Bahria University,
Karachi Campus

LAB EXPERIMENT NO.
02

LIST OF TASKS

TASK NO	OBJECTIVE
1.	**Develop a Python application to generate data visualizations Scenario: You** are a data analyst working with a large dataset containing various types of data. Your task is to create a Python application that uses the Pandas, Matplotlib, and Seaborn libraries to perform exploratory data analysis and generate interactive visualizations.
2.	**Implement a text summarization model using Transformers Scenario:** As a natural language processing (NLP) researcher, Your task is to utilize the Transformers library in Python to build and train a summarization model. The model should be able to take a long text document as input and generate a concise summary that captures the key information and main ideas.
3.	**Convert images to sketches using OpenCV Scenario:** Your task is to create a Python script that uses the OpenCV library to convert regular images into sketches. The script should allow users to select an image file, apply appropriate filters and transformations to convert it into a sketch-like image, and save the resulting image to disk.
4.	**Build a web scraper using Beautiful Soup Scenario:** Your task is to develop a Python script that uses the Beautiful Soup library to scrape product information from competitor websites. The script should be able to extract data such as product names, descriptions, prices, and images from the target websites and store the data in a structured format (e.g., CSV or JSON) for further analysis.
5.	**Automate WhatsApp messaging using PyWhatKit Scenario:** Your task is to create a Python script that uses the PyWhatKit library to automate the sending of messages and images through WhatsApp. The script should allow users to schedule the sending of messages or images to one or more contacts at specific times or intervals.
6.	**Develop a text-to-speech application using pyttsx3 Scenario:**Your task is to create a Python application that uses the pyttsx3 library to convert text into spoken words. The application should allow users to input text, select voice settings (e.g., language, gender, rate), and generate audio output that can be played or saved to a file.

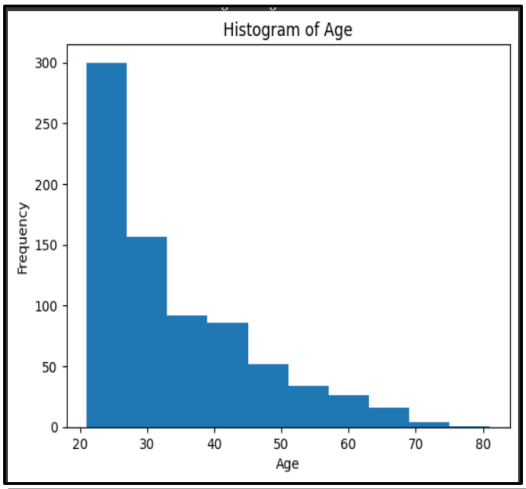
Submitted On:
4/5/2024

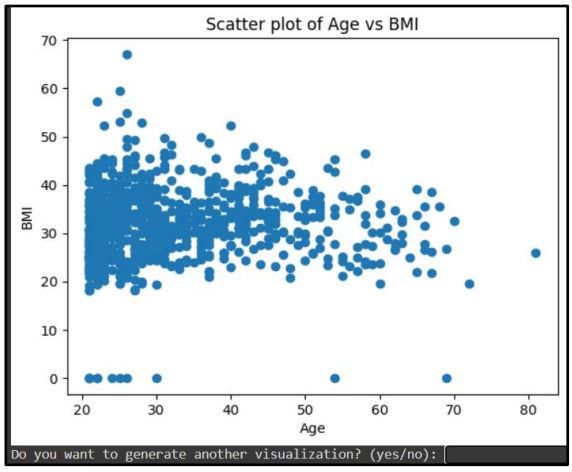
TASK NO 1:Develop a Python application to generate data visualizations Scenario:

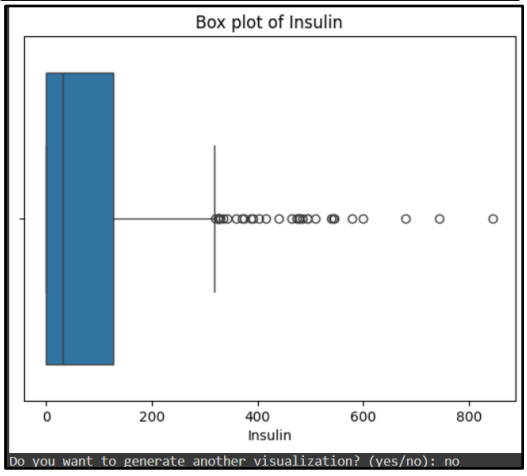
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
def load_dataset(file_path):
    try:
        return pd.read_csv(file_path)
    except FileNotFoundError:
        print("File not found.")
        return None
def explore_data(df):
    print("\nSummary statistics:")
    print(df.describe())
def generate_visualizations(df):
    print("\nAvailable visualizations:")
    print("1. Histogram")
    print("2. Scatter plot")
    print("3. Box plot")
    choice = input("Enter the corresponding number to generate
graph: ")
    if choice == '1':
        column = input("Enter the column name for histogram: ")
        plt.hist(df[column])
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.title("Histogram of " + column)
        plt.show()
    elif choice == '2':
        x_column = input("Enter the column name for x-axis: ")
        y_column = input("Enter the column name for y-axis: ")
        plt.scatter(df[x_column], df[y_column])
        plt.xlabel(x_column)
        plt.ylabel(y_column)
        plt.title("Scatter plot of " + x_column + " vs " + y_column)
        plt.show()
    elif choice == '3':
        column = input("Enter the column name for box plot: ")
        sns.boxplot(x=df[column])
        plt.xlabel(column)
        plt.title("Box plot of " + column)
        plt.show()
    else:
        print("Invalid choice.")
    again = input("Do you want to generate another
visualization? (yes/no): ")
    if again.lower() == 'yes':
        generate_visualizations(df)
def main():
    print("Welcome to Data Visualization App")
    file_path = input("Enter the path to your dataset (CSV
format): ")
    df = load_dataset(file_path)
    if df is not None:
        explore_data(df)
        generate_visualizations(df)
if __name__ == "__main__":
    main()
```

OUTPUT:

```
Welcome to Data Visualization App
Enter the path to your dataset (CSV format): /content/diabetes.csv

Summary statistics:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000

Available visualizations:
1. Histogram
2. Scatter plot
3. Box plot
Enter the corresponding number to generate graph: 1
Enter the column name for histogram: Age
```







TASK NO 2:Implement a text summarization model using Transformers Scenario:

```
from transformers import pipeline

summarizer = pipeline("summarization",model="Falconsai/text_summarization")
try:
  article_path = input("Enter file path:")
except :
  print("Invalid file name!!!")

with open(article_path, "r") as file:
  text = file.read()
print(summarizer(text,max_length=200,min_length=100,do_sample=False))
```

Text File:

file.txt

```
1 Types of Artificial Intelligence
2 First of all, the categorization of Artificial Int
3
4 Type 1: Reactive machines - These machines can rea
5
6 Type 2: Limited memory - These AI systems are capa
7
8 Type 3: Theory of mind - This refers to understand
9
10 Type 4: Self-awareness - This is the highest and m
```

Output:

```
Enter file path:file.txt
[{'summary_text': 'Type 1: Reactive machines - These machines react to situations . A famous example can be
```

TASK NO 3: Convert images to sketches using OpenCV Scenario:

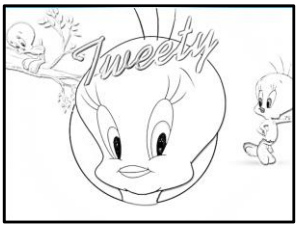
```
import cv2
from google.colab.patches import cv2_imshow
def pencil_sketch(image_path):
    image = cv2.imread(image_path)
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    inverted_gray_image = 255 - gray_image
    blurred_image = cv2.GaussianBlur(inverted_gray_image, (21, 21), 0)
    inverted_blurred_image = 255 - blurred_image
    sketch = cv2.divide(gray_image, inverted_blurred_image, scale=256.0)

    return sketch

input_image_path = input("Enter Image name:")
sketch_image = pencil_sketch(input_image_path)
cv2_imshow(cv2.imread(input_image_path))
cv2_imshow(sketch_image)
cv2.imwrite("pen1.jpg",sketch_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

OUTPUT:

```
••• Enter Image name:tweety.jpg
```

Input Image :

Output Image:



TASK NO 4:

Build a web scraper using Beautiful Soup Scenario:

```
import requests
import requests
from bs4 import BeautifulSoup
import csv
def scrapeWeb(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    scrape = []
    for scr in soup.find_all('fieldset', class_='gra1'):
        name = scr.find('div', class_='firsthomecontent').text.strip()
        scrape.append({'name': name })
    return scrape
def save_to_csv(file, filename):
    with open(filename, 'w', newline='') as csvfile:
        fieldnames = ['name']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        for s in file:
            writer.writerow(s)
urls = ['https://www.javatpoint.com/html-tutorial']
names = []
for url in urls:
  names.extend(scrapeWeb(url))
  save_to_csv(names, random.csv')
```

OUTPUT:

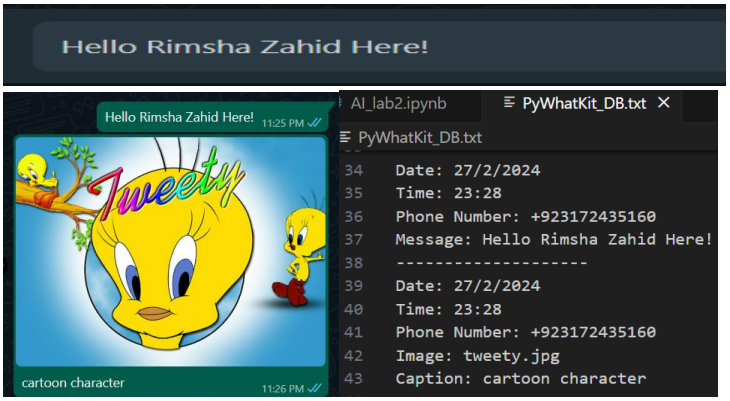
	A
1	name
2	Splunk
3	Aptitude
4	Artificial
5	DBMS

TASK NO 5: Automate WhatsApp messaging using PyWhatKit Scenario:

```
import pywhatkit as kit
from datetime import datetime, timedelta
def send_message(phone_num, message, scheduled_time):
    try:
        kit.sendwhatmsg_instantly(phone_num, message,  )
        print(f"Message scheduled to send at {scheduled_time}")
    except Exception as e:
        print(f"Error occurred: {str(e)}")
def send_image(phone_num, image_path, caption,scheduled_time):
    try:
        kit.sendwhats_image(phone_num, image_path, caption)
        print(f"Image send sucessfully ")
```

LAB NO #2

OUTPUT:




```
    except Exception as e:
        print(f"Error occurred: {str(e)}")
def main():
    phone_number = input("Enter Phone Number")
    message = input("Enter message to send")
    image_path = input("Enter image path")
    caption = input("Enter image caption")
    scheduled_time = datetime.now() + timedelta(seconds=0)
    send_message(phone_number, message, scheduled_time)
    send_image(phone_number, image_path, caption, scheduled_time)
if __name__ == "__main__":
    main()
```

TASK NO 6:Develop a text-to-speech application using pyttsx3 Scenario:

```
import pyttsx3
engine = pyttsx3.init()
def text_audio(text,voice,rate):
    voices = engine.getProperty('voices')
    engine.setProperty('rate', rate)
    engine.setProperty('volume', 0.9)
    engine.getProperty('voices')
    engine.setProperty('voice', voices[voice].id)
    engine.say(text)
    engine.runAndWait()
def download(file_name):
    engine.save_to_file(text, 'audio.mp3')
text = input("Enter text: ")
voice = int(input("Enter 0) Male 1)\t Female: "))
rate = int(input("Enter rate: "))
text_audio(text,voice,rate)
download= input("Do you want to download the audio\n 0)
Download 1) Don't want to download")
if download == 0:
    file_name = input("Enter audio file name: ")
    download(file_name)
```