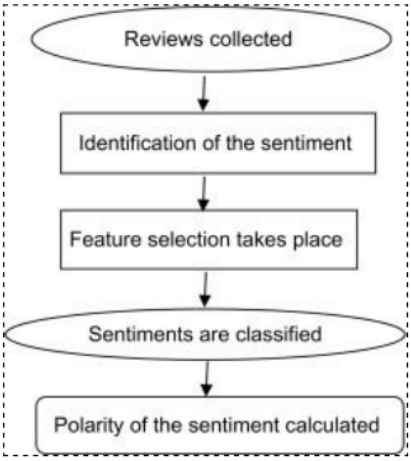


FIG 1: Flow Chart of workflow

BML MUNJAL UNIVERSITY™

Project Methodology

Fig 2: Project Methods

The Project involved the above tasks that needed to be done for the sentiment analysis. These steps had a lot of work in each and the explanation of all task carried are as follows:

1. **Data collection :** This is the first Task of the project which included searching of a perfect dataset from kaggle used for training the model this dataset included a lot of meta data collected together apart of this the data was collected from www.yelp.com by the data scraping program which uses selenium web driver. This data was for the testing purpose of the model.

2. **Text Preparation :** This Task included the preparation of the data which needs to be done . the following were the processes involved in this task:
   - removing unused columns
   - filling the null values
   - concatenating the dataframes
   - removing punctuations
   - removing emojis
   - removing stopwords
   - tokenization
   - stemming
   - calculating tf-idf values

3. **Sentiment Detection :** This task deals with the creation for labels according to the star rating given by the user on their feedback. This uses a comparison for creating labels for the sentiment where 1 is for positive and -1 for negative sentiment .

4. **Sentiment Classification :** This task involved creating a machine learning model which uses some classification algorithm in their background. They classify them by the features matrix feeded to them which contains the numerical conversion of documents that contain all the textual feedback. In this task I have used 3 different models which I compare at the end to check which one is better. The models used are Support vector machine, logistic regression and naives bayes.

5. **Presentation of output :** The output presentation has been done by plotting some charts to show data such as pie charts and the predicted result have been shown by plotting confusion matrix for each model moreover a classification report and accuracy score is printed for each model.

BML MUNJAL UNIVERSITY™

Result & Discussion

The Data Scraping program was successful in collecting reviews and has collected 5195 amounts of individual reviews with other details such as hotel url, hotel name , star rating of reviews. and stored the following in csv format.

There are basically 3 approaches in doing sentimental analysis for any group of data which are:

1. rule based
2. machine learning approach
3. hybrid

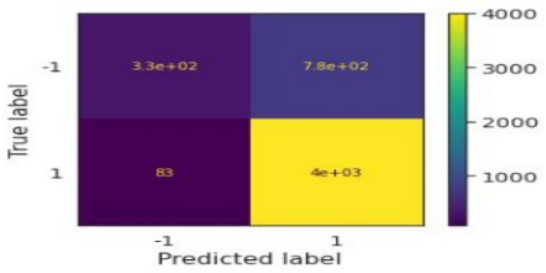
Out of all 3 mentioned above, I have used the 2nd approach in solving my project which have lead me to the following results:

- AMOUNT OF DATA IN TRAIN.CSV: 10000
- AMOUNT OF DATA IN TEST.CSV: 5195
- TOTAL DATA: 15195
- ACCURACY OF SVM: 83.11 %
- ACCURACY OF NAIVES BAYES: 79.13 %
- ACCURACY OF LOGISTIC REGRESSION: 83.36 %

Classification reports and confusion matrix:

1. **logistic regression**

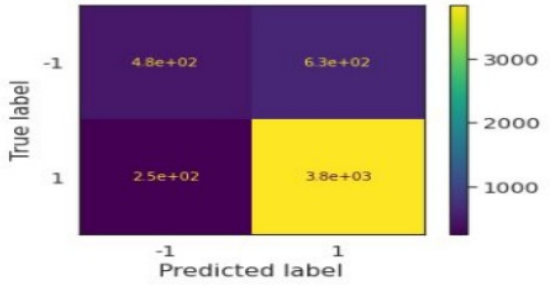
```
[[ 328  781]
 [  83 4003]]
              precision    recall  f1-score   support

          -1       0.80      0.30      0.43      1109
           1       0.84      0.98      0.90      4086

    accuracy                           0.83      5195
   macro avg       0.82      0.64      0.67      5195
weighted avg       0.83      0.83      0.80      5195

0.8336862367661213
```



BML MUNJAL UNIVERSITY™

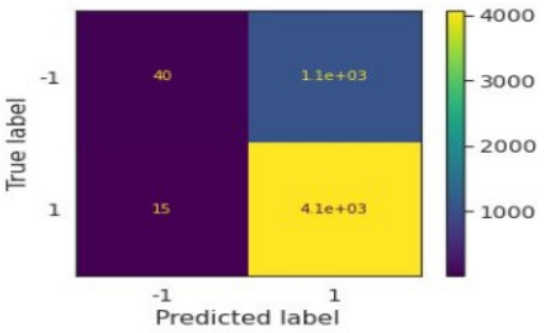
2. Support Vector Machine:

```
[[ 478  631]
 [ 246 3840]]
              precision    recall  f1-score   support

          -1       0.66      0.43      0.52      1109
           1       0.86      0.94      0.90      4086

    accuracy                           0.83      5195
   macro avg       0.76      0.69      0.71      5195
weighted avg       0.82      0.83      0.82      5195

0.8311838306063523
```



3. Naives Bayes:

```
[[  40 1069]
 [  15 4071]]
              precision    recall  f1-score   support

          -1       0.73      0.04      0.07      1109
           1       0.79      1.00      0.88      4086

    accuracy                           0.79      5195
   macro avg       0.76      0.52      0.48      5195
weighted avg       0.78      0.79      0.71      5195

0.7913378248315688
```



BML MUNJAL UNIVERSITY™

Appendices

1. **List of Mathematical Formulae:**

- tf-idf

$$\text{tf-idf}(t, d) = \text{tf}(t,d) \times \text{idf}(t, d)$$

$$\text{idf}(t, d) = \log \frac{n_d}{1 + \text{df}(d, t)},$$

where n_d is the total number of documents, and df(d, t) is the number of documents d that contain the term t.

Fig 3 : Tf-Idf

2. **Source code -> The Source code and the functionality are as follows:**

- DATA SCRAPING :

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import csv

driver=webdriver.Chrome('/home/honey/Desktop/the right doctors/chromedriver')
driver.get("https://www.yelp.com/")
search =driver.find_element_by_id("find_desc")
search_location=driver.find_element_by_id("dropperText_Mast")

x=input("enter what you want to find:")
y=input("enter location:")
search.send_keys(x)
search_location.send_keys(Keys.CONTROL + "a")
search_location.send_keys(y)
search.send_keys(Keys.RETURN)

hotels_urls=[]
hotels=[]
for i in range(6,36):
    hotels.append(driver.find_element_by_xpath("/html/body/div[2]/div[3]/div[2]/div/div[1]/div[1]/div[2]/div[2]/ul/li[
    hotels_urls.append(driver.find_element_by_xpath("/html/body/div[2]/div[3]/div[2]/div/div[1]/div[1]/div[2]/div[2]/u

reviews=[]
ratings=[]
name=[]
url_data=[]
```

Fig 4.1 : Datascrap.py

BML MUNJAL UNIVERSITY™

```
for url in hotels_urls[1:]:
    driver.get(url)
    try:

        for j in range(10):
            time.sleep(3)
            for i in range(1,21):
                url_data.append(url)
                name.append(driver.find_element_by_xpath("/html/body/div[2]/div[4]/div/div[4]/div/div/div[2]/div/div/d
                ratings.append(driver.find_element_by_xpath("/html/body/div[2]/div[4]/div/div[4]/div/div/div[2]/div/di
                try:
                    reviews.append(driver.find_element_by_xpath("/html/body/div[2]/div[4]/div/div[4]/div/div/div[2]/di
                except:
                    reviews.append(driver.find_element_by_xpath("/html/body/div[2]/div[4]/div/div[4]/div/div/div[2]/di
                print("data extracted.......")

            driver.find_element_by_xpath("/html/body/div[2]/div[4]/div/div[4]/div/div/div[2]/div/div/div[1]/div/div[1]
    except:
        continue
driver.close()


print(len(name))
print(len(reviews))
print(len(ratings))
print(len(url_data))
with open('test.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["Hotel Name", "reviews.text" , "reviews.rating", "hotel Url"])
    for j in range(0 , len(name)):
        try:
            writer.writerow([name[j], reviews[j], ratings[j], url_data[j]])
        except:
            continue
        print("details updated on csv.....")
```

Fig 4.2: DataScrap.py

- SENTIMENT ANALYSIS:

Importing Dataset and Libraries

In [1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Set global styles for plots
sns.set_style(style='white')
sns.set_context(context='notebook', font_scale=1.3, rc={'figure.figsize': (16,9)})

df1=pd.read_csv("/content/drive/My Drive/train.csv")
df2=pd.read_csv("/content/drive/My Drive/test.csv")

df2.head()
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is dep
recated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

Out[1]:

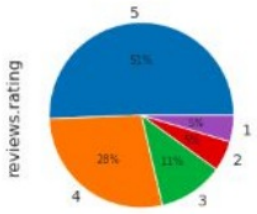
	Hotel Name	reviews.text	reviews.rating	hotel Url
0	Jacob's Pickles	It has been months since I have had Jacob's Pi...	5	https://www.yelp.com/biz/jacobs-pickles-new-yo...
1	Jacob's Pickles	I don't think I even have the words. This plac...	5	https://www.yelp.com/biz/jacobs-pickles-new-yo...
2	Jacob's Pickles	This place is very good; it's just a little he...	5	https://www.yelp.com/biz/jacobs-pickles-new-yo...
3	Jacob's Pickles	Customer service : super friendly. If they had...	5	https://www.yelp.com/biz/jacobs-pickles-new-yo...
4	Jacob's Pickles	Ordered brunch delivery during COVID pandemic ...	4	https://www.yelp.com/biz/jacobs-pickles-new-yo...

Fig 5: Importing data set and creating dataframe

BML MUNJAL UNIVERSITY™

```
In [3]: df2["reviews.rating"].value_counts().plot(kind='pie', autopct='%1.0f%%')
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc436653240>
```




```
In [4]: df1["reviews.rating"].value_counts().plot(kind='pie', autopct='%1.0f%%')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc436130a20>
```

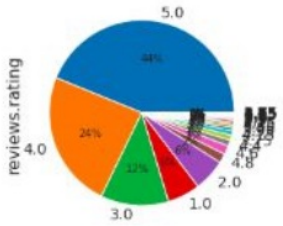


Fig 6: Displaying star rating of both dataframe

Removing the Unused Colums

```
In [5]: df1=df1.drop(['id', 'dateAdded', 'dateUpdated', 'address', 'categories',
       'primaryCategories', 'city', 'country', 'keys', 'latitude', 'longitude',
       'name', 'postalCode', 'province', 'reviews.date', 'reviews.dateSeen',
       'reviews.sourceURLs', 'reviews.title','reviews.userCity', 'reviews.userProvince', 'reviews.username',
       'sourceURLs', 'websites'], axis=1)
df1
```

Out[5]:

	reviews.rating	reviews.text
0	5.0	Our experience at Rancho Valencia was absolute...
1	5.0	Amazing place. Everyone was extremely warm and...
2	5.0	We booked a 3 night stay at Rancho Valencia to...
3	2.0	Currently in bed writting this for the past hr ...
4	5.0	I live in Md and the Aloft is my Home away fro...
...	...	...
9995	3.0	It is hard for me to review an oceanfront hote...
9996	4.0	I live close by, and needed to stay somewhere ...
9997	4.0	Rolled in 11:30 laid out heads down woke up to...
9998	1.0	Absolutely terrible..I was told I was being gi...
9999	1.0	Filthy, outdated, noisy neighbours, but this w...

10000 rows × 2 columns

Fig 7.1 : Data Cleaning

BML MUNJAL UNIVERSITY™

```
In [6]: df2=df2.drop(['Hotel Name','hotel Url'], axis=1)
        df2
```

Out[6]:

	reviews.text	reviews.rating
0	It has been months since I have had Jacob's Pi...	5
1	I don't think I even have the words. This plac...	5
2	This place is very good; it's just a little he...	5
3	Customer service : super friendly. If they had...	5
4	Ordered brunch delivery during COVID pandemic ...	4
...	...	...
5190	Love the atmosphere and the food is to die for...	5
5191	This is easily the best soul food out there (p...	5
5192	I'm so amazed about the service here Elliott a...	5
5193	The food was outstanding. As a real southern g...	5
5194	Had a lovely ladies brunch here. We couldn't m...	4

5195 rows × 2 columns

Fig 7.2: Data Cleaning

Cleaning Data

```
In [11]: import re
         def preprocessor(text):
             text = re.sub('<[^>]*>', '', text)
             emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)', text)
             text = re.sub('[\W]+', ' ', text.lower()) +\
                 ' '.join(emoticons).replace('-', '')
             return text

         df["reviews.text"]=df['reviews.text'].apply(preprocessor)
```

Fig 7.3: Data Cleaning

Creating Label Column in DataFrame

```
In [9]: a=[]
        for i in df["reviews.rating"]:
            if i <= 3:
                a.append(-1)
            else:
                a.append(1)
        df["label"]=a
        df
```

Out[9]:

	reviews.rating	reviews.text	label
0	5.0	Our experience at Rancho Valencia was absolute...	1
1	5.0	Amazing place. Everyone was extremely warm and...	1
2	5.0	We booked a 3 night stay at Rancho Valencia to...	1
3	2.0	Currently in bed writing this for the past hr ...	-1
4	5.0	I live in Md and the Aloft is my Home away fro...	1
...	...	...	...
15190	5.0	Love the atmosphere and the food is to die for...	1
15191	5.0	This is easily the best soul food out there (p...	1
15192	5.0	I'm so amazed about the service here Elliott a...	1
15193	5.0	The food was outstanding. As a real southern g...	1
15194	4.0	Had a lovely ladies brunch here. We couldn't m...	1

15195 rows × 3 columns

Fig8.1: Creating Labels

```
In [10]: df["label"].value_counts().plot(kind='pie', autopct='%1.0f%%')
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc436e260f0>
```

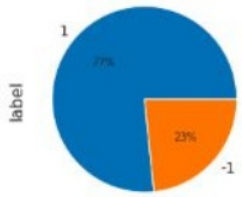


Fig 8.2: checking labels proportion

Adding Both DataFrames

```
In [7]: df = pd.concat([df1, df2],ignore_index=True)
        df
```

Out[7]:

	reviews.rating	reviews.text
0	5.0	Our experience at Rancho Valencia was absolute...
1	5.0	Amazing place. Everyone was extremely warm and...
2	5.0	We booked a 3 night stay at Rancho Valencia to...
3	2.0	Currently in bed writing this for the past hr ...
4	5.0	I live in Md and the Aloft is my Home away fro...
...	...	...
15190	5.0	Love the atmosphere and the food is to die for...
15191	5.0	This is easily the best soul food out there (p...
15192	5.0	I'm so amazed about the service here Elliott a...
15193	5.0	The food was outstanding. As a real southern g...
15194	4.0	Had a lovely ladies brunch here. We couldn't m...

15195 rows × 2 columns

Filling Null Colums

```
In [8]: df=df.fillna(" no review")
```

Fig 9.1: Data Preparation

```
In [12]: import nltk
         nltk.download('stopwords')
         [nltk_data] Downloading package stopwords to /root/nltk_data...
         [nltk_data]   Package stopwords is already up-to-date!
Out[12]: True
```

Fig 9.2: Downloading StopWords

```
In [13]: from sklearn.feature_extraction.text import CountVectorizer
         from nltk.corpus import stopwords
         from sklearn.feature_extraction.text import TfidfTransformer
         from nltk.stem.porter import PorterStemmer

         porter = PorterStemmer()

         def tokenizer(text):
             return text.split()


         def tokenizer_porter(text):
             return [porter.stem(word) for word in text.split()]

         vectorizer=CountVectorizer(stop_words=stopwords.words('english'), tokenizer=tokenizer_porter)
         X=vectorizer.fit_transform(df["reviews.text"].values)
         np.set_printoptions(precision=2)


         tfidf = TfidfTransformer(use_idf=True, norm='l2', smooth_idf=True)
         X=tfidf.fit_transform(X).toarray()

         /usr/local/lib/python3.6/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your stop_words may be
         inconsistent with your preprocessing. Tokenizing the stop words generated tokens ['abov', 'ani', 'becaus', 'befor',
         'doe', 'dure', 'ha', 'hi', "it'", 'onc', 'onli', 'ourselv', "she'", "should've", 'themselv', 'thi', 'veri', 'wa', 'w
         hi', "you're", "you've", 'yourselv'] not in stop_words.
           'stop_words.' % sorted(inconsistent))

In [14]: y=df["label"].values
```

Fig 9.3: Count Vectorization and Tf-Idf

Spliting testing and training datasets

```
In [15]: x_train=X[0:10000]
         x_test=X[10000:]
         y_train=df["label"].values[0:10000]
         y_test=df["label"].values[10000:]
```

Fig 10: Data Splitting

Creating Training Models

```
In [16]: from sklearn.linear_model import LogisticRegression
         LR = LogisticRegression()
         LR.fit(x_train, y_train)

Out[16]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, l1_ratio=None, max_iter=100,
                            multi_class='auto', n_jobs=None, penalty='l2',
                            random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                            warm_start=False)

In [17]: from sklearn import svm
         SVM = svm.LinearSVC()
         SVM.fit(x_train, y_train)

Out[17]: LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
                   intercept_scaling=1, loss='squared_hinge', max_iter=1000,
                   multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
                   verbose=0)

In [18]: from sklearn.naive_bayes import MultinomialNB
         nb = MultinomialNB()
         nb.fit(x_train, y_train)

Out[18]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

Fig 11: Creating Data Models

Checking Results

```
In [19]: pred=LR.predict(x_test)

In [20]: from sklearn.metrics import classification_report, confusion_matrix, accuracy_score


         print(confusion_matrix(y_test,pred))
         print(classification_report(y_test,pred))
         print(accuracy_score(y_test, pred))
```

```
[[ 328  781]
 [  83 4003]]
              precision    recall  f1-score   support

          -1       0.80      0.30      0.43      1109
           1       0.84      0.98      0.90      4086

    accuracy                           0.83      5195
   macro avg       0.82      0.64      0.67      5195
weighted avg       0.83      0.83      0.80      5195

0.8336862367661213
```

Fig 12.1: Checking Results

```
In [21]: pred=SVM.predict(x_test)
         from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

         print(confusion_matrix(y_test,pred))
         print(classification_report(y_test,pred))
         print(accuracy_score(y_test, pred))
```

```
[[ 478  631]
 [ 246 3840]]
              precision    recall  f1-score   support

          -1       0.66      0.43      0.52      1109
           1       0.86      0.94      0.90      4086

    accuracy                           0.83      5195
   macro avg       0.76      0.69      0.71      5195
weighted avg       0.82      0.83      0.82      5195

0.8311838306063523
```

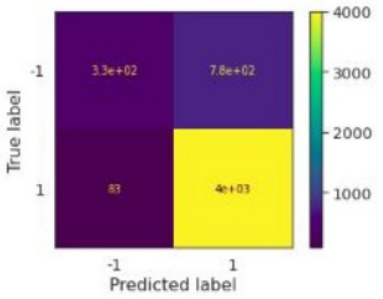
```
In [22]: pred=nb.predict(x_test)
         from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

         print(confusion_matrix(y_test,pred))
         print(classification_report(y_test,pred))
         print(accuracy_score(y_test, pred))
```

```
[[  40 1069]
 [  15 4071]]
              precision    recall  f1-score   support

          -1       0.73      0.04      0.07      1109
           1       0.79      1.00      0.88      4086

    accuracy                           0.79      5195
   macro avg       0.76      0.52      0.48      5195
weighted avg       0.78      0.79      0.71      5195

0.7913378248315688
```

Fig 12.2: checking Results

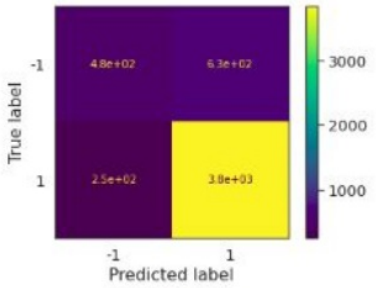
BML MUNJAL UNIVERSITY™

```
In [23]: from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(LR, x_test, y_test)
plt.show()
```




```
In [24]: from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(SVM, x_test, y_test)
plt.show()
```

Fig 12.3: Checking Results




```
In [25]: from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(nb, x_test, y_test)
plt.show()
```

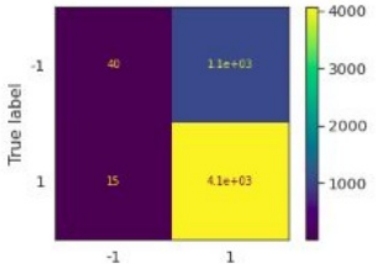


Fig 12.4: Checking Results

References

1. https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20is%20the%20interpretation,or%20services%20in%20online%20feedback.
2. https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17
3. https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python
4. https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184
5. https://www.learndatasci.com/tutorials/predicting-reddit-news-sentiment-naive-bayes-text-classifiers/
6. https://www.kaggle.com/datasets
7. https://scikit-learn.org/
8. https://www.coursera.org/learn/scikit-learn-logistic-regression-sentiment-analysis/home/welcome
9. https://www.coursera.org/learn/twitter-sentiment-analysis/home/welcome

Web Scraping and Sentiment Analysis

ORIGINALITY REPORT

7%	3%	1%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.akgim.edu.in Internet Source	3%
2	Submitted to University of Greenwich Student Paper	1%
3	Submitted to Pathfinder Enterprises Student Paper	1%
4	Submitted to Technische Universiteit Delft Student Paper	1%
5	Hiteshi Tandon, Tanmoy Chakraborty, Vandana Suhag. "A model of atomic compressibility and its application in QSAR domain for toxicological property prediction", Journal of Molecular Modeling, 2019 Publication	1%
6	Submitted to UNITEC Institute of Technology Student Paper	1%

Exclude quotes	On	Exclude matches	< 6 words
Exclude bibliography	On		