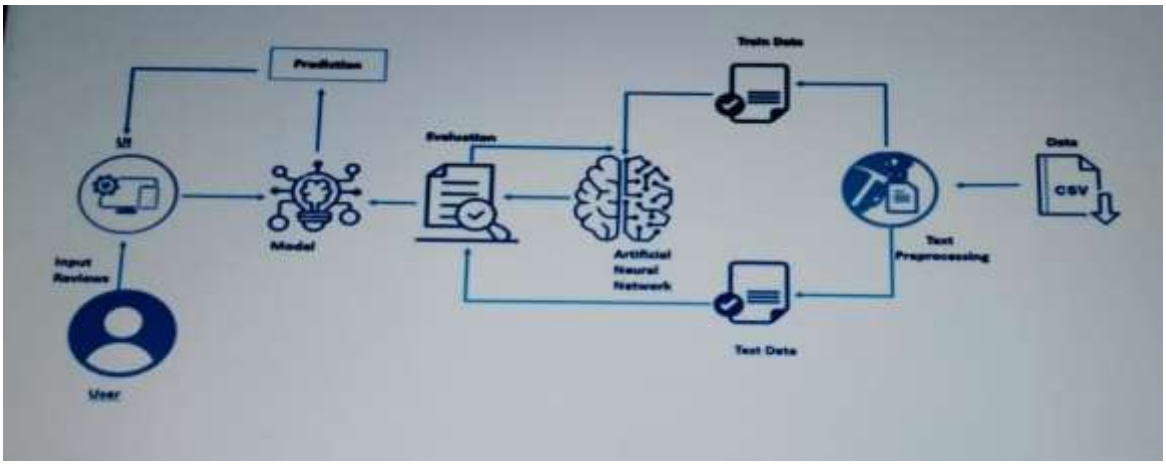
AMAZON KINDLE STORE REVIEWS ANALYSIS USING IBM WATSON SERVICES

PROJECT OBJECTIVE

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

BLOCK DIAGRAM



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

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

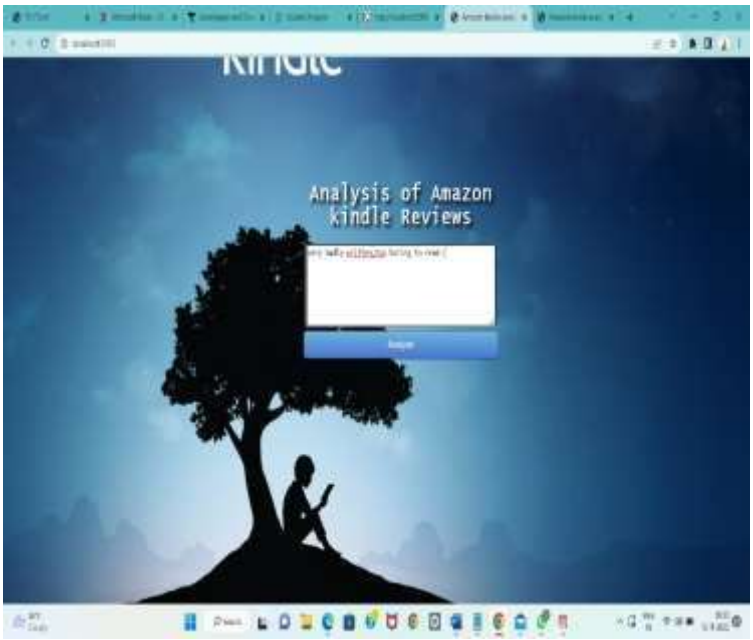
Import the model building Libraries
Initializing the model
Adding Input Layer
Adding Hidden Layer
Adding Output Layer
Configure the Learning Process
Training and testing the model
Optimize the Model o Save the Model

• Application Building

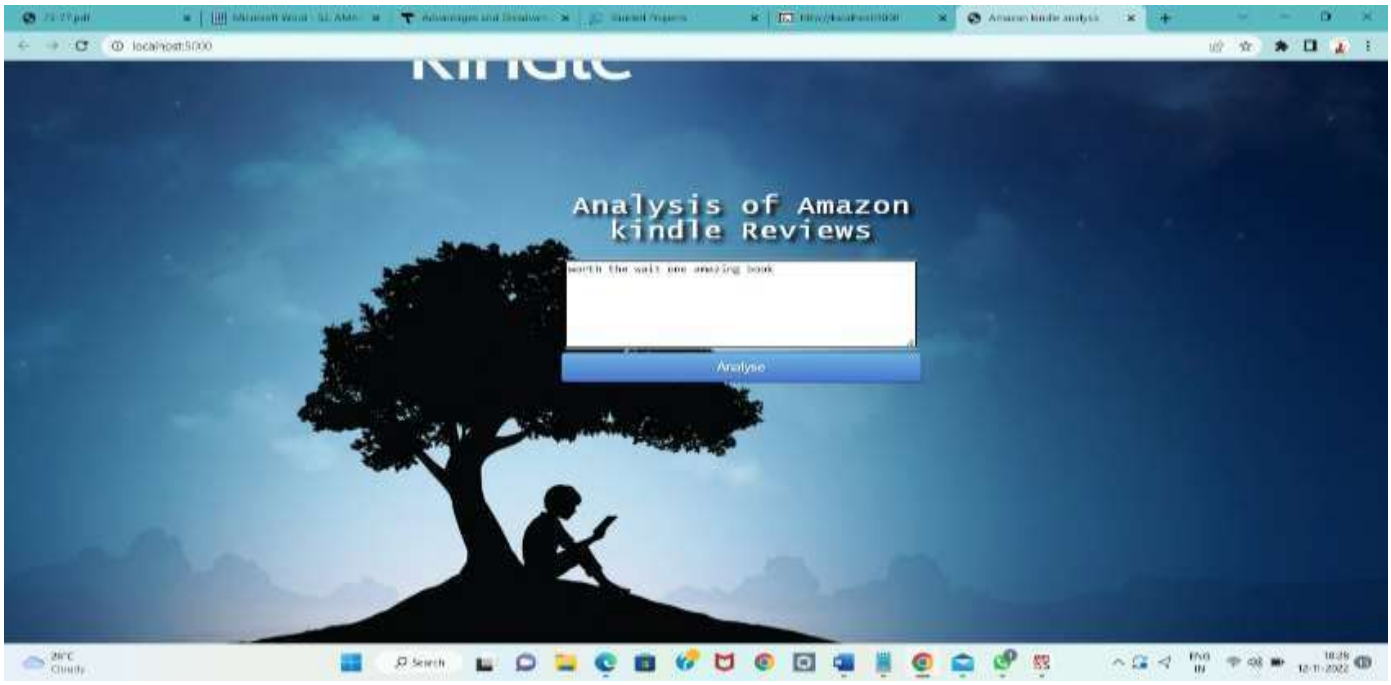
Create an HTML file
Build Python Code

6.RESULT

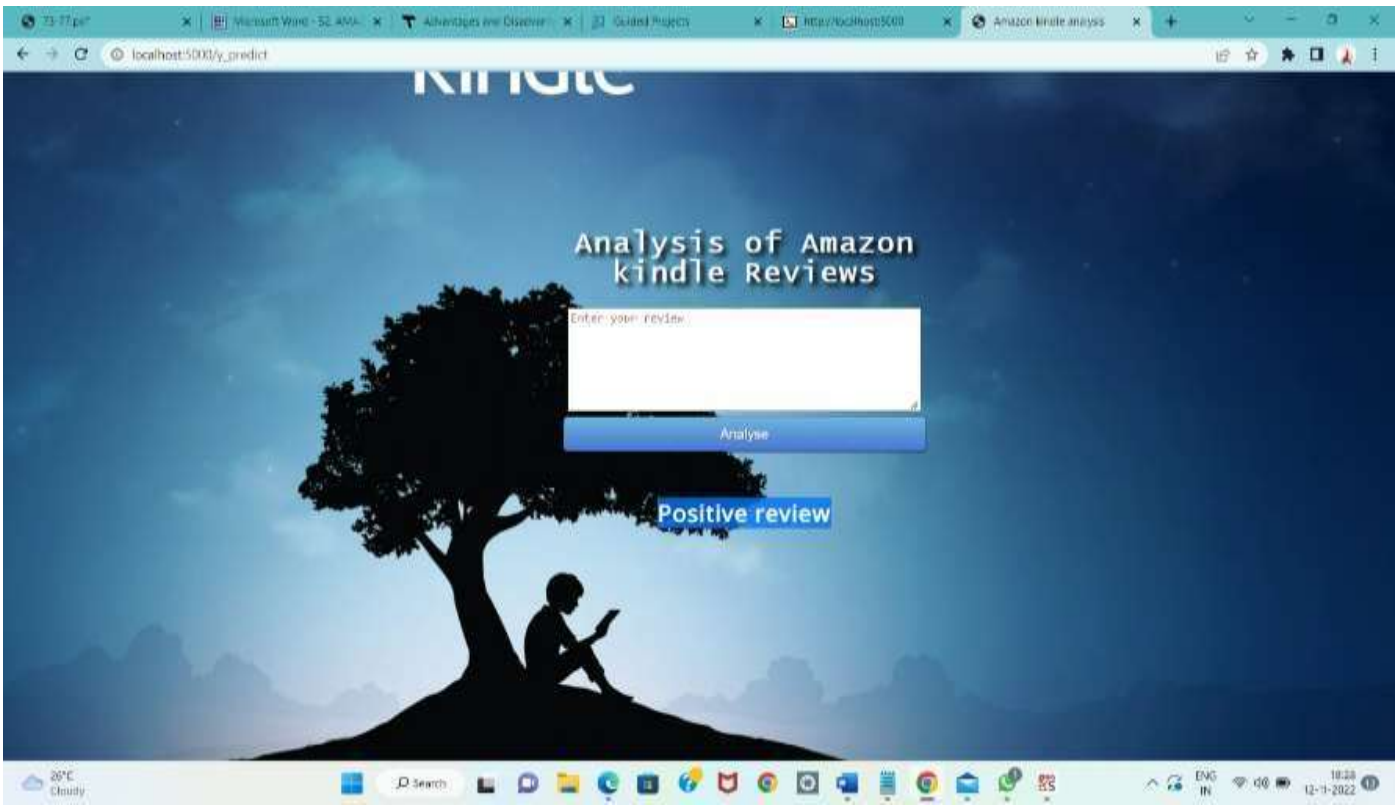
HOMEPAGE



REVIEW ANALYSIS PAGE **OUTPUT**



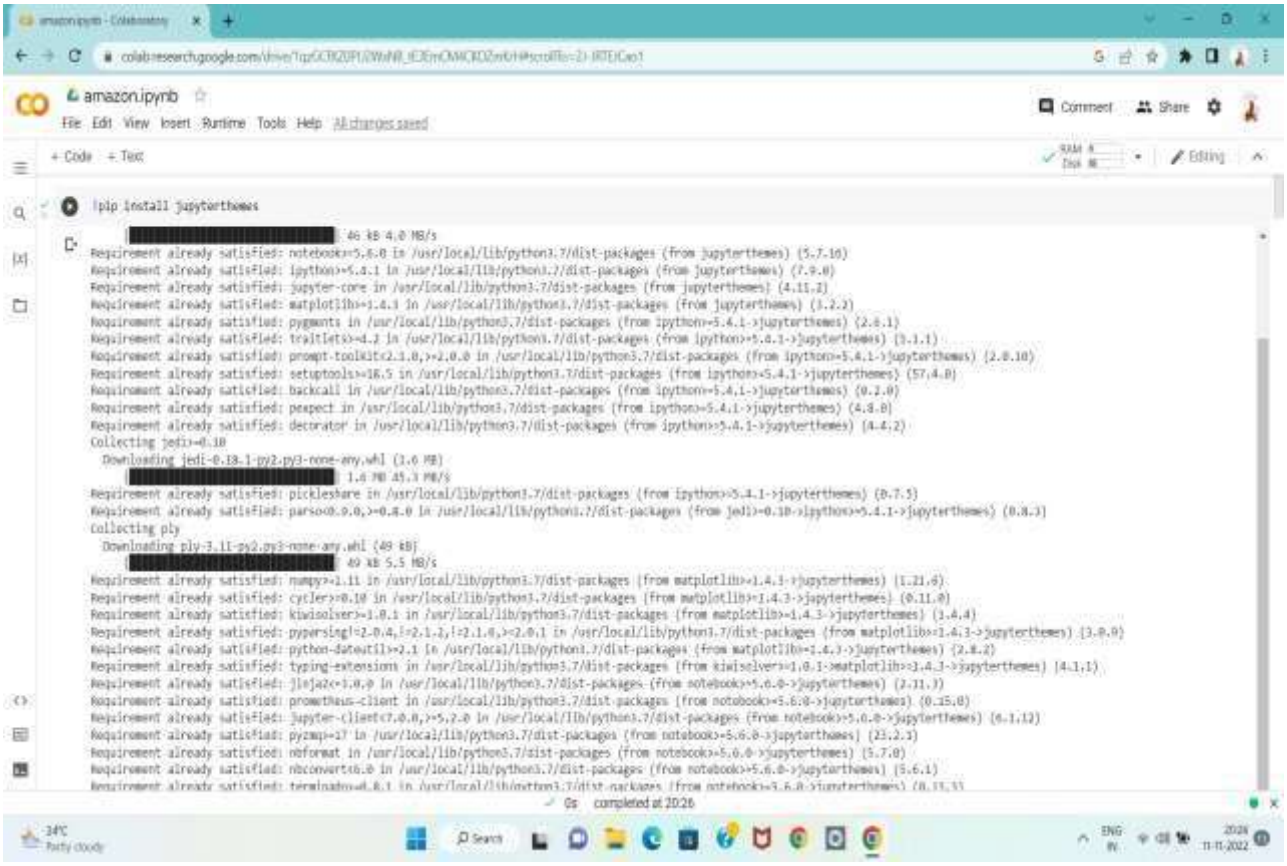
REVIEW ANALYSIS

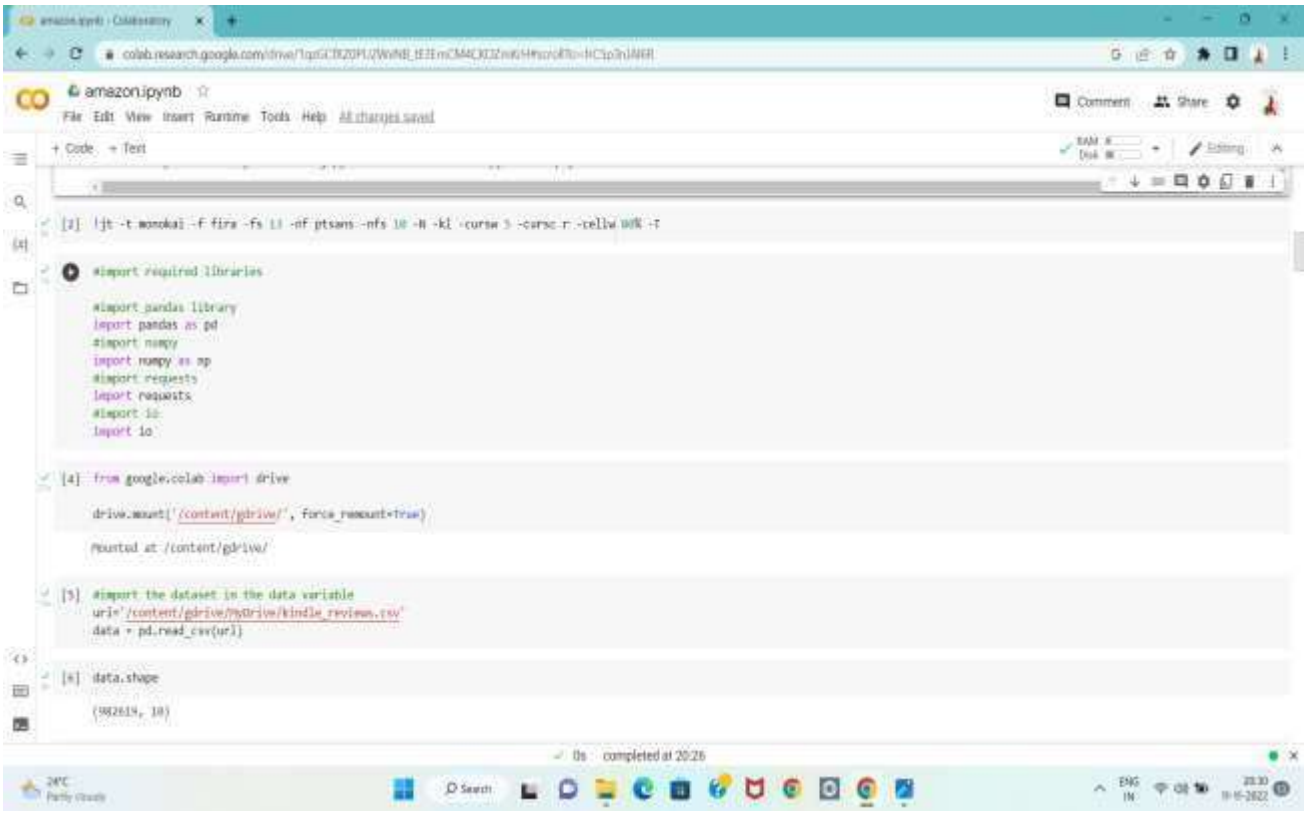


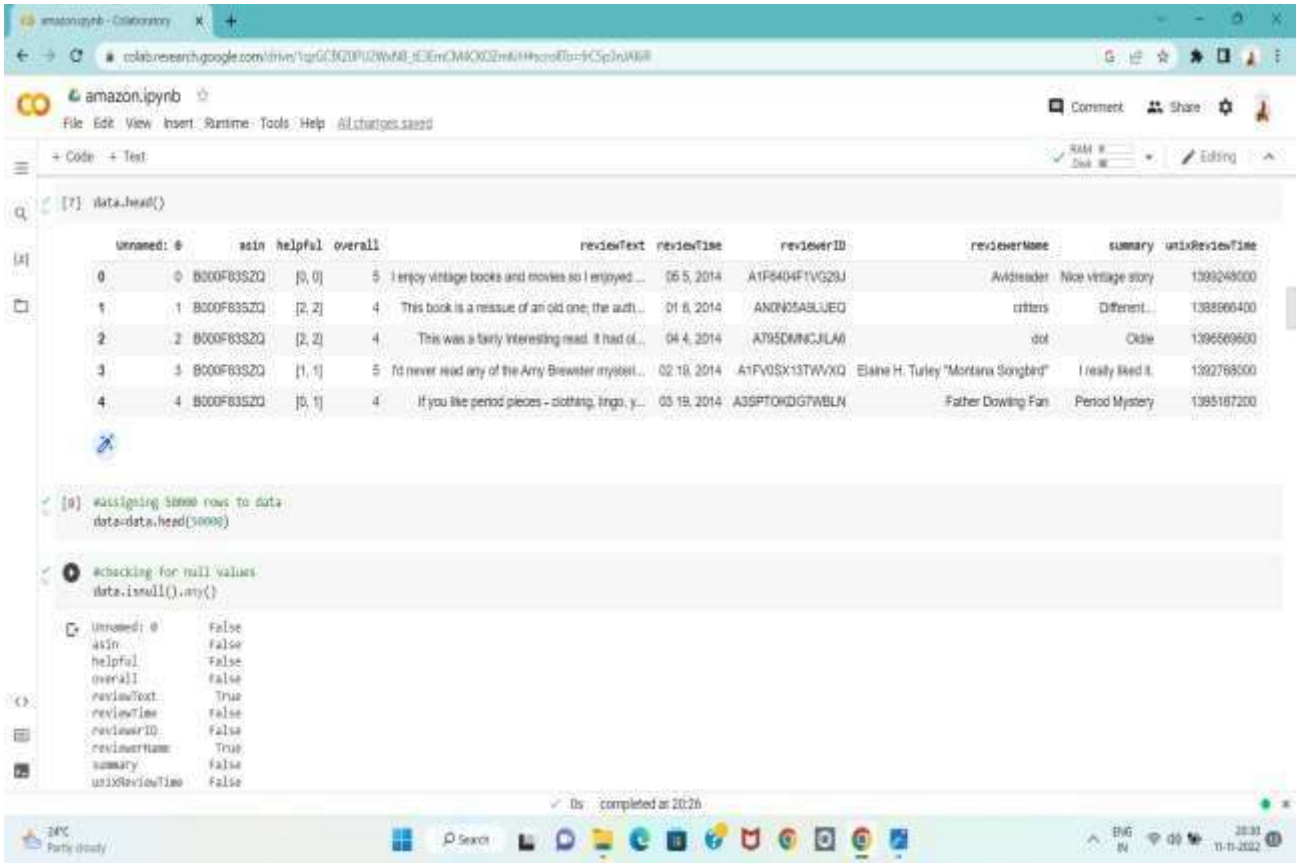
OUTPUT

9.CODE SNIPPETS

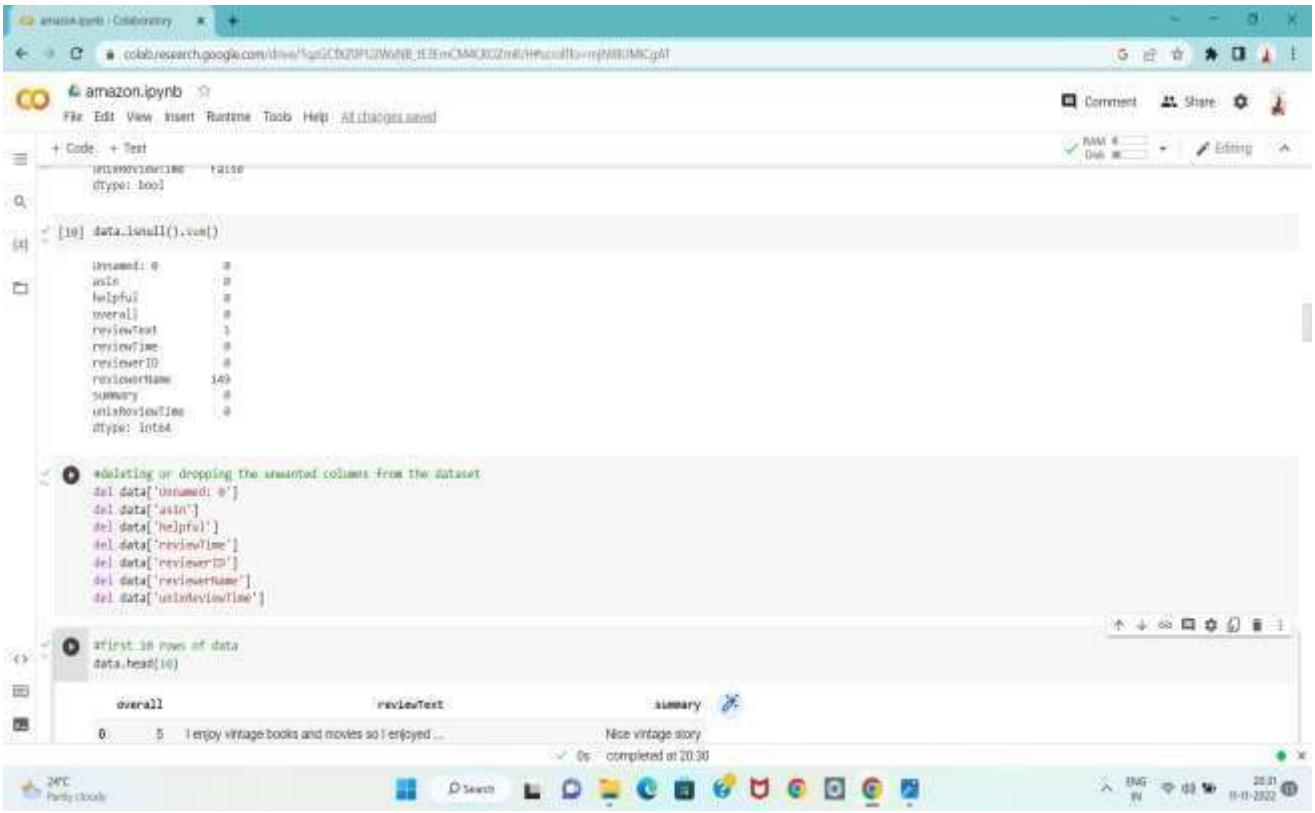
A.MODELBUILDING

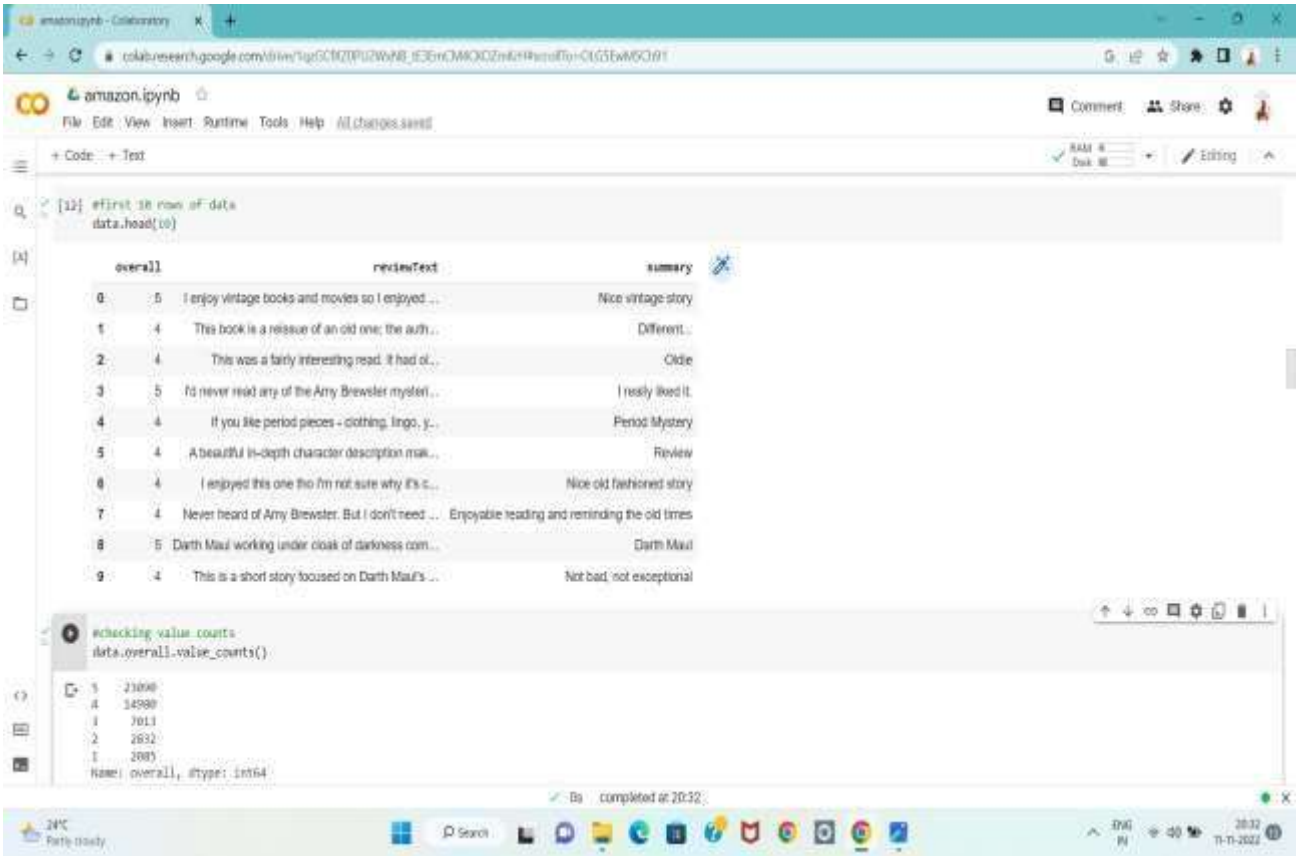


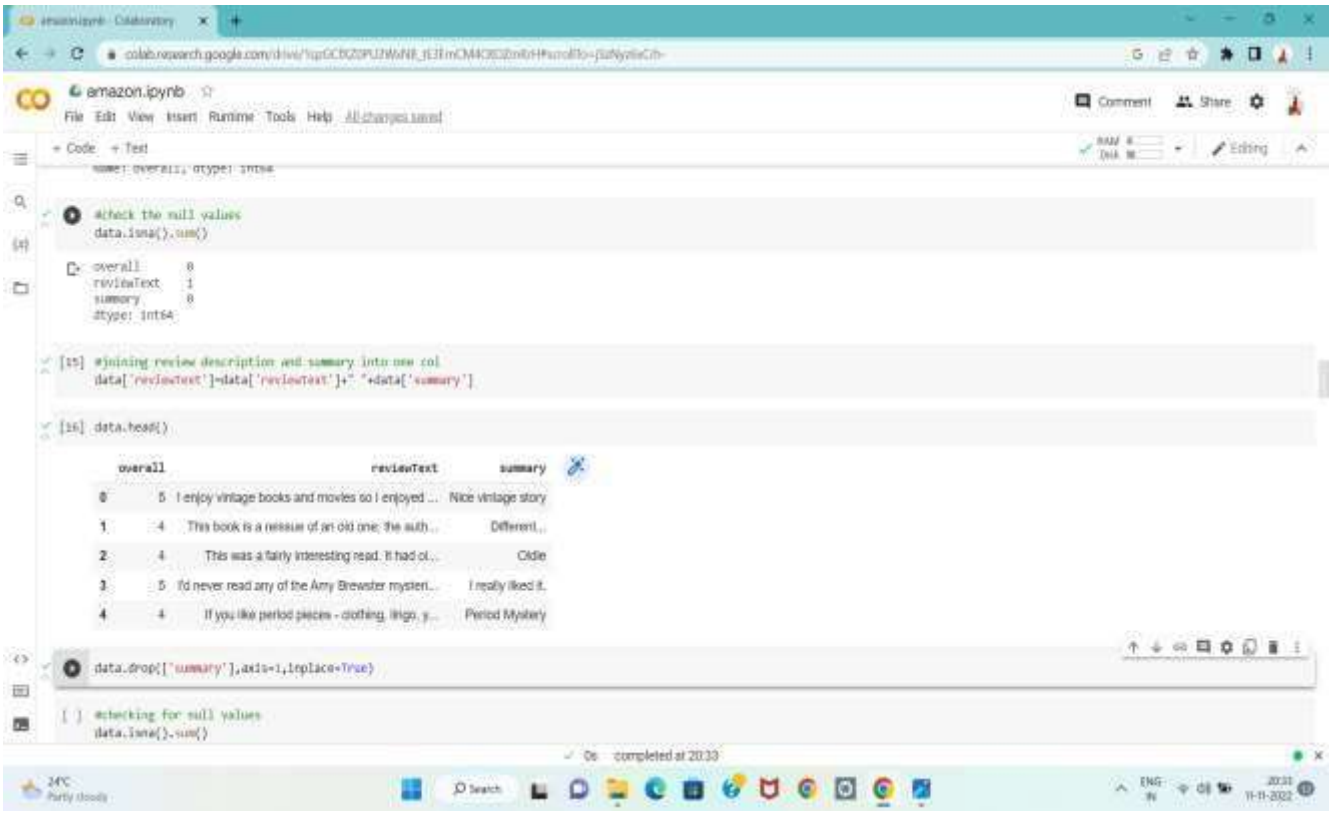


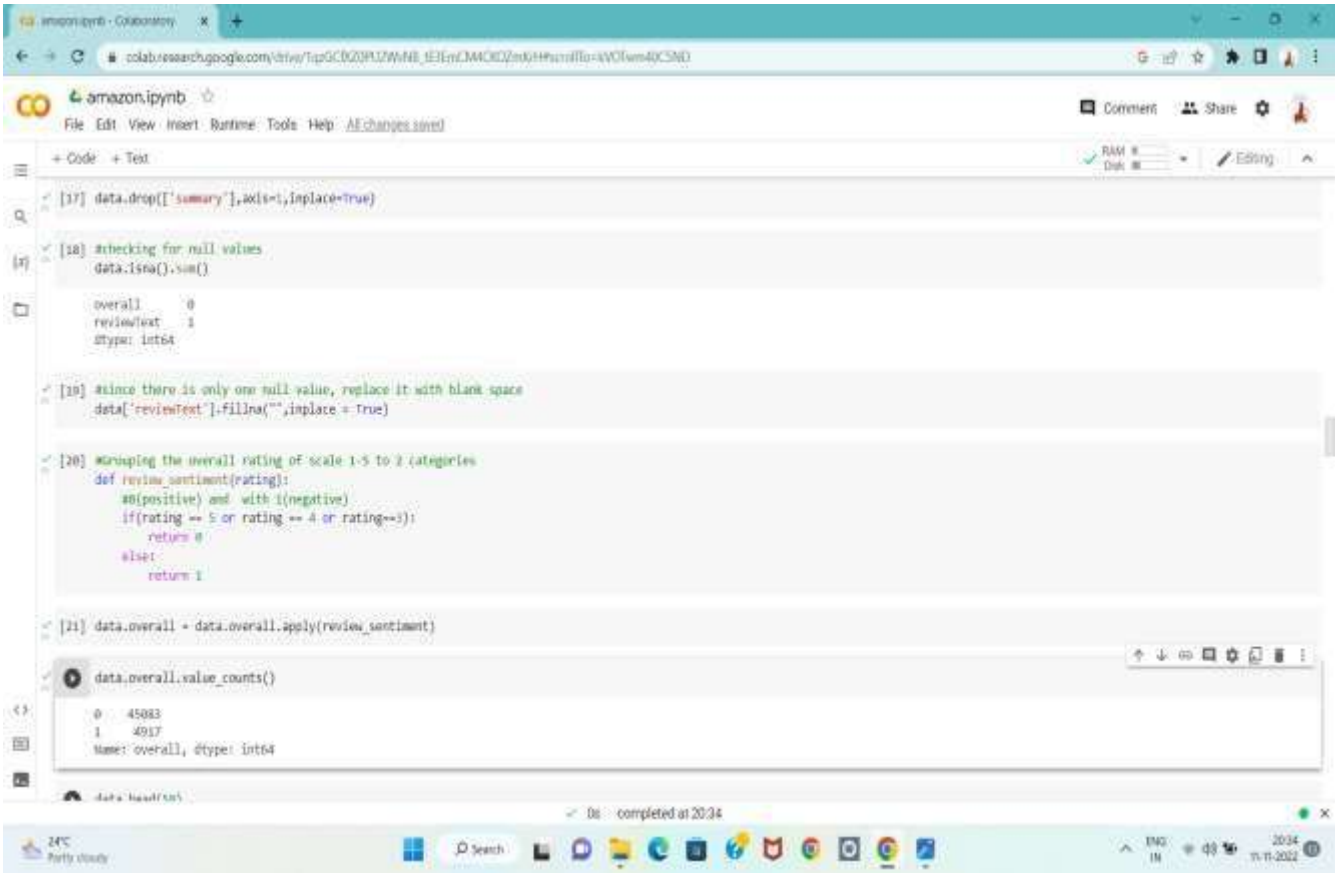


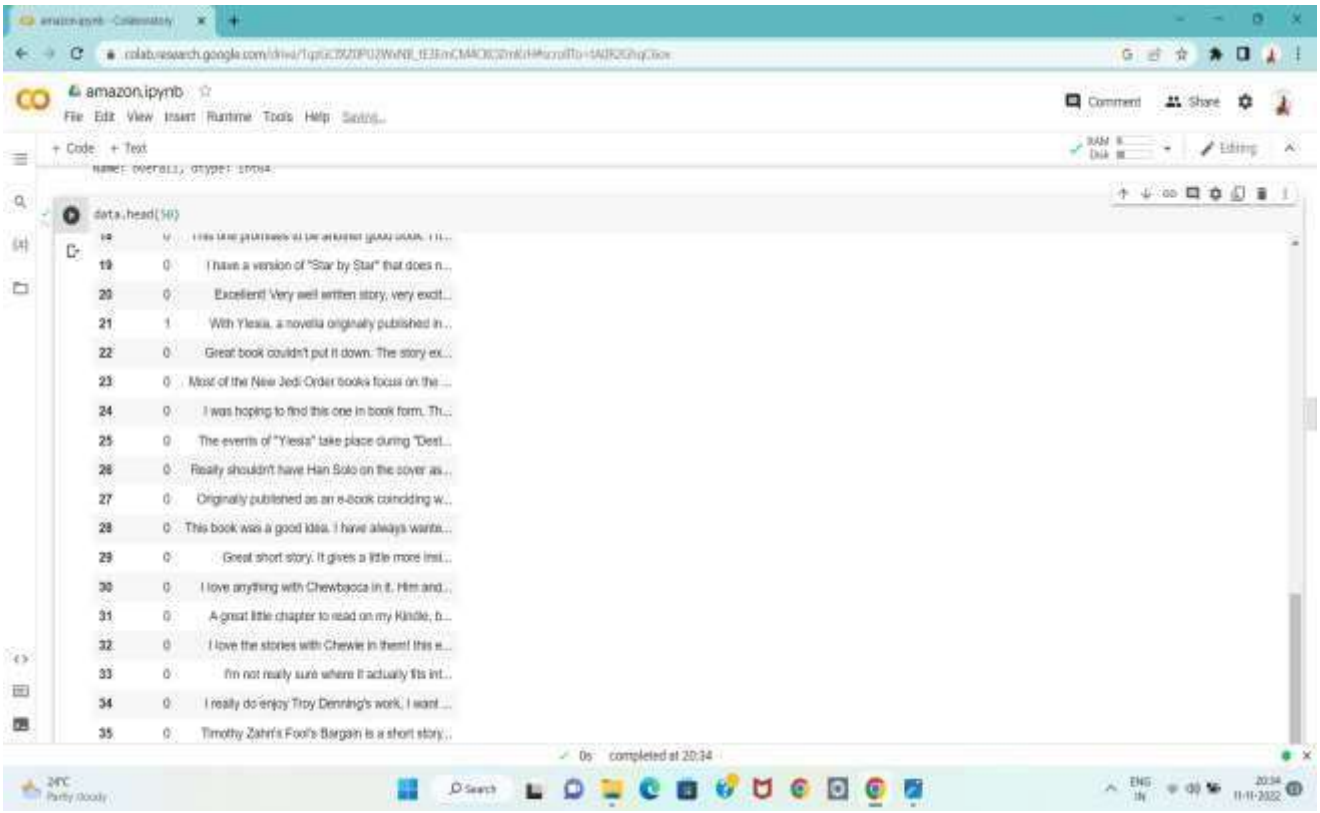
	Unnamed: 0	asin	helpful	overall	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime
0	0	B000F83SZQ	[0, 0]	5	I enjoy vintage books and movies so I enjoyed ...	05 5, 2014	A1F6404F1VG29J	Avidreader	Nice vintage story	1399248000
1	1	B000F83SZQ	[2, 2]	4	This book is a reissue of an old one; the auth...	01 6, 2014	ANDN05ABLJJEQ	critters	Different...	1388966400
2	2	B000F83SZQ	[2, 2]	4	This was a fairly interesting read. It had o...	04 4, 2014	A795DMNCJILA6	dot	Oldie	1396569600
3	3	B000F83SZQ	[1, 1]	5	I'd never read any of the Amy Brewster myster...	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

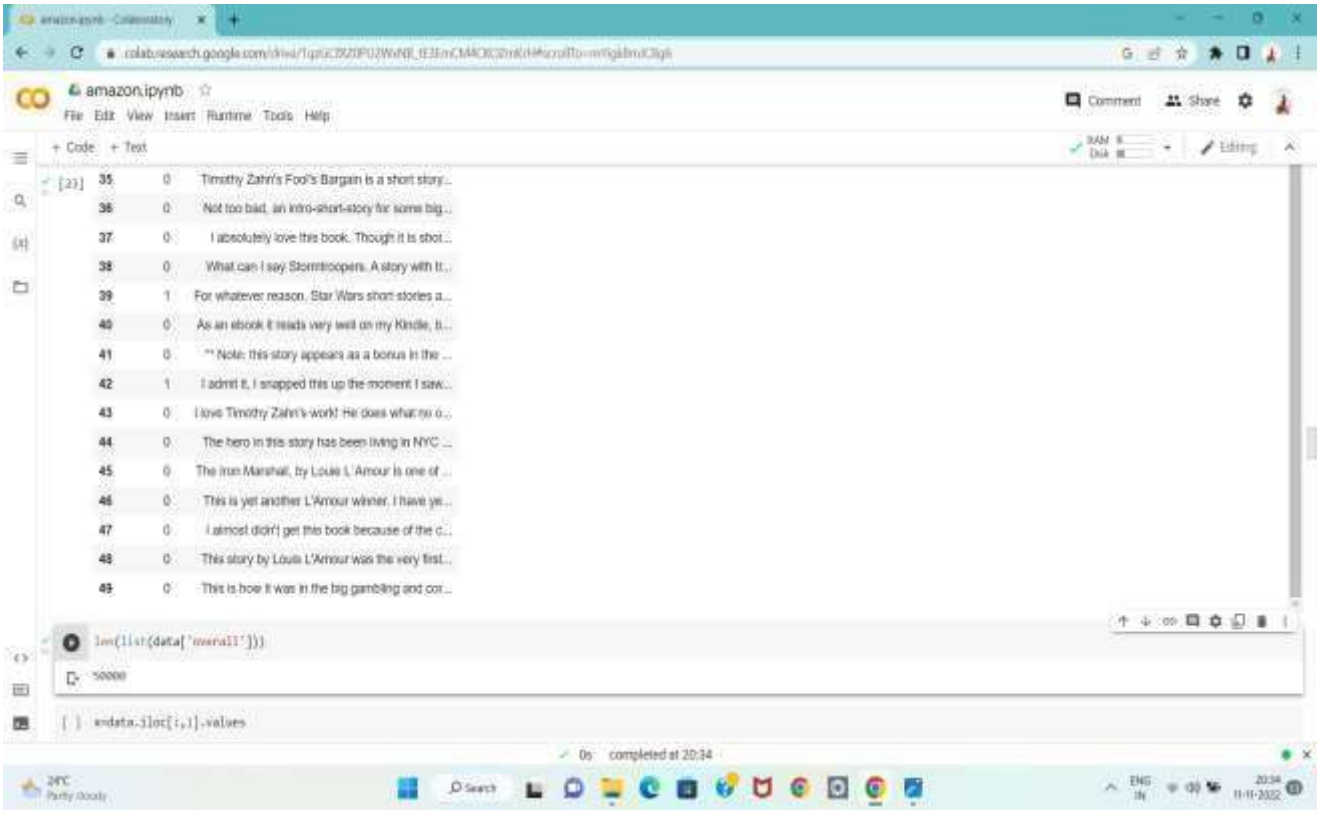

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
dtype: bool
[10] data.isnull().sum()
unixReviewTime 0
dtype: int64
#deleting or dropping the unwanted columns from the dataset
del data['Unnamed: 0']
del data['asin']
del data['helpful']
del data['reviewTime']
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

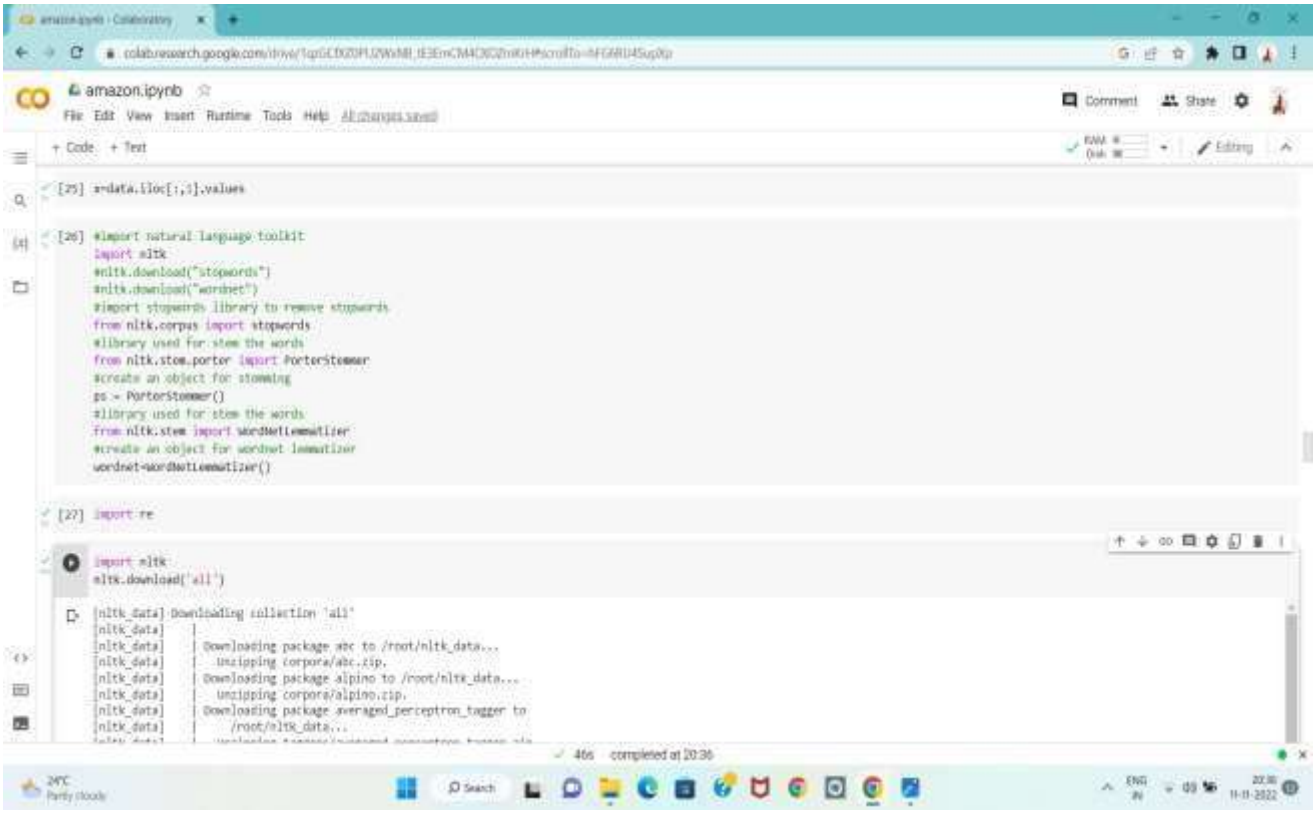

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
[12] #first 10 rows of data
data.head(10)
overall reviewText summary
0 5 I enjoy vintage books and movies so I enjoyed ... Nice vintage story
1 4 This book is a reissue of an old one; the auth... Different...
2 4 This was a fairly interesting read. It had ol... Oldie
3 5 I'd never read any of the Amy Brewster myster... 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 21000
4 14980
1 7013
2 2032
3 2005
Name: overall, dtype: int64
completed at 20:32

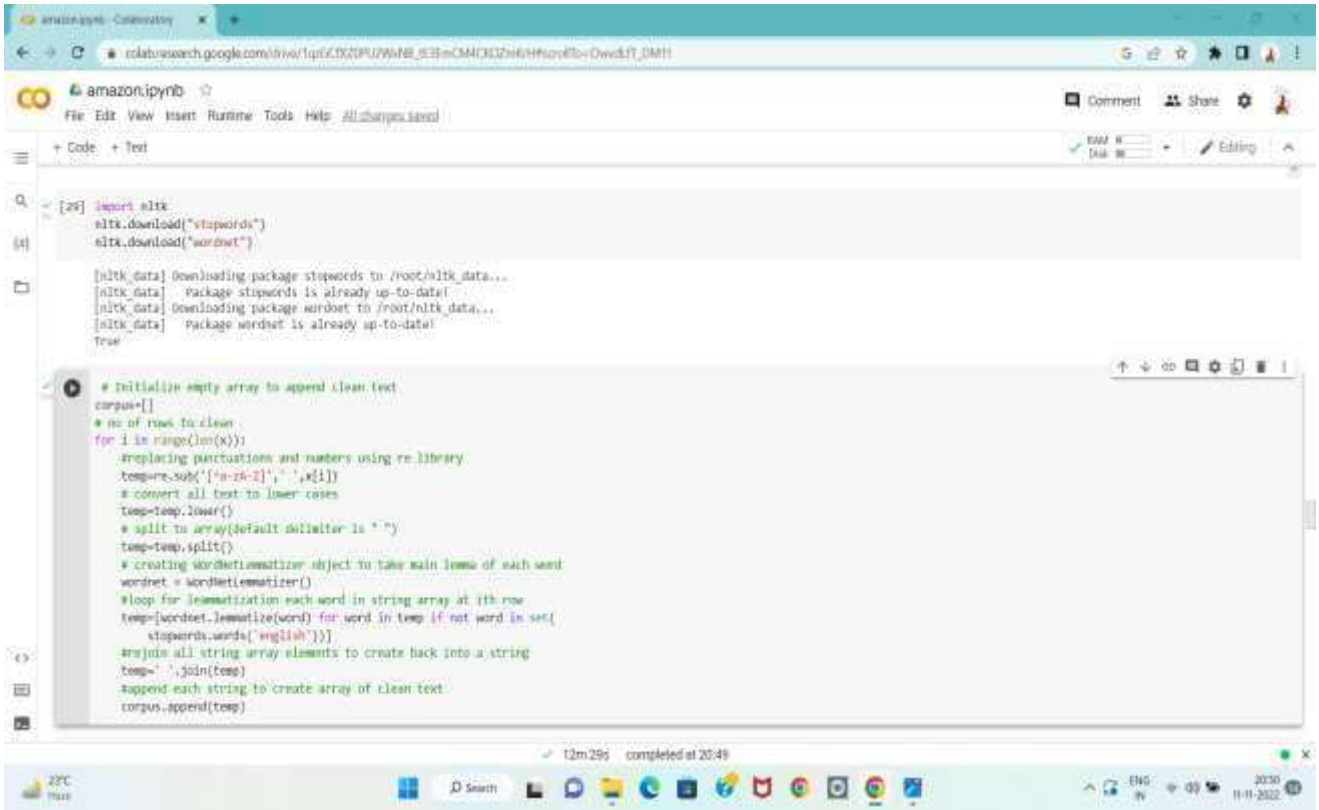

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
#check the null values
data.isna().sum()
overall 0
reviewText 1
summary 0
dtype: int64
[15] #joining review description and summary into one col
data['reviewText']=data['reviewText']+" "+data['summary']
[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()
0s completed at 20:33
24°C Partly cloudy
Search
ENG IN
20:33 11-11-2022

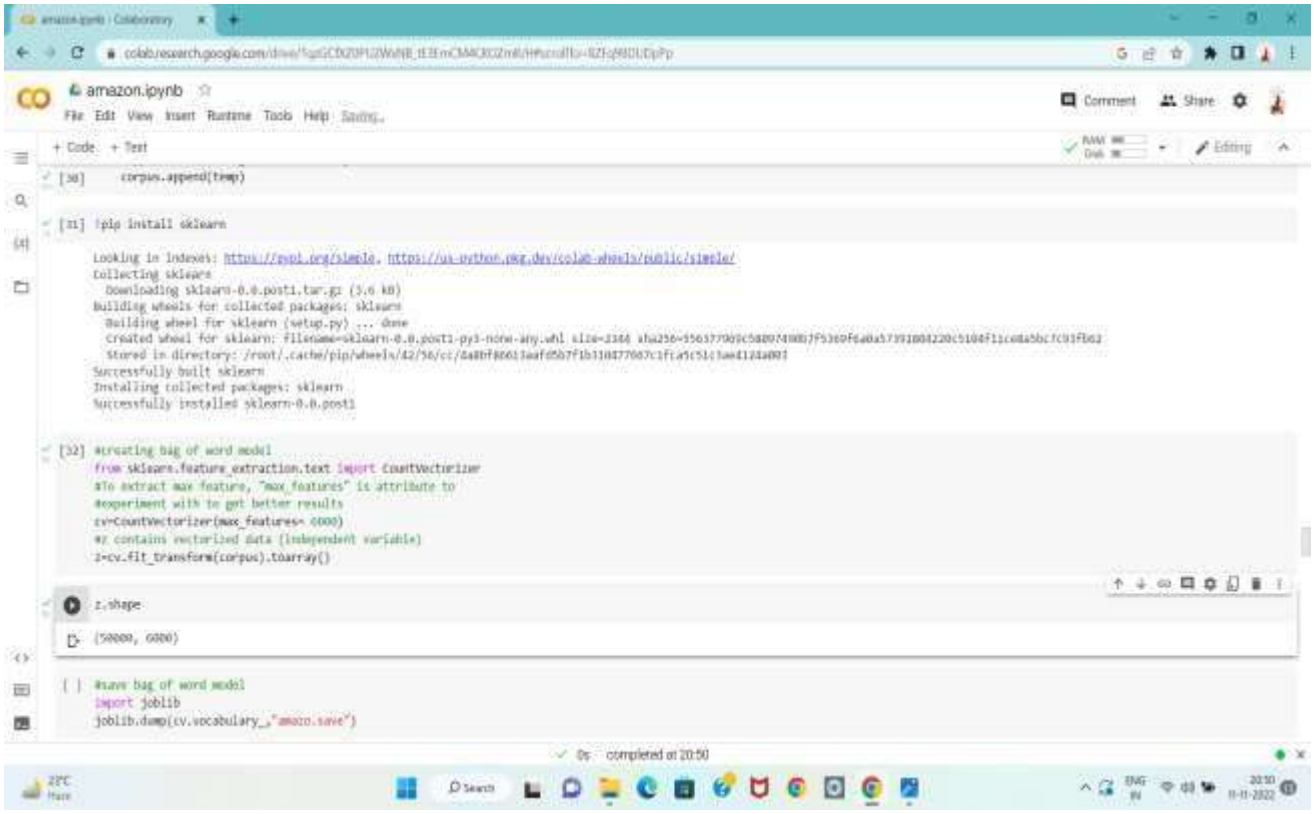

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
[17] data.drop(['summary'],axis=1,inplace=True)
[18] #checking for null values
data.isna().sum()
overall 0
reviewText 1
dtype: int64
[19] #since there is only one null value, replace it with blank 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
0s completed at 20:34
24°C Partly cloudy
Search
ENG IN
20:34 11-11-2022

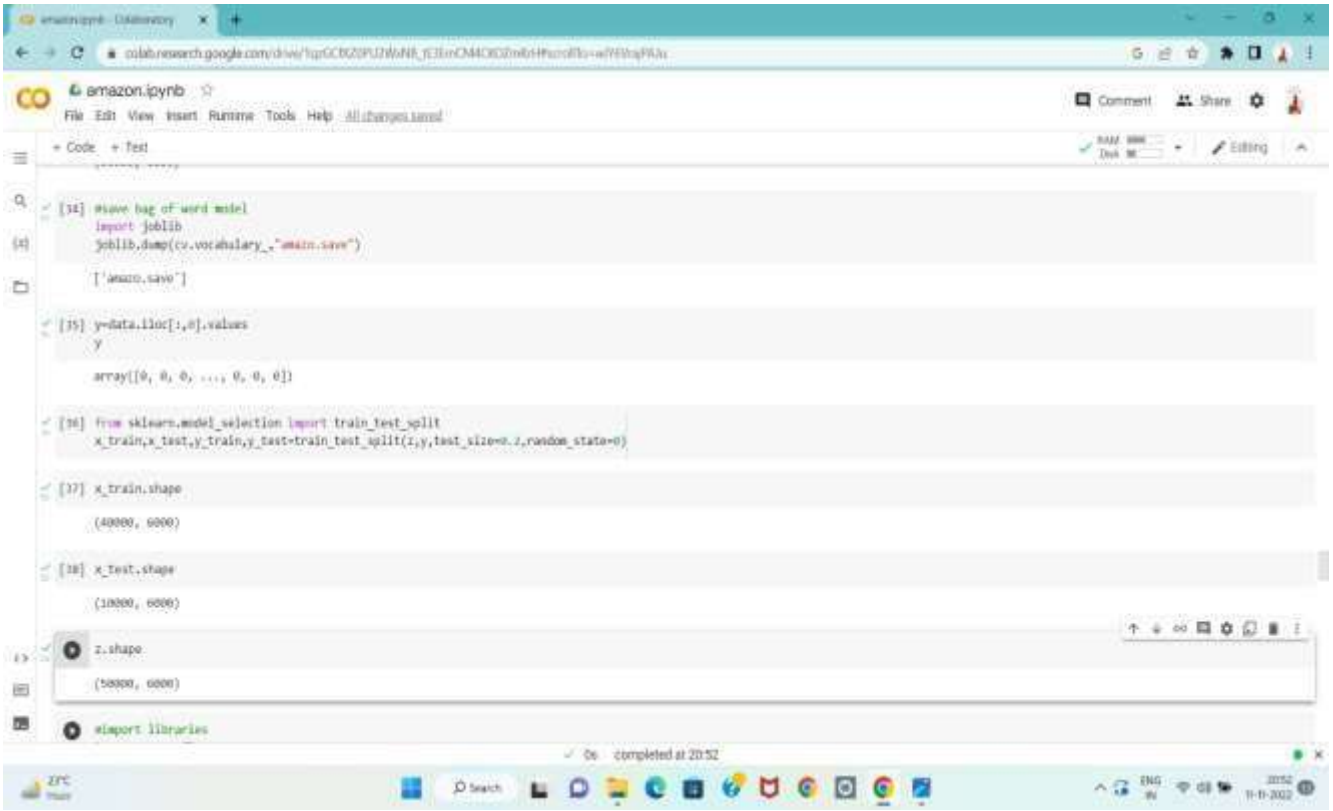

amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
data.head(50)
19 0 I have a version of "Star by Star" that does n...
20 0 Excellent! Very well written story, very excit...
21 1 With Yiesla, a novella originally published in...
22 0 Great book couldn't put it down. The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

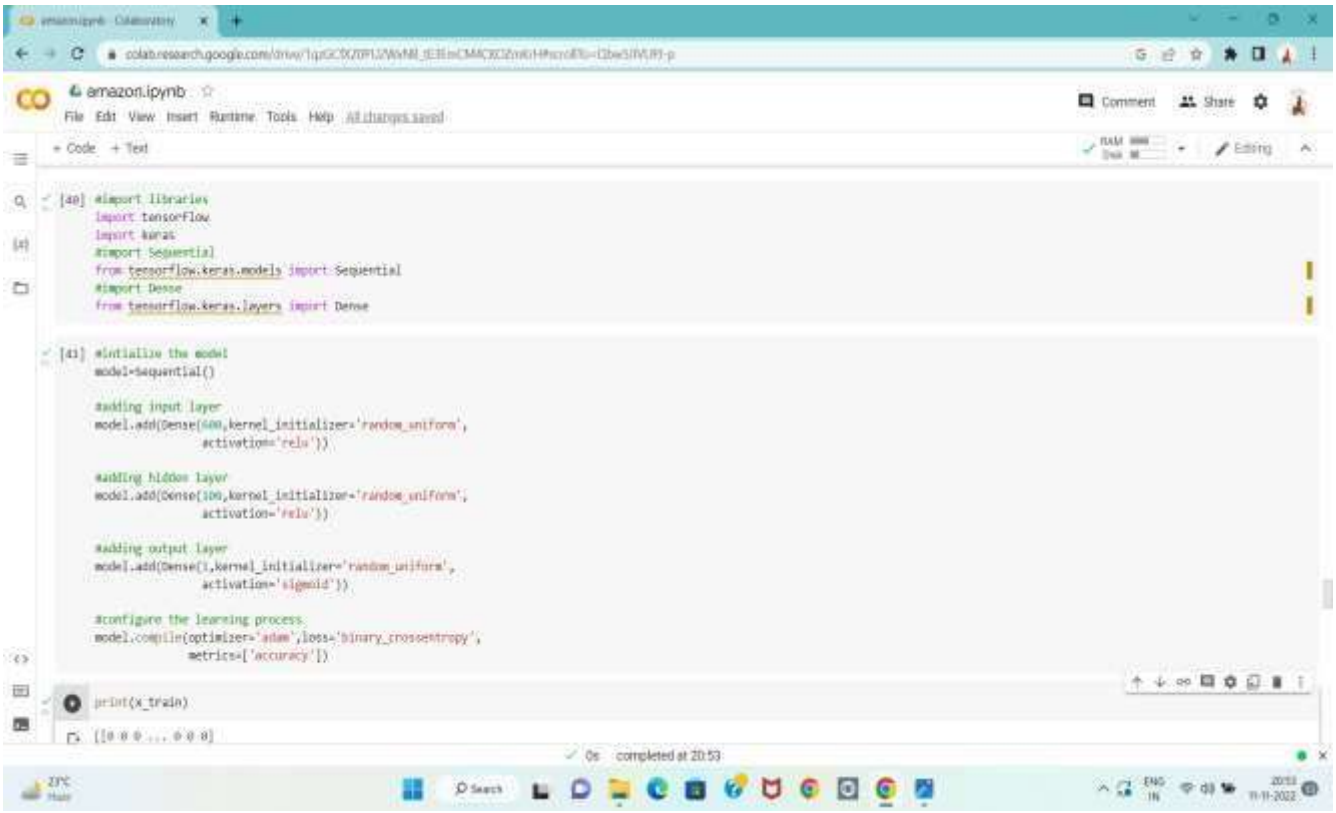

amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
[23] 35 0 Timothy Zahn's Fool's Bargain is a short story...
36 0 Not too bad, an intro-short-story for some big...
37 0 I absolutely love this book. Though it is shor...
38 0 What can I say Stormtroopers. A story with lo...
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 Marshal, 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
[ ] xvdata.iloc[:,1].values
0s completed at 20:34

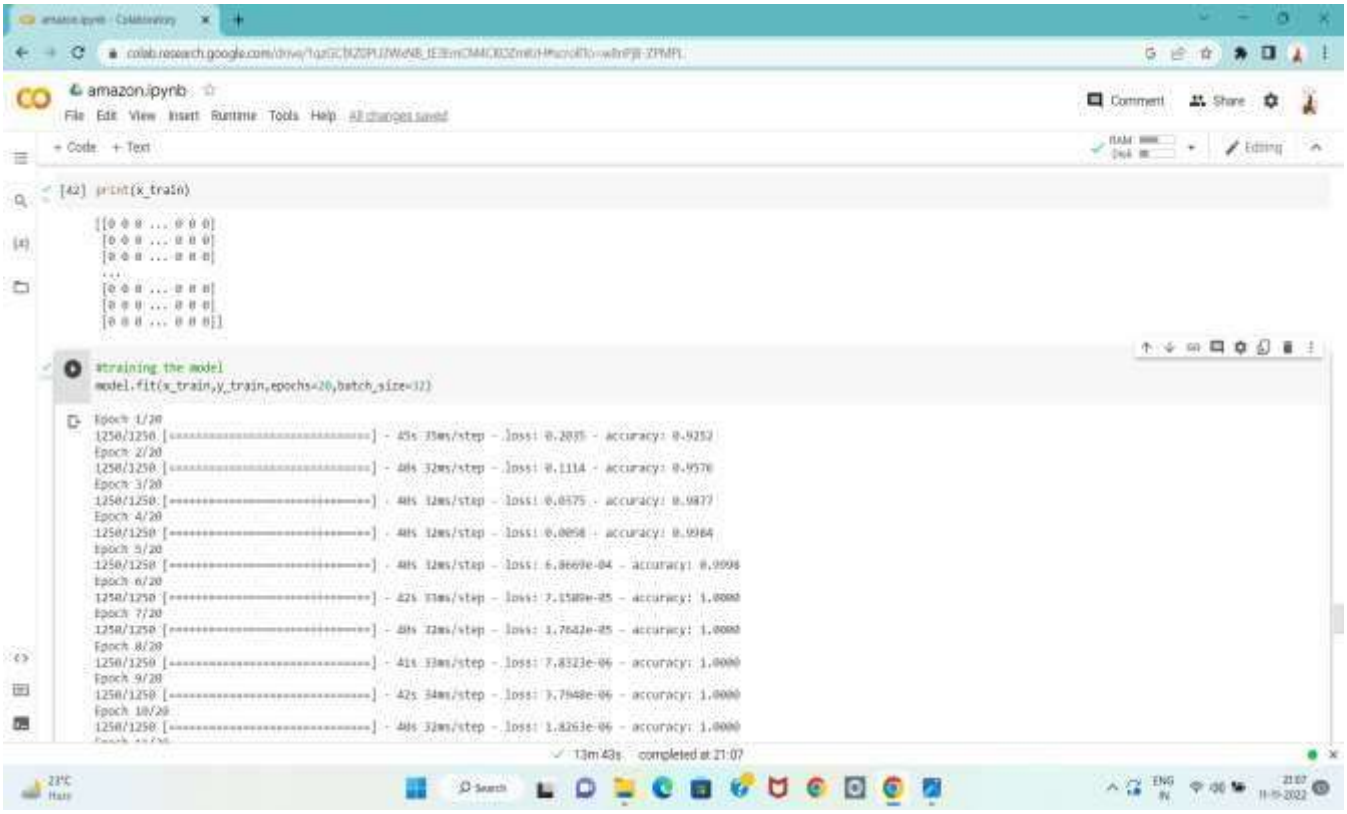

amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
[25] x=data.iloc[:,1].values
[26] #import natural language toolkit
import nltk
#nltk.download("stopwords")
#nltk.download("wordnet")
#import stopwords library to remove stopwords
from nltk.corpus import stopwords
#library 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] |
[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

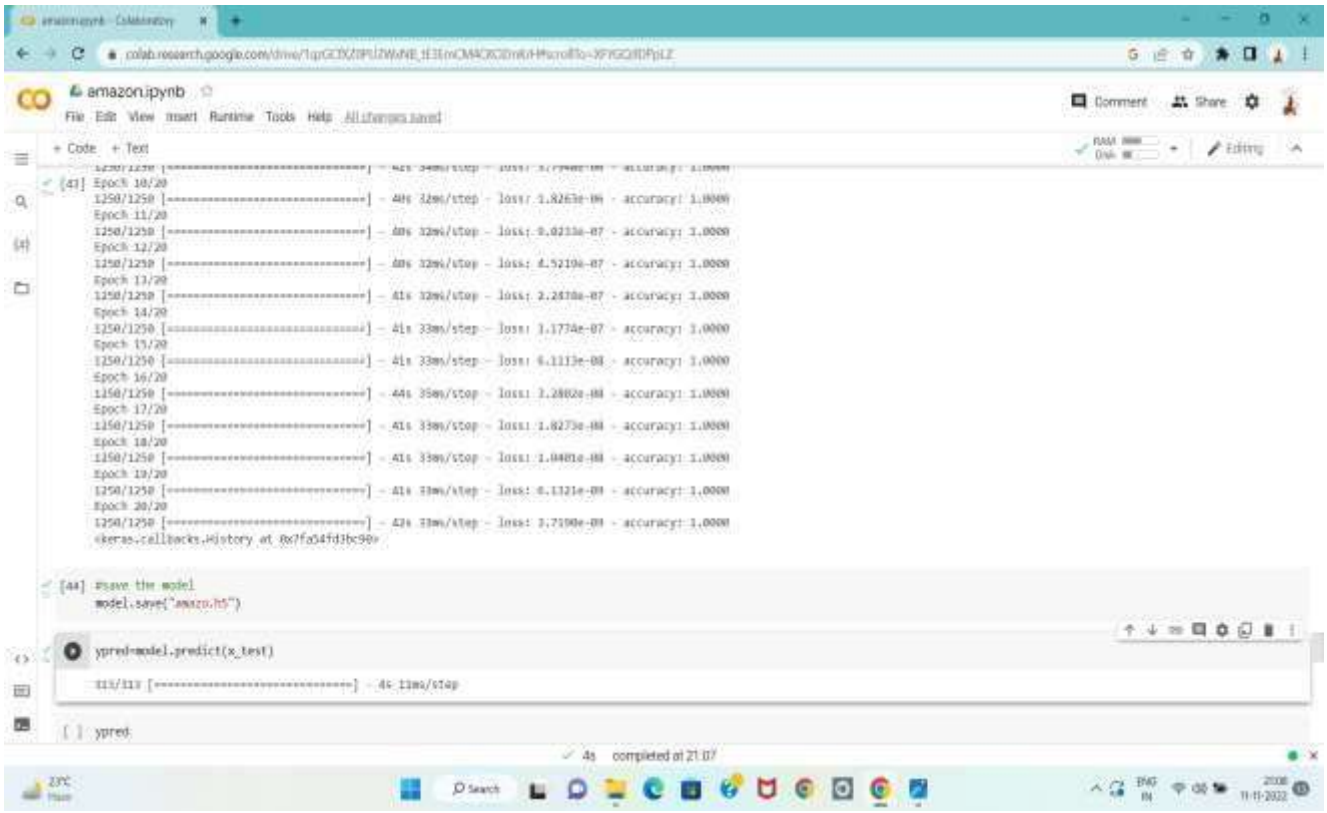

amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
[28] import nltk
nltk.download("stopwords")
nltk.download("wordnet")
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet 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

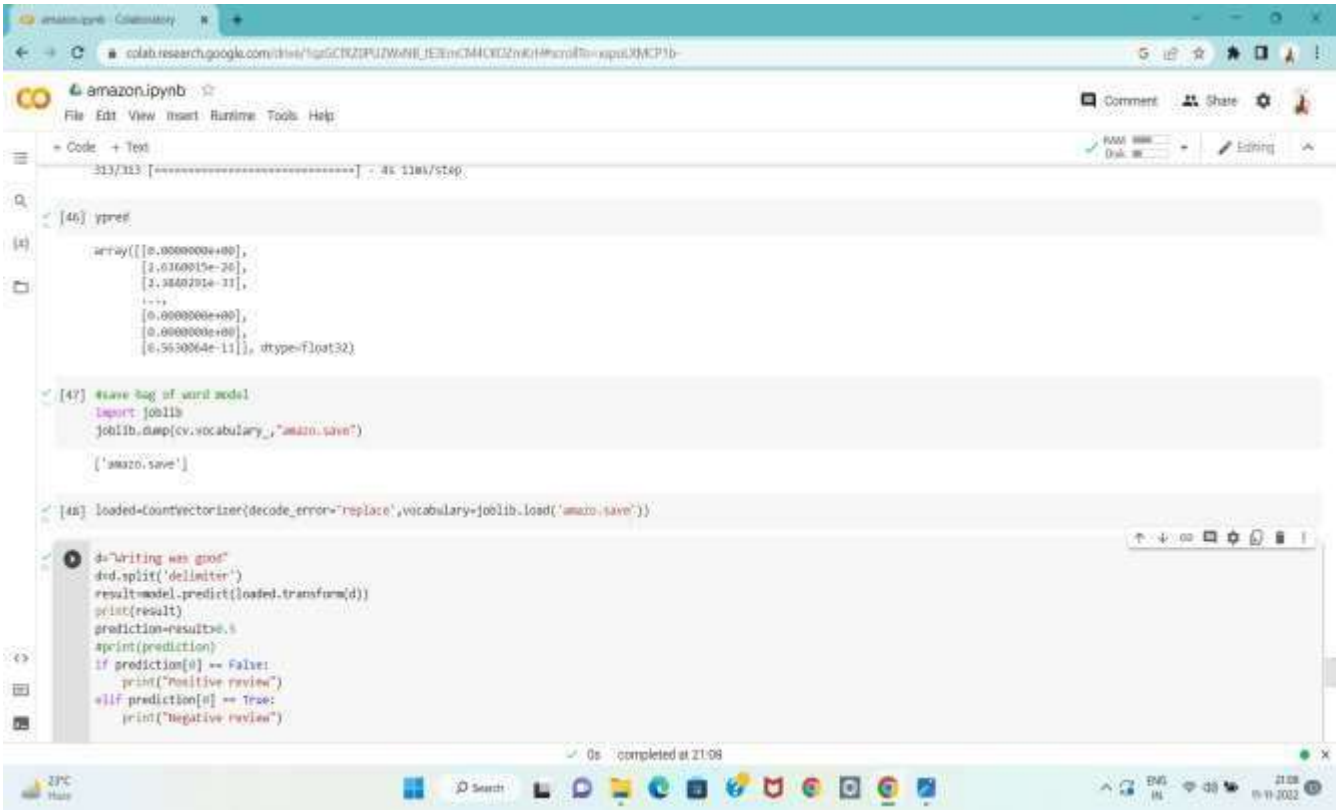

amazon.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment
Share
+ Code + Test
Editing
[30] corpus.append(temp)
[31] !pip install sklearn
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting sklearn
Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
Building wheels for collected packages: sklearn
Building wheel for sklearn (setup.py) ... done
Created wheel for sklearn: filename=sklearn-0.0.post1-py3-none-any.whl size=2344 sha256=95637709...
Stored in directory: /root/.cache/pip/wheels/42/56/cc/...
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")
0s completed at 20:50

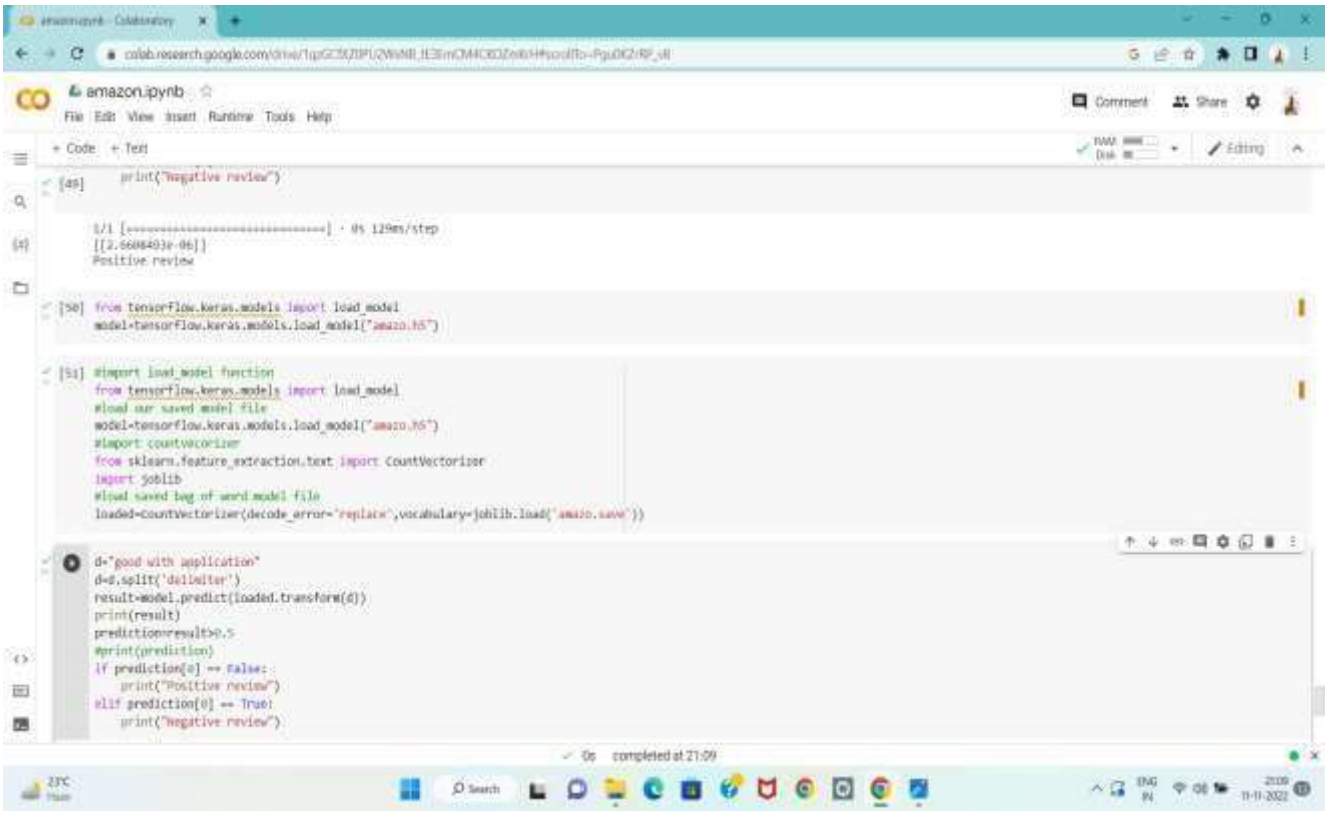

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Test
Editing
[34] #save bag of word model
import joblib
joblib.dump(cv.vocabulary_,"amazo.save")
['amazo.save']
[35] y=data.iloc[:,0].values
y
array([0, 0, 0, ..., 0, 0, 0])
[36] from sklearn.model_selection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

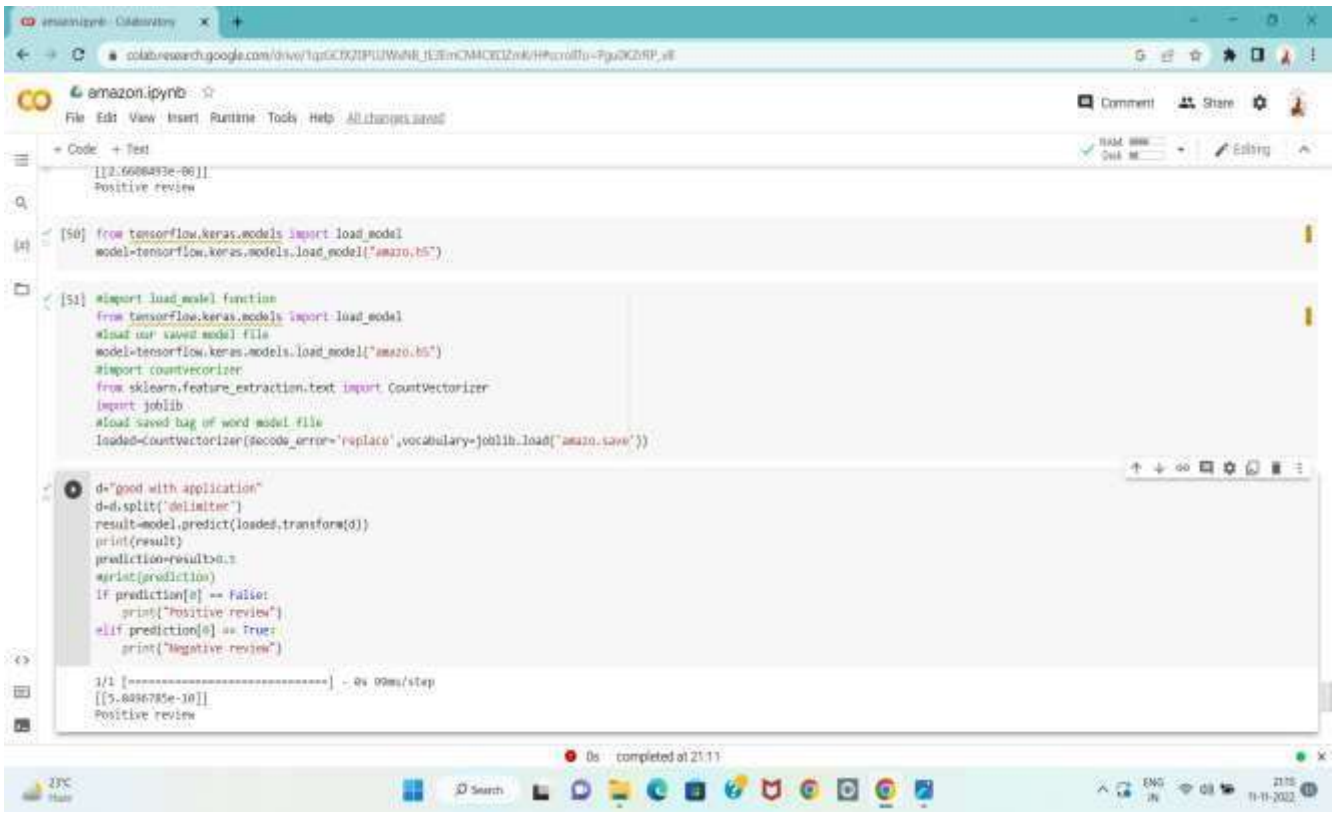

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
[40] #import libraries
import tensorflow
import keras
#import Sequential
from tensorflow.keras.models import Sequential
#import Dense
from tensorflow.keras.layers import Dense
[41] #intialize the model
model=Sequential()
#adding input layer
model.add(Dense(600,kernel_initializer='random_uniform',
activation='relu'))
#adding hidden layer
model.add(Dense(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
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
[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: 0.0058 - 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
13m 43s completed at 21:07


amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Epoch 10/20
1250/1250 [==========] - 40s 32ms/step - loss: 1.8263e-06 - accuracy: 1.0000
Epoch 11/20
1250/1250 [==========] - 40s 32ms/step - loss: 9.0213e-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: 2.2670e-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
4s completed at 21:07


amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
313/313 [==========] - 4s 11ms/step
[46] ypred
array([[8.0000000e+00],
[1.0360015e-26],
[2.3840291e-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")
0s completed at 21:08

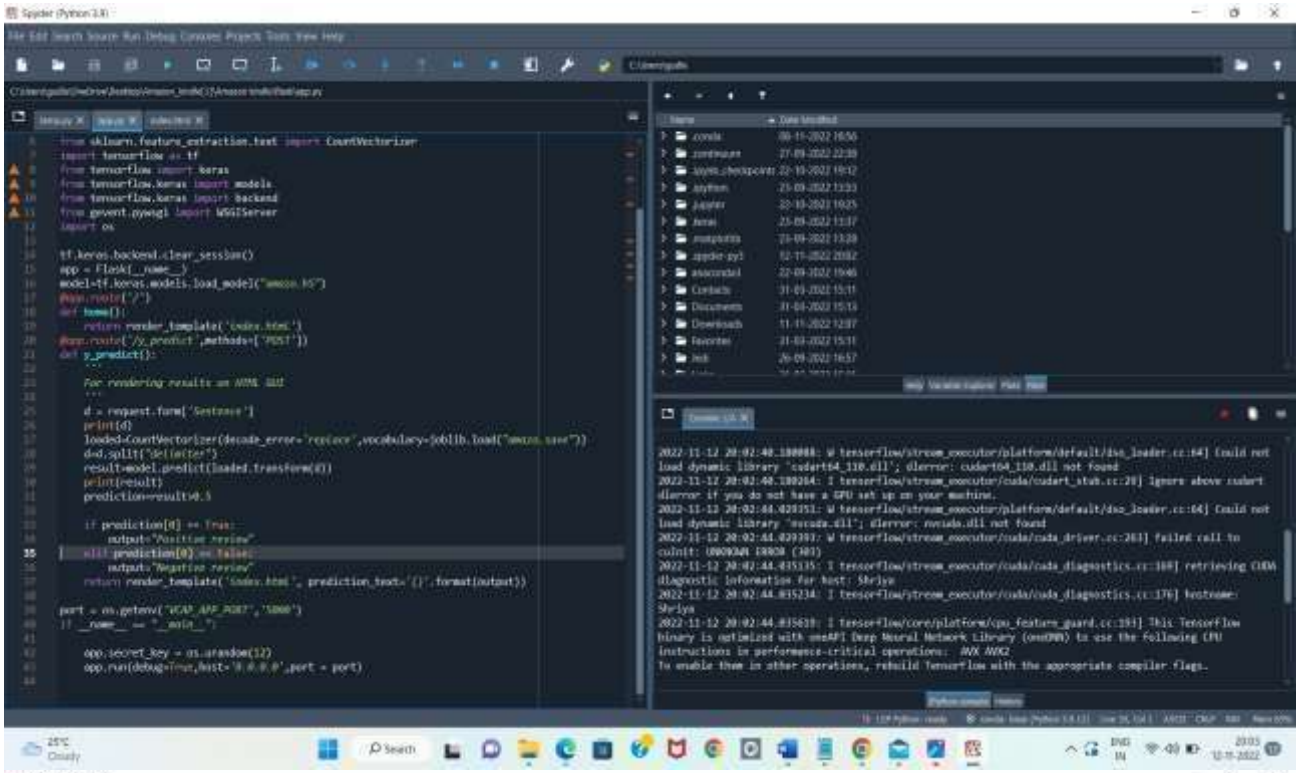

amazon.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
Editing
print("Negative review")
1/1 [==============================] - 0s 129ms/step
[[2.6608493e-06]]
Positive review
from tensorflow.keras.models import load_model
model=tensorflow.keras.models.load_model("amazo.h5")
#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.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")
0s completed at 21:09

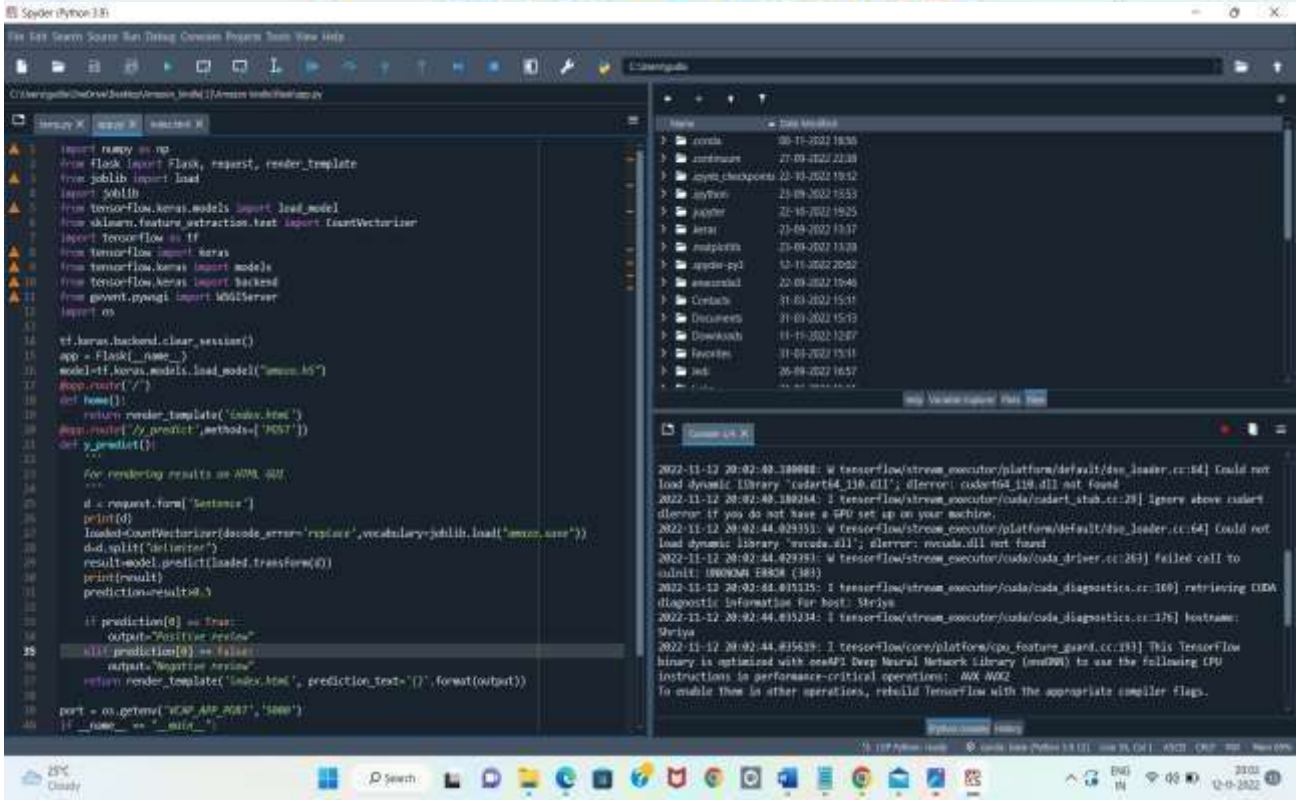

amazo.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Editing
[[2.6608493e-06]]
Positive review
from tensorflow.keras.models import load_model
model=tensorflow.keras.models.load_model("amazo.h5")
#import load_model function
from tensorflow.keras.models import 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.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
0s completed at 21:11

APPLICATION BUILDING

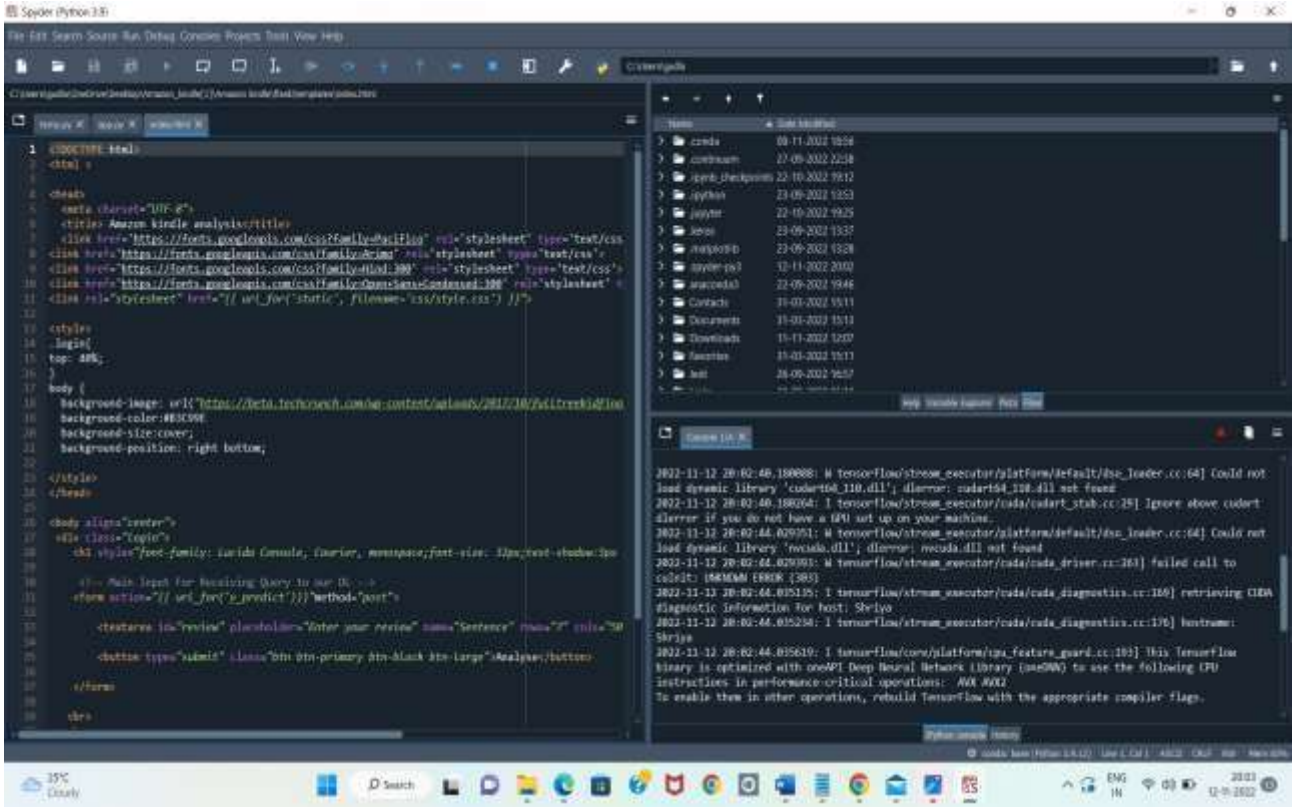
HTML CODE AND PYTHON CODE

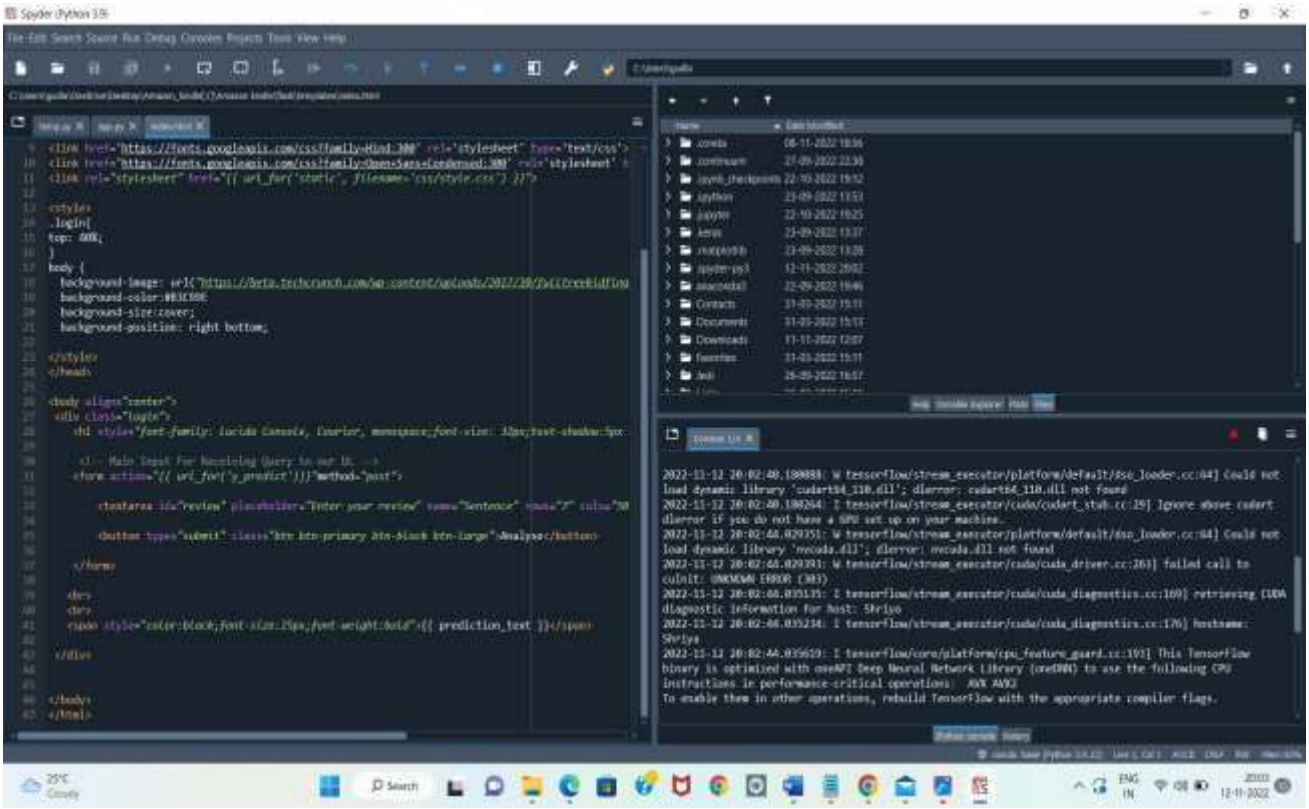
1.App.py code[FLASK]





2.Html code


Spyder (Python 3.9)
2022-11-12 20:02:40.180888: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-12 20:02:40.180264: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-12 20:02:44.029351: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-12 20:02:44.029393: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-12 20:02:44.035135: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Shriya
2022-11-12 20:02:44.035234: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Shriya
2022-11-12 20:02:44.035619: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.


Spyder (Python 3.9)
2022-11-12 20:02:40.180888: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-12 20:02:40.180264: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-12 20:02:44.029351: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-12 20:02:44.029393: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-12 20:02:44.035135: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Shriya
2022-11-12 20:02:44.035234: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Shriya
2022-11-12 20:02:44.035619: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.