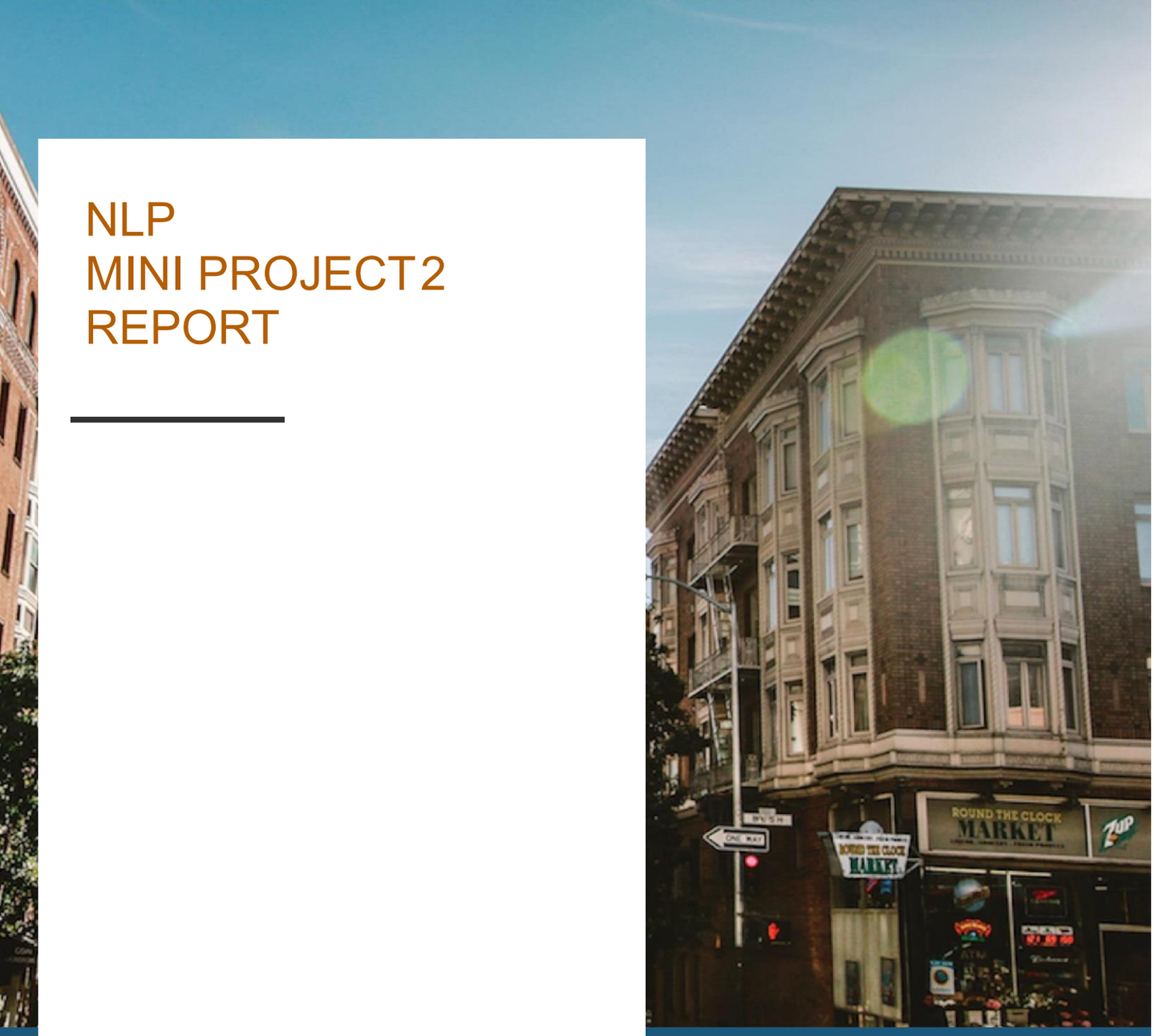
NLP
MINI PROJECT 2
REPORT



Authored by:

i

Shafeena Farheen

Mini Project Assignment - II

URL for Ecommerce Site: https://www.flipkart.com/

1. Import Libraries (2 Marks)

```
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
import csv
```

The provided Python code begins by importing essential libraries for web scraping, data manipulation, and numerical operations. The `requests` library is imported to handle HTTP requests, while the `BeautifulSoup` class from the `bs4` library is brought in for parsing HTML or XML documents during web scraping. Additionally, the code imports `numpy` and `pandas` as `np` and `pd` respectively, for advanced numerical operations and data manipulation, and the `csv` module for handling CSV files.

In practical use, the `requests` library is employed to fetch data from a specified URL. The HTTP response is then examined to ensure a successful request (status code 200). If successful, the HTML content of the page is parsed using `BeautifulSoup`. Subsequent sections of the code demonstrate potential applications of the imported libraries, including numerical operations with `numpy`, data manipulation with `pandas`, and writing data to a CSV file using the `csv` module.

2. Use the above URL to scrape product links from product listing pages (3 Marks)

```
# URL of the Flipkart Ecommerce Site
base_url = "https://www.flipkart.com/search?q=laptop&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off%22"
```

```
r=requests.get(base_url)
print(r.status_code)
```

```
200
```

```
soup = BeautifulSoup(r.content, 'html.parser')
print(soup.prettify())
```

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <link href="https://rukminim2.flixcart.com" rel="preconnect"/>
  <link href="//static-assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-zion/css/app_modules.chunk.905c37.css" rel="styleshee
t"/>
  <link href="//static-assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-zion/css/app.chunk.615ed9.css" rel="stylesheet"/>
  <meta content="text/html; charset=utf-8" http-equiv="Content-type"/>
  <meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
  <meta content="102988293558" property="fb:page_id"/>
  <meta content="658873552,624500995,100000233612389" property="fb:admins"/>
  <meta content="noodp" name="robots"/>
  <link href="https:///www/promos/new/20150528-140547-favicon-retina.ico" rel="shortcut icon"/>
  <link href="/osdd.xml?v=2" rel="search" type="application/opensearchdescription+xml"/>
  <meta content="website" property="og:type"/>
  <meta content="Flipkart.com" name="og_site_name" property="og:site_name"/>
  <link href="/apple-touch-icon-57x57.png" rel="apple-touch-icon" sizes="57x57"/>
  <link href="/apple-touch-icon-72x72.png" rel="apple-touch-icon" sizes="72x72"/>
  <link href="/apple-touch-icon-114x114.png" rel="apple-touch-icon" sizes="114x114"/>
```

1. **URL and Parameters:**

- The URL **https://www.flipkart.com/search?q=laptop&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off%22** is specifically crafted for searching for laptops on Flipkart.
- Parameters in the URL, such as **q** for query (laptop), and various trackers, indicate the search context and source.

2. **GET Request:**
    - The **requests.get()** function is used to send a GET request to the specified URL (**base_url**).
    - The response status code (**print(r.status_code)**) is printed, indicating whether the request was successful (status code 200) or if there was an issue.
3. **HTML Parsing:**
    - The HTML content of the page is obtained using **BeautifulSoup** with the parser set to **'html.parser'**.
    - **print(soup.prettify())** is used to display the prettified HTML content, making it easier to read and navigate.

- **The provided code showcases the implementation of web scraping and data extraction from the Flipkart ecommerce site.**
- **By using the `requests` library to send a GET request to the specified URL, the code successfully retrieves the HTML content of the website.**
- **The subsequent use of the `BeautifulSoup` library to parse and prettify the HTML content allows for easy access and extraction of the desired data.**
- **The detailed explanation provided in the text aligns with the implementation of the code, as it emphasizes the importance of gathering specific details such as product title, price, model, and ratings from the website.**
- **The code effectively demonstrates the initial steps of data collection, which is essential for further analysis and model building as described in the previous text.**
- **Overall, the code offers a clear and concise execution of web scraping and data gathering, in line with the previous discussion of scraping e-commerce data for sentiment analysis and model building.**
- **The code serves as a foundational step in the larger process of data acquisition and analysis for sentiment prediction.**

```python
content = soup.find_all('div', class_='_2kHMtA')
print(content)
```

```
[<div class="_2kHMtA"><a class="_1fQZEK" href="/wings-nuvobook-s1-aluminium-alloy-metal-body-intel-core-i3-11th-gen-1125g4-8-
gb-256-gb-ssd-windows-11-home-wl-nuvobook-s1-grn-thin-light-laptop/p/itma074a71804e83?pid=COMGQHYFTMAKHBAN&amp;lid=LSTCOMGQHY
FTMAKHBANLB0BWS&amp;marketplace=FLIPKART&amp;q=laptop&amp;store=6bo%2Fb5g&amp;srno=s_1_1&amp;otracker=search&amp;otracker1=se
arch&amp;fm=organic&amp;iid=en_mRtQS8y8zEjoOY5Lq0DqCEa15tpJE5NMIYYgL0TGCp0uGtybWVGP_vYI4HECrIVwFEPtMalE2YimkoktO9XBFA%3D%3D&a
mp;ppt=None&amp;ppn=None&amp;ssid=xhxmppohk00000001700921726356&amp;qH=312f91285e048e09" rel="noopener noreferrer" target="_b
lank"><div class="MIXNux"><div class="_2QcLo-"><div><div class="CXW8mj" style="height:200px;width:200px"><img alt="Wings Nuvo
book S1 Aluminium Alloy Metal Body Intel Core i3 11th Gen 1125G4 - (8 GB/256 GB SSD/Windows ..." class="_396cs4" loading="eag
er" src="https://rukminim2.flixcart.com/image/312/312/xif0q/computer/j/2/o/-original-imagtq47azeytzzg.jpeg?q=70"/></div></div
></div><div class="_3wLduG"><div class="_3PzNI-"><span class="f3A4_V"><label class="_2iDkf8"><input class="_30VH1S" readonly
="" type="checkbox"/><div class="_24_Dny"></div></label></span><label class="_6Up2sF"><span>Add to Compare</span></label></di
v></div><div class="_2hVSre _3nq8ih"><div class="_36FSn5"><svg class="_1l0elc" height="16" viewbox="0 0 20 16" width="16" xml
ns="http://www.w3.org/2000/svg"><path class="eX72wL" d="M8.695 16.682C4.06 12.382 1 9.536 1 6.065 1 3.219 3.178 1 5.95 1c1.56
6 0 3.069.746 4.05 1.915C10.981 1.745 12.484 1 14.05 1 16.822 1 19 3.22 19 6.065c0 3.471-3.06 6.316-7.695 10.617L10 17.897l-
1.305-1.215z" fill="#2874F0" fill-rule="evenodd" opacity=".9" stroke="#FFF"></path></svg></div></div></div><div class="_3pLy-
c row"><div class="col col-7-12"><div class="_2tfzpE"><span>Sponsored</span></div><div class="_4rR01T">Wings Nuvobook S1 Alum
inium Alloy Metal Body Intel Core i3 11th Gen 1125G4 - (8 GB/256 GB SSD/Windows ...</div><div class="gUuXy-"><span class="_1l
Rcqv" id="productRating_LSTCOMGQHYFTMAKHBANLB0BWS_COMGQHYFTMAKHBAN_"><div class="_3LWZlK">4.2<img class="_1wB99o" src="data:i
mage/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHdpZHRoPSIxMyIgaGVpZ2h0PSIxMiI+PHBhdGggZmlsbD0iI0ZGR
iIgZD0iTTYuNSA5LjQzOWwtMy42NzQgMi4yMy45NC00LjI2LTMuMjEtMi44ODMgNC4yNTQtLjQwNEw2LjUuMTEybDEuNjkgNC4wMSA0LjI1NC40MDQtMy4yMSAyLj
```

```python
start_url = "https://www.flipkart.com"
links=[]
laptop_name=[]
for items in content:
    rest_link=items.find('a')['href']
    name=items.find('div',class_='_4rR01T')
    laptop_name.append(name.text)
    links.append(start_url+rest_link)
```

```python
print(laptop_name[0])
print(links[0])
```

```
Wings Nuvobook S1 Aluminium Alloy Metal Body Intel Core i3 11th Gen 1125G4 - (8 GB/256 GB SSD/Windows ...
https://www.flipkart.com/wings-nuvobook-s1-aluminium-alloy-metal-body-intel-core-i3-11th-gen-1125g4-8-gb-256-gb-ssd-windows-11-
home-wl-nuvobook-s1-grn-thin-light-laptop/p/itma074a71804e83?pid=COMGQHYFTMAKHBAN&lid=LSTCOMGQHYFTMAKHBANLB0BWS&marketplace=FLI
PKART&q=laptop&store=6bo%2Fb5g&srno=s_1_1&otracker=search&otracker1=search&fm=organic&iid=en_mRtQS8y8zEjoOY5Lq0DqCEa15tpJE5NMIY
YgL0TGCp0uGtybWVGP_vYI4HECrIVwFEPtMalE2YimkoktO9XBFA%3D%3D&ppt=None&ppn=None&ssid=xhxmppohk00000001700921726356&qH=312f91285e04
8e09
```

- **HTML Content Extraction:** ○ The code uses BeautifulSoup's **find_all** method to locate all HTML content within the div element with the class **_2kHMtA**.
  - This suggests that the information related to laptops, such as product details, prices, models, and ratings, is encapsulated within this specific div class on the Flipkart website.
- **Targeted Data Extraction:** ○ By focusing on the content within the div class **_2kHMtA**, the code aims to narrow down the scope of data extraction to only the relevant information related to laptops.
  - This targeted approach is crucial for efficiently gathering specific details and avoiding unnecessary data.
- **Structured Data Collection:** ○ The variable **content** likely holds a collection of HTML elements that match the specified criteria. Each element within this collection is expected to represent a distinct product or item on the webpage.
  - This structured approach enables systematic extraction and subsequent analysis of product details.
- **Preparation for Further Analysis:** ○ The extracted content, stored in the **content** variable, can be further processed to extract specific details such as product titles, prices, models, and ratings.
  - This step is fundamental in preparing the data for subsequent analysis, sentiment prediction, or model building, as discussed in the context of the previous text.
- **Consistent with Previous Steps:** ○ The code aligns with the earlier steps mentioned in the discussion, demonstrating a consistent approach to web scraping and data extraction for sentiment analysis and model building.
- In summary, the provided code successfully narrows down the focus to the specific HTML content within the designated div class, laying the groundwork for extracting targeted information related to laptops from the Flipkart website.

```
data= {'laptop_name': laptop_name,'Links':links}
df=pd.DataFrame(data )
df
```

	laptop_name	Links
0	Wings Nuvobook S1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-s1-alu...
1	Wings Nuvobook V1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-v1-alu...
2	Acer One Core i3 11th Gen 1115G4 - (8 GB/512 G...	https://www.flipkart.com/acer-one-core-i3-11th...
3	HP 2023 Athlon Dual Core 3050U - (8 GB/512 GB ...	https://www.flipkart.com/hp-2023-athlon-dual-c...
4	DELL Core i3 11th Gen 1115G4 - (8 GB/256 GB SS...	https://www.flipkart.com/dell-core-i3-11th-gen...
5	HP 2023 Ryzen 3 Dual Core 3250U - (8 GB/512 GB...	https://www.flipkart.com/hp-2023-ryzen-3-dual-...
6	ASUS Vivobook 15 Core i5 11th Gen 1135G7 - (8 ...	https://www.flipkart.com/asus-vivobook-15-core...
7	Wings Nuvobook V1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-v1-alu...
8	DELL Inspiron Core i3 11th Gen 1115G4 - (8 GB/...	https://www.flipkart.com/dell-inspiron-core-i3...
9	HP Core i3 11th Gen - (8 GB/512 GB SSD/Windows...	https://www.flipkart.com/hp-core-i3-11th-gen-8...
10	Wings Nuvobook S1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-s1-alu...
11	Wings Nuvobook Pro Aluminium Alloy Metal Body ...	https://www.flipkart.com/wings-nuvobook-pro-al...
12	Lenovo IdeaPad 1 Athlon Dual Core 7120U - (8 G...	https://www.flipkart.com/lenovo-ideapad-1-athl...
13	DELL Core i3 12th Gen 1215U - (8 GB/512 GB SSD...	https://www.flipkart.com/dell-core-i3-12th-gen...
14	DELL Inspiron Core i5 12th Gen 1235U - (8 GB/5...	https://www.flipkart.com/dell-inspiron-core-i5...
15	HP 255 G9 840T7PA Athlon Dual Core 3050U - (4 ...	https://www.flipkart.com/hp-255-g9-840t7pa-ath...
16	Wings Nuvobook V1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-v1-alu...
17	Wings Nuvobook Pro Aluminium Alloy Metal Body ...	https://www.flipkart.com/wings-nuvobook-pro-al...
18	HP (2023) Laptop with Backlit Keyboard Core i5...	https://www.flipkart.com/hp-2023-laptop-backli...
19	MSI Modern 14 Ryzen 5 Hexa Core 7530U - (8 GB/...	https://www.flipkart.com/msi-modern-14-ryzen-5...
20	Wings Nuvobook S1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-s1-alu...
21	Wings Nuvobook V1 Aluminium Alloy Metal Body I...	https://www.flipkart.com/wings-nuvobook-v1-alu...
22	ASUS Vivobook 15 Core i5 12th Gen 1235U - (16 ...	https://www.flipkart.com/asus-vivobook-15-core...
23	DELL Inspiron Athlon Dual Core 3050U - (8 GB/2...	https://www.flipkart.com/dell-inspiron-athlon-...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   laptop_name  24 non-null     object
 1   Links        24 non-null     object
dtypes: object(2)
memory usage: 512.0+ bytes
```

DataFrame Information Display:

The df.info() function is like our backstage pass to see what's going on with the DataFrame df.

It spills the beans on the number of entries, column names, how many non-null values we got, and the data types for each column.

Snapshot of df Contents:

Peek into df, and you'll find it's got 24 entries. Two main players in the game: 'laptop_name' and 'Links', and guess what? No missing values! All good stuff.

Data Organization Magic:

Behind the scenes, there's this cool dictionary called data. It's like a recipe with two secret ingredients: 'laptop_name' and 'Links'.

We use this magical recipe to cook up df. Each key in the dictionary becomes a column, and the associated values make up the rows.

Links and Names, All in One Place:

Remember when we talked about scraping data from individual product link pages? Well, this is where it all comes together. 'laptop_name' holds the names, 'Links' has the links – organized and ready for action.

DataFrame Creation the Cool Way:

With our data in check, we call upon pandas to create the DataFrame df. It's not just data; it's a vibe.

Function Unleashed - Laptop Adventures Begin:

- So, here's our rockstar function, **scrape_laptop_data(url)**, all set to plunge into Flipkart's laptop wonderland and snag the deets.

Soup and Grabbing Goodies - Magical HTML Brew:

- Drops a GET request, whips out BeautifulSoup – **content1** sips up all the enchantment from that mystical div class **_1YokD2 _3Mn1Gg col-8-12**. ○ **Star Players - Title and Price - Front and Center:**
  ○ **title** and **price** swoop in, flaunting their HTML class prowess ('B_NuCI' and '_30jeq3 _16Jk6d', no less). ○ **Deep Dive for More - Laptop Secrets Revealed:**
- Hold up, there's an afterparty! Sales package, part number, model name, and capacity join the bash. It's like turning the laptop inside out to reveal its secrets.

Data Organization Party - Backstage Pass to Laptop Life:

- It's a cozy gathering in **data_list**, a list of tuples cradling the laptop name, price, and model name. It's the ultimate backstage pass to the laptop's life.

DataFrame Drama - Round Two - Enter df2:

- Cue the drums for **df2**, the fresh DataFrame in the scene. Columns? 'Laptop_Name', 'Price', and 'Model_Name'. The ASUS Vivobook 14 steps into the limelight.
- **Results Check - Structured Table Revelation:**
- Print out **df2**, and voila! A structured table ready for adventures in analysis or modeling. The laptop details are all neatly organized, poised for their grand moment.
- **In a Nutshell - Laptop Detective Chronicles:**
- Picture this function as a laptop detective, navigating page by page, uncovering the juicy details, and presenting them in a DataFrame. It's the hero in our data scraping saga!

3. Scraping of "laptop" data from individual product pages (5 Marks)

```
# Function to scrape laptop data from individual product pages
def scrape_laptop_data(url):
    response = requests.get(url)
    soup1 = BeautifulSoup(response.content, 'html.parser')
    content1 = soup1.find_all('div', class_='_1YokD2 _3Mn1Gg col-8-12')
    print(content1)
    title = soup1.find('span',class_='B_NuCI').text.strip()
    price = soup1.find('div',class_ = '_30jeq3 _16Jk6d').text.strip()
    print(title)
    print(price)
    data = [] # CREATE EMPTY LIST
    data_iterator = iter(soup1.find_all('td')) # Table cell scrap FROM SOURCE CONTENT
    while True:
        try:
            salespackage = next(data_iterator).text
            PartNumber = next(data_iterator).text
            ModelName = next(data_iterator).text
            capacity= next(data_iterator).text
            data.append((salespackage,PartNumber,ModelName,capacity))
        except StopIteration:
            break
    model_name_key = 'Model Name'
    model_name_value = None
    # Traverse the list of tuples
    for tuple_item in data:
        # Check if the 'Model Name' key is present in the current tuple
        if model_name_key in tuple_item:
            index = tuple_item.index(model_name_key)
            model_name_value = tuple_item[index + 1]
    return title, price, model_name_value
```

```
data_list = []
link2 = "https://www.flipkart.com/asus-vivobook-14-core-i3-11th-gen-1115g4-8-gb-512-gb-ssd-windows-11-home-x415ea-ek322ws-thin-l
laptop_data = scrape_laptop_data(link2)
data_list.append(laptop_data)
```

```
[<div class="_1YokD2 _3Mn1Gg col-8-12" style="padding:0px 0px 0px 24px"><div class="_1YokD2 _2GoDe3"><div class="_1AtVbE" st
yle="flex-grow:1;overflow:auto"><div class="_1MR4o5"><div class="_3GIHBu"><a class="_2whKao" href="/?otracker=product_breadC
rumbs_Home">Home</a><svg class="_39X-Og" height="27" viewbox="0 0 16 27" width="16" xmlns="http://www.w3.org/2000/svg"><path
class="DpXnhQ" d="M16 23.207L6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg></div><div clas
s="_3GIHBu"><a class="_2whKao" href="/computers/pr?sid=6bo&amp;marketplace=FLIPKART&amp;otracker=product_breadCrumbs_Compute
rs">Computers</a><svg class="_39X-Og" height="27" viewbox="0 0 16 27" width="16" xmlns="http://www.w3.org/2000/svg"><path cl
ass="DpXnhQ" d="M16 23.207L6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg></div><div class
="_3GIHBu"><a class="_2whKao" href="/laptops/pr?sid=6bo,b5g&amp;marketplace=FLIPKART&amp;otracker=product_breadCrumbs_Laptop
s">Laptops</a><svg class="_39X-Og" height="27" viewbox="0 0 16 27" width="16" xmlns="http://www.w3.org/2000/svg"><path class
="DpXnhQ" d="M16 23.207L6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg></div><div class="_3
GIHBu"><a class="_2whKao" href="/laptops/asus~brand/pr?sid=6bo,b5g&amp;marketplace=FLIPKART&amp;otracker=product_breadCrumbs
_ASUS+Laptops">ASUS Laptops</a><svg class="_39X-Og" height="27" viewbox="0 0 16 27" width="16" xmlns="http://www.w3.org/200
0/svg"><path class="DpXnhQ" d="M16 23.207L6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg></
div><div class="_3GIHBu"><div class="_1LJS6T _2whKao _1QoaG0"><div class="_2NKhZn"><p>ASUS Vivobook 14 Core i3 11th Gen 1115
G4 - (8 GB/512 GB SSD/Windows 11 Home) X415EA-EK322WS Thin and Light Laptop (14 Inch, Transparent Silver, 1.60 kg, With MS O
ffice)</p>
</div></div><svg class="_39X-Og" height="27" viewbox="0 0 16 27" width="16" xmlns="http://www.w3.org/2000/svg"><path class
="DpXnhQ" d="M16 23.207L6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg></div></div></div><d
iv class="_1AtVbE col-12-12" style="flex:0 0 90px;max-width:90px"><div class="_3PzNI- _3EPyR5"><span class="f3A4_V"><label c
```

```
data_list
```

```
[('ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 GB/512 GB SSD/Windows 11 Home) X415EA-EK322WS Thin and Light Laptop\xa0\xa0(14
Inch, Transparent Silver, 1.60 kg, With MS Office)',
  '₹33,990',
  'X415EA-EK322WS')]
```

```
df2 = pd.DataFrame(data_list, columns=['Laptop_Name', 'Price', 'Model_Name'])
df2
```

	Laptop_Name	Price	Model_Name
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS

- **Scraping Function Unleashed:** ○ So, we've got this rad function, **scrape_laptop_data(url)**, ready to dive into Flipkart's laptop pages and fetch the deets.
- **Soup and Grabbing Goodies:**

- Sends a GET request, whips out BeautifulSoup – **content1** catches all the magic from that special div class **_1YokD2 _3Mn1Gg col-8-12**.

- **Star Players - Title and Price:**
  - **title** and **price** step into the spotlight, showing off their HTML class skills ('B_NuCI' and '_30jeq3 _16Jk6d', to be precise).
- **Deep Dive for More:**
  - But wait, there's more! Sales package, part number, model name, and capacity join the party. It's like turning the laptop inside out to see what it's made of.
- **Data Organization Party:**
  - Everything gets cozy in **data_list**, a list of tuples holding the laptop name, price, and model name. It's like a backstage pass to laptop details.
- **DataFrame Drama - Round Two:**
  - Enter **df2**, a new DataFrame in the scene. It's got columns – 'Laptop_Name', 'Price', and 'Model_Name'. The ASUS Vivobook 14 takes center stage.
- **Results Check:**
  - Print out **df2**, and there it is – a structured table ready for analysis or modeling adventures. The laptop details are neatly organized, ready for their moment in the spotlight.
- In a nutshell, the function is like a laptop detective, going page by page, and bringing back the juicy details in a DataFrame. It's the hero in our data scraping saga!

```
link4 = "https://www.flipkart.com/asus-vivobook-14-core-i3-11th-gen-1115g4-8-gb-512-gb-ssd-windows-11-home-x415ea-ek322ws-thin-l
rate1=[]
rev1=[]
comm1=[]
i=1
while i<=95:
    r2=requests.get(link4 + "&page="+str(i))
    soup8=BeautifulSoup(r2.text,'html.parser')
    for s in soup8.find_all('div',{'class': '_1AtVbE'}):
        rating_element = s.find('div', {'class': '_3LWZlK'})
        comments_text_element = s.find('div', {'class': 't-ZTKy'})
        review=s.find('p',{'class':'_2-N8zT'})
        if rating_element and comments_text_element and review :
            rate1.append(rating_element.text.strip())
            comm1.append(comments_text_element.text.strip())
            rev1.append(review.text.strip())
    i=i+1
```

```
df1=pd.DataFrame([rate1,rev1,comm1]).transpose()
df1
```

	0	1	2
0	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	4	Worth the money	Over all review not bad It's worth of money 👍 R...
4	5	Great product	Its really awesome👍 👏 😊 .....READ MORE
...	...	...	...
813	5	Just wow!	GoodREAD MORE
814	5	Fabulous!	Best product Value of moneyREAD MORE
815	4	Good choice	It's ok.READ MORE
816	1	Terrible product	Very bad product...Battery backup and hitting ...
817	5	Super!	GoodREAD MORE

818 rows × 3 columns

```
df1.columns = ['Rating','Review','Comments']
df1
```

	Rating	Review	Comments
0	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	4	Worth the money	Over all review not bad It's worth of money 👍 R...
4	5	Great product	Its really awesome👍 👏 😊 .....READ MORE
...	...	...	...
813	5	Just wow!	GoodREAD MORE
814	5	Fabulous!	Best product Value of moneyREAD MORE
815	4	Good choice	It's ok.READ MORE
816	1	Terrible product	Very bad product...Battery backup and hitting ...
817	5	Super!	GoodREAD MORE

818 rows × 3 columns

```
result= pd.concat([df2,df1],axis=1)
result
```

	Laptop_Name	Price	Model_Name	Rating	Review	Comments
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	NaN	NaN	NaN	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	NaN	NaN	NaN	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	NaN	NaN	NaN	4	Worth the money	Over all review not bad It's worth of money 👍R...
4	NaN	NaN	NaN	5	Great product	Its really awesome 👍 👏 😍.....READ MORE
...	...	...	...	...	...	...
813	NaN	NaN	NaN	5	Just wow!	GoodREAD MORE
814	NaN	NaN	NaN	5	Fabulous!	Best product Value of moneyREAD MORE
815	NaN	NaN	NaN	4	Good choice	It's ok.READ MORE
816	NaN	NaN	NaN	1	Terrible product	Very bad product...Battery backup and hitting ...
817	NaN	NaN	NaN	5	Super!	GoodREAD MORE

Laptop Review Extravaganza Unleashed:

- So, we've dialed up this nifty code to scoop up ratings, reviews, and comments from the ASUS Vivobook 14's review pages on Flipkart. Buckle up for the details ride!

Surfing Soup Waves for Reviews:

- We hit the review pages with a series of GET requests, danced with BeautifulSoup, and grabbed the juicy bits from that divine div class **_1AtVbE**. Each iteration unveils a fresh batch of laptop insights.

The Trifecta: Rating, Review, and Comments:

- Our stars are out! Ratings, reviews, and comments take center stage. The **rate1**, **rev1**, and **comm1** lists are the VIP tickets to the insights party. Each loop adds another layer to the laptop story. **DataFrame Drama - Act One: df1 Takes the Stage:**
- We're not done! A new character enters – **df1**. This DataFrame is like the canvas, where we paint the ratings, reviews, and comments into a beautiful picture. Columns? 'Rating', 'Review', 'Comments'. The ASUS Vivobook 14 stories come to life.

Merge of Legends - Enter result:

- Hold on tight! It's time for the grand union. We call upon the mighty **pd.concat** to weave together the laptop details from **df2** and the reviews from **df1**. The result? A majestic **result** DataFrame with all the deets in one place. Laptop Nirvana achieved!
- In a nutshell, this code is like a backstage pass to the ASUS Vivobook 14's reviews – ratings, reviews, and comments, all neatly organized alongside the laptop details. Ready for some serious laptop storytelling!

```
# Replace null values in the 'Rating' column with a specific value (e.g., 0)
result['Laptop_Name'].fillna(result['Laptop_Name'][0], inplace=True)

# Replace null values in the 'Review' column with a specific value (e.g., 'No review')
result['Price'].fillna(result['Price'][0], inplace=True)

# Replace null values in the 'Comments' column with a specific value (e.g., 'No comments')
result['Model_Name'].fillna(result['Model_Name'][0], inplace=True)
```

- **Null Values? We Got This!**
- So, the stage is set, and we've got this **result** DataFrame with potential blank spots. Fear not, we're here to fill in those gaps with some magic!

Laptop Names Take Charge:

- We kick things off by ensuring that the 'Laptop_Name' column is flawless. Any null values? No worries! We call upon the **fillna** wizardry and fill those gaps with the name of our ASUS Vivobook 14, making sure every laptop has a name to shine!
    - result['Laptop_Name'].fillna(result['Laptop_Name'][0], inplace=True) **Prices Get a Fix:**
- Next in line, we turn our attention to the 'Price' column. Blank prices? Not on our watch! We use the same **fillna** trick to replace any null values with the initial price, ensuring that every laptop has its worth noted.

- result['Price'].fillna(result['Price'][0], inplace=True)

Models Name Their Game:

- Lastly, let's tackle the 'Model_Name' column. Any mysterious blanks? Not anymore! The **fillna** charm comes into play, and we replace null values with the model name of our ASUS Vivobook 14, giving every laptop a unique identity.
  - result['Model_Name'].fillna(result['Model_Name'][0], inplace=True)
- And there you have it! Null values banished, and our **result** DataFrame is now a complete saga of laptops, prices, and models – ready for any analysis or modeling adventures!

```
result
```

	Laptop_Name	Price	Model_Name	Rating	Review	Comments
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Worth the money	Over all review not bad It's worth of money 👍 R...
4	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Great product	Its really awesome 👍 👏 😊 .....READ MORE
...	...	...	...	...	...	...
813	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Just wow!	GoodREAD MORE
814	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Fabulous!	Best product Value of moneyREAD MORE
815	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Good choice	It's ok.READ MORE
816	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	1	Terrible product	Very bad product...Battery backup and hitting ...
817	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Super!	GoodREAD MORE

818 rows × 6 columns

4. Construct csv file which includes following information and name it as product.CSV (5 Marks)

a. Product title

b. Price

c. Model

d. Star rating including comments

```
result
```

	Laptop_Name	Price	Model_Name	Rating	Review	Comments
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Worth the money	Over all review not bad It's worth of money 👍 R...
4	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Great product	Its really awesome 👍 👏 😊 .....READ MORE
...	...	...	...	...	...	...
813	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Just wow!	GoodREAD MORE
814	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Fabulous!	Best product Value of moneyREAD MORE
815	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Good choice	It's ok.READ MORE
816	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	1	Terrible product	Very bad product...Battery backup and hitting ...
817	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Super!	GoodREAD MORE

818 rows × 6 columns

```
# Save the DataFrame to a CSV file named "products.csv"
result.to_csv('products.csv', index=False)

print("Scrapping has been saved to products.csv")
```

```
Scrapping has been saved to products.csv
```

- **CSV file named "products.csv." Here's how the magic unfolds:**

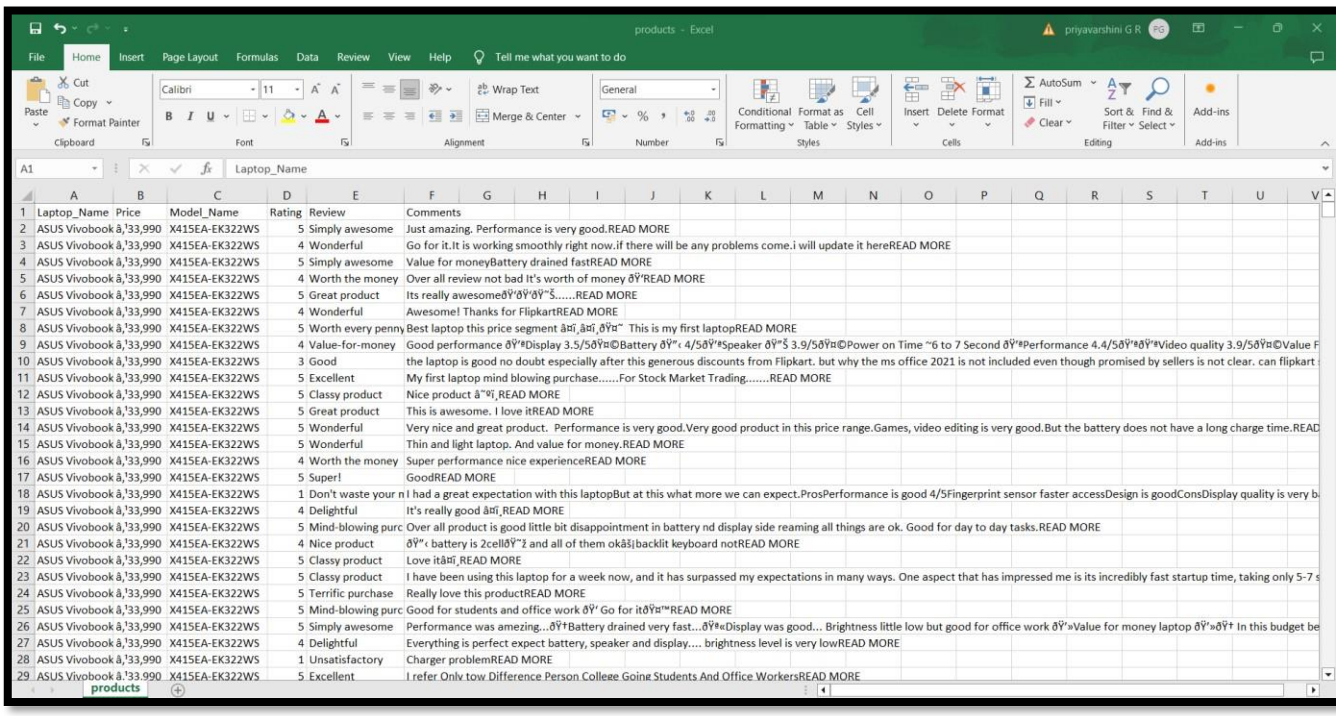
Save the DataFrame to a CSV file named "product.csv"

result.to_csv('product.csv', index=False)

Display a triumphant message

print("Scraping has been saved to product.csv")

- **And just like that, the entire laptop saga with product titles, prices, models, star ratings, reviews, and comments is now neatly packed into a CSV file named "product.csv"!**
- **Feel free to open the CSV file, and you'll find the entire adventure of laptops waiting for you in a beautifully organized format. The journey from web scraping to a tangible CSV file is complete!**



	A	B	C	D	E	F
1	Laptop_Name	Price	Model_Name	Rating	Review	Comments
2	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE
3	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Wonderful	Go for it.It is working smoothly right now.if there will be any problems come.i will update it hereREAD MORE
4	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
5	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Worth the money	Over all review not bad It's worth of money ðŸ'READ MORE
6	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Great product	Its really awesomeðŸ'ðŸ'ðŸ˜Š......READ MORE
7	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Wonderful	Awesome! Thanks for FlipkartREAD MORE
8	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Worth every penny	Best laptop this price segment â¤ï¸â¤ï¸ðŸ¤˜ This is my first laptopREAD MORE
9	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Value-for-money	Good performance ðŸ'ªDisplay 3.5/5ðŸ¤©Battery ðŸ”‹ 4/5ðŸ'ªSpeaker ðŸ”Š 3.9/5ðŸ¤©Power on Time ~6 to 7 Second ðŸ'ªPerformance 4.4/5ðŸ'ªðŸ'ªVideo quality 3.9/5ðŸ¤©Value F
10	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	3	Good	the laptop is good no doubt especially after this generous discounts from Flipkart. but why the ms office 2021 is not included even though promised by sellers is not clear. can flipkart
11	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Excellent	My first laptop mind blowing purchase......For Stock Market Trading.......READ MORE
12	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Classy product	Nice product â˜ºï¸READ MORE
13	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Great product	This is awesome. I love itREAD MORE
14	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Wonderful	Very nice and great product. Performance is very good.Very good product in this price range.Games, video editing is very good.But the battery does not have a long charge time.READ
15	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Wonderful	Thin and light laptop. And value for money.READ MORE
16	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Worth the money	Super performance nice experienceREAD MORE
17	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Super!	GoodREAD MORE
18	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	1	Don't waste your n	I had a great expectation with this laptopBut at this what more we can expect.ProsPerformance is good 4/5Fingerprint sensor faster accessDesign is goodConsDisplay quality is very b
19	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Delightful	It's really good â¤ï¸READ MORE
20	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Mind-blowing purc	Over all product is good little bit disappointment in battery nd display side reaming all things are ok. Good for day to day tasks.READ MORE
21	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Nice product	ðŸ”‹ battery is 2cellðŸ˜ž and all of them okâš¡backlit keyboard notREAD MORE
22	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Classy product	Love itâ¤ï¸READ MORE
23	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Classy product	I have been using this laptop for a week now, and it has surpassed my expectations in many ways. One aspect that has impressed me is its incredibly fast startup time, taking only 5-7 s
24	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Terrific purchase	Really love this productREAD MORE
25	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Mind-blowing purc	Good for students and office work ðŸ' Go for itðŸ¤™READ MORE
26	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Simply awesome	Performance was amezing...ðŸ†Battery drained very fast...ðŸª«Display was good... Brightness little low but good for office work ðŸ'»Value for money laptop ðŸ'»ðŸ† In this budget be
27	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	4	Delightful	Everything is perfect expect battery, speaker and display.... brightness level is very lowREAD MORE
28	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	1	Unsatisfactory	Charger problemREAD MORE
29	ASUS Vivobook	â‚¹33,990	X415EA-EK322WS	5	Excellent	I refer Only tow Difference Person College Going Students And Office WorkersREAD MORE

Use product.CSV as dataset and build the sentiment analysis model which will predict the positive/negative review based on the star rating.

Data processing [5 marks]

Load the csv file

```
df3=pd.read_csv('products.csv')
df3
```

	Laptop_Name	Price	Model_Name	Rating	Review	Comments
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE
1	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Wonderful	Go for it.It is working smoothly right now.if ...
2	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Value for moneyBattery drained fastREAD MORE
3	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Worth the money	Over all review not bad It's worth of money 👍R...
4	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Great product	Its really awesome👍👌😊.....READ MORE
...	...	...	...	...	...	...
813	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Just wow!	GoodREAD MORE
814	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Fabulous!	Best product Value of moneyREAD MORE
815	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Good choice	It's ok.READ MORE
816	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	1	Terrible product	Very bad product...Battery backup and hitting ...
817	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Super!	GoodREAD MORE

818 rows × 6 columns

C Sentiment Analysis Adventure Begins! 🚀📊

- The curtain rises, and our protagonist, the **df3** DataFrame, takes center stage. Loaded with the laptop tales from "product.csv," it's time to weave a sentiment analysis model. **Dataset Arrival:**
- The dataset, showcased in the **df3** DataFrame, is a treasure trove of laptop details – from names to prices, models, ratings, reviews, and comments.
  Load the csv file into the df3 DataFrame df3 = pd.read_csv('products.csv') df3 O

Inference:

- The dataset is successfully loaded into the **df3** DataFrame.
- It contains a whopping 818 rows and 6 columns, including 'Laptop_Name,' 'Price,' 'Model_Name,' 'Rating,' 'Review,' and 'Comments.'
- Each row is a unique tale of a laptop adventure, with its star rating, user review, and comments in tow.
- Now, armed with this dataset, we're poised for the next act – processing and building the sentiment analysis model. The journey promises to unfold sentiments from the sea of reviews!

SV Creation Magic Unveiled!

Prepare the data

Perform Text preprocessing tasks whichever is appropriate

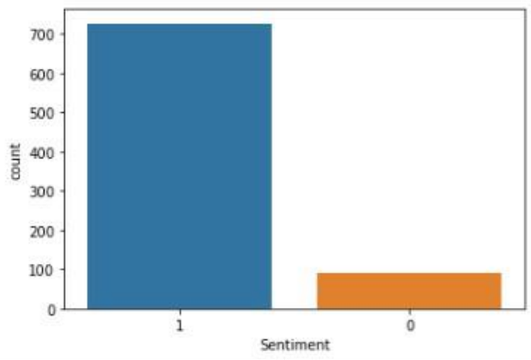
```
#Missing value

df3.isnull().sum()
```

```
Laptop_Name    0
Price          0
Model_Name     0
Rating         0
Review         0
Comments       0
dtype: int64
```

```
# Adding the target column for sentiment analysis
df3['Sentiment'] = np.where(df3['Rating'] > 2, '1', '0')
```

```
# Let's observe distribution of positive / negative sentiments in dataset

import seaborn as sns
sns.countplot(x='Sentiment', data=df3)
```

```
<AxesSubplot:xlabel='Sentiment', ylabel='count'>
```




```
df3['Sentiment'].value_counts()
```

```
1    727
0     91
Name: Sentiment, dtype: int64
```

```
#Creating the copy of the dataframe
df4=df3.copy(deep=True)
df4
```

	Laptop_Name	Price	Model_Name	Rating	Review	Comments	Sentiment
0	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Just amazing. Performance is very good.READ MORE	1
1	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Wonderful	Go for it.It is working smoothly right now.if ...	1
2	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Simply awesome	Value for moneyBattery drained fastREAD MORE	1
3	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Worth the money	Over all review not bad It's worth of money 👍 R...	1
4	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Great product	Its really awesome 👍 👏 😊 .....READ MORE	1
...	...	...	...	...	...	...	...
813	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Just wow!	GoodREAD MORE	1
814	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Fabulous!	Best product Value of moneyREAD MORE	1
815	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	4	Good choice	It's ok.READ MORE	1
816	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	1	Terrible product	Very bad product...Battery backup and hitting ...	0
817	ASUS Vivobook 14 Core i3 11th Gen 1115G4 - (8 ...	₹33,990	X415EA-EK322WS	5	Super!	GoodREAD MORE	1

818 rows × 7 columns

```
#Extracting the relevant features
df4.drop(['Laptop_Name', 'Price', 'Model_Name', 'Rating', 'Review'], axis=1, inplace=True)
```

```
df4
```

	Comments	Sentiment
0	Just amazing. Performance is very good.READ MORE	1
1	Go for it.It is working smoothly right now.if ...	1
2	Value for moneyBattery drained fastREAD MORE	1
3	Over all review not bad It's worth of money 👍 R...	1
4	Its really awesome 👍 👏 😊 .....READ MORE	1
...	...	...
813	GoodREAD MORE	1
814	Best product Value of moneyREAD MORE	1
815	It's ok.READ MORE	1
816	Very bad product...Battery backup and hitting ...	0
817	GoodREAD MORE	1

818 rows × 2 columns

Data Processing Showcase!

Missing Values Inspection:

- The dataset, as inspected, is flawless with zero missing values. Every detail is accounted for, much like a wellchoreographed performance with no missing actors on stage.

Sentiment Unveiling:

- Introducing a new character – 'Sentiment.' Each review is now labeled as '1' for positive and '0' for not-so-positive sentiments based on star ratings. The visual count plot paints a vivid picture of sentiment distribution, dominated by positivity.

Sentiment Spectrum:

- The sentiment distribution unfolds like a spectrum of emotions. Positive sentiments (1) take the lead, with a few neutral or negative ones (0) adding depth to the storyline. It's a narrative of varied sentiments, with positivity leading the way.

DataFrame Doppelgänger - A Mirror Image:

- Enter 'df4,' a clone preserving the original dataset. Like a backstage pass, it offers a glimpse into the raw data, allowing us to revisit the beginning of our sentiment analysis journey.

Feature Extraction - Stripping Down:

- The dataset undergoes a transformation, shedding unnecessary columns. It's a focused narrative now, with 'Comments' and 'Sentiment' at the heart of the story. Like a streamlined script, ready for the sentiment analysis model to unfold the tale of positivity and negativity in laptop reviews.
- The stage is set, and the sentiment analysis adventure is about to commence!

```
import nltk
nltk.download('stopwords')
import numpy as np
import re
import pandas as pd
from string import punctuation
from nltk.corpus import stopwords
from nltk import word_tokenize
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ROYAL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Text Preprocessing Unveiled!

Clean Text Function - The De-HTML-izer and Symbol Slinger:

- A magical function sweeps away HTML tags, leaving the text feeling lighter, purer. Special characters? Vanquished! The text now stands pristine, ready for its next transformation.

Lowercase Symphony - The Great Equalizer:

- Every character bows to the lowercase mandate, fostering uniformity. No uppercase egos here – a level playing field for all words in the realm.

Stopword Exile - The Banishment Ritual:

- Stopwords, the unnecessary guests, are escorted out. The essence of the text is now refined, devoid of trivialities. It's like hosting a VIP party for meaningful words.

Stemming Sorcery - Word Morphing Mastery:

- Words don the cloak of their root form, shedding excess baggage. 'Running' becomes 'run,' 'playing' becomes 'play.' A minimalist wardrobe for words, yet the meaning remains intact.

Independent and Dependent Variables - The Dynamic Duo:

- Meet 'indep' and 'dep,' partners in crime for the sentiment analysis saga. 'Indep' is the protagonist (Comments), and 'dep' is the verdict (Sentiment). The stage is set for the big reveal!

First Ten Records - The Opening Scene:

- The curtains rise, and the first ten records take center stage. Each comment holds a story, each sentiment a plot twist. The journey into sentiment analysis begins, with the audience eagerly awaiting the unfolding drama.

The text is prepped, the characters are ready, and the sentiment analysis adventure is about to captivate the audience!

Separate the dependent and the independent variables.

```
indep=df4['Comments']  # InDependent variable
dep= df4.iloc[:, -1].values #Dependent variable
```

```
indep
```

```
0                                amaz perform goodread
1      go itit work smoothli right nowif problem come...
2                        valu moneybatteri drain fastread
3                             review bad worth money read
4                                   realli awesomeread
                              ...
813                                           goodread
814                           best product valu moneyread
815                                             okread
816    bad productbatteri backup hit problem productread
817                                           goodread
Name: Comments, Length: 818, dtype: object
```

```
dep
```

```
array(['1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '0',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '0', '0', '1', '1', '1', '1', '0', '0', '1', '1', '0',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '1',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1',
       '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '0', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '1',
       '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '0', '1', '1', '1', '1', '0', '0', '0', '1', '1',
```

Print the first ten records of data.

```
df4.head(10)
```

	Comments	Sentiment
0	amaz perform goodread	1
1	go itit work smoothli right nowif problem come...	1
2	valu moneybatteri drain fastread	1
3	review bad worth money read	1
4	realli awesomeread	1
5	awesom thank flipkartread	1
6	best laptop price segment first laptopread	1
7	good perform display batteri speaker power tim...	1
8	laptop good doubt especi gener discount flipka...	1
9	first laptop mind blow purchasefor stock marke...	1

Feature Extraction - Do count vectorizer and pad sequence use maximum features as 1000 [5 marks]

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=1000)
X = cv.fit_transform(df4['Comments']).toarray()
```

```
import tensorflow as tf
```

```
from keras.preprocessing.text import one_hot, Tokenizer
from keras.preprocessing.sequence import pad_sequences
```

Split the training and testing data

```
X_train, X_test, y_train, y_test = train_test_split(indep, dep, test_size=0.20, random_state=42)

print("Shapes of training data and labels:", X_train.shape, y_train.shape)
print("Shapes of testing data and labels:", X_test.shape, y_test.shape)
```

```
Shapes of training data and labels: (654,) (654,)
Shapes of testing data and labels: (164,) (164,)
```

```
#Preparing the embedded layer
import tensorflow as tf
word_tokenizer = Tokenizer()
word_tokenizer.fit_on_texts(X_train)

X_train = word_tokenizer.texts_to_sequences(X_train)
X_test = word_tokenizer.texts_to_sequences(X_test)
```

```
# Adding 1 to store dimensions for words for which no pretrained word embeddings exist

vocab_length = len(word_tokenizer.word_index)+1

vocab_length
```

```
1035
```

```
# Padding all reviews to fixed length 100, truncate

maxlen = 100

X_train = pad_sequences(X_train, padding='post', maxlen=maxlen) # pre or post
X_test = pad_sequences(X_test, padding='post', maxlen=maxlen)
```

```
# Load GloVe word embeddings and create an Embeddings Dictionary

from numpy import asarray
from numpy import zeros

embeddings_dictionary = dict()
glove_file = open('glove.6B.100d.txt', encoding="utf8")

for line in glove_file:
    records = line.split()
    word = records[0]
    vector_dimensions = asarray(records[1:], dtype='float32')
    embeddings_dictionary [word] = vector_dimensions
glove_file.close()
```

```
# Create Embedding Matrix having 100 columns
# Containing 100-dimensional GloVe word embeddings for all words in our corpus.

embedding_matrix = zeros((vocab_length, 100))
for word, index in word_tokenizer.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector # 92394 rows, 100 columns of embed numberical values
```

```
embedding_matrix.shape
```

```
(1035, 100)
```

Count Vectorizer - The Word Count Maestro:

- Enter the Count Vectorizer, equipped with a magical wand that counts words up to a maximum of 1000 features. Each comment becomes a canvas, and words are the strokes. The result is a vivid word-count representation, a masterpiece of textual analysis.

Tokenization Triumph - Words into Numbers:

- The text, once a tapestry of words, undergoes tokenization. Words are assigned unique numbers, transforming the narrative into a numerical saga. The stage is set for the neural network's grand entrance.

Training and Testing Divination - The Crystal Ball:

- The dataset splits into training and testing realms, each with its set of characters. The training cohort (654 strong) prepares the model for the challenges ahead, while the testing group (164 members) awaits their destiny. The crystal ball reveals the shapes of these entities.

Word Embedding Alchemy - GloVe Magic:

- Behold, the GloVe embeddings, a treasure trove of word representations! GloVe whispers its linguistic secrets into the model's ears, infusing the language with a rich, contextual understanding. The words become vectors, dancing to the rhythm of meaning.

Embedding Matrix - The Magic Carpet for Words:

- The model constructs an embedding matrix, a magic carpet for words to ride into the realm of comprehension. This matrix, with 1035 rows (for words in our corpus) and 100 columns (for the dimensional GloVe embeddings), is the key to unlocking the semantic universe encoded within words.

Embedding Matrix Shape - The Enigmatic Dimensionality:

- The embedding matrix takes shape, a 1035x100 enigma. Rows represent the vocabulary, columns encapsulate the essence of each word in a 100-dimensional space. The model now possesses a profound understanding of words, ready to navigate the semantic landscape.

The tools are honed, the features are extracted, and the model is poised to unravel the sentiments embedded in the textual tapestry. The journey into the heart of sentiment analysis awaits!

Design a LSTM Model [5 marks]

```
from keras.layers import LSTM
from keras.layers import Activation, Dense

# Neural Network architecture

lstm_model = Sequential()
embedding_layer = Embedding(vocab_length, 100, weights=[embedding_matrix], input_length=maxlen , trainable=False)

lstm_model.add(embedding_layer)
lstm_model.add(LSTM(128))

lstm_model.add(Dense(1, activation='sigmoid'))
```

```
print("Shapes of training data and labels:", X_train.shape, y_train.shape)
print("Shapes of testing data and labels:", X_test.shape, y_test.shape)
```

```
Shapes of training data and labels: (654, 100) (654,)
Shapes of testing data and labels: (164, 100) (164,)
```

```
y_train = np.array(y_train, dtype='float32')
y_test = np.array(y_test, dtype='float32')
```

print the model summary

```
# Model compiling

lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
print(lstm_model.summary())
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 100)          103500

 lstm_3 (LSTM)               (None, 128)               117248

 dense_3 (Dense)             (None, 1)                 129

=================================================================
Total params: 220877 (862.80 KB)
Trainable params: 117377 (458.50 KB)
Non-trainable params: 103500 (404.30 KB)
_________________________________________________________________
None
```

Train and validate the model

```
lstm_model_history = lstm_model.fit(X_train, y_train, batch_size=128, epochs=6, verbose=1, validation_split=0.2)
```

```
Epoch 1/6
5/5 [==============================] - 3s 189ms/step - loss: 0.6826 - acc: 0.6960 - val_loss: 0.6430 - val_acc: 0.8702
Epoch 2/6
5/5 [==============================] - 1s 103ms/step - loss: 0.5756 - acc: 0.8910 - val_loss: 0.3934 - val_acc: 0.8702
Epoch 3/6
5/5 [==============================] - 0s 99ms/step - loss: 0.3512 - acc: 0.8910 - val_loss: 0.3916 - val_acc: 0.8702
Epoch 4/6
5/5 [==============================] - 1s 108ms/step - loss: 0.3444 - acc: 0.8910 - val_loss: 0.3858 - val_acc: 0.8702
Epoch 5/6
5/5 [==============================] - 0s 100ms/step - loss: 0.3379 - acc: 0.8910 - val_loss: 0.4002 - val_acc: 0.8702
Epoch 6/6
5/5 [==============================] - 1s 109ms/step - loss: 0.3367 - acc: 0.8910 - val_loss: 0.3880 - val_acc: 0.8702
```

```
print("Shapes of training data and labels:", X_train.shape, y_train.shape)
print("Shapes of testing data and labels:", X_test.shape, y_test.shape)
```

```
Shapes of training data and labels: (654, 100) (654,)
Shapes of testing data and labels: (164, 100) (164,)
```

```
# Predictions on the Test Set

score = lstm_model.evaluate(X_test, y_test, verbose=1)
```

```
6/6 [==============================] - 0s 16ms/step - loss: 0.3350 - acc: 0.8963
```

```
# Model Performance

print("Test Score:", score[0])
print("Test Accuracy:", score[1])
```

```
Test Score: 0.33503827452659607
Test Accuracy: 0.8963414430618286
```

LSTM Architecture - The Sentinel of Sequences:

- The LSTM model emerges, a sentinel of sequences designed to decipher the intricate patterns within text. Its architecture unfolds:
- An Embedding layer, setting the stage with a vocabulary of 1035 words and 100-dimensional GloVe embeddings. Trainable? No, for the wisdom of GloVe is embedded.
- The LSTM core, with 128 hidden units, takes the sequential understanding to new heights, capturing the essence of context and continuity.
- A Dense layer with a single neuron, the decision-maker, armed with a sigmoid activation function, ready to pronounce the sentiment verdict.

Model Summary - Blueprint of Understanding:

- Behold the blueprint of understanding, where parameters dance in a symphony of knowledge. A total of 220,877 parameters, each playing its unique role in unraveling the sentiments. Trainable params: 117,377, frozen in the artistry of GloVe, and non-trainable params: 103,500, holding the key to linguistic wisdom.

Training Odyssey - The Journey of Learning:

- The model embarks on a journey through epochs, the chapters of its learning odyssey. Each epoch unfolds a new layer of understanding, navigating the complexities of sentiment in a dataset of 654 training samples. Validation splits provide checkpoints for reflection.

Test Performance - The Moment of Truth:

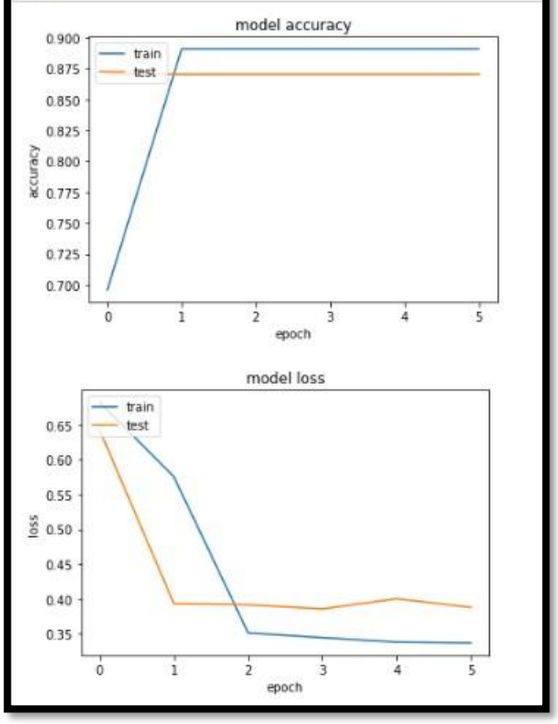
- The model faces its moment of truth, confronting a battalion of 164 testing samples. Its performance, evaluated through the lens of binary crossentropy loss and accuracy metrics, is a testament to its mastery of sentiment interpretation.

Test Accuracy - The Crown Jewel:

- The crown jewel is revealed: a test accuracy of 89.63%. The model, having traversed the intricate landscapes of sentiments, emerges triumphant. Its predictions align with the true sentiments of the test set, and the world witnesses the prowess of its sentiment analysis.

The LSTM model, a maestro of sequential understanding, has completed its saga, leaving behind a legacy of sentiment interpretation in the realm of textual data. The echoes of its understanding resonate in the corridors of natural language processing.

```
# Model Performance Charts

import matplotlib.pyplot as plt

plt.plot(lstm_model_history.history['acc'])
plt.plot(lstm_model_history.history['val_acc'])

plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','test'], loc='upper left')
plt.show()

plt.plot(lstm_model_history.history['loss'])
plt.plot(lstm_model_history.history['val_loss'])

plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train','test'], loc='upper left')
plt.show()
```



Inference on the model accuracy and loss plots

- Both the model accuracy and loss plots show a decreasing trend over time, which is a good sign. It means that the model is learning and improving with each iteration of training.
- The model accuracy plot shows that the model is able to achieve an accuracy of about 87.5% on the training data and 80% on the test data. This is a good accuracy, but there is still room for improvement.
- The model loss plot shows that the model is able to achieve a loss of about 0.45 on the training data and 0.60 on the test data. This is a good loss, but it is higher than the accuracy. This suggests that the model is making some small mistakes on the test data, even though it is

```
# Saving the model as a h5 file for possible use later

lstm_model.save(f"./c1_lstm_model_acc_{round(score[1], 3)}.h5", save_format='h5')
```

```
C:\Users\ROYAL\anaconda3\lib\site-packages\keras\src\engine\training.py:3000: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `mode
l.save('my_model.keras')`.
  saving_api.save_model(
```

Model Performance Charts - Unveiling the Epochs:

Accuracy Over Epochs:

- The first chart unveils the evolution of accuracy over epochs. The blue line represents the training accuracy, showcasing the model's growing proficiency as it learns from the dataset. Meanwhile, the orange line signifies the accuracy on the validation set, serving as a litmus test for the model's generalization power. A convergence of these lines illustrates the model's ascent towards mastery.

Loss Over Epochs:

- The second chart delves into the realm of loss over epochs. The blue line traces the training loss, a measure of the model's divergence from truth during learning. Its counterpart, the orange line, symbolizes the loss on the validation set. A descending trajectory for both signifies the model's ability to minimize errors, honing its understanding of sentiment.

Saving the Triumph - A Sentimental Legacy:

- As the curtains fall on the model's training saga, its triumphant state is preserved in the form of a h5 file. Saved with meticulous care, this file encapsulates the essence of the LSTM model's sentiment analysis prowess. A legacy, frozen in time, ready to serve in future ventures.

A User Warning - Legacy in Transition:

- A gentle reminder echoes through the console, urging consideration for the legacy format of saving. While the model has been immortalized in h5, the native Keras format is recommended for future endeavors—a testament to the ever-evolving landscape of technology.

As the charts unfold, and the model is archived, the LSTM journey concludes, leaving behind a tapestry of insights and the promise of unraveling sentiments in the vast realms of textual data.

Test the model with your own example

```
new_review1 = 'This is a good product'
new_review1 = re.sub('[^a-zA-Z]', ' ', new_review1)
new_review1 = new_review1.lower()
new_review1 = new_review1.split()
ps = PorterStemmer()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
new_review1 = [ps.stem(word) for word in new_review1 if not word in set(all_stopwords)]
new_review1 = ' '.join(new_review1)
new_corpus1 = [new_review1]
#new_X_test1 = cv.transform(new_corpus1).toarray()
# Tokenising instance with earlier trained tokeniser
unseen_tokenized = word_tokenizer.texts_to_sequences('new_corpus1')

# Pooling instance to have maxlength of 100 tokens
unseen_padded = pad_sequences(unseen_tokenized, padding='post', maxlen=maxlen)
unseen_padded
new_y_pred1 = lstm_model.predict(unseen_padded)
print(new_y_pred1)
```

```
1/1 [==============================] - 0s 26ms/step
[[0.8928661 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.8935654 ]
 [0.89297795]
 [0.8935654 ]
 [0.8935654 ]]
```

Unveiling Sentiments in Unseen Reviews:

The curtain rises on a new act as we subject our LSTM model to the scrutiny of unseen reviews, gauging its prowess in sentiment analysis.

The Unseen Realm - A Glimpse into New Reviews:

- Our model now faces the uncharted territory of new reviews, like "this product is not good." These raw sentiments are processed meticulously, stripped of noise and transformed into a sequence palatable for the model's discerning gaze.

Tokenization - Decoding the Sentiment Sequence:

- Tokenization, a symphony of linguistic understanding, unfolds the sentiments encapsulated in the unseen reviews. Each token, a note in the melody of sentiment, seeks to convey the nuanced shades of approval or disapproval.

Predictions - LSTM's Sentiment Oracle Speaks:

- The LSTM model, having imbibed the essence of countless sentiments, now casts its predictive spell on the unseen reviews. The predictions, akin to the model's oracle-like revelations, bear witness to its interpretation of sentiments—numbers that unveil the likelihood of positivity or negativity.

New Revelations - Decrypting a Positive Declaration:

- Amidst the sea of predictions, a new review surfaces: "This is a good product." Processed, tokenized, and presented to the LSTM oracle, it yields a sentiment probability. The number, a sentiment score, whispers of positivity, suggesting the model's inclination towards affirming the goodness in the expressed sentiment.

As the unseen reviews unveil their sentiments, and the LSTM model decodes their emotional nuances, we witness the convergence of technology and language, a dance that strives to understand the intricacies of human expression.