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
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),BOLLEPALLYRAMYA(H.NO:19 UK1A0553),***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.PROJECT OBJECTIVE

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 PROBLEM STATEMENT

Given a dataset containing of various attributes,use the features available in dataset and define a supervised classification algorithm which can identify wether 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 SOLLUTION

All Information in the world can be broadly classified into mainly two categories, facts and opinions. Factsare 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.

Preparing Review Database

↓

POS Tagging

↓

Feature Extraction

↓

Opinion Word Extraction

↓

Opinion Word Polarity Identification

↓

Opinion Sentences Polarity Identification

↓

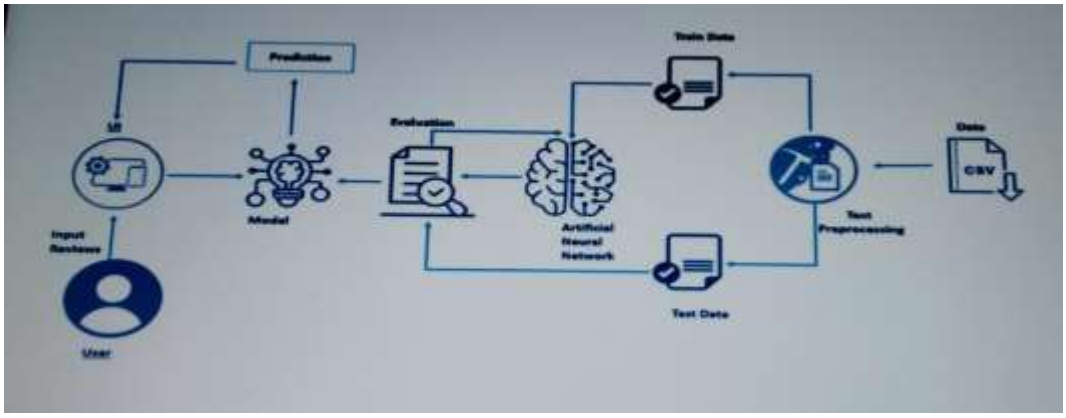
Summary Generation

<u>Basic Step of Feature Extraction</u>

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

3.3 PROJECT FLOW

below the project flow to be followed while developing the project.

- User interacts with the UI (User Interface) to enter the review
- Entered review is analysed by the model which is integrated
- Once the model analyses the input prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

- Data Collection.
    - Collect the dataset or Create the dataset
- Text Preprocessing.
    - Import the Libraries.
    - Importing the dataset.
    - Remove Punctuations
    - Convert each word into lower case.
    - Stemming.
    - Splitting Data into Train and Test.
- Model Building
    - Import the model building Libraries
    - Initializing the model
    - Adding Input Layer
    - Adding Hidden Layer
    - Adding Output Layer
    - Configure the Learning Process
    - Training and testing the model
    - Optimize the Model
    - Save the Model
- Application Building
    - Create an HTML file
    - Build Python Code

4.SYSTEM DESIGN

SYSTEM FLOW

In this system, we are finding ratings for feature reviews only and not for service and product review. Since on Amazon reviews there are ratings available for each review, the sentiment for the product review and service review will be equivalent to the review given by the customer so the computational task of finding the sentiments is reduced in case of service and product reviews.

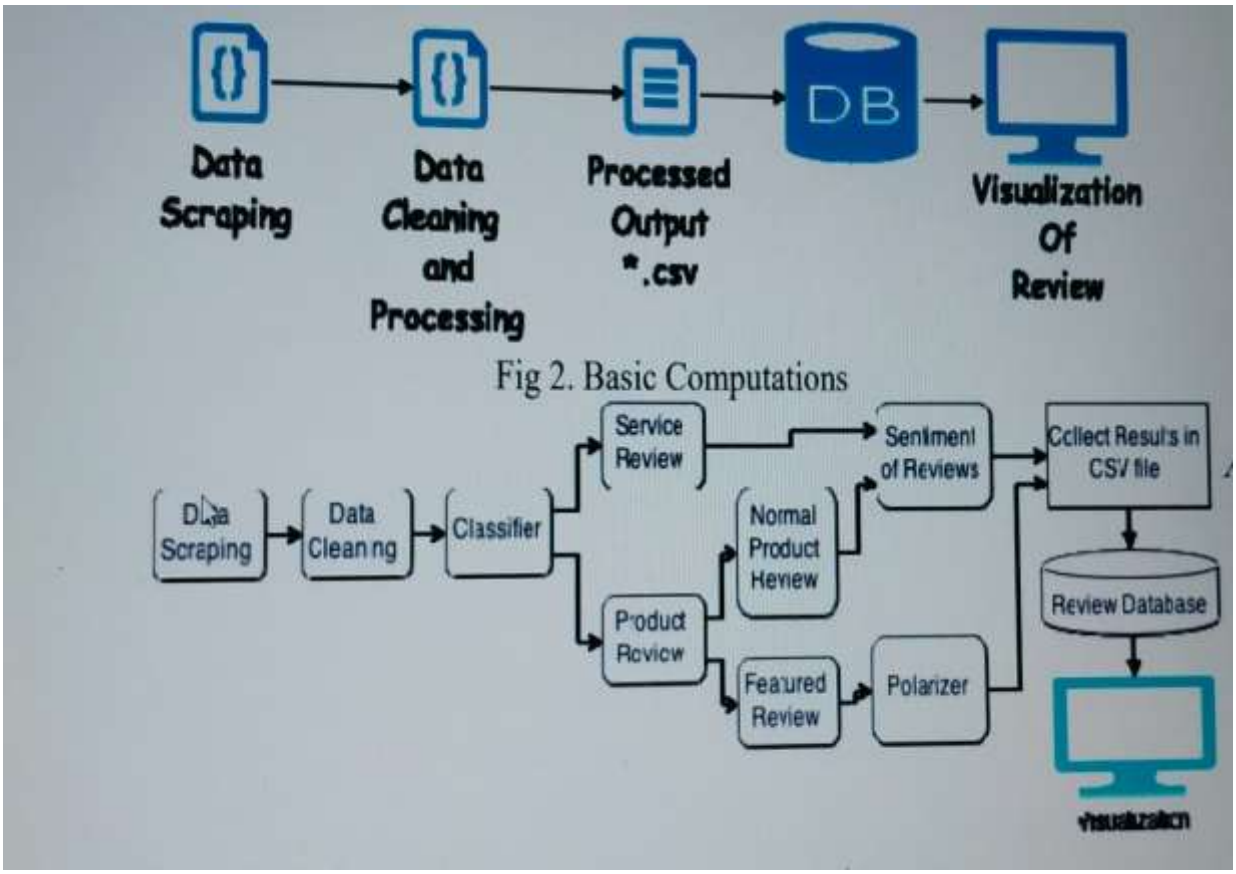


Fig 3.Detaialed block diagram of system

In Fig 2, we have our basic workflow that we have followed to get the opinion mining of our test case of iPhone 5 reviews on amazon.com. The Steps are as follows:

1. **Data Scraping** :

Crawl the amazon review url to extract all required details from it. We need to take care of the text so as to satisfy the required format, for e.g.tags have a special meaning to the browser i.e. break read or next line, we need to explicitly convert each tag to spaces or else the crawling result will be improper. When working with online reviews there is always a question in our mind, how can I trust the review. This is not a problem with amazon reviews, amazon reviewers can up vote or down vote a review, this collectively is available as helpful count. We have taken a special care in extracting the data from web pages smallest necessary data is extracted for processing. The following is the list of items that we have extracted: Review of Title, Helpful Count, User

Review and Date of Review. Caution: Websites uses utf-8 character set for encoding characters, but sometimes this encoding can give errors during web scraping as scraping involves matching strings and patterns. Solution to this is simply enforce the string to be coded in utf-8 format.

2. Data Cleaning and Processing :

The data extracted need to be cleaned so that we get proper text review on which analysis can be performed. Cleaning of crawled data is done by removal of all special characters (such as: ":/.,'#$*^&-) in order to retrieve best results. After cleaning the crawled content copy it into a csv file. The next step is processing the cleaned data, firstly review is classified as service, feature or product review. If the review is a feature review then feature extraction is done using POS Tagging and grammar rule all stated below. After feature extraction the feature opinion polarization is obtained.

3. All processed output is stored in one csv file for further use.

4. The file is then loaded into the database for use in visualization and summarization.

5. Finally the summarization of sentiments is generated as charts and displayed to the user as an attractive dashboard.

5.METHODOLOGY

Amazon is one of the largest E-commerce sites as for that there are innumerous reviews that can be seen. We used data named Amazon Kindle Reviews which was provided in Kaggle. The dataset is a .csv file consisting of labeled data having the following columns –

"reviewerID": D 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 (raw)
"unixreviewTime": unix timestamp

For the dataset we selected, it consists of more than 50,000 kindle book reviews. From the format used analyzing the review polarity we used review Text & Overall from it. We can see an overview of our methodology:

A.DATA COLLECTION

We acquired our dataset in .csv format that was already labeled from kaggle. As we have a large amount of reviews, manually analysing was quite impossible for us. Therefore we preprocessed our data and used elementary data analysis techniques to pre-process the datasets. As amazon reviews come in 5-star rating based generally 2 star ratings are either positive or negative. So we wrote a function that considers 5,4 and 3 ratings to be positive and 2,1 ratings to be negative reviews and proceed to the next step.

B.PRE-PROCESSING

Tokenization: It is the process of separating a sequence of strings into individuals such as words, keywords, phrases, symbols and other elements known as tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens work as the input for different processes like parsing and text mining.

Stemming: Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.

A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”. Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words**.**

Removing Stop Words: Stop words are those objects in a sentence which are not necessary in any sector in text mining. So we generally ignore these words to enhance the accuracy of the analysis. In different formats there are different stop words depending on the country, language etc. In English format there are several stop words.

C.FEATURE EXTRACTION

Bag-of-Words: It is one of the most fundamental methods to transform tokens into a set of features. The BoW model is used in document classification, where each word is used as a feature for training the classifier. For example, in a task of review based sentiment analysis, the presence of words like 'fabulous', 'excellent' indicates a positive review, while words like 'annoying', 'poor' point to a negative review .

There are 3 steps while creating a BoW model :

1. Text pre-processing
2. Creating the vocabulary
3. Creating matrix of features- Text Vectorisation

D. CNN MODEL:

We use a Convolutional Neural Network (CNN) as they have proven to be successful at document classification problems. A conservative CNN configuration is used with 32 filters (parallel fields for processing words) and a kernel size of 8 with a rectified linear ('relu') activation function. This is followed by a pooling layer that reduces the output of the convolutional layer by half. Next, the 2D output from the CNN part of the model is flattened to one long 2D vector to represent the 'features' extracted by the CNN. The back-end of the model is a standard Multilayer Perceptron layer to interpret the CNN features. The output layer uses a sigmoid activation function to output a value between 0 and 1 for the negative and positive sentiment in the review.

E.TRAIN AND TEST DATA:

We are pretending that we are developing a system that can predict the sentiment of a textual book review as either positive or negative.This means that after the model is developed, we will need to make predictions on new textual reviews. This will require all of the same data preparation to be performed on those new reviews as is performed on the training data for the model. We will ensure that this constraint is built into the evaluation of our models by splitting the training and test datasets prior to any data preparation. This means that any knowledge in the data in the test set that could help us better prepare the data (e.g. the words used) are unavailable in the preparation of data used for training the model. That being said, we will use 80% train, 20% as a test of the data

F.EVALUATION MODEL:

evaluation is an integral part of the model development process. It helps to find the best model that represents the data and how well the chosen model will work in the future.The output is predicted by analyzing the test data as input along with test data output and then the output is displayed.

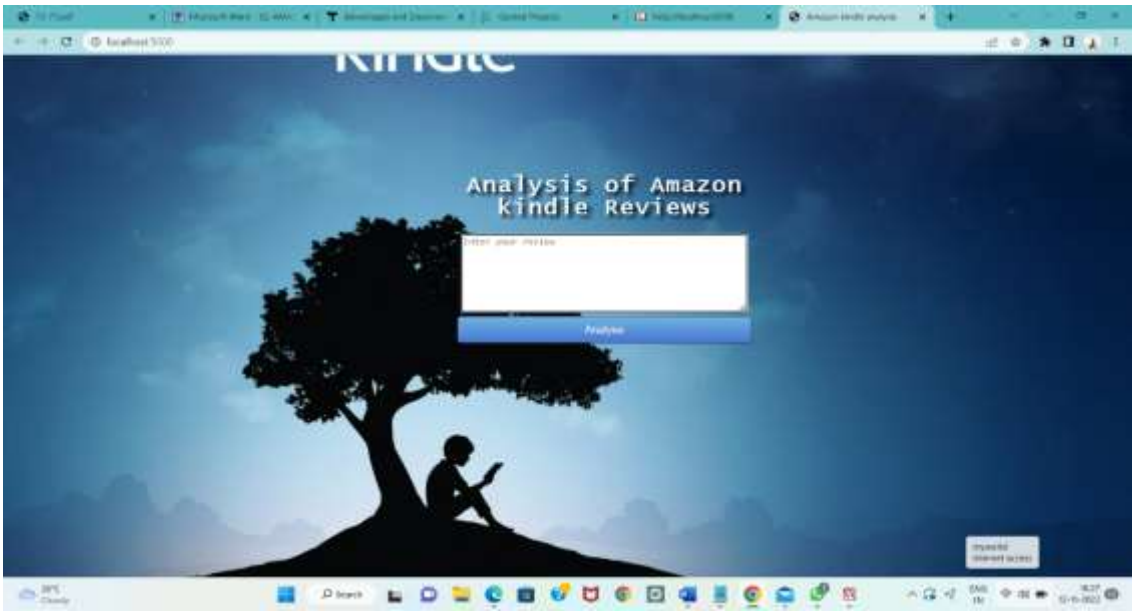
Accuracy: Accuracy predicts how often the classifier makes the correct prediction. Accuracy is the ratio between the number of correct predictions and the total number of predictions.

G.INTERFACE:

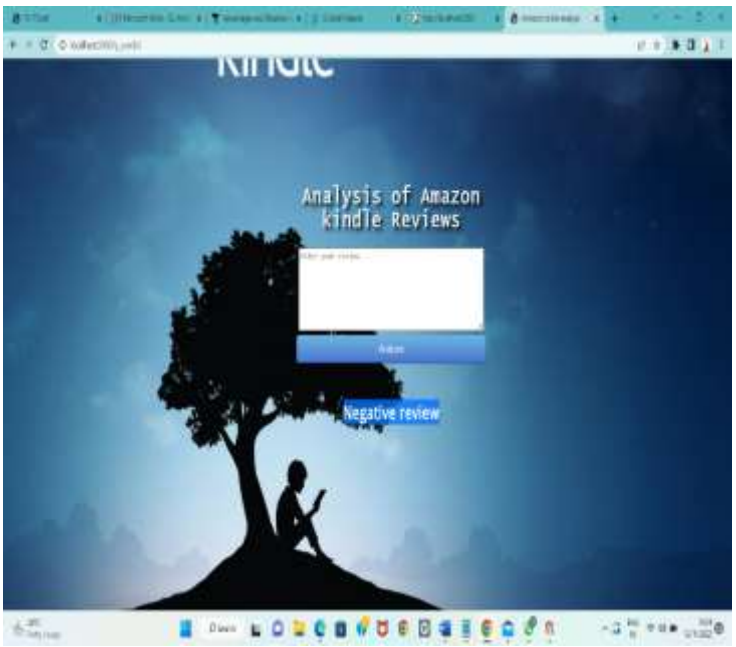
A web interface is built to take input and display an output.

Flask web framework is used to build a web interface and other libraries are used to integrate the model.

6.RESULT

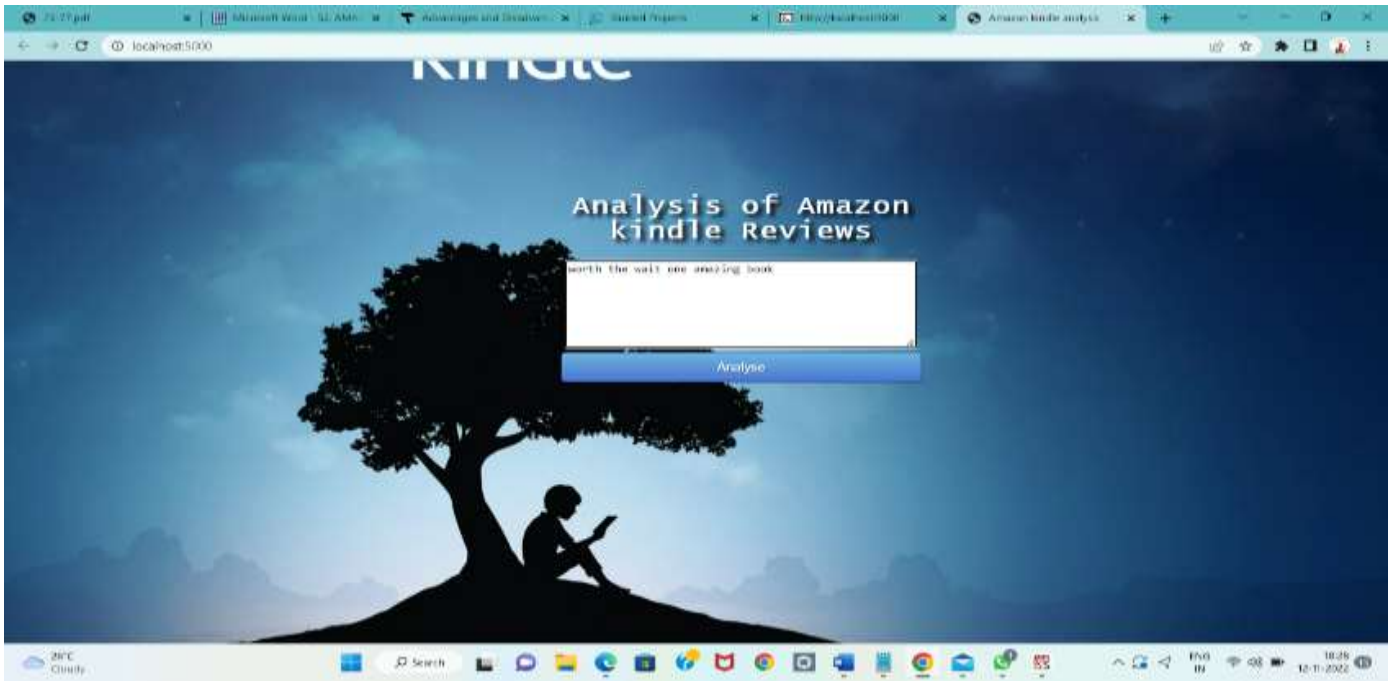


HOMEPAGE

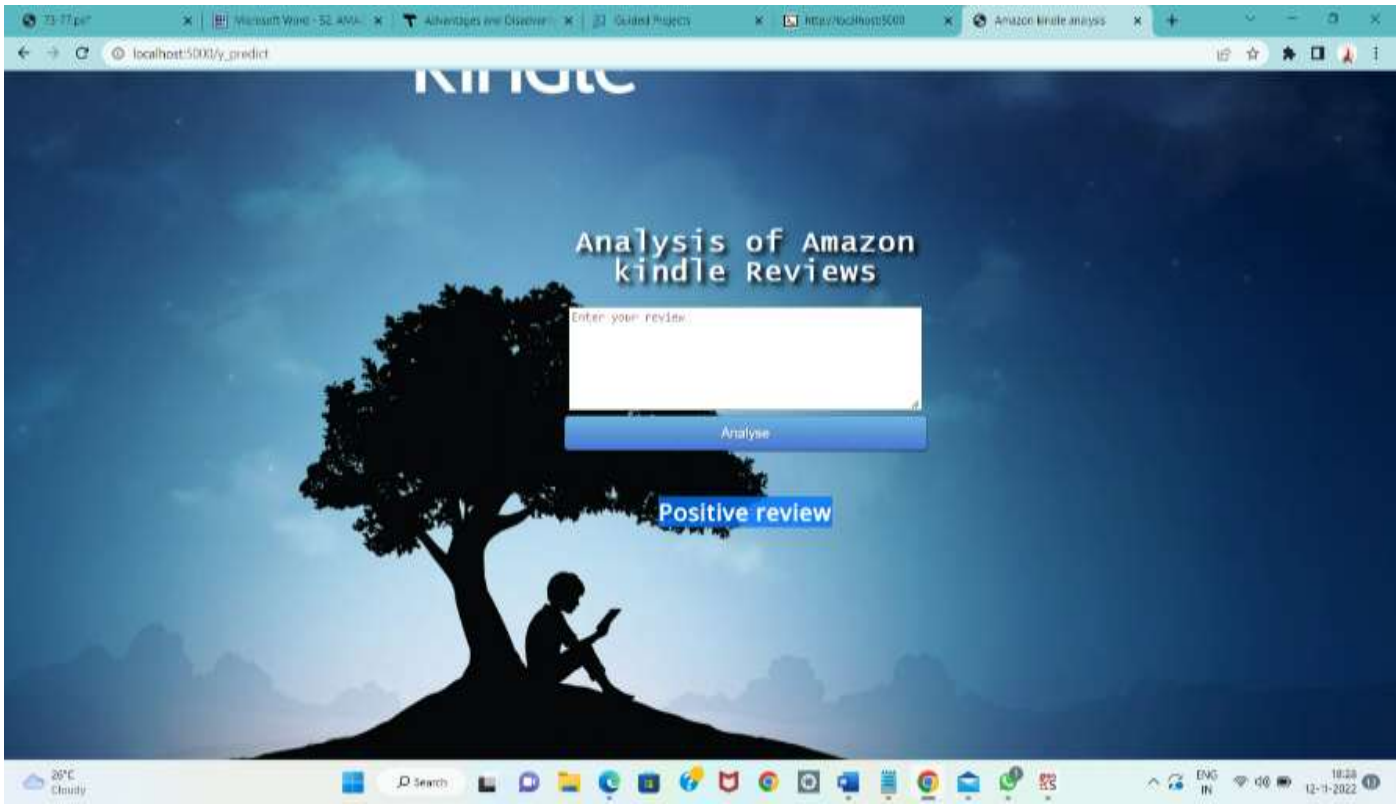


REVIEW ANALYSIS PAGE

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

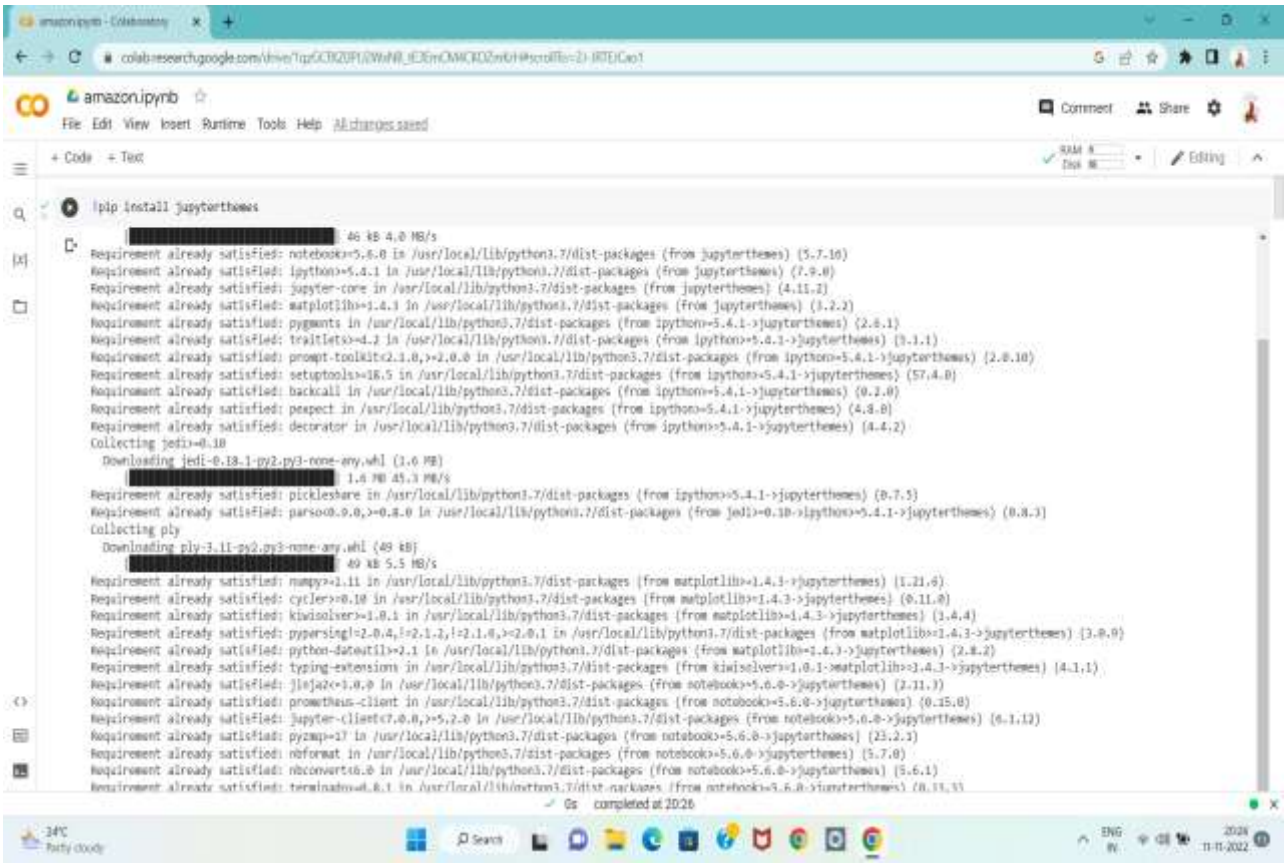
8.CONCLUSION AND FUTURE SCOPE

It is completely impossible to use only raw text as input for making predictions. Hence,we saw that the pre-processing 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.

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.

9.CODE SNIPPETS

A.MODELBUILDING



amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[2] !jt -t monokai -f fira -fs 13 -nf ptsans -nfs 10 -N -kl -cursw 5 -cursc r -cellw 98% -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)
```

✓ 0s completed at 20:26

amazon.ipynb

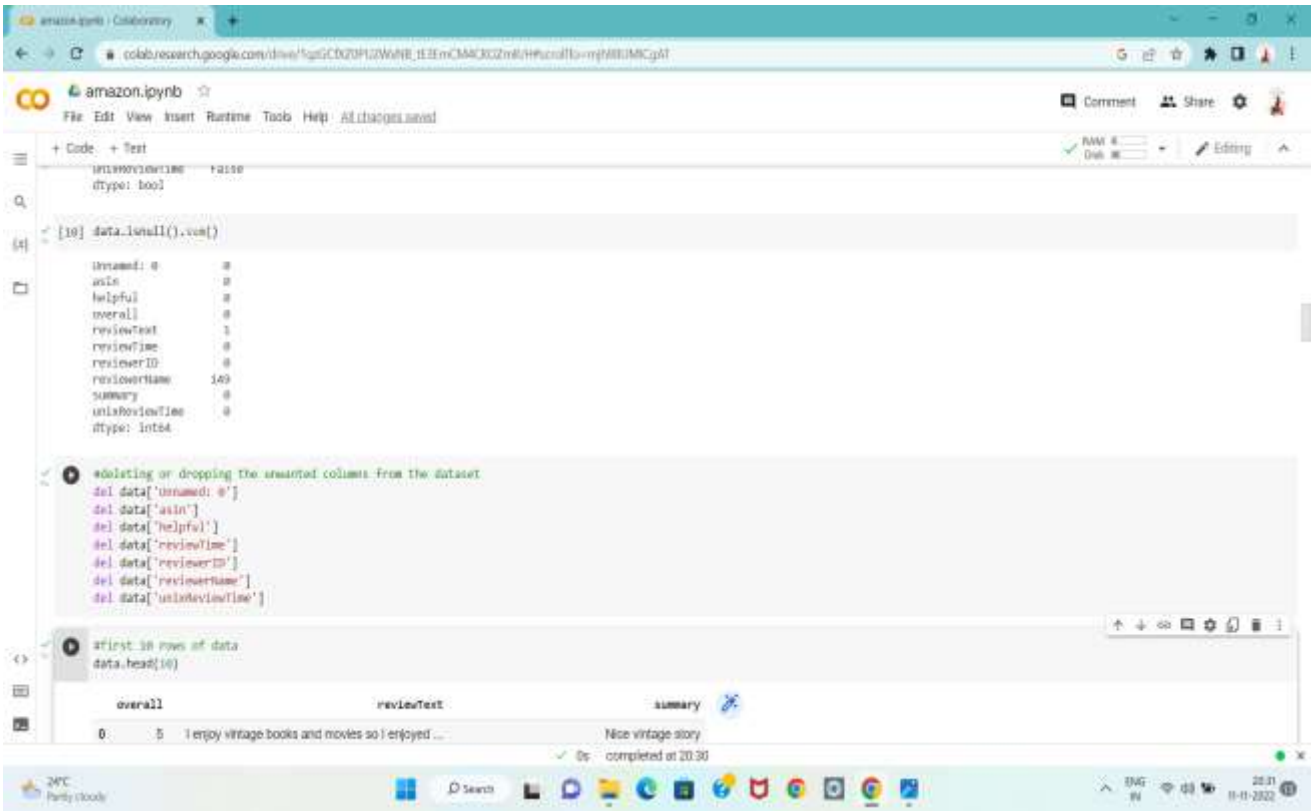
File Edit View Insert Runtime Tools Help All changes saved

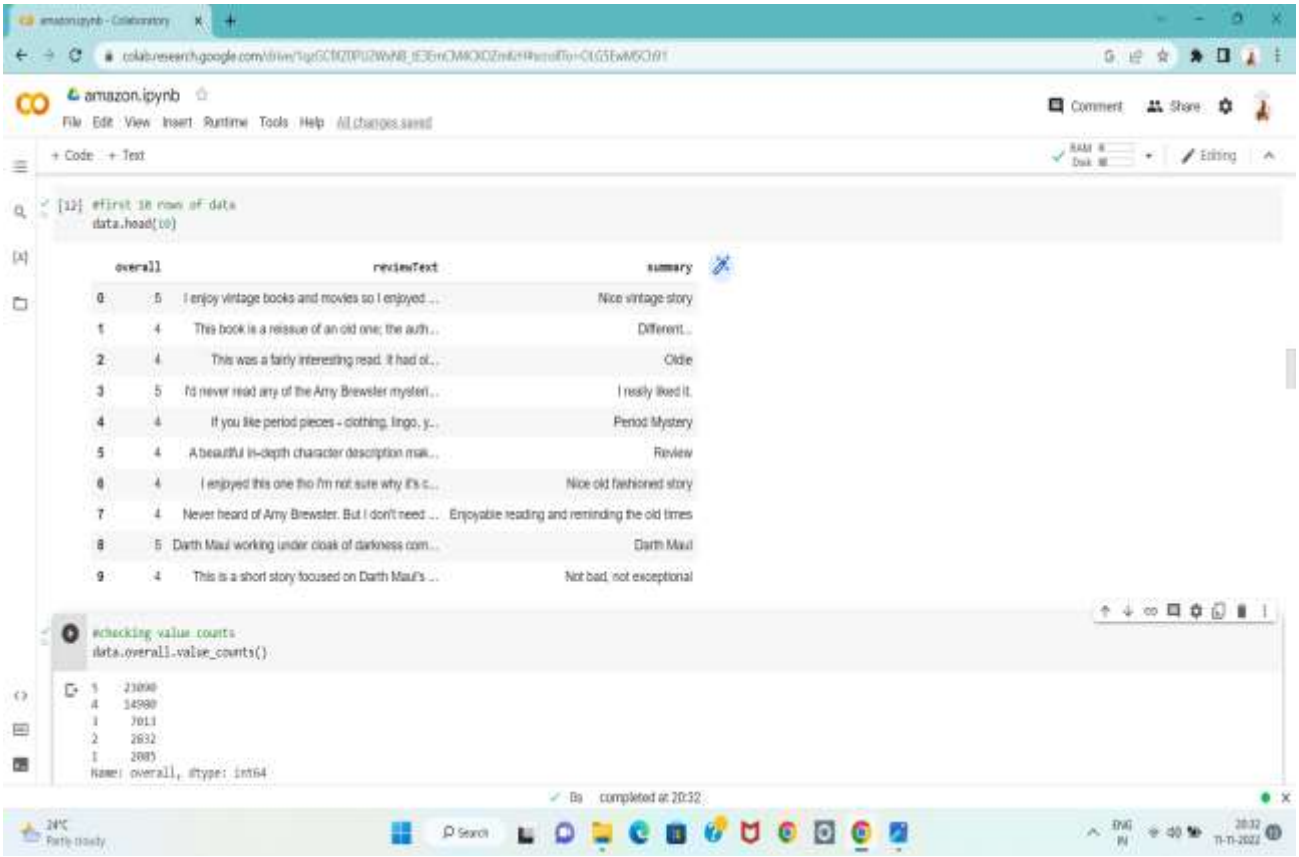
```
[7] data.head()
```

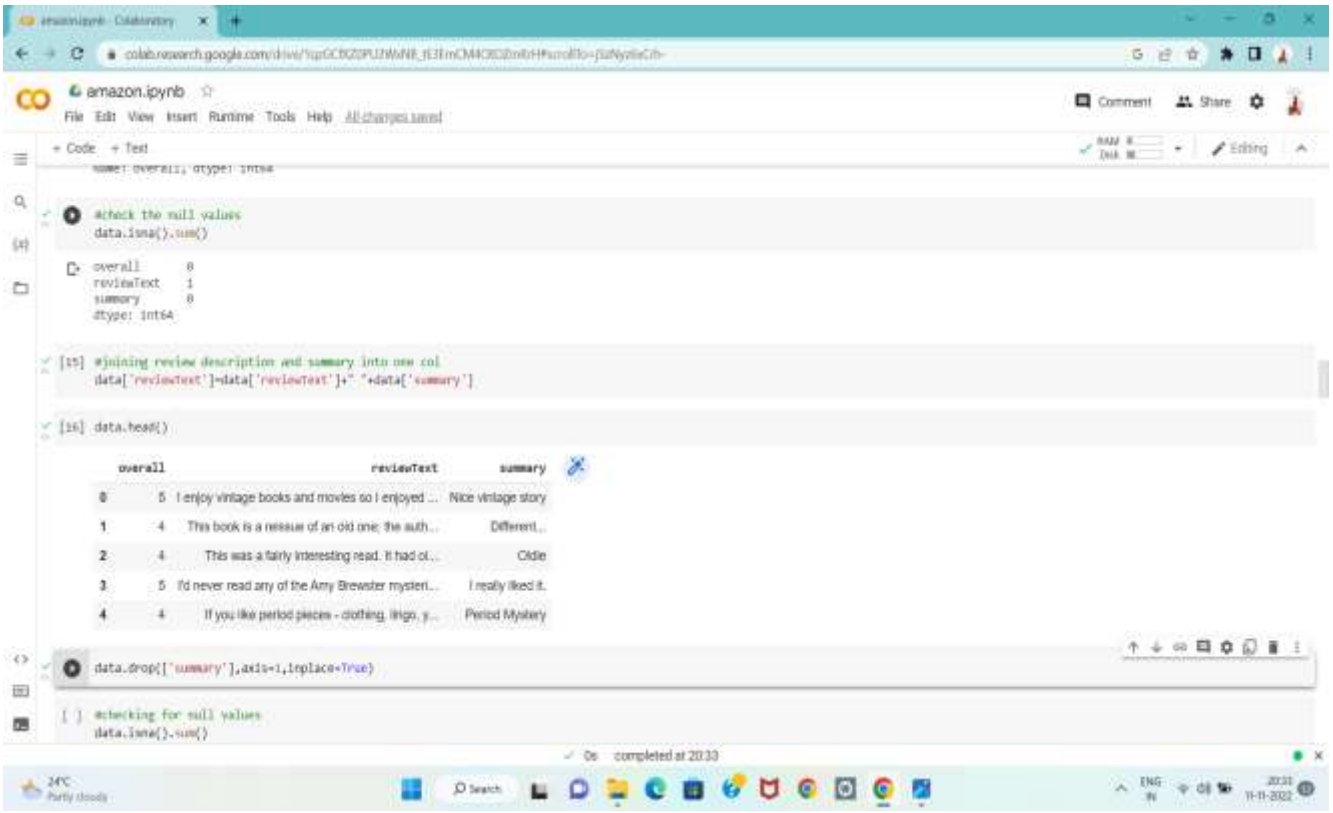
	Unnamed: 0	asin	helpful	overall	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime
0	0	B000F83SZQ	[0, 0]	5	I enjoy vintage books and movies so I enjoyed ...	05 5, 2014	A1F6404F1VG29J	Avidreader	Nice vintage story	1399249000
1	1	B000F83SZQ	[2, 2]	4	This book is a reissue of an old one; the auth...	01 6, 2014	AN0N05A9LIJEQ	critters	Different...	1388966400
2	2	B000F83SZQ	[2, 2]	4	This was a fairly interesting read. It had ol...	04 4, 2014	A795DMNCJILA6	dot	Oldie	1396569600
3	3	B000F83SZQ	[1, 1]	5	I'd never read any of the Amy Brewster mystert...	02 19, 2014	A1FV0SX13TWVXQ	Elaine H. Turley "Montana Songbird"	I really liked it.	1392768000
4	4	B000F83SZQ	[0, 1]	4	If you like period pieces - clothing, lingo, y...	03 19, 2014	A3SPTOKDG7WBLN	Father Dowling Fan	Period Mystery	1395187200

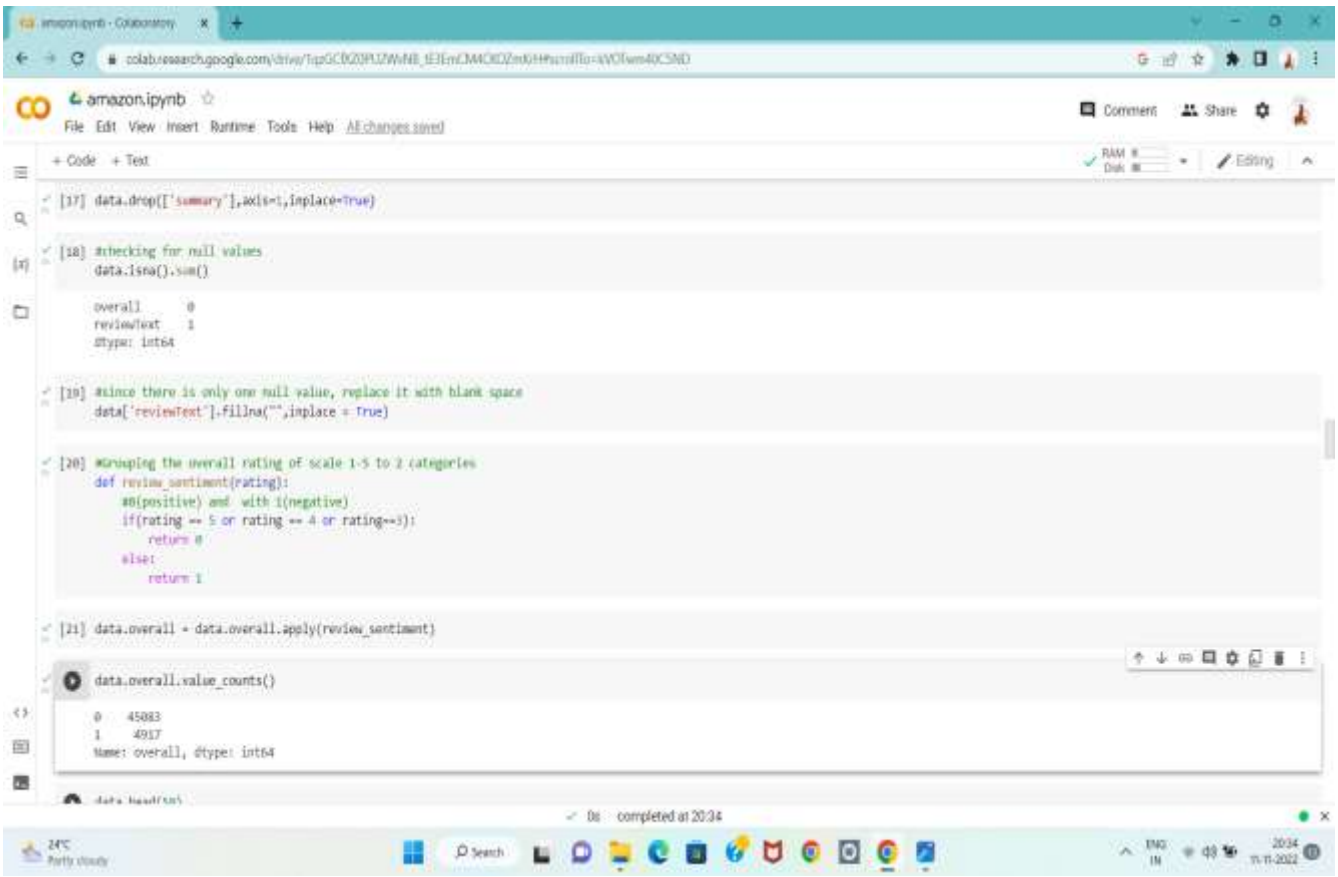
```
[8] #assigning 50000 rows to data
data=data.head(50000)

#checking for null values
data.isnull().any()

Unnamed: 0      False
asin            False
helpful         False
overall         False
reviewText       True
reviewTime      False
reviewerID      False
reviewerName     True
summary         False
unixReviewTime  False
```

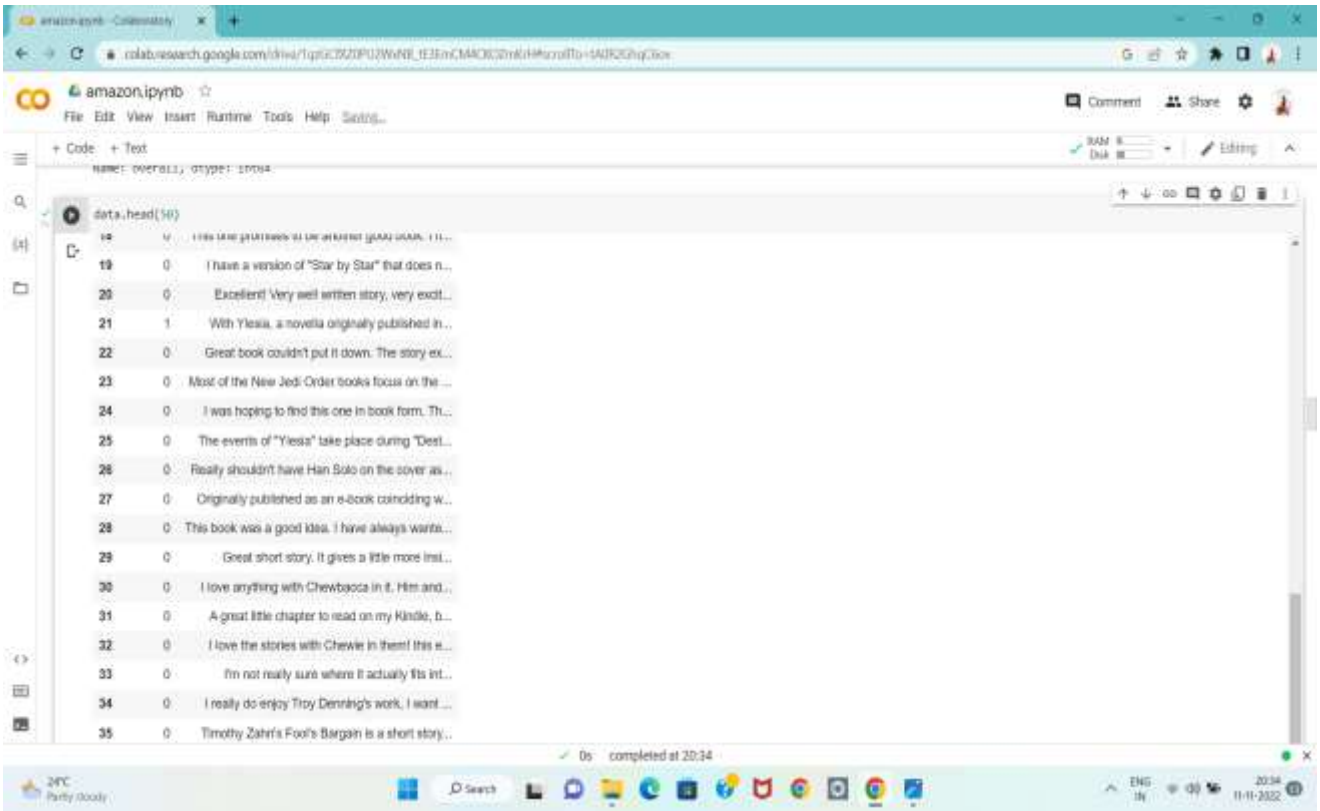
✓ 0s completed at 20:26

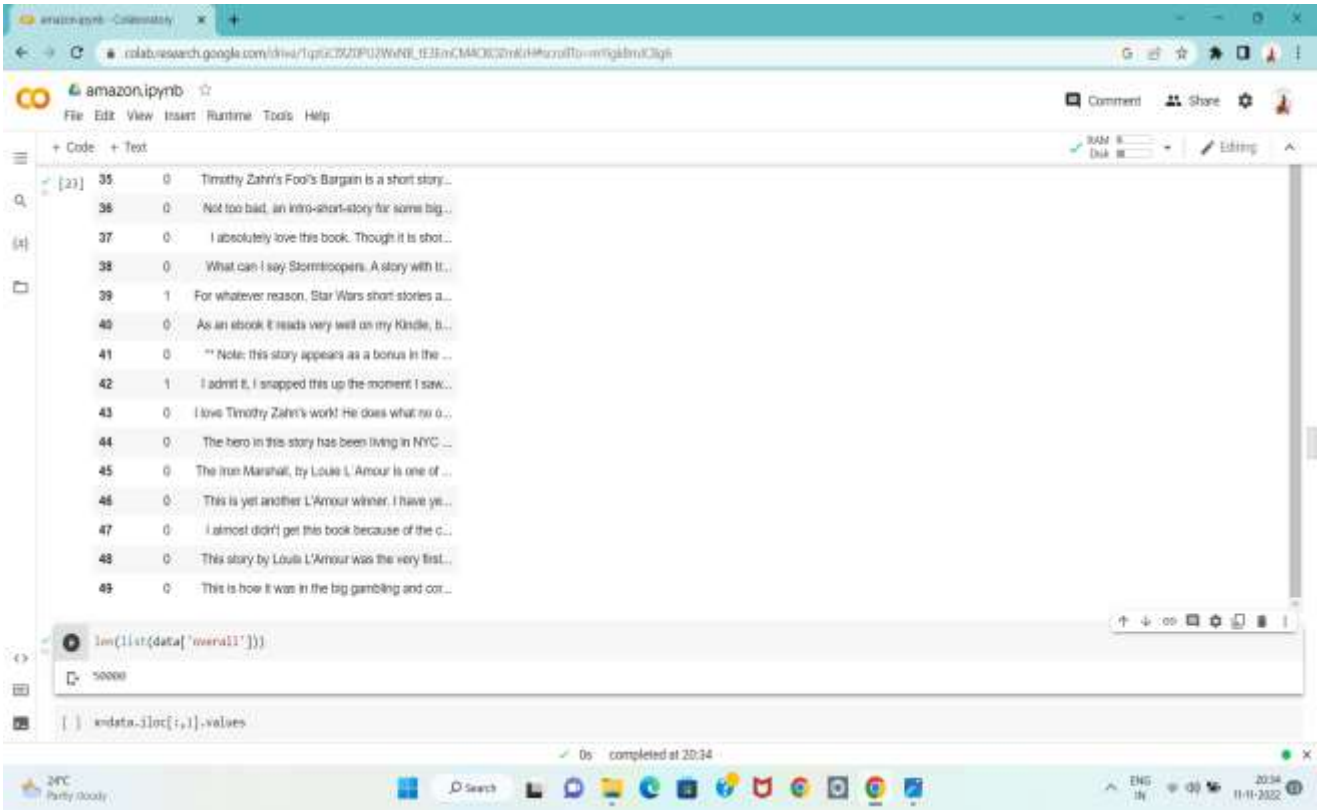

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
dtype: bool
[10] data.isnull().sum()
unnamed: 0 0
asin 0
helpful 0
overall 0
reviewText 1
reviewTime 0
reviewerID 0
reviewerName 149
summary 0
unixReviewTime 0
dtype: int64
#deleting or dropping the unwanted columns from the dataset
del data['unnamed: 0']
del data['asin']
del data['helpful']
del data['reviewTime']
del data['reviewerID']
del data['reviewerName']
del data['unixReviewTime']
#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
completed at 20:30

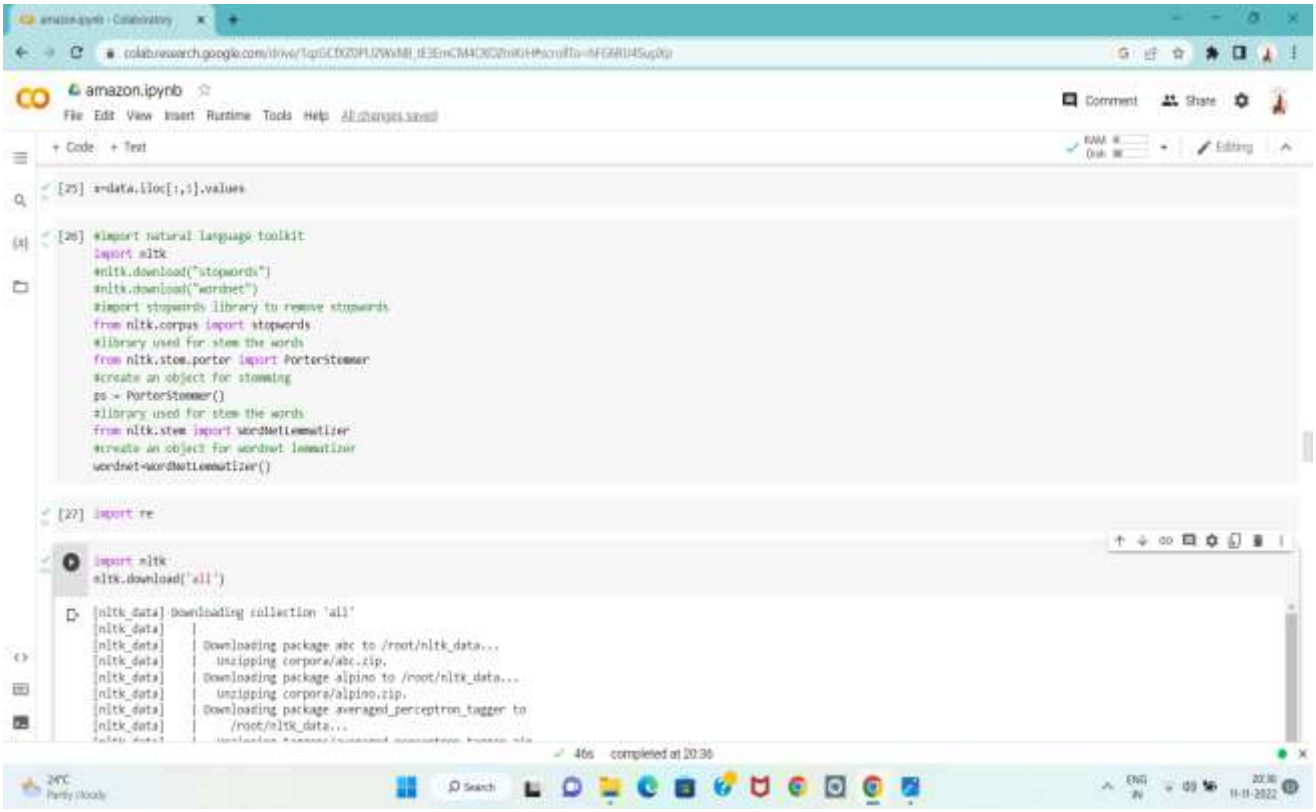

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[12] #first 10 rows of data
data.head(10)
overall reviewText summary
0 5 I enjoy vintage books and movies so I enjoyed ... Nice vintage story
1 4 This book is a reissue of an old one; the auth... Different...
2 4 This was a fairly interesting read. It had ol... Oldie
3 5 I'd never read any of the Amy Brewster mysteri... I really liked it.
4 4 If you like period pieces - clothing, lingo, 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 23890
4 14980
1 7013
2 2832
3 2085
Name: overall, dtype: int64
completed at 20:32

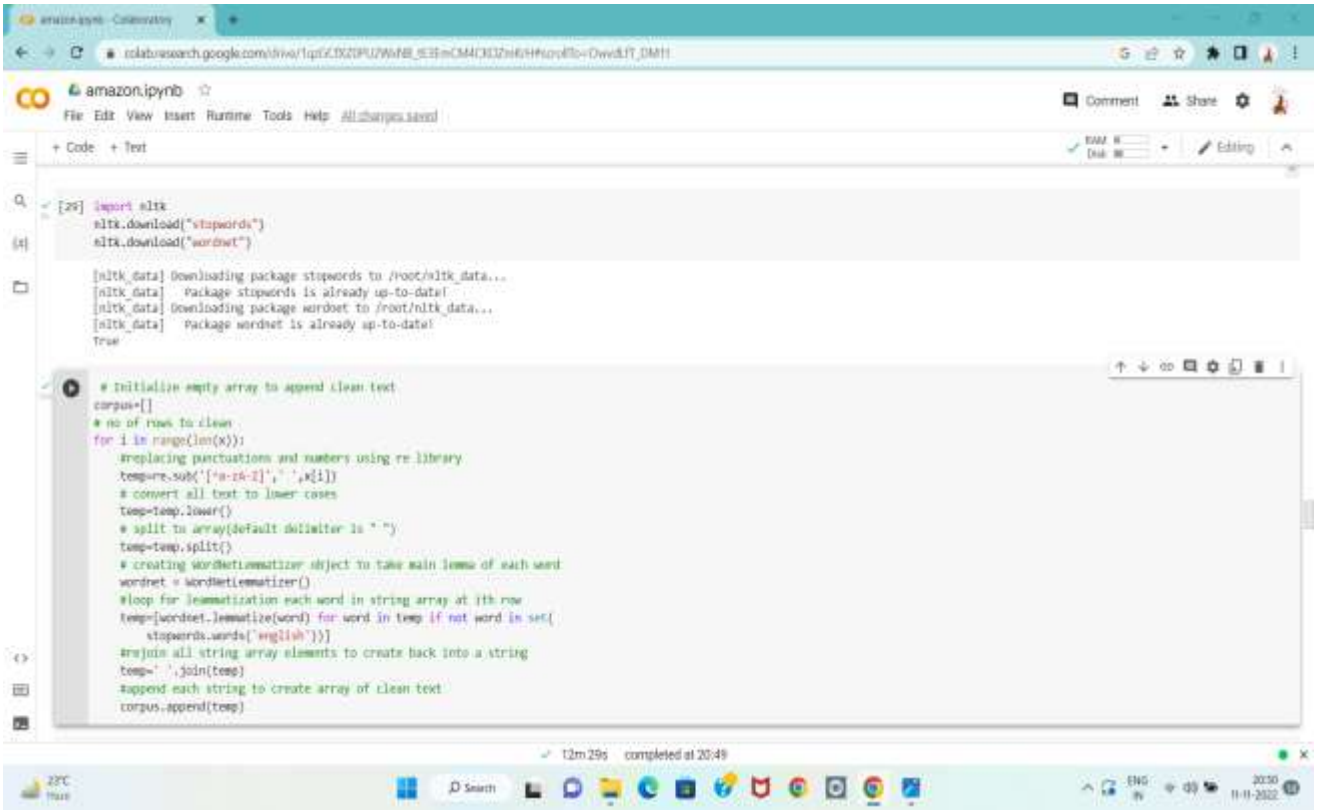

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
#check the null values
data.isna().sum()
overall 0
reviewText 1
summary 0
dtype: int64
[15] #joining review description and summary into one col
data['reviewText']=data['reviewText']+" "+data['summary']
[16] data.head()
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 Amy Brewster mysteri... I really liked it.
4 4 If you like period pieces - clothing, lingo, y... Period Mystery
data.drop(['summary'],axis=1,inplace=True)
[ ] #checking for null values
data.isna().sum()
completed at 20:33

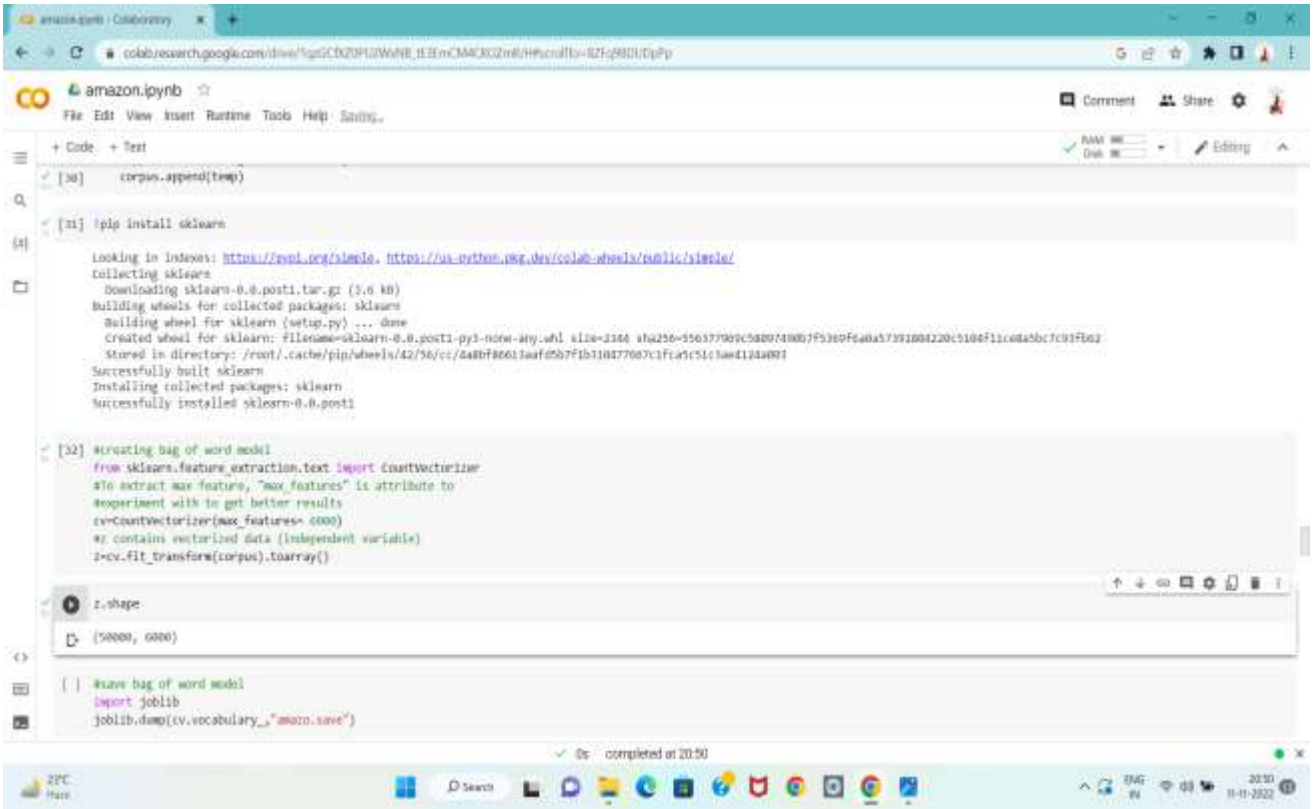

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[17] data.drop(['summary'],axis=1,inplace=True)
[18] #checking for null values
data.isna().sum()
overall 0
reviewText 1
dtype: int64
[19] #since there is only one null value, replace it with blank space
data['reviewText'].fillna("",inplace = True)
[20] #Grouping the overall rating of scale 1-5 to 2 categories
def review_sentiment(rating):
#0(positive) and with 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
completed at 20:34

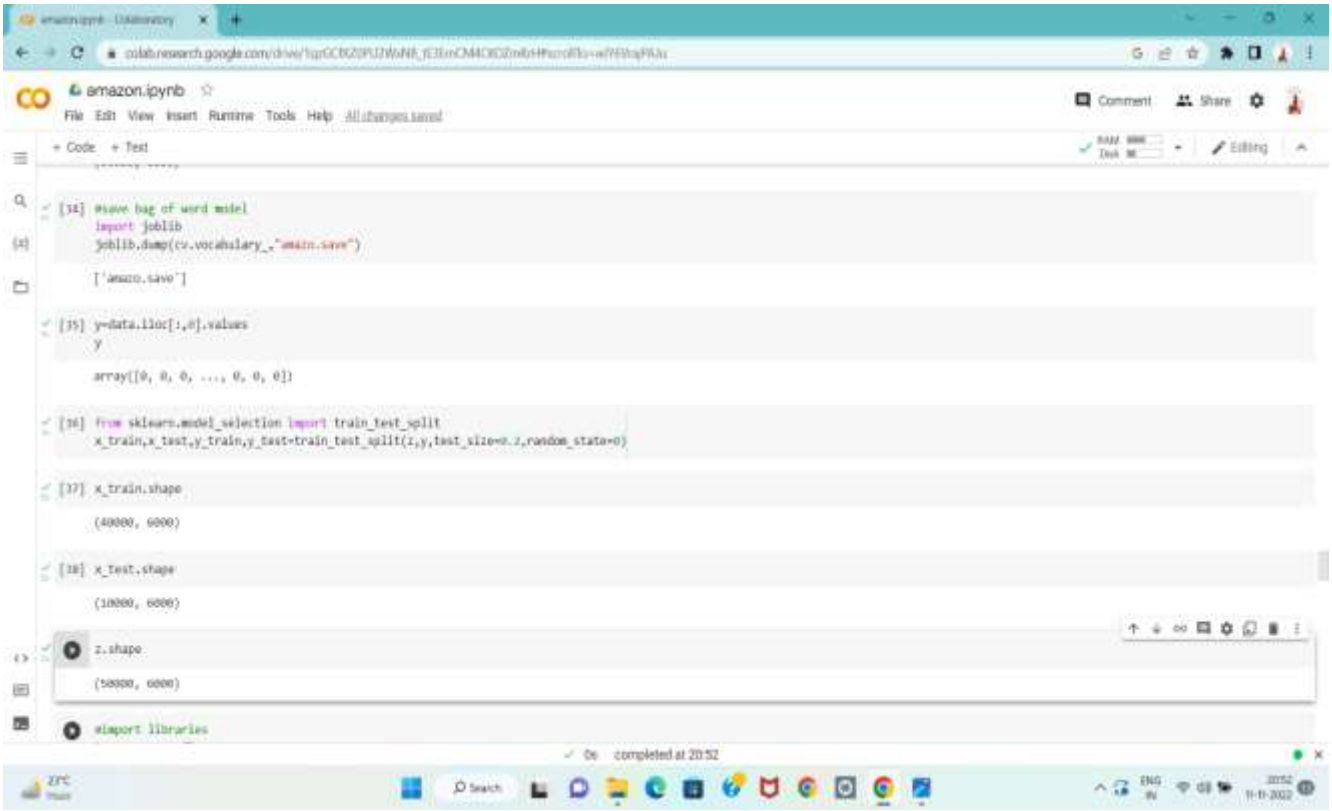

amazon.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
data.head(50)
19 0 I have a version of "Star by Star" that does n...
20 0 Excellent! Very well written story, very excit...
21 1 With Ylesia, a novella originally published in...
22 0 Great book couldn't put it down. The story ex...
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 wante...
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 Fool's Bargain is a short story...
completed at 20:34

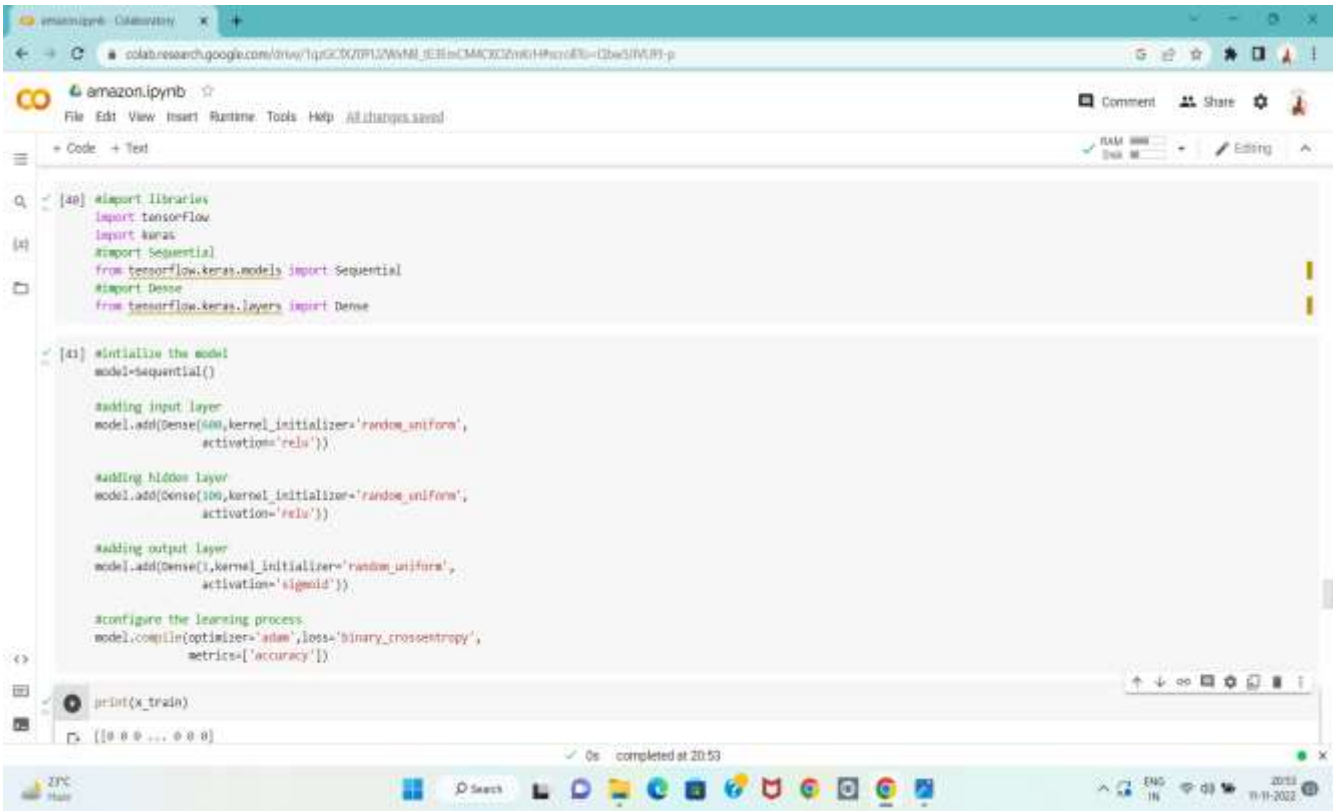

amazon.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[23] 35 0 Timothy Zahn's Fool's Bargain is a short story...
36 0 Not too bad, an intro-short-story for some big...
37 0 I absolutely love this book. Though it is shor...
38 0 What can I say Stormtroopers. A story with th...
39 1 For whatever reason, Star Wars short stories a...
40 0 As an ebook it reads very well on my Kindle, b...
41 0 ** Note: this story appears as a bonus in the ...
42 1 I admit it, I snapped this up the moment I saw...
43 0 I love Timothy Zahn's work! He does what no o...
44 0 The hero in this story has been living in NYC ...
45 0 The Iron Marshall, by Louis L' Amour is one of ...
46 0 This is yet another L'Amour winner. I have ye...
47 0 I almost didn't get this book because of the c...
48 0 This story by Louis L'Amour was the very first...
49 0 This is how it was in the big gambling and cor...
len(list(data['overall']))
50000
x=data.iloc[:,1].values
completed at 20:34


amazon.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[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')
[nltk_data] Downloading collection 'all'
[nltk_data] | Downloading package abc to /root/nltk_data...
[nltk_data] | Unzipping corpora/abc.zip.
[nltk_data] | Downloading package alpino to /root/nltk_data...
[nltk_data] | Unzipping corpora/alpino.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger to
[nltk_data] | /root/nltk_data...
46s completed at 20:36


amazon.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[26] import nltk
nltk.download("stopwords")
nltk.download("wordnet")
[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-zA-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'))]
#rejoin 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


amazon.ipynb
File Edit View Insert Runtime Tools Help Saving...
+ Code + Test
[30] corpus.append(temp)
[31] !pip install sklearn
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.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
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 CountVectorizer
#To extract max feature, "max_features" is attribute to
#experiment with to get better results
cv=CountVectorizer(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_,"amazo.save")
completed at 20:50

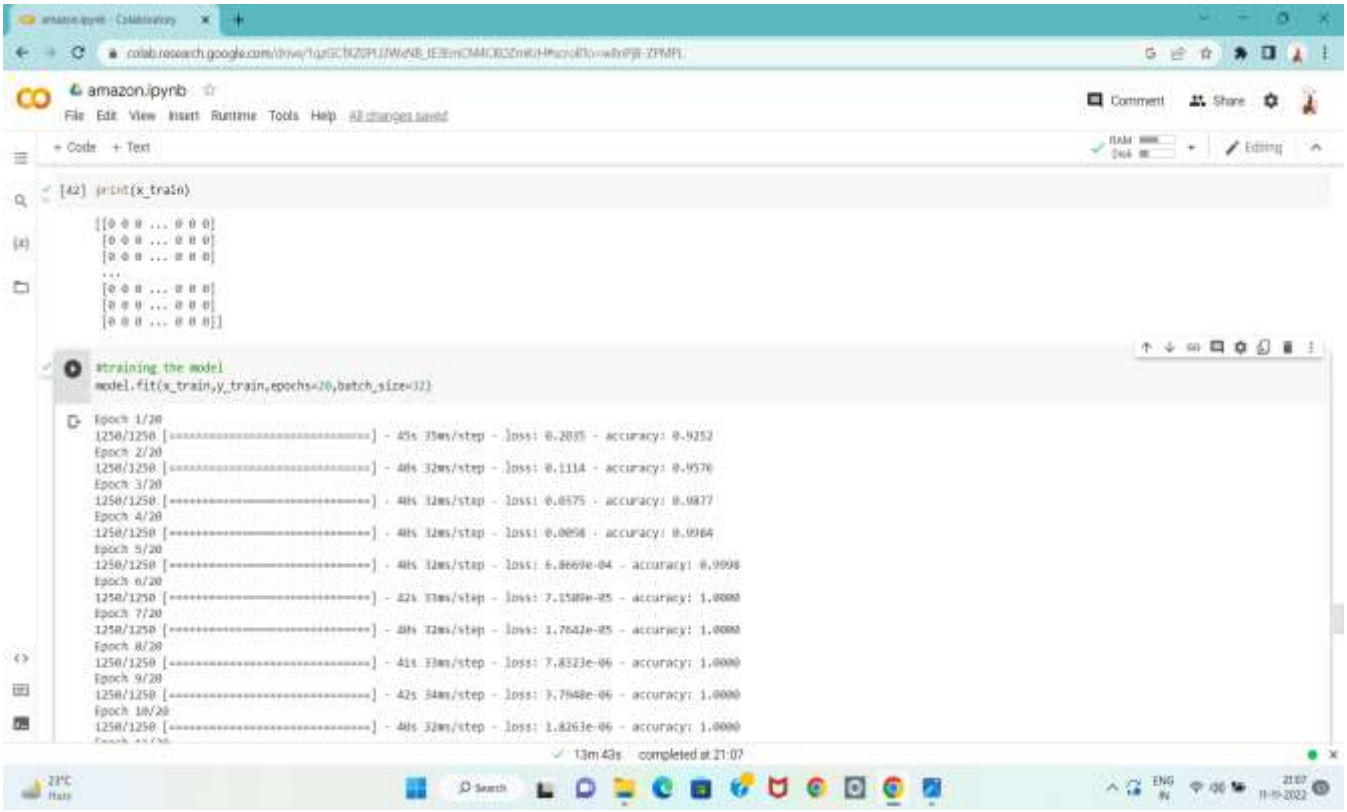

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Test
[34] #save bag of word model
import joblib
joblib.dump(cv.vocabulary_,"amazo.save")
['amazo.save']
[35] y=data.iloc[:,0].values
y
array([0, 0, 0, ..., 0, 0, 0])
[36] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(z,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
completed at 20:52



```
[40] #import libraries
     import tensorflow
     import keras
     #import Sequential
     from tensorflow.keras.models import Sequential
     #import Dense
     from tensorflow.keras.layers import Dense
```

```
[41] #intialize the model
     model=Sequential()

     #adding input layer
     model.add(Dense(600,kernel_initializer='random_uniform',
                     activation='relu'))

     #adding hidden layer
     model.add(Dense(300,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]
```



```
[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 [==============================] - 45s 35ms/step - loss: 0.2035 - accuracy: 0.9252
Epoch 2/20
1250/1250 [==============================] - 40s 32ms/step - loss: 0.1114 - accuracy: 0.9570
Epoch 3/20
1250/1250 [==============================] - 40s 32ms/step - loss: 0.0575 - accuracy: 0.9877
Epoch 4/20
1250/1250 [==============================] - 40s 32ms/step - loss: 0.0050 - accuracy: 0.9984
Epoch 5/20
1250/1250 [==============================] - 40s 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 [==============================] - 40s 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 [==============================] - 40s 32ms/step - loss: 1.8263e-06 - accuracy: 1.0000
```

```
[43] Epoch 10/20
     1250/1250 [==============================] - 40s 32ms/step - loss: 1.8263e-06 - accuracy: 1.0000
     Epoch 11/20
     1250/1250 [==============================] - 40s 32ms/step - loss: 9.0233e-07 - accuracy: 1.0000
     Epoch 12/20
     1250/1250 [==============================] - 40s 32ms/step - loss: 4.5219e-07 - accuracy: 1.0000
     Epoch 13/20
     1250/1250 [==============================] - 41s 32ms/step - loss: 2.2478e-07 - accuracy: 1.0000
     Epoch 14/20
     1250/1250 [==============================] - 41s 33ms/step - loss: 1.1774e-07 - accuracy: 1.0000
     Epoch 15/20
     1250/1250 [==============================] - 41s 33ms/step - loss: 6.1113e-08 - accuracy: 1.0000
     Epoch 16/20
     1250/1250 [==============================] - 44s 35ms/step - loss: 3.2802e-08 - accuracy: 1.0000
     Epoch 17/20
     1250/1250 [==============================] - 41s 33ms/step - loss: 1.8273e-08 - accuracy: 1.0000
     Epoch 18/20
     1250/1250 [==============================] - 41s 33ms/step - loss: 1.0401e-08 - accuracy: 1.0000
     Epoch 19/20
     1250/1250 [==============================] - 41s 33ms/step - loss: 6.1321e-09 - accuracy: 1.0000
     Epoch 20/20
     1250/1250 [==============================] - 42s 33ms/step - loss: 3.7190e-09 - accuracy: 1.0000
     <keras.callbacks.History at 0x7fa54fd3bc90>

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

     ypred=model.predict(x_test)

     313/313 [==============================] - 4s 11ms/step

[ ] ypred
```

```
     313/313 [==============================] - 4s 11ms/step

[46] ypred

     array([[0.0000000e+00],
            [2.6360015e-26],
            [2.3840201e-31],
            ...,
            [0.0000000e+00],
            [0.0000000e+00],
            [6.5630064e-11]], dtype=float32)

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

     ['amazo.save']

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

     d="Writing was good"
     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")
```

```
[49]    print("Negative review")

1/1 [==============================] - 0s 129ms/step
[[2.6688493e-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 countvecorizer
     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.save'))
```

```
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")
```

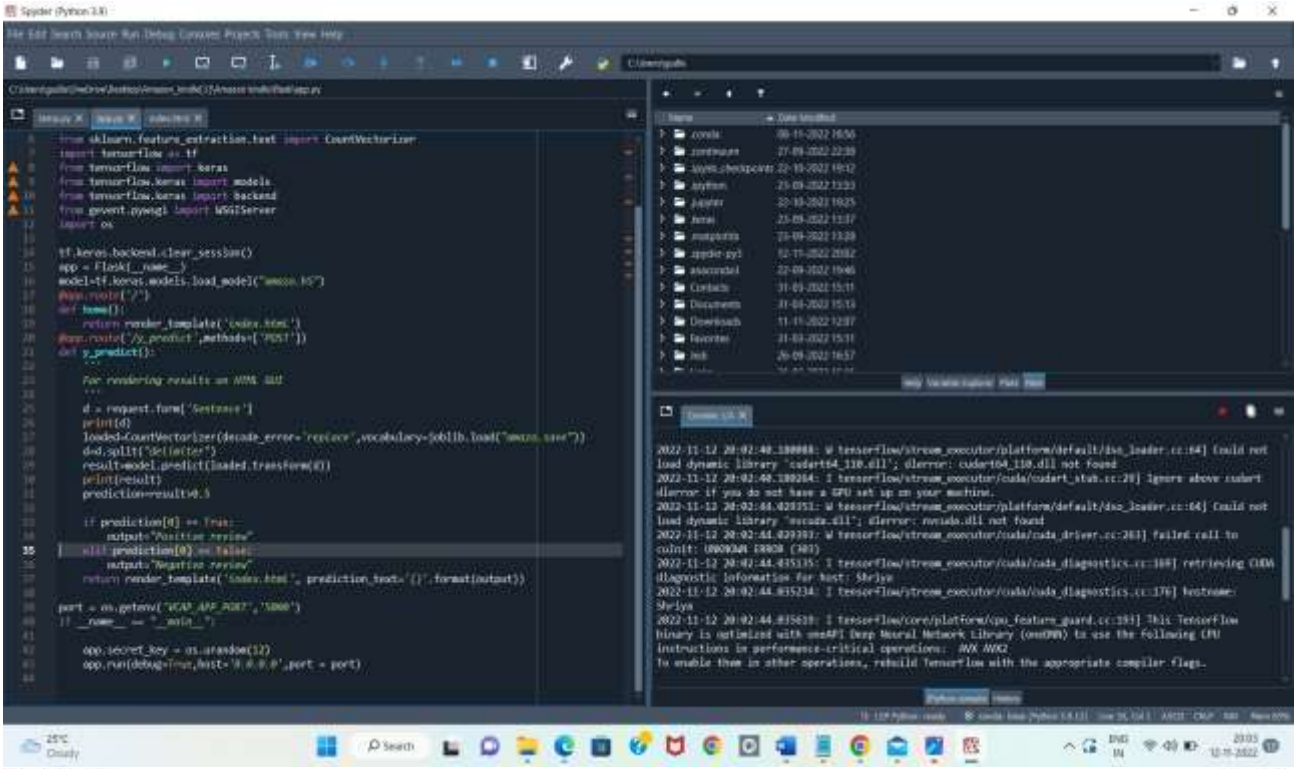

```
[[2.6688493e-06]]
Positive review

[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 countvecorizer
     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.save'))

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")
```

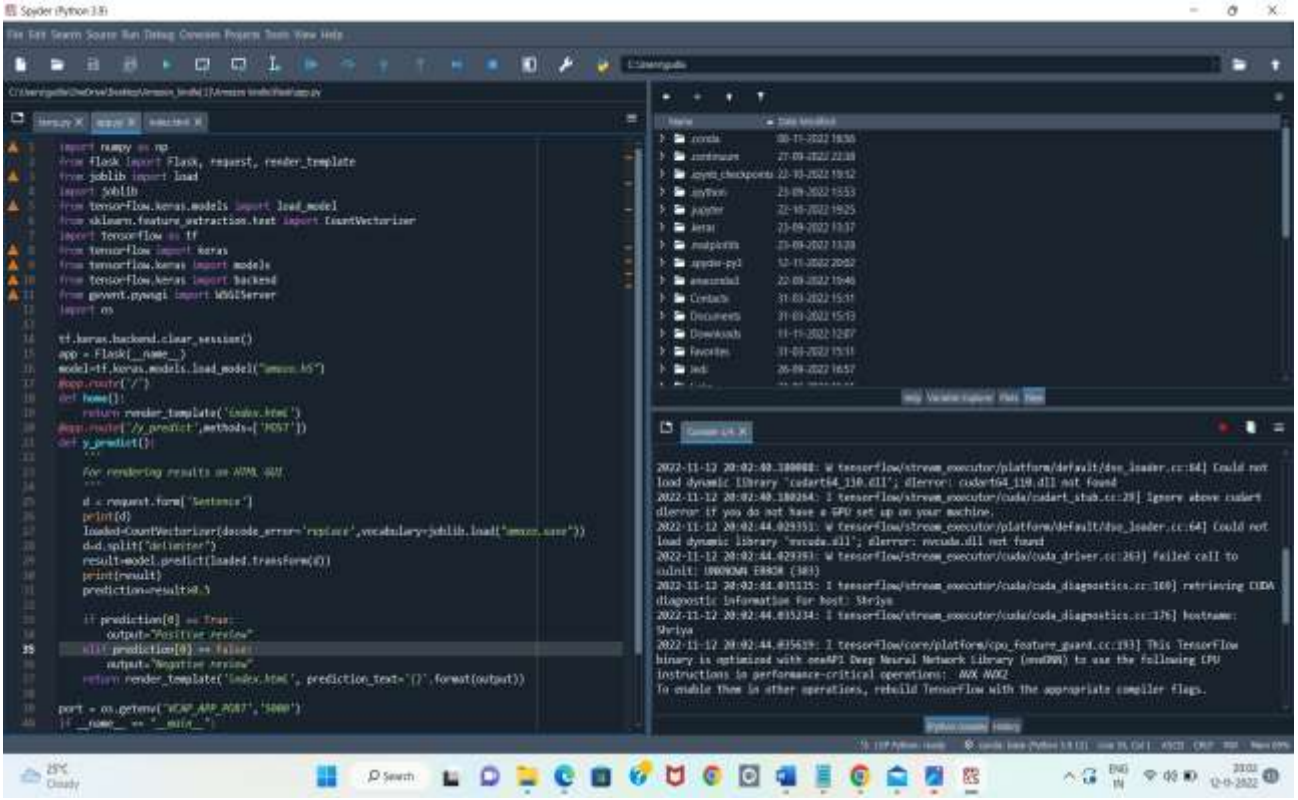

```
1/1 [==============================] - 0s 99ms/step
[[5.8496785e-10]]
Positive review
```

APPLICATION BUILDING

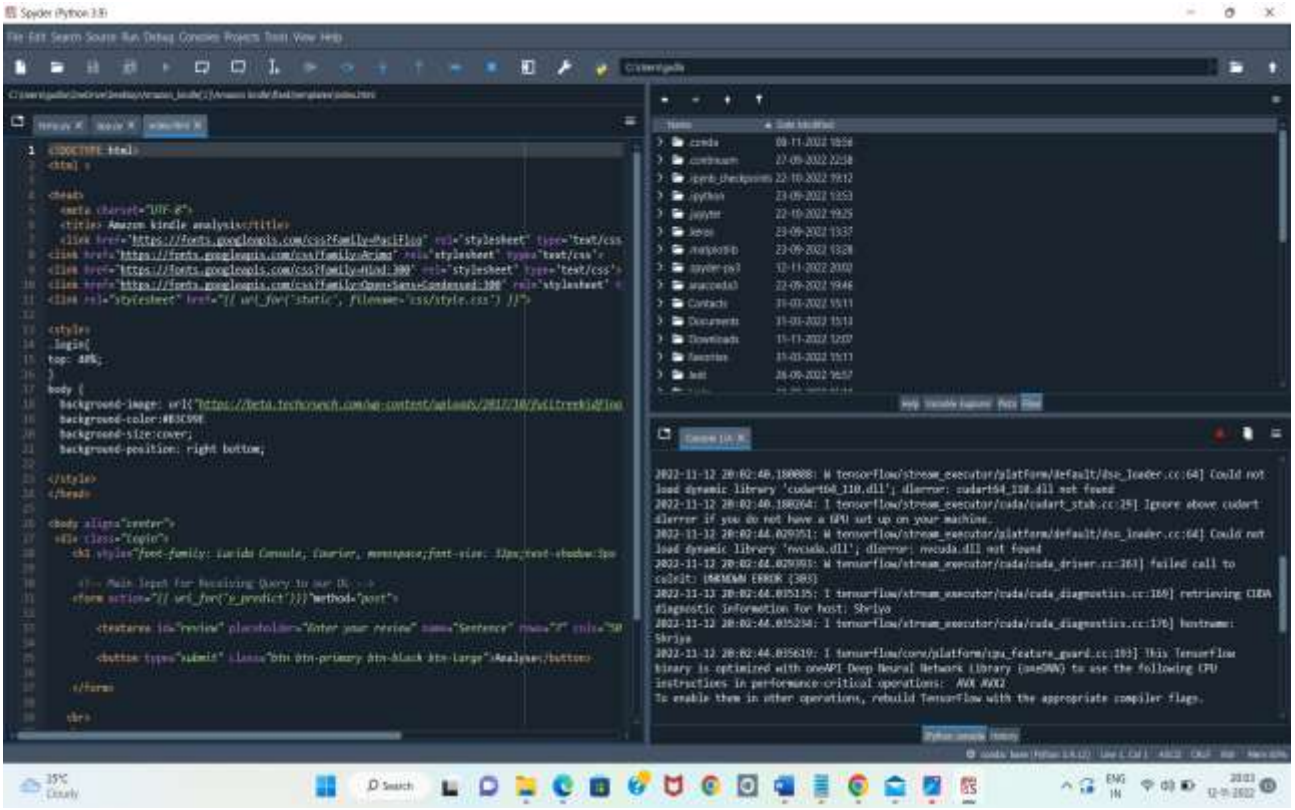
HTML CODE AND PYTHON CODE

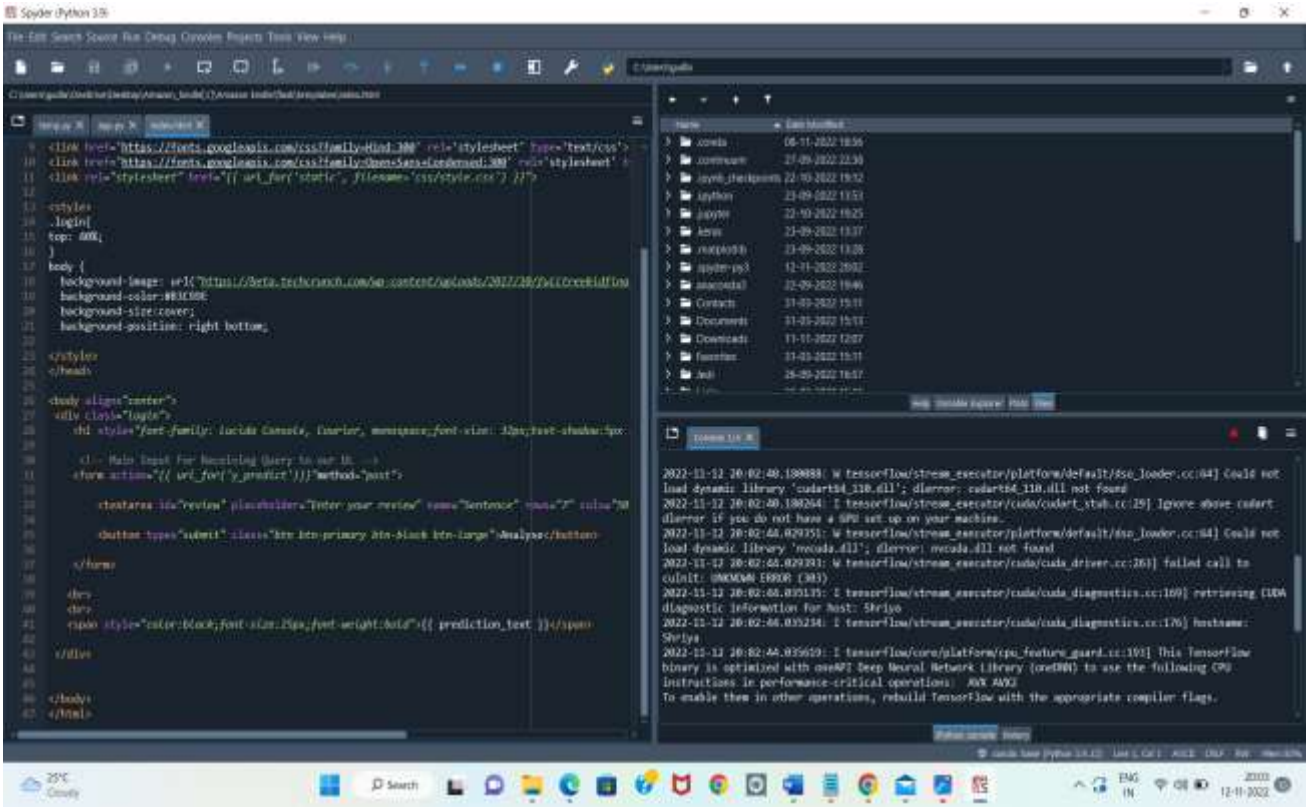
1.App.py code[FLASK]





2.Html code





10.HELP FILE

PROJECT EXECUTION:

STEP-1: Go to **Start,** search and launch **ANACONDA NAVIGATOR.**

STEP-2: After launching of **ANACONDA NAVIGATOR,** launch **JUPYTER NOTEBOOK.**

STEP-3: Open **"Major project code" IPYNB file.**

STEP-4: Then run all the cells.

STEP-5: All the **data preprocessing**, **training and testing**, **model building**, **accuracy** of the model can be showcased.

STEP-6: And a pickle file will be generated.

STEP-7: Create a Folder named **FLASK** on the **DESKTOP.** Extract the pickle file into this Flask Folder.

STEP-8: Extract all the html files (home.html, index.html, chance.html, nochance.html) and python file(app.py) into the **FLASK Folder.**

STEP-9: Then go back to **ANACONDA NAVIGATOR** and the launch the **SPYDER.**

STEP-10: After launching Spyder, give the path of **FLASK FOLDER** which you have created on the DESKTOP.

STEP-11: Open all the app.py and html files present in the Flask Folder.

STEP-12: After running of the app.py, open **ANACONDA PROMPT** and follow the below steps:

cd File Path□click enter python app.py□click enter (We could see running of files).

STEP-13: Then open **BROWSER,** at the URL area type **—localhost:5000".**

STEP-14: Home page of the project will be displayed.

STEP-15: Click on **—Go to Predict".** Directly it will be navigated to index page.

STEP-16:A index page will be displayed where the user needs to give the inputs and then click on **—Predict"**. Output will be generated whether a person is having liver disease or not.