Assignment C1

import requests

import bs4

headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'

}

request1 = requests.get('https://www.flipkart.com/poco-c31-royal-blue-64-gb/p/itm19effae969b86', headers=headers)

request1.content

#Fetching reviews/comments

reviews = soup.findAll('div',{'class' : 't-ZTKy'});

for review in reviews:

print(review.get\_text() + "\n\n")

#Average overall rating

ratings = soup.find('div',{'class':'ipqd2A'}).get\_text();

print(ratings)

#Fetching tags

tags = soup.find('span',{'class':'\_6EBuvT'}).get\_text();

tags

#Fetching customer names

customer\_name = soup.findAll('p',{'class':'\_2NsDsF AwS1CA'});

for cust\_name in customer\_name:

print(cust\_name.get\_text() + "\n")

#specification

specifications = soup.find\_all('div', {'class': '\_7eSDEz'})

for specification in specifications:

print(specification.get\_text())

answer = specification.find('td', {'class':'\_7eSDEz'}) # Replace 'your-class-name-here' with the actual class name

print(answer + "\n\n")

Assignment B1

import numpy as np

import pandas as pd

df=pd.read\_csv(r"C:\Users\PC WORLD\Desktop\DSBDA PRACTICALS\dataset\_Facebook.csv",sep=";")

df

df.describe()

df.shape

#subset1

df1=df[['Page total likes','Category','Post Month','Post Weekday']].loc[0:15]

#subset2

df2=df[['Page total likes','Category','Post Month','Post Weekday']].loc[16:30]

#subset3

df3=df[['Page total likes','Category','Post Month','Post Weekday']].loc[31:50]

#merge data

merging=pd.concat([df1,df2,df3])

merging

#sort data

sort\_values=df.sort\_values('Page total likes',ascending=False)

sort\_values

#Transposing data

df.transpose()

#shaping

shaping=df.shape

shaping

#reshaping

pivot\_table=pd.pivot\_table(df,index=['Type','Category'],values='comment')

print(pivot\_table)

Assignment B3

import matplotlib.pyplot as plt

from wordcloud import WordCloud,STOPWORDS

import sys,os

os.chdir(sys.path[0])

text = open(r'C:\Users\PC WORLD\Music\d1.txt', mode="r", encoding="utf-8").read()

stopwords = STOPWORDS

print(len(stopwords))

wc = WordCloud(background\_color="white", stopwords=stopwords, height=600, width=400)

wc.generate(text)

wc.to\_file(r'C:\Users\PC WORLD\Music\wcoutput.png')

Assignment B4

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

plt.rcParams['figure.figsize']=(7,4)

import warnings

warnings.filterwarnings('ignore')

df=pd.read\_csv(r"C:\Users\PC WORLD\Desktop\DSBDA PRACTICALS\dataset\_Facebook.csv",sep=";")

df.head()

df.info()

df[['Lifetime Post Consumers','Type']].groupby(['Type']).median().sort\_values(by="Lifetime Post Consumers",ascending=False)

plt.figure(figsize=(6, 4))

df[['Lifetime Post Consumers', 'Type']].groupby(['Type']).median().sort\_values(by="Lifetime Post Consumers", ascending=False).plot(kind='bar')

plt.xlabel('Type')

plt.ylabel('Median Lifetime Post Consumers')

plt.title('Median Lifetime Post Consumers by Type')

plt.show()

plt.figure(figsize=(6, 4))

df[['Post Hour', 'Type']].groupby(['Type']).median().sort\_values(by="Post Hour", ascending=False).plot(kind='bar')

plt.xlabel('Type')

plt.ylabel('Post Hour ')

plt.title('Post Hour by Type')

plt.show()

fuels=['natural gas','coal','petrol','disel','oil']

data=[20,30,10,20,20]

plt.pie(data,labels=fuels,colors=['#0f0f0f','#e00f0e','#ff00ff','#f0f0f0','#fff000'])

plt.title('chart title')

plt.show()

Assignment B2

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.impute import SimpleImputer

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

df = pd.read\_csv('D:Heart.csv')

# 1) Data Cleaning

# Handle missing values

imputer = SimpleImputer(strategy='mean')

df[['age', 'trestbps', 'chol', 'thalach', 'oldpeak']] = imputer.fit\_transform

df

# Handle duplicates

df.drop\_duplicates(inplace=True)

# 3) Data Transformation

# Feature scaling

scaler = StandardScaler()

df[['age', 'trestbps', 'chol', 'thalach', 'oldpeak']] = scaler.fit\_transform(df[['age', 'trestbps', 'chol', 'thalach', 'oldpeak']])

# 5) Data Model Building

# Split the dataset into features (X) and target variable (y)

actual=np.concatenate((np.ones(45),np.zeros(450),np.ones(5)))

predicted= np.concatenate((np.ones(100),np.zeros(400)))

from sklearn.metrics import ConfusionMatrixDisplay

ConfusionMatrixDisplay.from\_predictions(actual,predicted)

from sklearn.metrics import classification\_report

from sklearn.metrics import accuracy\_score

print(classification\_report(actual,predicted))

accuracy\_score(actual,predicted)