

**AMAZON KINDLE STORE REVIEWS ANALYSIS USING
IBM
WATSON SERVICES**

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,
HYDERABAD**

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE
AND ENGINEERING**

Submitted by

SRIYAGUDLA

19UK1A0518

VAMSHI MAMIDALA

19UK1A0516

RAMYA BOLLEPALLY

19UK1A0553

Under the esteemed guidance of

Mr. A. ASHOK KUMAR

(Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE**

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal –

506005 **2019– 2023**

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
ENGINEERING VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005 2019 – 2023**



CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled “**AMAZON KINDLE STORE REVIEWS ANALYSIS USING IBM WATSON SERVICES**” Is being submitted by- **GUDLASRIYA(H.NO:19UK1A0518),MAMIDALAVAMSHI(H.NO:19UK1A0516),BOLL EPALLYRAMYA(H.NO:19UK1A0553)**,in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2022-23, is a record of work carried out by them under the guidance and supervision.

Project Guide
Mr. A. ASHOK KUMAR
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **A.ASHOK KUMAR**, Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SRIYAGUDLA (19UK1A0518)

VAMSHI MAMIDALA(19UK1A0516)

RAMYA BOLLEPALLY(19UK1A0553)

ABSTRACT

Amazon Kindle Store is an e-book e-commerce store for all the book reading hobbyists. Online reviews are a category of product information created by users based on personal handling experience. Online shopping websites endow with platforms for consumers to review products and carve up opinions. The problem is most of the comments from customer reviews about the products are contradicted to their ratings. Many customers will post their comments and forgot to rate the product or not engrossed to rate it.

Sentiment mining plays a very important role in business to understand the opinion of customers to improve the products. Customer also depends on the opinion of others who have bought the products already. Reviews or feedback becomes the deciding factor to buy or sell a product. A rating of the products gives a speedy clarification to pact with the product. We will be using Natural language processing to analyse the sentiment (positive or a negative) of the given review.

TABLE OF CONTENTS:-

1.INTRODUCTION	1
1.1 OVERVIEW.....	1
1.2 PURPOSE.....	1
2.LITERATURE SURVEY.....	2
2.1 EXISTING PROBLEM.....	2
2.2 PROPOSED SOLUTION.....	2-3
3.THEORITICAL ANALYSIS.....	4
3.1 BLOCK DIAGRAM.....	4
3.2 HARDWARE /SOFTWARE DESIGNING	4
4.EXPERIMENTAL INVESTIGATIONS	5
5.FLOWCHART.....	6
6.RESULTS.....	7-8
7.ADVANTAGES AND DISADVANTAGES.....	9
8.APPLICATIONS.....	9
9.CONCLUSION.....	10
10.FUTURE SCOPE.....	10
11.BIBILOGRAPHY	11
12.APPENDIX (SOURCE CODE)	12-25

1.INTRODUCTION

1.1.OVERVIEW

The objective of this paper is to categorize the positive and negative feedback of the customers over different products and build a supervised learning model to polarize large amounts of reviews. A study on amazon last year revealed more than 80% of online shoppers trust reviews as much as personal recommendations. Any online item with a large amount of positive reviews provides a powerful comment of the legitimacy of the item. Conversely, books, or any other online item, without reviews puts potential prospects in a state of distrust. Quite simply, more reviews look more convincing. People value the consent and experience of others and the review on a material is the only way to understand others' impression on the product. Opinions, collected from users' experiences regarding specific products or topics, straightforwardly influence future customer purchase decisions. Similarly, negative reviews often cause sales loss. For those understanding the feedback of customers and polarizing accordingly over a large amount of data is the goal. There are some similar works done over amazon dataset. In opinion mining over a small set of dataset of Amazon kindle product reviews to understand the polarized attitudes towards the product.

- Know fundamental concepts and techniques of natural language processing (NLP)
- Gain a broad understanding of text data.
- Know how to pre-process/clean the data using different text preprocessing techniques.
- Know how to build a neural network.
- Know how to build a web application using the Flask framework

1.2.PURPOSE

As the commercial sites of the world are almost fully online platforms, people are trading products through different e-commerce websites. And for that reason reviewing products before buying is also a common scenario. Also nowadays, customers are more inclined towards the reviews to buy a product. So analyzing the data from those customer reviews to make the data more dynamic is an essential field nowadays. In this age of increasing machine learning and deep learning based algorithms, reading thousands of reviews to understand a product is rather time consuming where we can polarize a review on a particular category to understand its popularity among the buyers all over

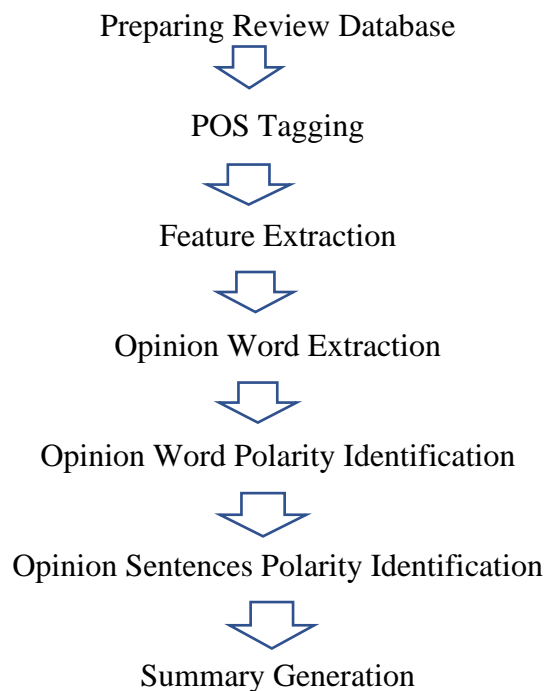
2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Given a dataset containing of various attributes, use the features available in dataset and define a supervised classification algorithm which can identify whether they getting reviews correct predicted reviews or not. The problem is most of the comments from customer reviews about the products are contradicted to their ratings. Many customers will post their comments and forgot to rate the product or not engrossed to rate it.

2.2 PROPOSED SOLUTION

All Information in the world can be broadly classified into mainly two categories, facts and opinions. Facts are objective statements about entities and worldly events. On the other hand opinions are subjective statements that reflect people's sentiments or perceptions about the entities and events. Maximum amount of existing research on text and information processing is focused on mining and getting the factual information from the text or information. Before we had WWW we were lacking a collection of opinion data, in an individual needs to make a decision, he/she typically asks for opinions from friends and families. When an organization needs to find opinions of the general public about its products and services, it conducted surveys and focused groups. But after the growth of Web, especially with the drastic growth of the user generated content on the Web, the world has changed and so has the methods of gaining ones opinion. One can post reviews of products at merchant sites and express views on almost anything in Internet forums, discussion groups, and blogs, which are collectively called the user generated content. As the technology of connectivity grew so as the ways of interpreting and processing of users opinion information has changed. Some of the machine learning techniques like Naïve Bayes, Maximum Entropy and Support Vector Machines has been discussed in the paper. Extracting features from user opinion information is an emerging task.

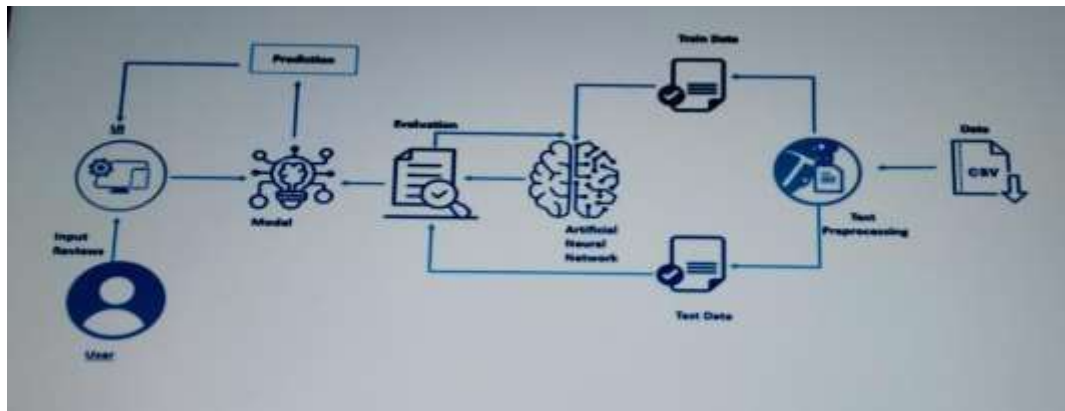


Basic Step of Feature Extraction

a generic model of feature extraction from opinion information is shown, firstly the information database is created, next POS tagging is done on the review, next the features are extracted using grammar rules such as adjective + noun or so on, as nouns are features and adjectives are sentiment words. Next Opinion words are extracted followed by its polarity identification. Some models also calculate sentence polarity for accuracy. Lastly the results are combined to obtain a summary. Many algorithms can be used in opinion mining such as Naive Bayes Classification, Probabilistic Machine Learning approach to classify the reviews as positive or negative, have been used to get the sentiment of opinions of different domains such as movie, Amazon reviews of products. In our work we have used reviews of iPhone 5 extracted from Amazon website. We studied all the reviews and got to know that there are many reviews in which the user talks about the service provided by amazon and its sellers. So we decided to classify reviews into service, product and feature based reviews. We also found that the sentiment of each review is very obvious, the review rating provided by the user mirrors what the user writes as his/her review, i.e. if the user writes something bad definitely the overall rating the user gives is either 1 or 2 out of 5. This is from our study of a set of amazon reviews on iPhone 5. Our work mainly concentrates on feature extraction and finding out the sentiment of the particular feature. We have used POS tagging technique on sentence level. In our approach we have made certain rules using the tags of particular word and using list of words with respective sentiment value to find the feature and then getting the appropriate sentiment from it. The Sentiment model that we have proposed is designed based on the uncertainty of the amazon reviews. Our work also include summarization in the form of charts for overall view of the sentiments of the users on the product or a particular feature.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

The following is the Hardware required to complete this project:

- Internet connection to download and activate
- Administration access to install and run Anaconda Navigator
- Minimum 10GB free disk space
- Windows 8.1 or 10 (64-bit or 32-bit version) OR Cloud: Get started free, *Cloud account required.

Minimum System Requirements To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

- 500 megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

The following are the software required for the project:

- Google Colaboratory Notebook and Jupyter Notebook
- Spyder and Pycharm Community
- Microsoft Excel 2013

4.EXPERIMENTAL INVESTIGATION

In this project, we have used Amazon Kindle Store Reviews Dataset. This dataset is a csv file consisting of labelled data and having the following columns-

“reviewerID”: ID of the reviewer

“asin”: ID of the product

“reviewerName”: name of the reviewer

“helpful”: helpfulness rating of the review

“reviewText”: text of the review

“overall”: rating of the product

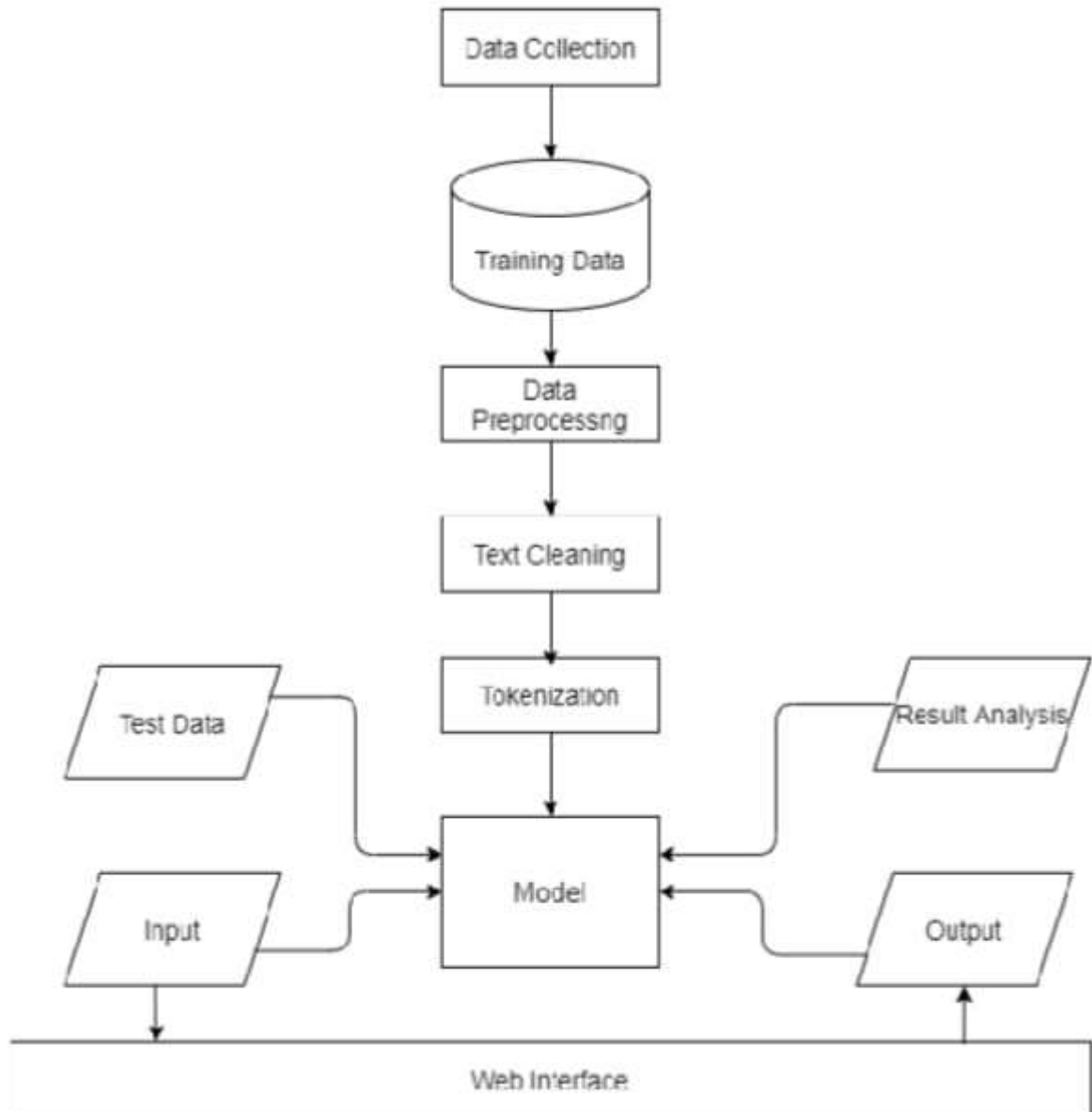
“summary”: summary of the review

“reviewTime”: time of the review

“unixreviewTime”: unix timestamp

For the dataset we selected, it consists of more than 50,000 kindle book reviews. From the format used analysing the review polarity we used review text & Overall from it.

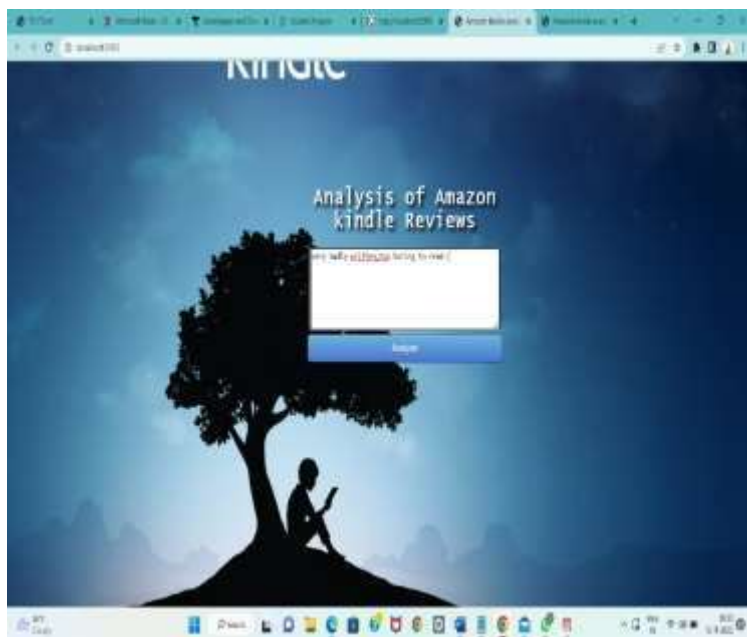
5.FLOWCHART



6.RESULT



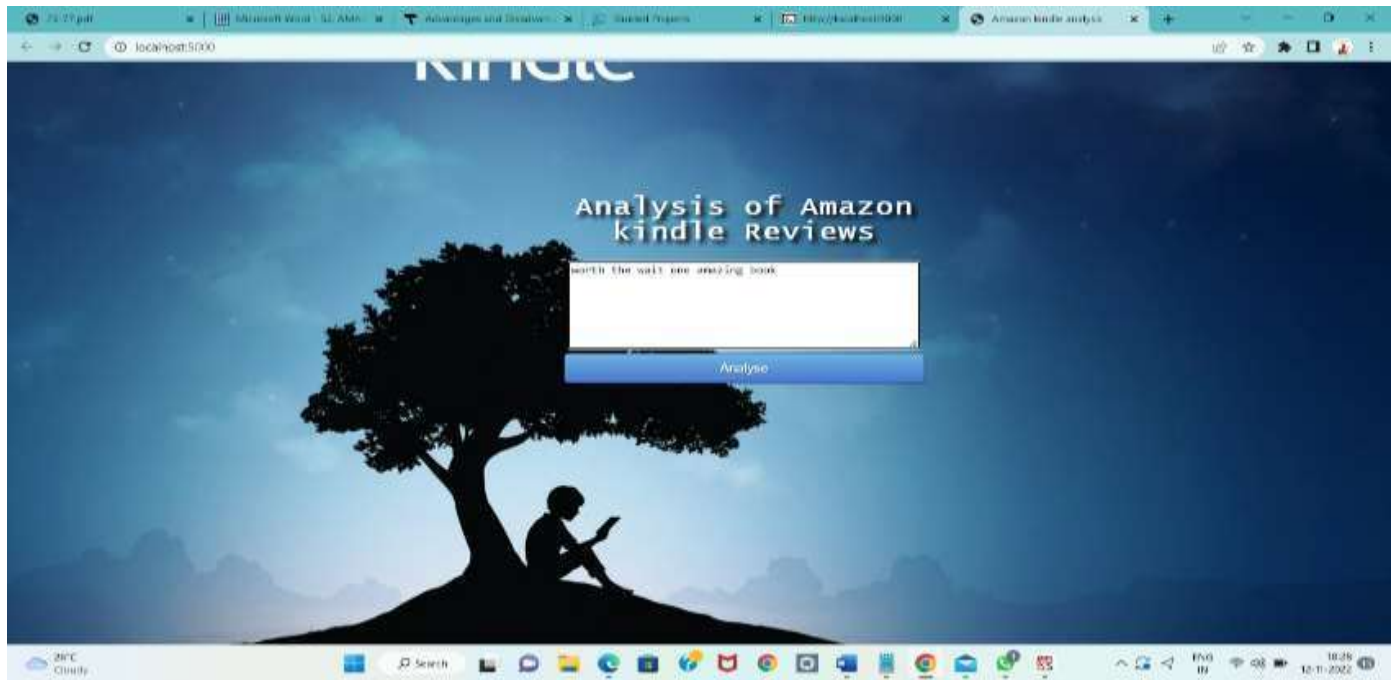
HOMEPAGE



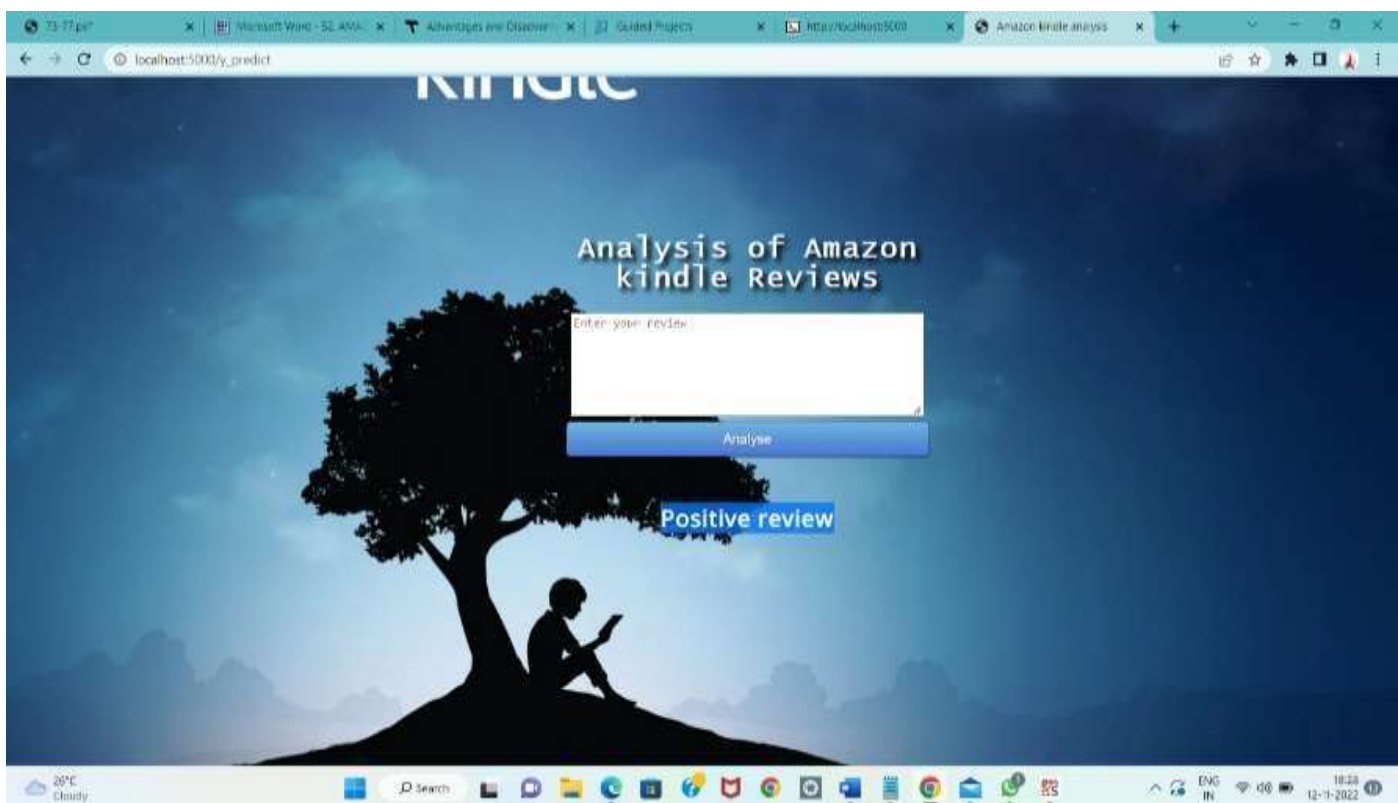
REVIEW ANALYSIS PAGE



OUTPUT



REVIEW ANALYSIS



OUTPUT

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

So many books to choose from
Free books
Access to libraries online collections
Cheaper books
Internet, music, and games
Dictionary
Translations
Electronic markers
No book light required
Large print
Long battery life
Search function
Paperless
Convenience

DISADVANTAGES

It's harder to share
No color
Eye strain and retention
Its electronic

8.APPLICATIONS

- 1.Kindle unlimited and amazon prime numbers can select and download kindle books directly in the app
- 2.choose from over six million kindle books
- 3.understand challenging books
- 4.Improve your reading comprehension

9.CONCLUSION

It is completely impossible to use only raw text as input for making predictions. Hence, we saw that the preprocessing step played a major role in the complete process of NLP. To get better results, accuracy and make the machine take all the text as tokens, pre-processing of data is to be done carefully looking at the type of contents present in it. The most important thing is to be able to extract the relevant features from the given source of data. This kind of data can often come as a good complementary source in order to extract more learning features and increase the predictive power of the models. And the user is able to predict that the given comment is positive or negative.

10.FUTURE SCOPE

In future, the work can be extended to perform multi-class classification of reviews which will provide a delineated nature of review to the consumer, hence better judgment of the product. It can also be used to predict the rating of a product from the review. This will provide users with a reliable rating because sometimes the rating received by the product and the sentiment of the review do not provide justice to each other. The proposed extension of work will be very beneficial for the e-commerce industry as it will augment user satisfaction and trust.

11.BIBLIOGRAPHY

- [1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
- [2] M. S. Elli and Y.-F. Wang. Amazon reviews, business analytics with sentiment analysis.
- [3] C. Rain. Sentiment analysis in amazon reviews using probabilistic machine learning. Swarthmore College, 2013.
- [4] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
- [5] Y. Xu, X. Wu, and Q. Wang. Sentiment analysis of yelps ratings based on text reviews, 2015.
- [6] Bhatt, Aashutosh, et al. "Amazon Review Classification and Sentiment Analysis." *International Journal of Computer Science and Information Technologies* 6.6 (2015): 5107- 5110.
- [7]Chen, Weikang, Chihhung Lin, and YiShu Tai."Text-Based Rating Predictions on Amazon Health & Personal Care Product Review." (2015)
- [8]Shaikh, Tahura, and DeepaDeshpande. "Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews.",(2016)
- [9]Nasr, Mona Mohamed, Essam Mohamed Shaaban, and Ahmed Mostafa Hafez. "Building Sentiment analysis Model using Graphlab." *IJSER*.,
- [10]Text mining for yelp dataset challenge; Mingshan Wang; University of California San Diego, (2017)

12.APPENDIX

Model Building

- # 1.Dataset
- ## 2.Jupyter Notebook

Application Building

- 1.HTML file
- 2.CSS file
- 3.Flask
- 4.IBM Watson.

SOURCE CODE:

HTML Code:

```
<!DOCTYPE html>
```

<html lang="en">

<head>

```
<meta charset="utf-8">
```

<title>

beautiful and simple website using html only -- fullywrold web tutorials

</title>

</head>

```
<body background="1.jpg" link="#000" alink="#017bf5" vlink="#000">
```


LOGO

[illegible]


```

from gevent.pywsgi import WSGIServer

import os

tf.keras.backend.clear_session()

app = Flask(__name__)

model=tf.keras.models.load_model("amazo.h5")

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/y_predict',methods=['POST'])

def y_predict():

    """

    For rendering results on HTML GUI

    """

    d = request.form['Sentence']

    print(d)

    loaded=CountVectorizer(decode_error='replace',vocabulary=joblib.load("amazo.save"))

    d=d.split("delimiter")

    result=model.predict(loaded.transform(d))

    print(result)

    prediction=result>0.5

    if prediction[0] == False:

        output="Positive review"

    elif prediction[0] == True:

        output="Negative review"

```

```

return render_template('index.html', prediction_text='{}'.format(output))

port = os.getenv('VCAP_APP_PORT','5000')

if __name__ == "__main__":

    __app.secret_key = os.urandom(12)

    app.run(debug=True,host='0.0.0.0',port = port)

```

CODE SNIPPETS

A.MODELBUILDING

```

!pip install jupyterthemes

Requirement already satisfied: notebook<5.6.0 in /usr/local/lib/python3.7/dist-packages (from jupyterthemes) (5.7.10)
Requirement already satisfied: ipython>=5.4.1 in /usr/local/lib/python3.7/dist-packages (from jupyterthemes) (7.9.0)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from jupyterthemes) (4.11.2)
Requirement already satisfied: matplotlib>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from jupyterthemes) (3.2.2)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (2.8.1)
Requirement already satisfied: traitlets>=4.1 in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (3.1.1)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (2.0.10)
Requirement already satisfied: setuptools>=16.5 in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (57.4.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (0.2.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (4.8.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (4.4.2)
Collecting jedi<=0.10
  Downloading jedi-0.18.1-py2.py3-none-any.whl (1.0 MB)
    1.6 MB 45.1 MB/s
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython>=5.4.1->jupyterthemes) (0.7.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dist-packages (from jedi<=0.10->ipython>=5.4.1->jupyterthemes) (0.8.3)
Collecting ply
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
    49 kB 5.5 MB/s
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=1.4.1->jupyterthemes) (1.21.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=1.4.1->jupyterthemes) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=1.4.1->jupyterthemes) (1.4.4)
Requirement already satisfied: pyparsing<=2.0.4,>=2.1.1,!=2.1.4,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=1.4.1->jupyterthemes) (3.0.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=1.4.1->jupyterthemes) (2.8.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib>=1.4.1->jupyterthemes) (4.1.1)
Requirement already satisfied: jls&gt;=0.0.0 in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (2.11.3)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (0.15.0)
Requirement already satisfied: jupyter-client>=7.0.0,>=5.7.0 in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (6.1.12)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (23.2.1)
Requirement already satisfied: nbformat in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (5.7.0)
Requirement already satisfied: nbconvert<6.0 in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (5.6.1)
Requirement already satisfied: terminado<0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>=5.6.0->jupyterthemes) (0.15.3)

```

```

[2] !ls -lt monokai -f fire -fs 11 -of ptsans -nfs 10 -H -kl -course 5 -course r -cellw 80% -t

# Import required libraries
import pandas library
import pandas as pd
import numpy
import numpy as np
import requests
import requests
import io
import io

[4] from google.colab import drive

drive.mount('/content/gdrive', force_remount=True)

Mounted at /content/gdrive/

[5] # Import the dataset in the data variable
url = '/content/gdrive/mydrive/kindle_reviews.csv'
data = pd.read_csv(url)

[6] data.shape

(982619, 10)
  
```

```

[7] data.head()

   unnamed: 0  asin  helpful  overall  reviewText  reviewTime  reviewerID  reviewTime  summary  unixReviewTime
0           0    B000F83S2Q  [0, 0]      5  I enjoy vintage books and movies so I enjoyed ...  05/5/2014  A1P8404F1VG2BU  Avid reader  Nice vintage story  1399249000
1           1    B000F83S2Q  [2, 2]      4  This book is a reissue of an old one; the auth...  01/6/2014  ANDW05ABLUED  citters  Different...  1388660400
2           2    B000F83S2Q  [2, 2]      4  This was a fairly interesting read. I had of...  04/4/2014  A795D8MNCJLAA  dot  Oldie  1396569600
3           3    B000F83S2Q  [1, 1]      5  I'd never read any of the Amy Brewer mysteri...  02/19/2014  A1PV05X13TWXQ  Elaine H. Turley "Montana Songbird"  I really liked it.  1392768500
4           4    B000F83S2Q  [0, 1]      4  If you like period pieces - clothing, kins, y...  03/19/2014  A35PT0KDG7WBLN  Father Dowling Fan  Period Mystery  1395187200

[8] # Assigning 5000 rows to data
data = data.head(5000)

[9] # Checking for null values
data.isnull().any()

   unnamed: 0  asin  helpful  overall  reviewText  reviewTime  reviewerID  reviewTime  summary  unixReviewTime
0           0    B000F83S2Q  [0, 0]      5  I enjoy vintage books and movies so I enjoyed ...  05/5/2014  A1P8404F1VG2BU  Avid reader  Nice vintage story  1399249000
1           1    B000F83S2Q  [2, 2]      4  This book is a reissue of an old one; the auth...  01/6/2014  ANDW05ABLUED  citters  Different...  1388660400
2           2    B000F83S2Q  [2, 2]      4  This was a fairly interesting read. I had of...  04/4/2014  A795D8MNCJLAA  dot  Oldie  1396569600
3           3    B000F83S2Q  [1, 1]      5  I'd never read any of the Amy Brewer mysteri...  02/19/2014  A1PV05X13TWXQ  Elaine H. Turley "Montana Songbird"  I really liked it.  1392768500
4           4    B000F83S2Q  [0, 1]      4  If you like period pieces - clothing, kins, y...  03/19/2014  A35PT0KDG7WBLN  Father Dowling Fan  Period Mystery  1395187200
  
```

amazon.ipynb - Collaboratory

colab.research.google.com/drive/1qG2C0Z0PU2WwNB_1E3BmCMAC0Zm6tHfucdftb-mjN8UIMCgAT

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Test

unishowreviewtime false
dtype: bool

[10] data.isnull().sum()

```

unamed: 0      0
asin      0
helpful   0
overall   0
reviewtext 1
reviewtime 0
reviewerID 0
reviewtime 140
summary    0
unishowreviewtime 0
dtype: int64

```

deleting or dropping the unwanted columns from the dataset

```

del data['unamed: 0']
del data['asin']
del data['helpful']
del data['reviewtime']
del data['reviewerID']
del data['reviewtime']
del data['unishowreviewtime']

```

first 10 rows of data

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

	overall	reviewtext	summary
0	5	I enjoy vintage books and movies so I enjoyed ...	Nice vintage story

24°C Partly cloudy

Search

completed at 20:30

amazon.ipynb - Collaboratory

colab.research.google.com/drive/1qG2C0Z0PU2WwNB_1E3BmCMAC0Zm6tHfucdftb-mjN8UIMCgAT

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Test

[12] first 10 rows of data

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

	overall	reviewtext	summary
0	5	I enjoy vintage books and movies so I enjoyed ...	Nice vintage story
1	4	This book is a reissue of an old one; the auth ..	Different ..
2	4	This was a fairly interesting read. It had ot ..	Oldie
3	5	I'd never read any of the Amy Brewster myster ..	I really liked it.
4	4	If you like period pieces - clothing, lingos, y ..	Period Mystery
5	4	A beautiful in-depth character description mak ..	Review
6	4	I enjoyed this one tho I'm not sure why it's c ..	Nice old fashioned story
7	4	Never heard of Amy Brewster. But I don't need ...	Enjoyable reading and reminding the old times
8	5	Darth Maul working under cloak of darkness com ..	Darth Maul
9	4	This is a short story focused on Darth Maul's ...	Not bad, not exceptional

checking value counts

```
data.overall.value_counts()
```

```

5    21090
4    14980
3     7011
2     2632
1       200
Name: overall, dtype: int64

```

24°C Partly cloudy

Search

completed at 20:32

amazon.ipynb - Colaboratory

colab.research.google.com/drive/1upGCDZPUZWNH_EJ3EnCM4OCZmdtHfHwmlfo=JufNylixCh-

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

name: overall, dtype: int64

check the null values
data.isna().sum()

```

overall    0
reviewText  1
summary    0
dtype: int64

```

[15] #joining review description and summary into one col
data['reviewText']=data['reviewText']+" "+data['summary']

[16] data.head()

	overall	reviewText	summary
0	5	I enjoy vintage books and movies so I enjoyed ...	Nice vintage story
1	4	This book is a reissue of an old one; the auth ...	Different...
2	4	This was a fairly interesting read. It had ol...	Oldie
3	5	I'd never read any of the Army Brewster myster...	I really liked it.
4	4	If you like period pieces - clothing, litig, y...	Period Mystery

data.drop(['summary'],axis=1,inplace=True)

checking for null values
data.isna().sum()

0

completed at 20:33

24°C Partly cloudy

Search

ENG IN 20:34 11-11-2022

amazon.ipynb - Colaboratory

colab.research.google.com/drive/1upGCDZPUZWNH_EJ3EnCM4OCZmdtHfHwmlfo=JufNylixCh-

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[17] data.drop(['summary'],axis=1,inplace=True)

[18] checking for null values
data.isna().sum()

```

overall    0
reviewText  1
dtype: int64

```

[19] since there is only one null value, replace it with blank space
data['reviewText'].fillna("",inplace = True)

[20] #grouping the overall rating of scale 1-5 to 2 categories
def review_sentiment(rating):
 #0(positive) and 1(negative)
 if(rating == 5 or rating == 4 or rating==3):
 return 0
 else:
 return 1

[21] data.overall = data.overall.apply(review_sentiment)

data.overall.value_counts()

```

0    45083
1    4917
Name: overall, dtype: int64

```

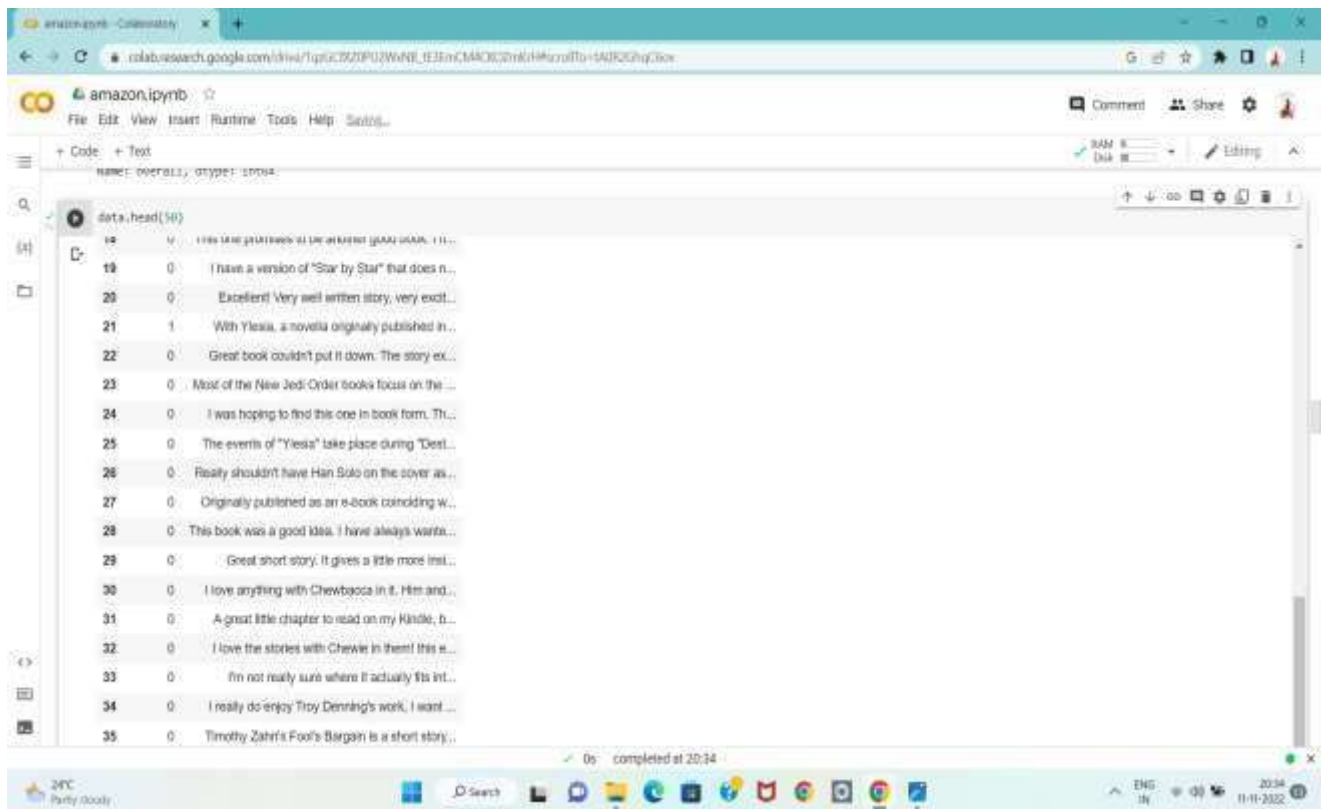
data.head(5)

completed at 20:34

24°C Partly cloudy

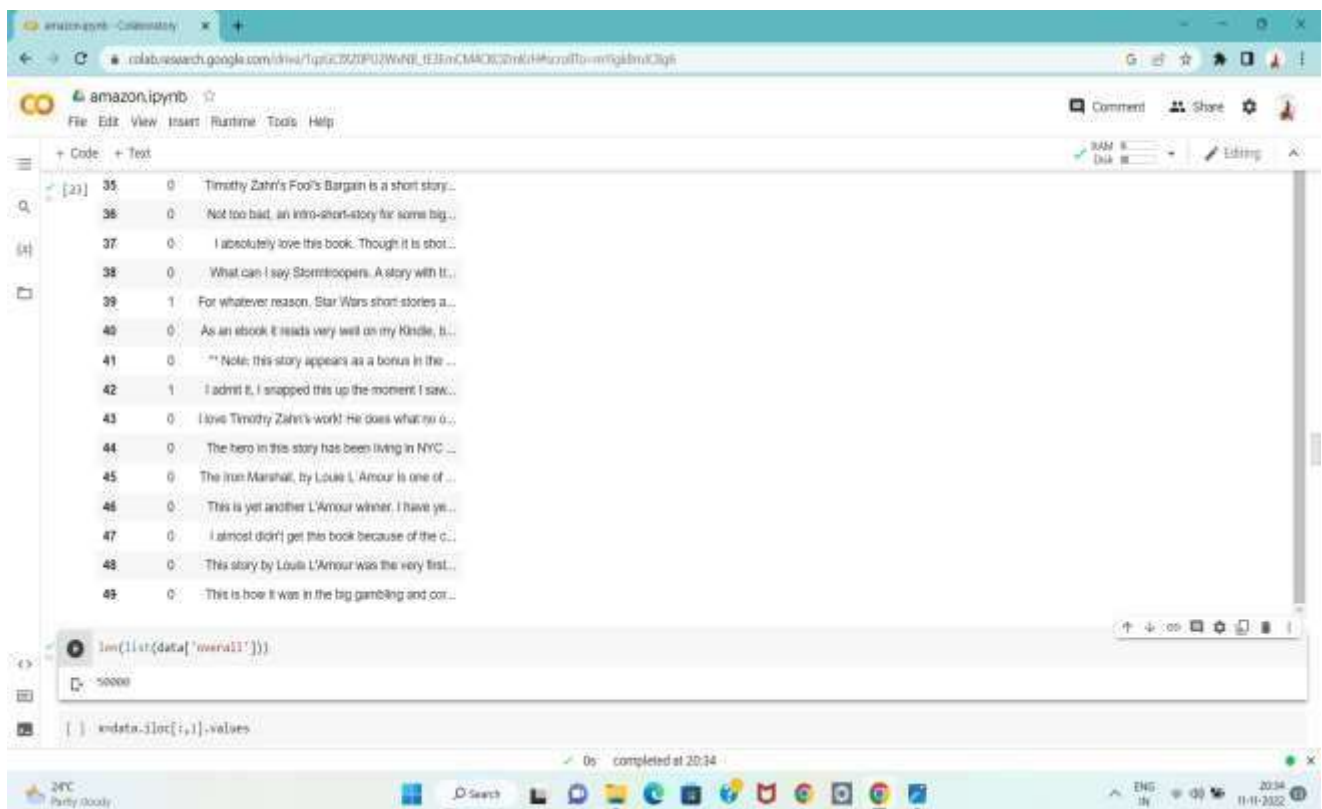
Search

ENG IN 20:34 11-11-2022



```
data.head(35)
```

Index	Rating	Text
18	5	I like this play more as an answer than as a...
19	0	I have a version of "Star by Star" that does n...
20	0	Excellent! Very well written story, very excit...
21	1	With Ylesia, a novella originally published in...
22	0	Great book couldn't put it down. The story ex...
23	0	Most of the New Jedi Order books focus on the ...
24	0	I was hoping to find this one in book form. Th...
25	0	The events of "Ylesia" take place during "Dest...
26	0	Really shouldn't have Han Solo on the cover as...
27	0	Originally published as an e-book coinciding w...
28	0	This book was a good idea. I have always want...
29	0	Great short story. It gives a little more insi...
30	0	I love anything with Chewbacca in it. Him and...
31	0	A great little chapter to read on my Kindle, b...
32	0	I love the stories with Chewie in them! this e...
33	0	I'm not really sure where it actually fits int...
34	0	I really do enjoy Troy Denning's work, I want...
35	0	Timothy Zahn's Foo's Bargain is a short story...



```
len(list(data['overall']))
```

```
50000
```

```
data.iloc[i,:].values
```

The screenshot shows an Amazon IPYNB notebook with the following code and output:

```
[25] x=data.iloc[:,1].values
```

```
[26] #import natural language toolkit
import nltk
nltk.download("stopwords")
nltk.download("wordnet")
#import stopwords library to remove stopwords
from nltk.corpus import stopwords
#library used for stem the words
from nltk.stem.porter import PorterStemmer
#create an object for stemming
ps = PorterStemmer()
#library used for stem the words
from nltk.stem import WordNetLemmatizer
#create an object for wordnet lemmatizer
wordnet=wordnetlemmatizer()
```

```
[27] import re
```

```
import nltk
nltk.download('all')
```

Output for [27]:

```
[nltk_data] Downloading collection 'all'
[nltk_data]
[nltk_data]   Downloading package abc to /root/nltk_data...
[nltk_data]     Unzipping corpora/abc.zip.
[nltk_data]   Downloading package alpine to /root/nltk_data...
[nltk_data]     Unzipping corpora/alpine.zip.
[nltk_data]   Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
```

46s completed at 20:36

The screenshot shows an Amazon IPYNB notebook with the following code and output:

```
[28] import nltk
nltk.download("stopwords")
nltk.download("wordnet")
```

Output for [28]:

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
# initialize empty array to append clean text
corpus=[]
# no of rows to clean
for i in range(len(x)):
    #replacing punctuations and numbers using re library
    temp=re.sub('[^a-z- ]',' ',x[i])
    # convert all text to lower cases
    temp=temp.lower()
    # split to array(default delimiter is " ")
    temp=temp.split()
    # creating wordnetlemmatizer object to take main lemma of each word
    wordnet = wordnetlemmatizer()
    #loop for lemmatization each word in string array at ith row
    temp=[wordnet.lemmatize(word) for word in temp if not word in set(
        stopwords.words('english'))]
    #join all string array elements to create back into a string
    temp=' '.join(temp)
    #append each string to create array of clean text
    corpus.append(temp)
```

12m 29s completed at 20:49

The screenshot shows the Amazon IPYNB interface with the following code and output:

```
[30] corpus.append(temp)
```

```
[31] !pip install sklearn
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.org/dev/colab-wheels/public/simple/>
collecting sklearn
Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
building wheels for collected packages: sklearn
Building wheel for sklearn (setup.py) ... done
created wheel for sklearn: filename=sklearn-0.0.post1-py3-none-any.whl size=1346 sha256=55637795c5d07480b7f93e0f6a0a57391804220c5104f11c0a45b0fc01f062
Stored in directory: /root/.cache/pip/wheels/42/56/c1/548bf8661aaf0b07fb110477067c1f45c513a64121a001
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0.post1

```
[32] #creating bag of word model
from sklearn.feature_extraction.text import CountVecorizer
#to extract max features, "max_features" is attribute to
#experiment with to get better results
cv=CountVecorizer(max_features= 6000)
#z contains vectorized data (independent variable)
z=cv.fit_transform(corpus).toarray()
```

z.shape
(50000, 6000)

```
[ ] #save bag of word model
import joblib
joblib.dump(cv.vocabulary_,"amazon.save")
```

0s completed at 20:50

The screenshot shows the Amazon IPYNB interface with the following code and output:

```
[31] #save bag of word model
import joblib
joblib.dump(cv.vocabulary_,"amazon.save")
```

```
[35] y=data.iloc[:,0].values
y
array([0, 0, 0, ..., 0, 0, 0])
```

```
[36] 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)
```

```
[37] x_train.shape
(40000, 6000)
```

```
[38] x_test.shape
(10000, 6000)
```

z.shape
(50000, 6000)

```
[ ] #import libraries
```

0s completed at 20:52

```
amazon.ipynb - Colaboratory
colabresearch.google.com/drive/1qGCXXZ0P1JWwMM_EEBhCMMCKXZm6H4Krcd0fo=Q3w5IVUR/p
amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[40] import libraries
import tensorflow
import keras
import Sequential
from tensorflow.keras.models import Sequential
import Dense
from tensorflow.keras.layers import Dense

[41] initialize the model
model=Sequential()

adding input layer
model.add(Dense(500,kernel_initializer='random_uniform',
activation='relu'))

adding hidden layer
model.add(Dense(100,kernel_initializer='random_uniform',
activation='relu'))

adding output layer
model.add(Dense(1,kernel_initializer='random_uniform',
activation='sigmoid'))

configure the learning process.
model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])

print(x_train)
[[0 0 0 ... 0 0 0]]
0s - completed at 20:53
```

```
amazon.ipynb - Colaboratory
colabresearch.google.com/drive/1qGCXXZ0P1JWwMM_EEBhCMMCKXZm6H4Krcd0fo=whnFpJ-ZPMFL
amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[42] print(x_train)
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

training the model
model.fit(x_train,y_train,epochs=20,batch_size=32)

Epoch 1/20
1250/1250 [=====] - 43s 35ms/step - loss: 0.2035 - accuracy: 0.9252
Epoch 2/20
1250/1250 [=====] - 48s 32ms/step - loss: 0.1114 - accuracy: 0.9576
Epoch 3/20
1250/1250 [=====] - 48s 32ms/step - loss: 0.0575 - accuracy: 0.9877
Epoch 4/20
1250/1250 [=====] - 48s 32ms/step - loss: 0.0098 - accuracy: 0.9984
Epoch 5/20
1250/1250 [=====] - 48s 32ms/step - loss: 6.8669e-04 - accuracy: 0.9998
Epoch 6/20
1250/1250 [=====] - 42s 33ms/step - loss: 7.1589e-05 - accuracy: 1.0000
Epoch 7/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.7642e-05 - accuracy: 1.0000
Epoch 8/20
1250/1250 [=====] - 41s 33ms/step - loss: 7.8323e-06 - accuracy: 1.0000
Epoch 9/20
1250/1250 [=====] - 42s 34ms/step - loss: 3.7948e-06 - accuracy: 1.0000
Epoch 10/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 11/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 12/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 13/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 14/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 15/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 16/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 17/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 18/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 19/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
Epoch 20/20
1250/1250 [=====] - 48s 32ms/step - loss: 1.8261e-06 - accuracy: 1.0000
13m 43s - completed at 21:07
```



```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qGCIXZ0PUZWNHJE3EnCMACRC2mKH#scrollTo=3PFGQRIpILZ

amazon.ipynb
File Edit View Insert Runtime Tools Help All templates saved

+ Code + Text
[41] Epoch 10/20 [=====] - 40s 32m/step - loss: 1.8263e-06 - accuracy: 1.0000
Epoch 11/20 [=====] - 40s 32m/step - loss: 9.0213e-07 - accuracy: 1.0000
Epoch 12/20 [=====] - 40s 32m/step - loss: 4.5210e-07 - accuracy: 1.0000
Epoch 13/20 [=====] - 41s 32m/step - loss: 2.2570e-07 - accuracy: 1.0000
Epoch 14/20 [=====] - 41s 33m/step - loss: 1.1774e-07 - accuracy: 1.0000
Epoch 15/20 [=====] - 41s 33m/step - loss: 6.1113e-08 - accuracy: 1.0000
Epoch 16/20 [=====] - 44s 35m/step - loss: 3.2802e-08 - accuracy: 1.0000
Epoch 17/20 [=====] - 41s 33m/step - loss: 1.8273e-08 - accuracy: 1.0000
Epoch 18/20 [=====] - 41s 33m/step - loss: 1.0481e-08 - accuracy: 1.0000
Epoch 19/20 [=====] - 41s 33m/step - loss: 6.1321e-09 - accuracy: 1.0000
Epoch 20/20 [=====] - 42s 33m/step - loss: 3.7390e-09 - accuracy: 1.0000
keras.callbacks.History at 0x7fa04fd1bc90

[44] #save the model
model.save("amazo.h5")

ypred=model.predict(x_test)
113/113 [=====] - 4s 11m/step

[ ] ypred

4s completed at 21:07
```

```
amazon.ipynb - Colaboratory
colab.research.google.com/notebook/1qGCIXZ0PUZWNHJE3EnCMACRC2mKH#scrollTo=xpuXKMP1b-

amazon.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text
313/313 [=====] - 4s 11m/step

[46] ypred

array([[0.0000000e+00],
       [0.0000015e+00],
       [0.0000010e+00],
       ...,
       [0.0000000e+00],
       [0.0000000e+00],
       [0.0000000e+00]], dtype=float32)

[47] #save bag of word model
import joblib
joblib.dump(cv.vocabulary_, "amazo.save")

['amazo.save']

[48] loaded=CountVecorizer(decode_error='replace',vocabulary=joblib.load('amazo.save'))

# "writing was good"
dvd.split('delimiter')
result=model.predict(loaded.transform(d))
print(result)
prediction=result%2
#print(prediction)
if prediction[0] == False:
    print("Positive review")
elif prediction[0] == True:
    print("Negative review")

0s completed at 21:08
```

```

[45] print("negative review")

1/1 [=====] - 0% 129ms/step
[[2.668493e-06]]
Positive review

[50] from tensorflow.keras.models import load_model
model=tensorflow.keras.models.load_model("amazo.h5")

[51] #import load_model function
from tensorflow.keras.models import load_model
#load our saved model file
model=tensorflow.keras.models.load_model("amazo.h5")
#import countvectorizer
from sklearn.feature_extraction.text import CountVectorizer
import joblib
#load saved bag of word model file
loaded=CountVectorizer(decode_error='replace',vocabulary=joblib.load("amazo.sav"))

d="good with application"
d=d.split('delimiter')
result=model.predict(loaded.transform(d))
print(result)
prediction=result>0.5
#print(prediction)
if prediction[0] == False:
    print("Positive review")
elif prediction[0] == True:
    print("negative review")

```

0s completed at 21:09

```

[[2.668493e-06]]
Positive review

[50] from tensorflow.keras.models import load_model
model=tensorflow.keras.models.load_model("amazo.h5")

[51] #import load_model function
from tensorflow.keras.models import load_model
#load our saved model file
model=tensorflow.keras.models.load_model("amazo.h5")
#import countvectorizer
from sklearn.feature_extraction.text import CountVectorizer
import joblib
#load saved bag of word model file
loaded=CountVectorizer(decode_error='replace',vocabulary=joblib.load("amazo.sav"))

d="good with application"
d=d.split('delimiter')
result=model.predict(loaded.transform(d))
print(result)
prediction=result>0.5
#print(prediction)
if prediction[0] == False:
    print("Positive review")
elif prediction[0] == True:
    print("Negative review")

1/1 [=====] - 0% 09ms/step
[[5.848785e-10]]
Positive review

```

0s completed at 21:11

