

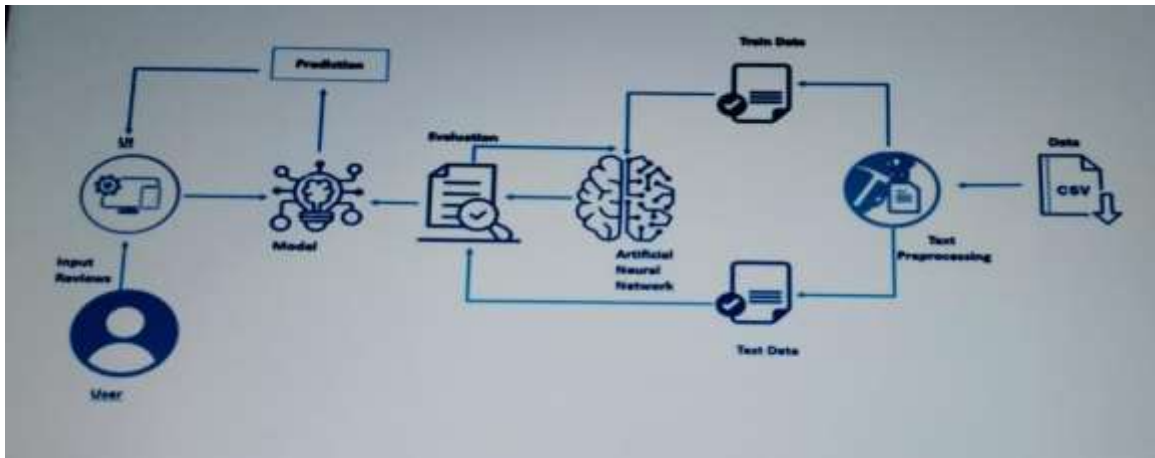
AMAZON KINDLE STORE REVIEWS ANALYSIS USING IBM WATSON SERVICES

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

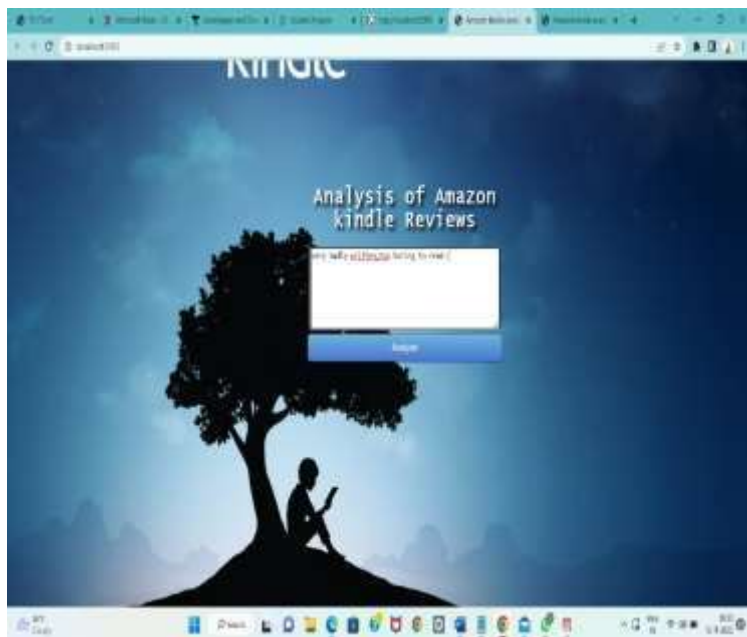
Create an HTML file

Build Python Code

6.RESULT



HOMEPAGE



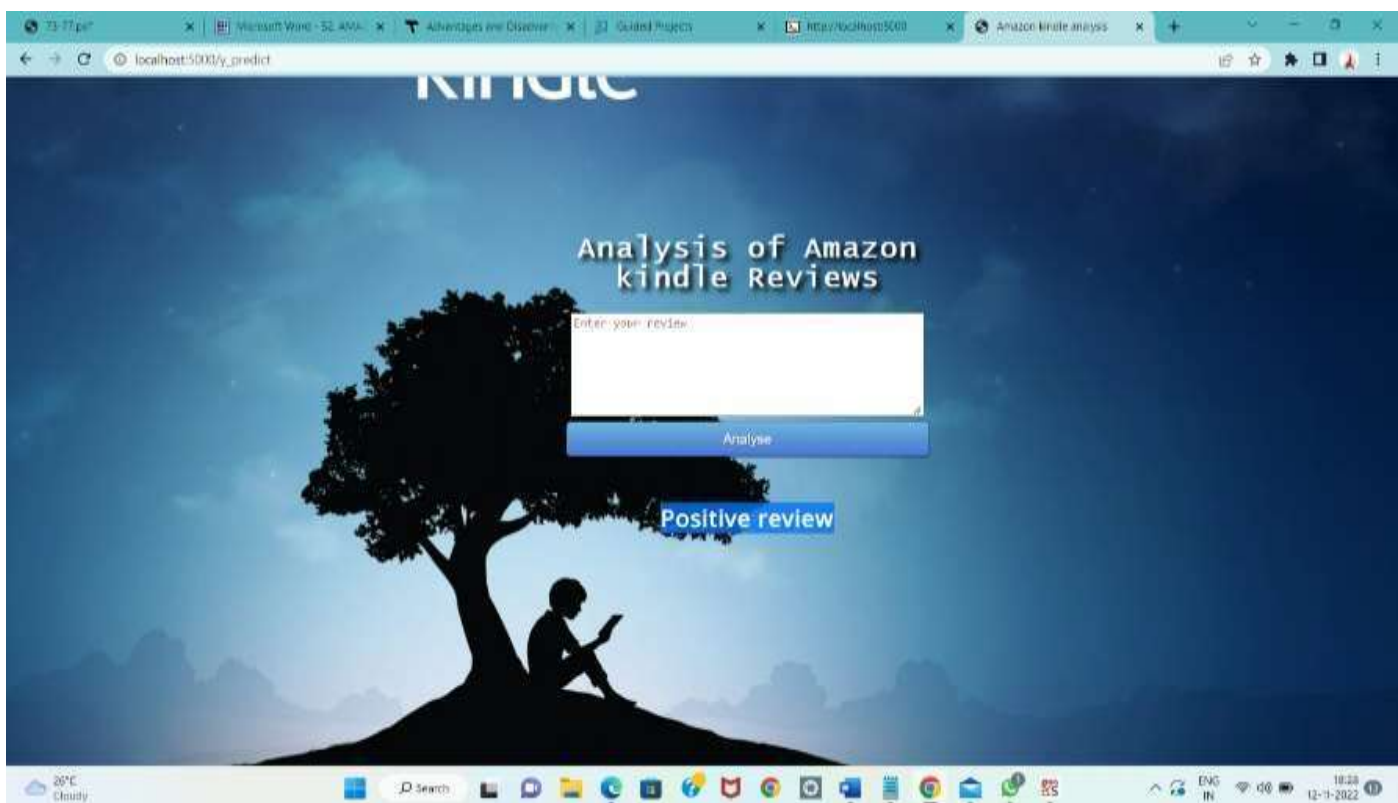
REVIEW ANALYSIS PAGE



OUTPUT



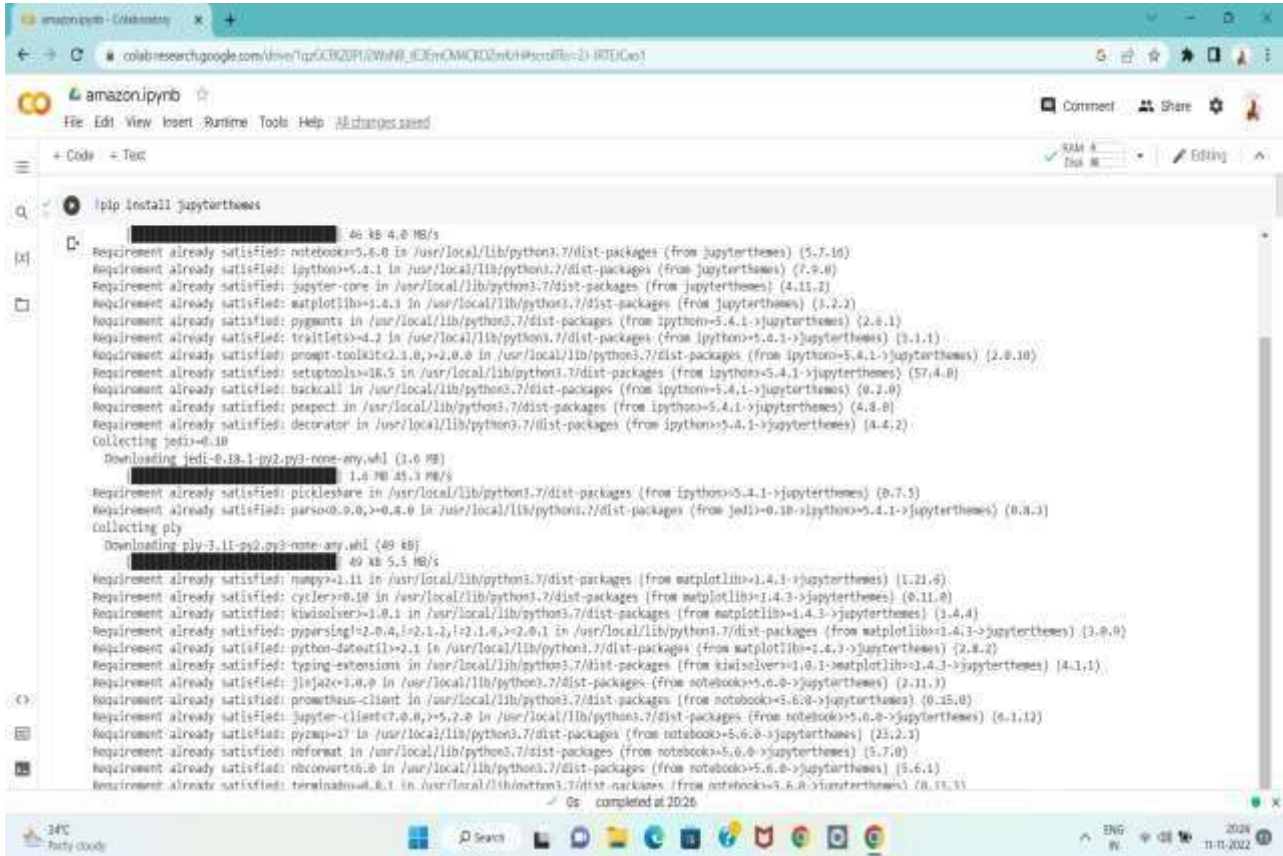
REVIEW ANALYSIS



OUTPUT

9.CODE SNIPPETS

A.MODEL BUILDING



```
amazon.ipython - ColabNotebook
colab-research.google.com/View?ipzGCKZUP12Wt#t#_tEDjYtCMACRQZmk6t#tcolab=2-10TDCap1

amazon.ipython
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

!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>4.2 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 (1.6 MB)
    1.6 MB 45.3 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>0.0.0,>=0.0.0 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,<2.1.2,>=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: Jinja2>=1.0.0 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) (6.6.1)
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>5.6.0->jupyterthemes) (0.15.3)
Completed at 20:26
```

```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qrGCBZ0P1QWwN8tE3EmCMACX0Zm6Hhsc0Rb-HCSpnJwRl

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

+ Code + Test
RAM 8 GB Disk 100 GB Editing

[2] !ls -t 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)

0s completed at 20:26
24°C Partly cloudy Search 28.33 11-10-2022
```

```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qrGCBZ0P1QWwN8tE3EmCMACX0Zm6Hhsc0Rb-HCSpnJwRl

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

+ Code + Test
RAM 8 GB Disk 100 GB Editing

[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  Avdreader  Nice vintage story  1399249000
1  1  B000F83S2Q  [2, 2]  4  This book is a mess of an old one; the auth...  01/8/2014  ANDW05ABLUED  citters  Different...  1388660400
2  2  B000F83S2Q  [2, 2]  4  This was a fairly interesting read. It had of...  04/4/2014  A795D8MNCJLAA  dot  Oldie  1396569600
3  3  B000F83S2Q  [1, 1]  5  It never read any of the Amy Brewer mysteri...  02/10/2014  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  A3SPT0KDG7WBLN  Father Dowling Fan  Period Mystery  1395187200

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

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

  unnamed: 0  asin  helpful  overall  reviewText  reviewTime  reviewerID  reviewTime  summary  unixReviewTime
0  False  False  False  False  True  False  False  False  False  False
1  False  False  False  False  True  False  False  False  False  False
2  False  False  False  False  True  False  False  False  False  False
3  False  False  False  False  True  False  False  False  False  False
4  False  False  False  False  True  False  False  False  False  False
5  False  False  False  False  True  False  False  False  False  False
6  False  False  False  False  True  False  False  False  False  False
7  False  False  False  False  True  False  False  False  False  False
8  False  False  False  False  True  False  False  False  False  False
9  False  False  False  False  True  False  False  False  False  False
```


amazon.ipynb - Colaboratory

colab.research.google.com/drive/1qG2C0Z0PU2WwNB_1E3EnCMAC0Zm6tHfucdRto-mjN8UIMCgAT

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

1 [10] data.isnull().sum()

```

overall: 0
asin: 0
helpful: 0
overall: 0
reviewText: 1
reviewTime: 0
reviewerID: 0
reviewTime: 140
summary: 0
unixreviewTime: 0
dtype: int64

```

2 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['reviewTime']
del data['unixreviewTime']

```

3 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 - Colaboratory

colab.research.google.com/drive/1qG2C0Z0PU2WwNB_1E3EnCMAC0Zm6tHfucdRto-mjN8UIMCgAT

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

1 [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

2 checking value counts

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

overall	count
5	21098
4	14988
3	7811
2	2832
1	288

Name: overall, dtype: int64

24°C Partly cloudy

Search

completed at 20:32

amazon.ipynb - Colaboratory

colab.research.google.com/drive/1upGCDZPUZWNH_EJ3EnCM4OCZnd6tHhPurdlfo=JufNyXicCh-

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

name: overall, dtype: int64

```
1 #check the null values
data.isna().sum()
```

```
2 overall      0
  reviewText   1
  summary      0
  dtype: int64
```

```
3 [15] #joining review description and summary into one col
data['reviewText']=data['reviewText']+ " "+data['summary']
```

```
4 [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, litigo, y...	Period Mystery

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

```
6 #checking for null values
data.isna().sum()
```

0 completed at 20:33

24°C Partly cloudy

amazon.ipynb - Colaboratory

colab.research.google.com/drive/1upGCDZPUZWNH_EJ3EnCM4OCZnd6tHhPurdlfo=JufNyXicCh-

amazon.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
7 [17] data.drop(['summary'],axis=1,inplace=True)
```

```
8 [18] #checking for null values
data.isna().sum()
```

```
overall      0
  reviewText   1
  dtype: int64
```

```
9 [19] #since there is only one null value, replace it with blank space
data['reviewText'].fillna("",inplace = True)
```

```
10 [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
```

```
11 [21] data.overall = data.overall.apply(review_sentiment)
```

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

```
0    45083
1     4917
Name: overall, dtype: int64
```

0 completed at 20:34

24°C Partly cloudy

amazon.ipynb - Colaboratory

colabresearch.google.com/drive/Top/c2920P02WwNf_E3JmCMAC62mkH#scrollTo=54R2GhpJox

amazon.ipynb

File Edit View Insert Runtime Tools Help Settings

+ Code + Text

name: overall, dtype: object

```
data.head(50)
```

18	0	I like these planetoids as per another good book. I li...
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...

completed at 20:34

24°C Party Boody

Search

ENG IN 2034 11-11-2332

amazon.ipynb - Colaboratory

colabresearch.google.com/drive/Top/c2920P02WwNf_E3JmCMAC62mkH#scrollTo=mmgadmUQgls

amazon.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[21]
```

35	0	Timothy Zahn's Foo'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 it...
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

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

completed at 20:34

24°C Party Boody

Search

ENG IN 2034 11-11-2332

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 alpino to /root/nltk_data...
[nltk_data]     Unzipping corpora/alpino.zip.
[nltk_data]   Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
```

Task 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)
```

Task 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/c1/548bf8661aaf0b07fb110477067c1f45c513a4121a001
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:

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

```
[ ] ['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

The screenshot shows a Jupyter Notebook interface with the following code cells:

```
[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(400,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]]
```

The notebook is titled "amazon.ipynb" and the code is saved. The output of the last cell shows a single row of zeros.

The screenshot shows the same Jupyter Notebook interface with the following code cells:

```
[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)
```

The output of the second cell shows the training progress over 20 epochs. The loss decreases from approximately 0.2035 to 1.8261e-06, and the accuracy increases from approximately 0.9252 to 1.0000.

Epoch	Loss	Accuracy
1/20	0.2035	0.9252
2/20	0.1114	0.9576
3/20	0.0575	0.9877
4/20	0.0098	0.9984
5/20	6.8669e-04	0.9998
6/20	7.1589e-05	1.0000
7/20	1.7642e-05	1.0000
8/20	7.8323e-06	1.0000
9/20	3.7948e-06	1.0000
10/20	1.8261e-06	1.0000

```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qGCIXZ0PUZWNHJE3EnCMACR2mKtHMcroRto-3PFGQRIpILZ

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

+ Code + Text
1250/1250 [=====] - 42s 348s/step - loss: 3.7940e-08 - accuracy: 1.0000
Epoch 10/20
1250/1250 [=====] - 40s 326s/step - loss: 1.8263e-08 - accuracy: 1.0000
Epoch 11/20
1250/1250 [=====] - 40s 326s/step - loss: 9.0213e-07 - accuracy: 1.0000
Epoch 12/20
1250/1250 [=====] - 40s 326s/step - loss: 4.5210e-07 - accuracy: 1.0000
Epoch 13/20
1250/1250 [=====] - 41s 326s/step - loss: 2.2570e-07 - accuracy: 1.0000
Epoch 14/20
1250/1250 [=====] - 41s 336s/step - loss: 1.1774e-07 - accuracy: 1.0000
Epoch 15/20
1250/1250 [=====] - 41s 336s/step - loss: 6.1113e-08 - accuracy: 1.0000
Epoch 16/20
1250/1250 [=====] - 44s 358s/step - loss: 3.2802e-08 - accuracy: 1.0000
Epoch 17/20
1250/1250 [=====] - 41s 336s/step - loss: 1.0273e-08 - accuracy: 1.0000
Epoch 18/20
1250/1250 [=====] - 41s 336s/step - loss: 1.0481e-08 - accuracy: 1.0000
Epoch 19/20
1250/1250 [=====] - 41s 336s/step - loss: 0.1321e-08 - accuracy: 1.0000
Epoch 20/20
1250/1250 [=====] - 42s 336s/step - loss: 3.7390e-09 - accuracy: 1.0000
kera.callbacks.History at 0x7fa04fd1bc90

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

ypred=model.predict(x_test)

113/113 [=====] - 4s 118s/step

[ ] ypred

4s completed at 21:07
```

```
amazon.ipynb - Colaboratory
colab.research.google.com/notebook/1qGCIXZ0PUZWNHJE3EnCMACR2mKtHMcroRto-3PFGQRIpILZ

amazon.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text
313/313 [=====] - 4s 118s/step

[46] ypred

array([[0.0000000e+00],
       [2.6360015e-20],
       [2.3840701e-11],
       ...,
       [0.0000000e+00],
       [0.0000000e+00],
       [0.5630064e-11]], 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
```



```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qjGZ3ZPU2WwMBLJE3mCMACDZek6H#scrollTo=Pgu0Q2RP_vll

amazon.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
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
```

```
amazon.ipynb - Colaboratory
colab.research.google.com/drive/1qjGZ3ZPU2WwMBLJE3mCMACDZek6H#scrollTo=Pgu0Q2RP_vll

amazon.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[[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")

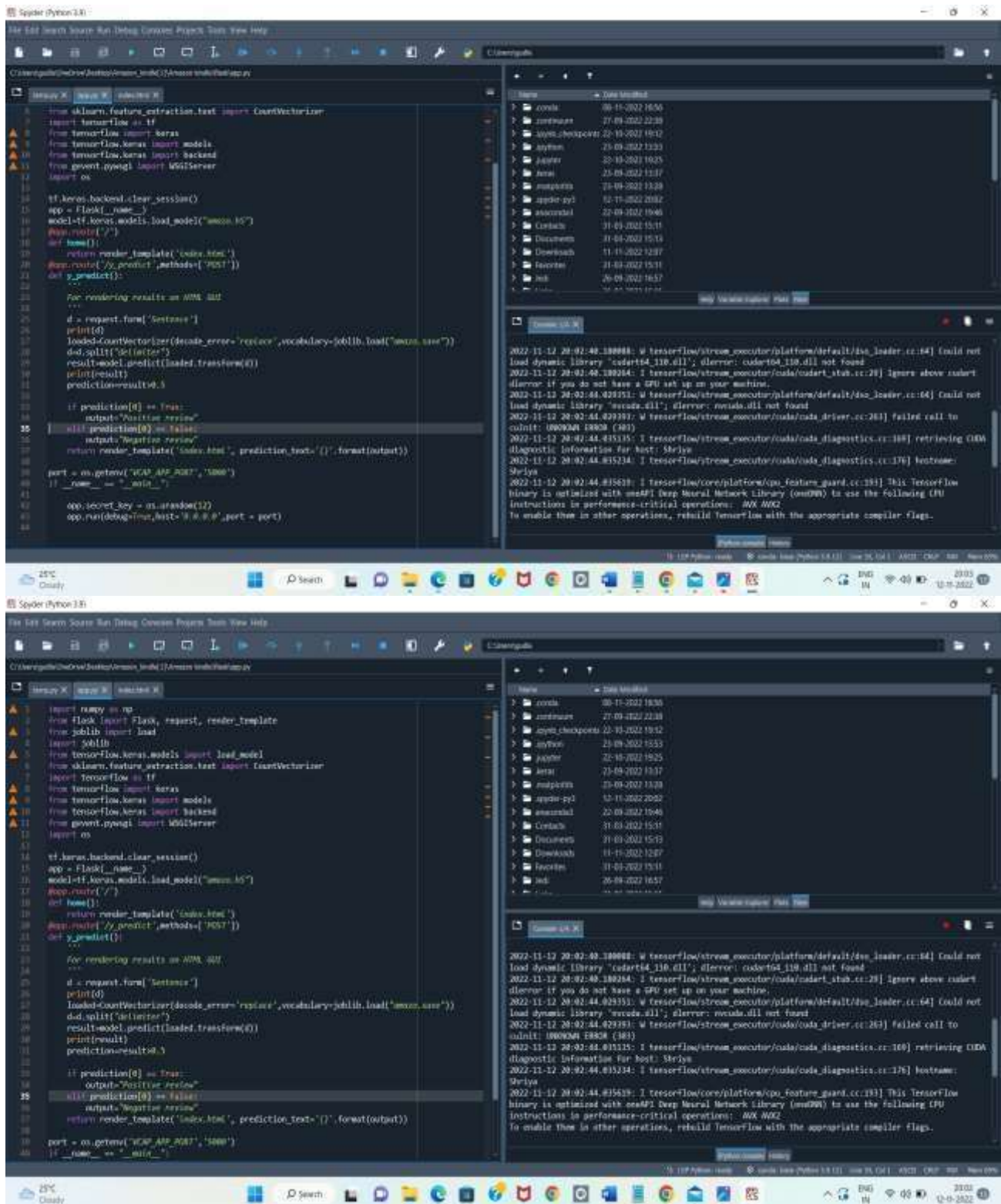
1/1 [=====] - 0% 99ms/step
[[5.8486785e-10]]
Positive review

0s completed at 21:11
```

APPLICATION BUILDING

HTML CODE AND PYTHON CODE

1.App.py code[FLASK]



```
1 from sklearn.feature_extraction.text import CountVectorizer
2 import tensorflow as tf
3 from tensorflow.keras import layers
4 from tensorflow.keras import models
5 from tensorflow.keras import backend
6 from google.colab import notebook
7 import os
8
9 tf.keras.backend.clear_session()
10 app = Flask(__name__)
11 model = tf.keras.models.load_model('mnist.h5')
12 @app.route('/')
13 def home():
14     return render_template('index.html')
15 @app.route('/predict', methods=['POST'])
16 def predict():
17     ...
18     for rendering results on HTML GUI
19     ...
20     d = request.form['textinput']
21     print(d)
22     loaded_count_vectorizer = CountVectorizer(encoding='utf-8', vocabulary=joblib.load('mnist_vec.pkl'))
23     d = split(d, delimiter=' ')
24     result = model.predict(loaded_count_vectorizer.transform(d))
25     print(result)
26     prediction = result[0]
27     if prediction[0] == True:
28         output = 'Positive Review'
29     else:
30         output = 'Negative Review'
31     return render_template('index.html', prediction_text='{}'.format(output))
32
33 port = os.getenv('PORT', 5000)
34 if __name__ == '__main__':
35     app.run(debug=True, host='0.0.0.0', port=port)
```

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 from joblib import load
4 import joblib
5 from tensorflow.keras.models import load_model
6 from sklearn.feature_extraction.text import CountVectorizer
7 import tensorflow as tf
8 from tensorflow.keras import layers
9 from tensorflow.keras import models
10 from tensorflow.keras import backend
11 from google.colab import notebook
12 import os
13
14 tf.keras.backend.clear_session()
15 app = Flask(__name__)
16 model = tf.keras.models.load_model('mnist.h5')
17 @app.route('/')
18 def home():
19     return render_template('index.html')
20 @app.route('/predict', methods=['POST'])
21 def predict():
22     ...
23     for rendering results on HTML GUI
24     ...
25     d = request.form['textinput']
26     print(d)
27     loaded_count_vectorizer = CountVectorizer(encoding='utf-8', vocabulary=joblib.load('mnist_vec.pkl'))
28     d = split(d, delimiter=' ')
29     result = model.predict(loaded_count_vectorizer.transform(d))
30     print(result)
31     prediction = result[0]
32     if prediction[0] == True:
33         output = 'Positive Review'
34     else:
35         output = 'Negative Review'
36     return render_template('index.html', prediction_text='{}'.format(output))
37
38 port = os.getenv('PORT', 5000)
39 if __name__ == '__main__':
```

2.Html code

