

Website for Analyzing Product Reviews

SOFTWARE ENGINEERING

CSE3001

By

Harish Bharadwaj – 18BCE0078

Adhil Mohammed – 18BCE0056

Under the guidance of

Prof. Manjula R

INDEX

1. ABSTRACT
2. KEYWORDS
3. INTRODUCTION
4. LITERATURE REVIEW
5. LIMITATIONS
6. PROPOSED SYSTEM
 - BLOCK DIAGRAM
 - WORK BREAKDOWN STRUCTURE
 - CLASS DIAGRAM
 - USE CASE DIAGRAM
 - SEQUENCE DIAGRAM
 - ACTIVITY DIAGRAM
 - STATECHART DIAGRAM
 - CODE SNIPPET
 - DESIGN OF TEST CASES
7. RESULT ANALYSIS AND DISCUSSION
8. CONCLUSION AND FUTURE ENHANCEMENT

Abstract

Nowadays online shopping is very popular. Millions of items are bought online everyday. Customers might face a problem of deciding whether a product is good or not. Some E-Commerce sites provide good quality products and some don't. This software can be used by customers who want to decide which product to buy. This software lets users compare each product by analyzing the reviews and telling how many people liked and disliked the product. This takes the link of e-commerce websites like flipkart and amazon as input. The software then scrapes the reviews from the link and then tells the users the percentage of people that liked that product based on the reviews. It can also be used by companies to check how their product is doing on the market. Two different ML algorithms are used - Naive Bayes and LSTM. The text is processed and passed to the model. The algorithm identifies the patterns based on the data passed. The accuracy of both the models are also compared.

Keywords

- LSTM - Long Short Term Memory Networks
- PRW- Product Review Website
- AI- Artificial Intelligence
- ML- Machine Learning
- NB - Naive Bayes

Introduction

It is an online website for analyzing reviews of a product. This takes the link of e-commerce websites like flipkart and amazon as input and then tell the users the percentage of people that liked that product based on the reviews. It also displays the details of the products like name. This helps the customer to decide which product is better. This software system is a Product Review Website(PRW) where a user can compare similar products based on the reviews given on other websites. PRW analyzes the user reviews of the product and provides a conclusion. Depending on the type of review the system classifies the reviews as either positive or negative.

Literature Review

Authors and Year	Title	Concept/Theoretical model/Framework	Methodology used/Implementation	Future Research
Liron Yao Yazhuo Guan	An Improved LSTM Structure for Natural Language Processing	LSTM	LSTM whose parameters are randomly discarded when they are passed backwards in the recursive projection layer.	Accuracy is 93%
Dr. Gorti Satyanarayana Murty, Shanmukha Rao Allu IJERT Vol 9 Issue 5 May 2020	Text based Sentiment Analysis using LSTM	LSTM with embedding layers	Movie review dataset that contains a total of 50,000 reviews out of which 25000 are positively polarized and 25000 are negatively polarized.	Different embedding models can be considered on large variety of the datasets.
Rohini V Merin Thomas IJSRD	Comparison of Lexicon based and Naïve Bayes Classifier in Sentiment Analysis	Naive Bayes and Lexicon Based Approach	Not mentioned	Lexicon Based 93% Naive Bayes 86%

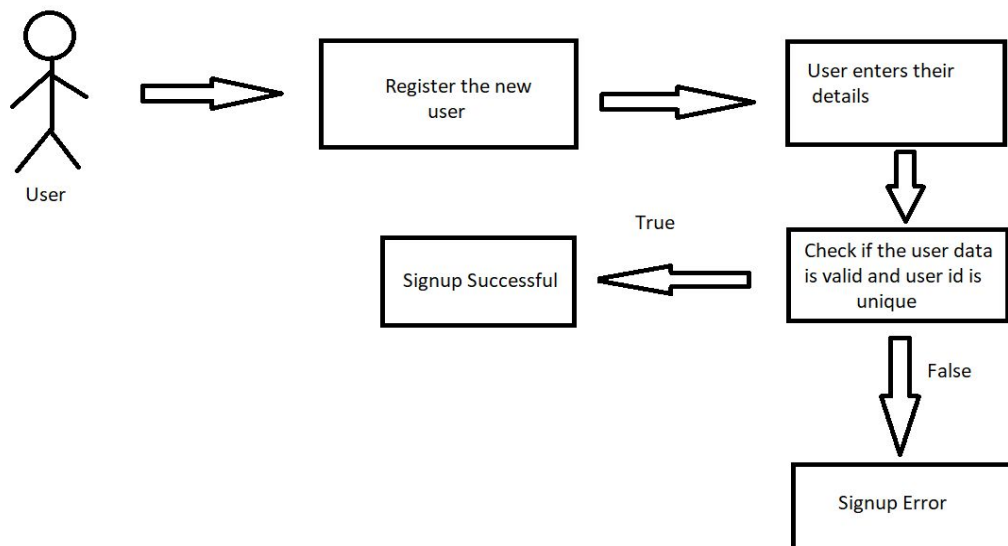
Piyush Lnu	Sentiment Analysis and Text Generation with LSTM	RNN	Scraped tweets from twitter	Improve by <ol style="list-style-type: none"> 1. We could increase the size of the dataset and scrap data across various social media platforms . 2. Fine Tuning the network architecture 3. Fine Tuning the network parameters
------------	--	-----	-----------------------------	--

Limitations

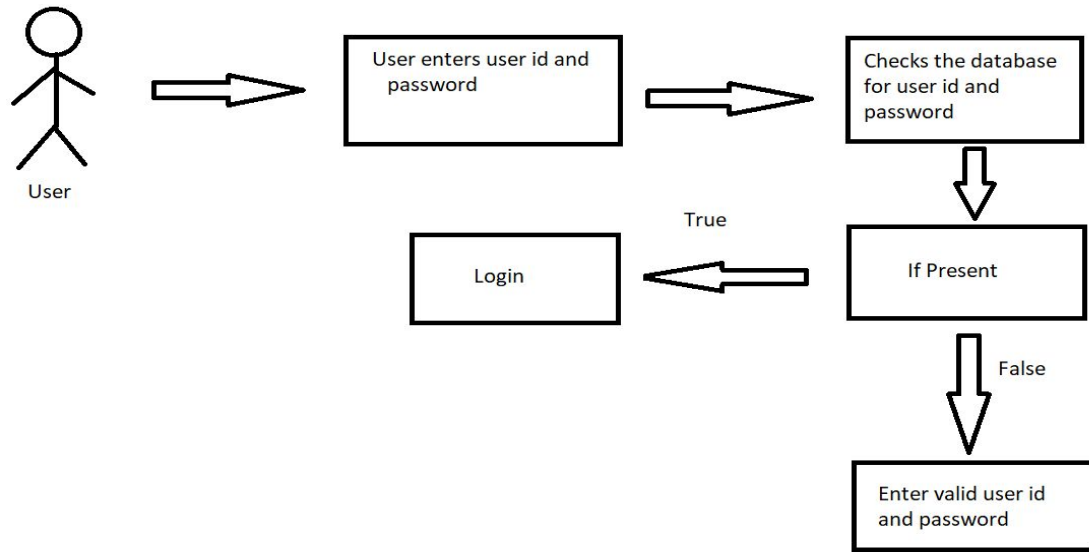
- 1 The product link should be from the ecommerce website supported by the system.
- 2 The model is not always accurate and is based on the reviews extracted.
- 3 The comparison of the products are to be of the same category type.

Proposed System

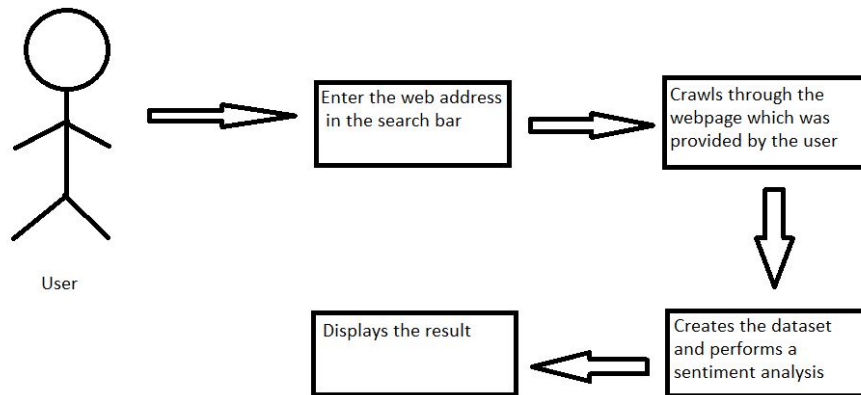
• Block Diagram



User Case Diagram for Signup System

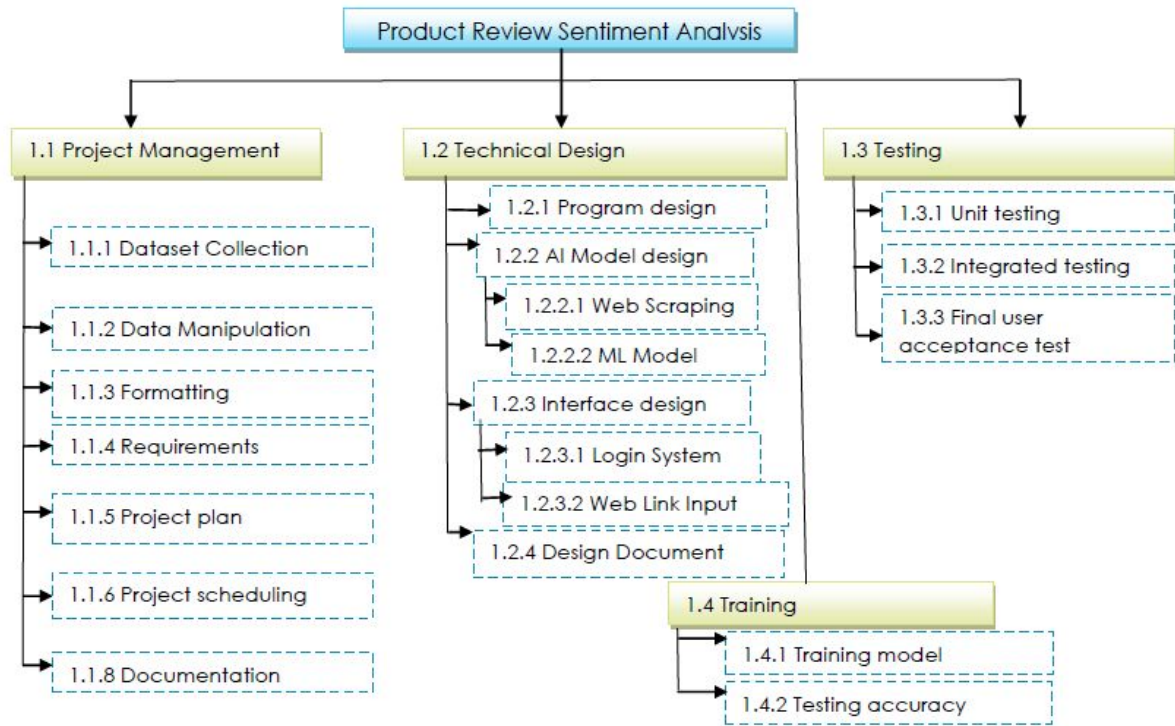


Block Diagram for Login System

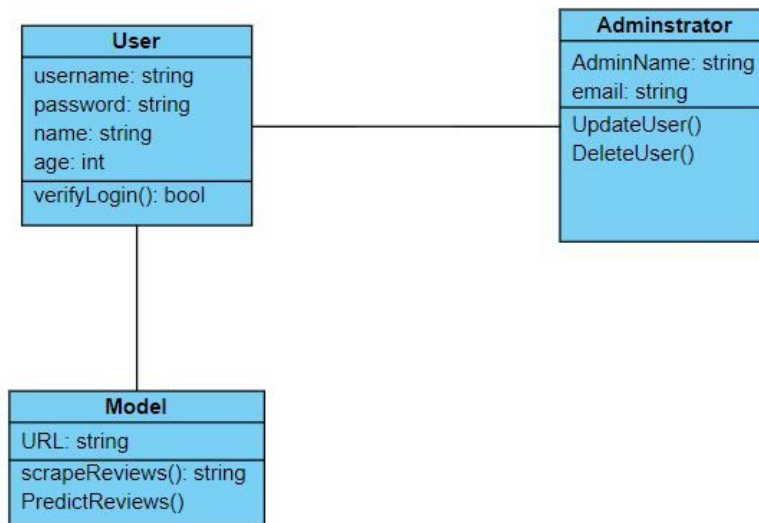


Block Diagram for Product Review Analysis

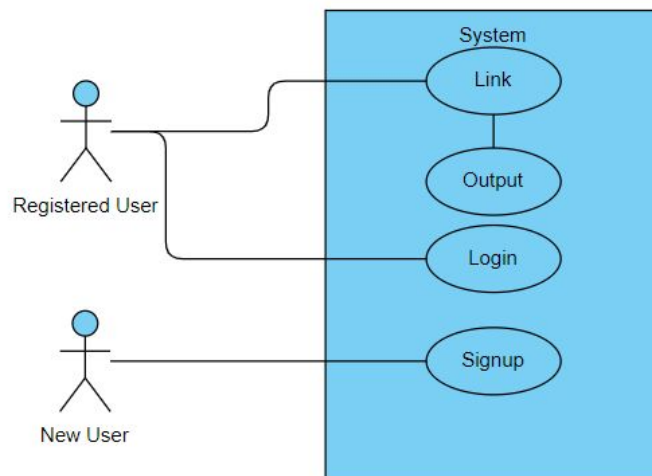
- **Work Breakdown Structure**



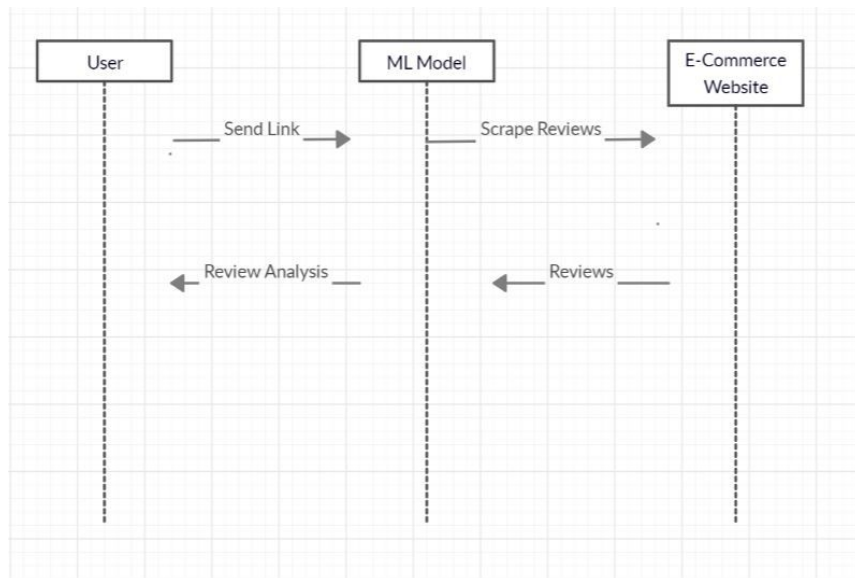
Class Diagram



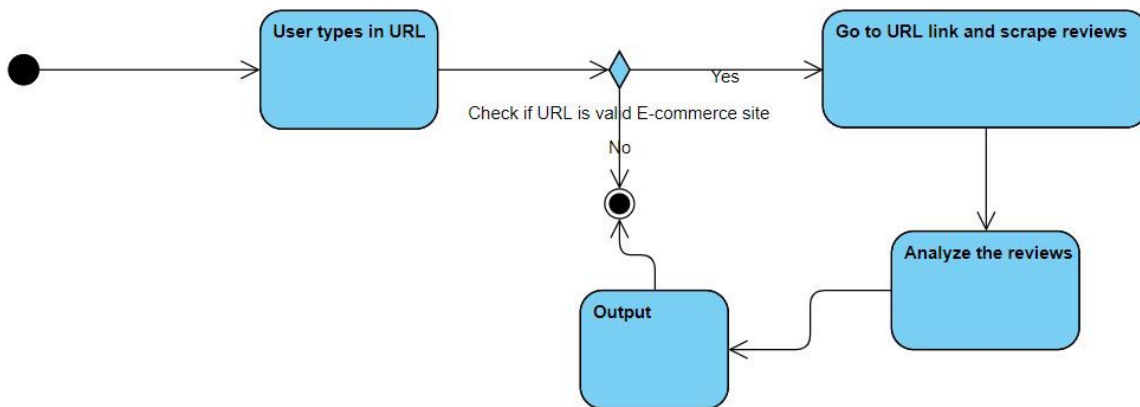
Use Case Diagram



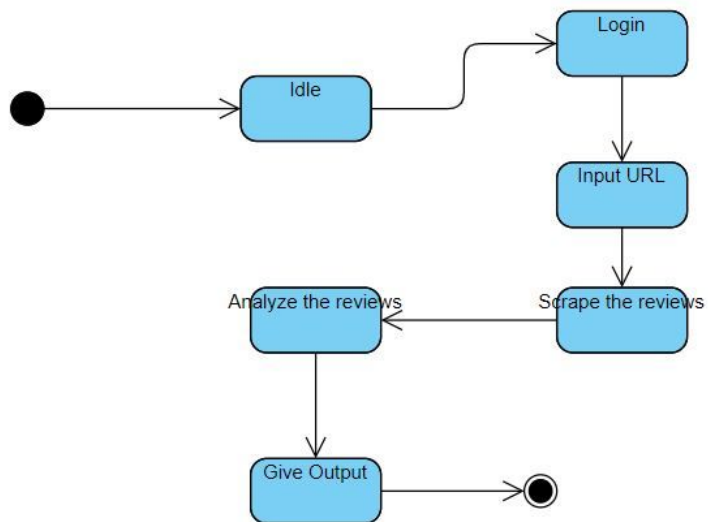
Sequence Diagram



Activity Diagram



StateChart Diagram



Code Snippet

Naive Bayes Model

```
import pandas as pd
import numpy as np
import nltk
import re
import unicodedata

from selenium import webdriver

from bs4 import BeautifulSoup

from urllib.request import urlopen as ureq

dataset=pd.read_csv('datasett.csv',encoding='latin-1')
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
def ProcessReview(sentence):
    review = re.sub('[^a-zA-Z]', ' ',sentence)
    review = review.lower()
    review = word_tokenize(review)
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review]
    return review
```

```

import nltk
nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
corpus=[]
for i in range(0,9999):
    corpus.append(ProcessReview(dataset['text'][i]))
corpus1=corpus
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer(tokenizer=lambda doc: doc, lowercase=False,max_features=1500)
X=cv.fit_transform(corpus1)
X=X.toarray()
Y=dataset.iloc[:,0].values
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB()
classifier.fit(X_train,Y_train)
Y_pred=classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(Y_test,Y_pred)
dataset1=pd.read_csv('products2.csv')
X_new=cv.transform(corpus2)
X_new=X_new.toarray()
Y_new=classifier.predict(X_new)

```

LSTM Model

```

import numpy as np
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential, load_model
from keras.layers import Dense, Embedding, LSTM, Bidirectional
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re

```

```
data = pd.read_csv("datasett.csv",encoding='latin-1')
```

```
tokenizer = Tokenizer(num_words=2000, split=' ')
```

```

tokenizer.fit_on_texts(data['text'])
X = tokenizer.texts_to_sequences(data['text'])
X = pad_sequences(X)
Y = data['sentiment']

```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
```

```

model = Sequential()

model.add( Embedding(2000, 128, input_length = X.shape[1], dropout=0.2))
model.add( Bidirectional(LSTM(196, dropout_U = 0.2, dropout_W = 0.2)))
model.add( Dense(2, activation = 'softmax'))

model.compile(loss = 'sparse_categorical_crossentropy', optimizer = 'adam', metrics =
['accuracy'])
model.fit(X_train, Y_train, nb_epoch = 7, batch_size = 100, verbose = 2)
import numpy as np
y=model.predict(X_test)
Y=np.argmax(y,axis=1)
from sklearn.metrics import confusion_matrix
confusion_matrix(Y_test,Y)
from selenium import webdriver
from bs4 import BeautifulSoup
from urllib.request import urlopen as ureq
my_url=str(input("enter url"))
uclient=ureq(my_url)
page_html=uclient.read()
uclient.close()
page_soup1=BeautifulSoup(page_html,"html.parser")
totpages=page_soup1.find("div",{ "class": "_2zg3yZ _3KSYCY"})
totpages=totpages.span.text.strip()
x=""
while(totpages[-1]!=" "):
    x=x+totpages[-1]
    totpages=totpages[0:len(totpages)-1]
x=int(x[:-1].replace(",",""))
print("pages",x)
filename="products2.csv"
headers="productname,review\n"
f=open(filename,"w",encoding='utf-8')
f.write(headers)
x= 15 if x>15 else x
for z in range(1,x+1):
    my_url=my_url+"&page="+str(z)
    uclient=ureq(my_url)
    page_html=uclient.read()
    uclient.close()
    page_soup=BeautifulSoup(page_html,"html.parser")
    containers=page_soup.findAll("div",{ "class": "_1PBCrt"})
    pn=page_soup.find("div",{ "class": "_3BTv9X"})
    pn=pn.img["alt"]
    for cont in containers:
        review=cont.findAll("p",{ "class": "_2xg6UI"})
        review=review[0].text
        revwords=review.split()

```

```

if len(revwords)<191:
    f.write(str(pn.replace(",","/"))+","+str(review.replace(",","/"))+"\n")
tp='IT DIDNT WORK WE BUY NEW IT DIE ON ME DON.T BUY IT: THE VHS WE BUY DID NOT
WORK THE MATCHS WERE BORING TOO BORING I KNOW DON.T HOW WWF CAN HAVE A
BAD PPV BUT THEY CAN WE BUY NEW IT WAS SO NEW BUT IT DIE ON ME DON.T BUY THIS
ON VHS THIS VHS WAS NEW BUT DIDNT PLAY I HAVE NOT SEEN ALL OF THE PPV BUT
WHAT I HAVE SEEN IT DIDNT PLAY GOOD NOT GOOD AT ALL IF YOU WANT TO WATCH IT
GET ON DVD THAT ALL HAVE TO SAY DON.T BUY THE VHS IF YOU WANT A GOOD PPV BUY
HELL IN THE CALL 2012 THAT ONE GOOD i was not happy to have a vhs i buy not work not
happy at all it was too old to play good it woods not work i was sad when it did.nt work i
woods not buy a wwf vhs for a 2 time the ppv was poor not that fun to watch it the wwf and
a bad ppv i want to see a good wwf ppv'
f.write(str(pn.replace(",","/"))+","+tp+"\n")
f.close()
data2 = pd.read_csv("products2.csv",encoding='latin-1')
tokenizer = Tokenizer(num_words=2000, split=' ')
tokenizer.fit_on_texts(data2['review'])
X2 = tokenizer.texts_to_sequences(data2['review'])
X2 = pad_sequences(X2)
y2=model.predict(X2)
result=np.argmax(y2,axis=1)
posrev=np.count_nonzero(result == 1)
negrev=len(result)-posrev-1
print(f'positive reviews: {posrev}\nnegative reviews: {negrev}')

```

Design of Test Cases

LOGIN SYSTEM

Test Case ID	Test Scenario	Test Process	Expected Results	Actual Results
1	Entering username already present in database for signup process	Click Signup Enter Details in form	Username already taken	Username already taken
2	In signup process	Click Signup	Enter valid age	Enter valid age

	Name: Adhil Age: H Username:adhilmd Password:****	Enter Details in form		
3	Name: Harish Age: 21 Username: hB2018 Password:****	Click Signup Enter Details in form	New account created	New account created
4	Entering wrong username or password while login	Click Login Enter Username and password	Username or password is wrong	Username or password is wrong
5	Entering right username and password in login	Click Login Enter Username and password	Goes to next page where we can enter URL	As Expected

URL Input bar

Test Case ID	Test Scenario	Expected Results	Actual Results
1	software engineering	Enter valid E-commerce website	Enter valid E-commerce website
2	www.google.com	Enter valid E-commerce website	Enter valid E-commerce website
3	https://www.flipkart. com/samsung-galax y-s9-midnight-black- 64-gb/product-revie ws/itm33a69rpszg n?pid=MOBF2VWVB	Starts scraping for reviews	Starts scraping for reviews

	GCT5QQN&lid=LSTM OBF2VWVBGCT5QQ N0ZJFUP&marketpla ce=FLIPKART		
--	--	--	--

Result Analysis and Discussion

Training Models:

Naive Bayes

```
: model.score(X_train,Y_train)
: 0.805975746968371
```

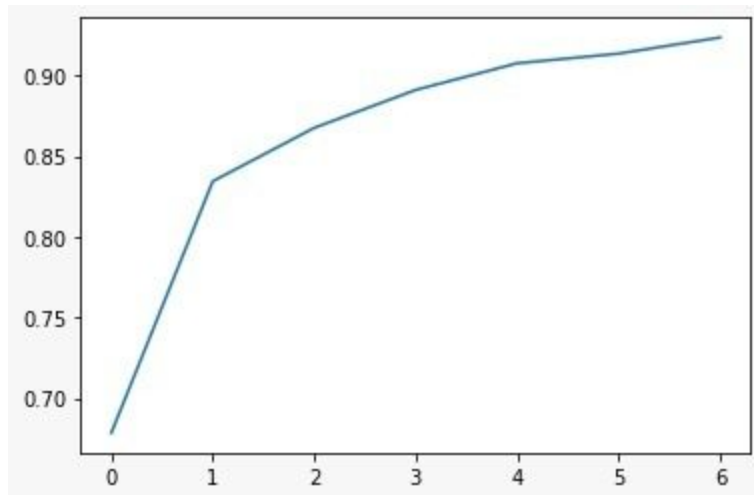
The NB model trains to an accuracy of 80.59%

Bi Directional LSTM

```
Epoch 1/7
- 126s - loss: 0.6125 - accuracy: 0.6785
Epoch 2/7
- 128s - loss: 0.3876 - accuracy: 0.8344
Epoch 3/7
- 131s - loss: 0.3262 - accuracy: 0.8675
Epoch 4/7
- 131s - loss: 0.2740 - accuracy: 0.8910
Epoch 5/7
- 130s - loss: 0.2442 - accuracy: 0.9075
Epoch 6/7
- 128s - loss: 0.2350 - accuracy: 0.9135
Epoch 7/7
- 127s - loss: 0.2093 - accuracy: 0.9235
```

The model trains for 7 epochs and we attain an accuracy of 92.35%

Graph accuracy vs epochs trained in Bi Directional LSTM:



Testing Models:

Naive Bayes Confusion matrix on testing dataset:

```
array([[665, 276],  
      [189, 870]], dtype=int64)
```

The accuracy on test dataset by NB Model is 76.75%

LSTM Confusion Matrix on testing dataset:

```
array([[783, 163],  
      [180, 874]], dtype=int64)
```

The accuracy on test dataset by Bi Directional LSTM Model is 82.85%

Thus the Bi Directional LSTM Model does better when compared to NB Model when it sees test cases that it has not seen before. So this model is better suitable for implementation and deployment.

OUTPUT:

Form Bar to enter URL


[Home](#) [SaveOrder](#) [OrderReminder](#) [ProductCategories](#)

You are logged in as harish99 [Logout](#)

SEARCH PRODUCT


enter url of the product enter e-com site name

Good reviewed Trending products :




Mac Book Pro

Cost:1,00,000






cannon 77D

Cost:60,000



Iphone 11pro

Cost:70,000



Login Page

[Home](#) [SaveOrder](#) [OrderReminder](#) [ProductCategories](#)

You are logged out!


username

password

Login to place order


[Signup](#)

Good reviewed Trending products :




Mac Book Pro

Cost:1,00,000



cannon 77D

Cost:60,000



Iphone 11pro

Cost:70,000

Conclusion

The software model has been implemented and deployed successfully. The model has satisfied all test cases and has been trained to provide results with high accuracy.

Future Enhancement

A feature could be added where the software compares the previous search of that product by the user during the last login and analyzes the price drop and also checks stock. We can add a feature for comparing prices for the same product on different websites.