Bahria University

Karachi Campus



**LAB EXPERIMENT NO.**

**02**

**LIST OF TASKS**

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| 01 | Develop a Python application to generate data visualizations Scenario |
| 02 | Implement a text summarization model using Transformers Scenario |
| 03 | Convert images to sketches using OpenCV Scenario |
| 04 | Build a web scraper using Beautiful Soup Scenario |
| 05 | Automate WhatsApp messaging using PyWhatKit Scenario |
| 06 | Develop a text-to-speech application using pyttsx3 |

Submitted On: 23/02/24

**Task 1 :** Develop a text-to-speech application using pyttsx3

import pyttsx3

def main():

    # Initialize the text-to-speech engine

    engine = pyttsx3.init()

    # Get available voices

    voices = engine.getProperty('voices')

    print("Available voices:")

    for idx, voice in enumerate(voices):

        print(f"{idx + 1}. {voice.name}")

    # Let user select voice

    voice\_idx = int(input("Select voice (enter the corresponding number): ")) - 1

    selected\_voice = voices[voice\_idx]

    engine.setProperty('voice', selected\_voice.id)

    # Get other settings

    rate = int(input("Enter speech rate (default is 200): "))

    engine.setProperty('rate', rate)

    # Get text input from user

    text = input("Enter the text to convert to speech: ")

    # Convert text to speech

    engine.say(text)

    # Save to file if user wants

    save\_option = input("Do you want to save the speech to a file? (yes/no): ").lower()

    if save\_option == 'yes':

        file\_name = input("Enter file name (include extension, e.g., output.mp3): ")

        engine.save\_to\_file(text, file\_name)

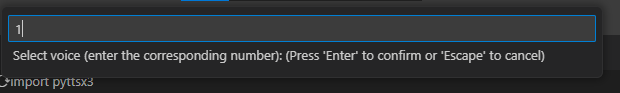
        print(f"Speech saved to {file\_name}")

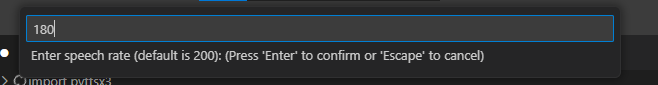
    # Play the speech

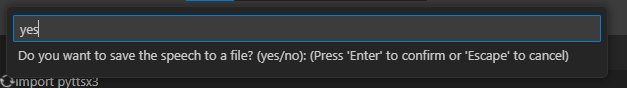
    engine.runAndWait()

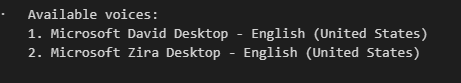
if \_\_name\_\_ == "\_\_main\_\_":

    main()









**Task 2 :** Automate WhatsApp messaging using PyWhatKit Scenario:





**Task 3 :** Convert images to sketches using OpenCV Scenario:

import cv2

def convert\_to\_sketch(image\_path, save\_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\_image = cv2.divide(gray\_image, inverted\_blurred\_image, scale=256.0)

    cv2.imwrite(save\_path, sketch\_image)

    print(f"Sketch saved to {save\_path}")

def main():

    image\_path = '1688459626\_newbabarazam.jpg'



    save\_path = '1688459626\_newbabarazam.jpg'

    convert\_to\_sketch(image\_path, save\_path)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Task 4 :**Implement a text summarization model using Transformers Scenario:

from transformers import pipeline

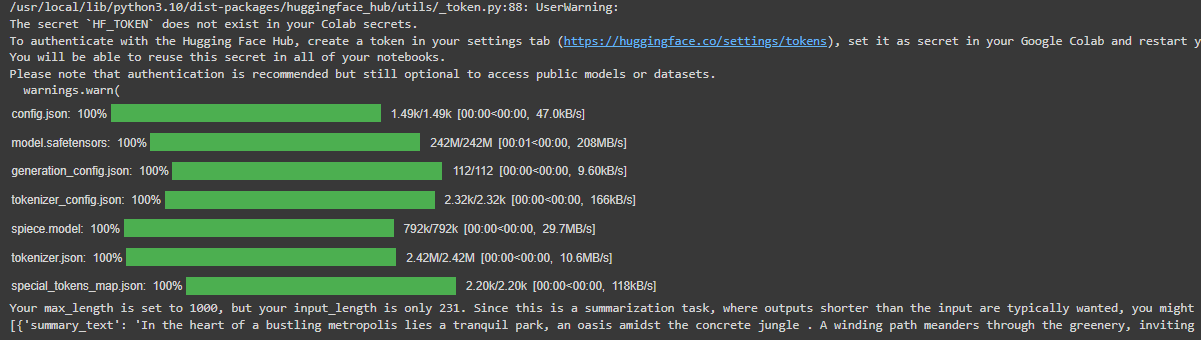
summarizer = pipeline("summarization",model = "Falconsai/text\_summarization")

ARTICLE = """

In the heart of a bustling metropolis lies a tranquil park, an oasis amidst the concrete jungle. Towering trees stretch their branches towards the sky, their leaves rustling in the gentle breeze like whispers of ancient secrets. A winding path meanders through the greenery, inviting wanderers to explore its hidden nooks and crannies. Benches, weathered by time and use, offer respite to weary souls seeking solace in nature's embrace. The air is filled with the symphony of bird songs, a melodic chorus that dances on the wind. Sunlight filters through the canopy above, dappling the ground below in a patchwork of light and shadow. In this sanctuary of green, time seems to slow, allowing visitors to pause, to breathe, to simply be. Away from the chaos of the city streets, here in this haven of peace, one can find a moment of tranquility, a moment of connection with the natural world that surrounds us.

"""

print(summarizer(ARTICLE,max\_length = 1000,min\_length = 30,do\_sample = False))



**Task 5 :** 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):

    df = pd.read\_csv('./train\_and\_test2.csv')

    return df

def explore\_data(df):

    print("Dataset information:")

    print(df.info())

    print("\nSummary statistics:")

    print(df.describe())

def visualize\_data(df):

    plt.figure(figsize=(8, 6))

    sns.countplot(x='Pclass', data=df)

    plt.title('Distribution of Passenger Classes')

    plt.xlabel('Class')

    plt.ylabel('Count')

    plt.show()

    plt.figure(figsize=(8, 6))

    sns.histplot(df['Age'].dropna(), bins=30, kde=True)

    plt.title('Distribution of Passenger Ages')

    plt.xlabel('Age')

    plt.ylabel('Count')

    plt.show()

    plt.figure(figsize=(8, 6))

    sns.barplot(x='Pclass', y='2urvived', data=df, ci=None)

    plt.title('Survival Rate by Passenger Class')

    plt.xlabel('Class')

    plt.ylabel('Survival Rate')

    plt.show()

def main():

    file\_path = "titanic.csv"

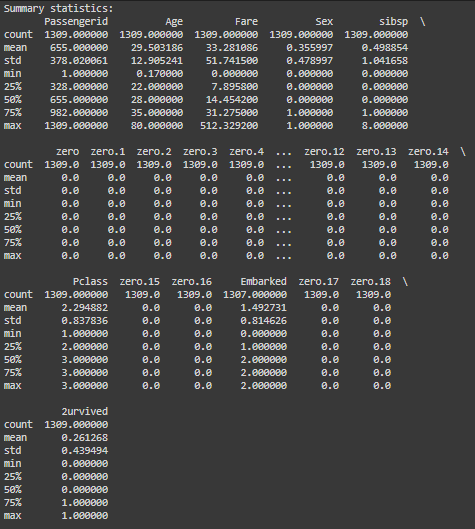
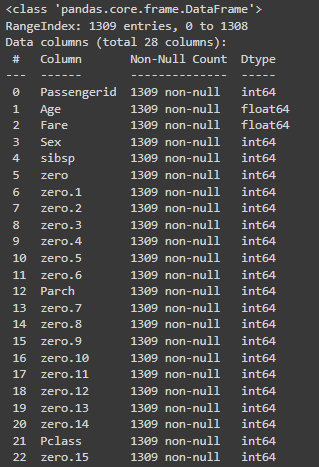
    df = load\_dataset(file\_path)

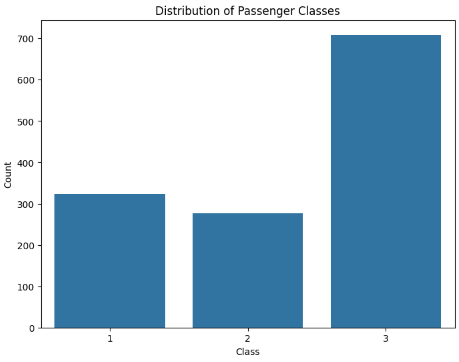
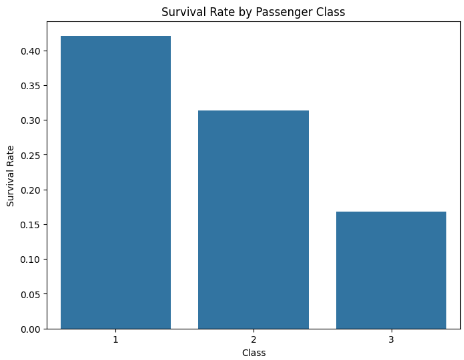
    explore\_data(df)

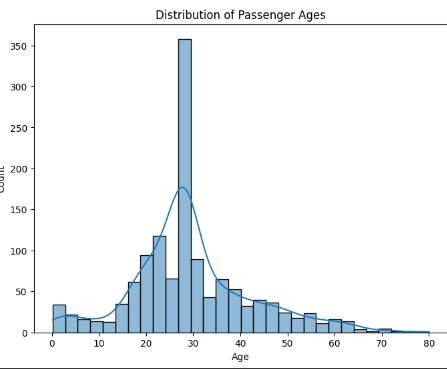
    visualize\_data(df)

if \_\_name\_\_ == "\_\_main\_\_":

    main()



**Task 6 :** Build a web scraper using Beautiful Soup Scenario:

**ARMY-AUCTIONJEEP CATEGORY:**

r = requests.get('https://www.pakwheels.com/used-cars/army-auction-jeep/430586')

html = r.text

soup = BeautifulSoup(html, "html.parser")

divs = soup.find\_all("ul",{"list-unstyled search-vehicle-info fs13"})

car\_detail = soup.find\_all("ul",{"class":"list-unstyled search-vehicle-info-2 fs13"})

divsd = soup.find\_all("a",{"class":"car-name ad-detail-path"})

divsd

key = "car name"

key0 = "category"

key1 = "city "

key2 = "make year"

key3= " total km"

key4 = "engine type"

key5 = "engine\_cc"

list\_of\_dicts\_armyacution = []

f = 0;

for i in divsd:

detail = i.text

detail = detail.strip("\n ")

detailed = detail.split()

detailed.pop()

detailed.pop()

detail = ' '.join(detailed)

# dictionary = {key: detail}

details = divs[f].text

details = details.strip("\n ")

further\_detail = car\_detail[f].text

further\_detail = further\_detail.strip("\n")

further\_details = further\_detail.split()

year = further\_details[0]

km = further\_details[1]

engine = further\_details[3]

e\_type = further\_details[4]

dictionary = {key: detail,key0:"army-auction-jeep",key1:details,key2: year,key3:km,key4: engine,key5:e\_type}

list\_of\_dicts\_armyacution.append(dictionary)

f +=1;

for i in list\_of\_dicts\_armyacution:

print(i)

**AUTOMATIC CARS CATEGORY:**

r = requests.get('https://www.pakwheels.com/used-cars/automatic/57336')

html = r.text

soup = BeautifulSoup(html, "html.parser")

divs = soup.find\_all("ul",{"list-unstyled search-vehicle-info fs13"})

car\_detail = soup.find\_all("ul",{"class":"list-unstyled search-vehicle-info-2 fs13"})

divsd = soup.find\_all("a",{"class":"car-name ad-detail-path"})

divsd

key = "car name"

key0 = "category"

key1 = "city "

key2 = "make year"

key3= " total km"

key4 = "engine type"

key5 = "engine\_cc"

list\_of\_dicts = []

f = 0;

for i in divsd

detail = i.text

detail = detail.strip("\n ")

detailed = detail.split()

detailed.pop()

detailed.pop()

detail = ' '.join(detailed)

# dictionary = {key: detail}

details = divs[f].text

details = details.strip("\n ")

further\_detail = car\_detail[f].text

further\_detail = further\_detail.strip("\n")

further\_details = further\_detail.split()

year = further\_details[0]

km = further\_details[1]

engine = further\_details[3]

e\_type = further\_details[4]

dictionary = {key: detail,key0:"automatic cars",key1:details,key2: year,key3:km,key4: engine,key5:e\_type}

list\_of\_dicts.append(dictionary)

f +=1;

for i in list\_of\_dicts:

print(i)

merged = list\_of\_dicts\_armyacution +list\_of\_dicts\_automatic + list\_of\_dicts\_armyacution2 +list\_of\_dicts\_automatic2+ list\_of\_dicts\_automatic3 + list\_of\_dicts\_automatic4+ list\_of\_dicts\_japanese + list\_of\_dicts\_japanese2+ list\_of\_dicts\_japanese3 + list\_of\_dicts\_japanese4 + list\_of\_dicts\_electric + list\_of\_dicts\_electric2 + list\_of\_dicts\_electric3 + list\_of\_dicts\_electric4 + list\_of\_dicts\_sports + list\_of\_dicts\_sports2 + list\_of\_dicts\_sports3 + list\_of\_dicts\_sports4

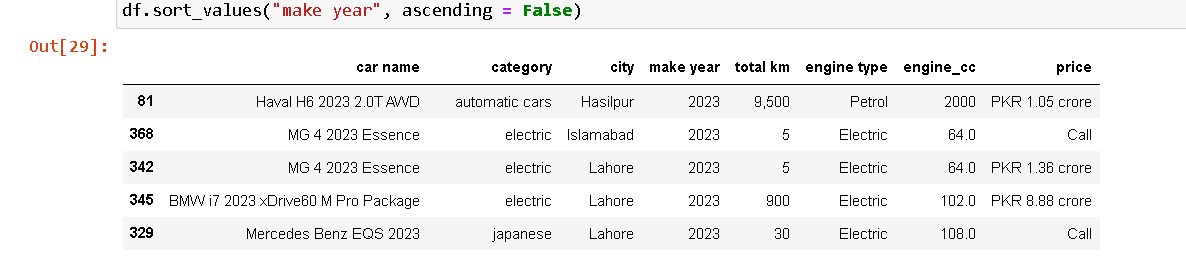
len(merged)

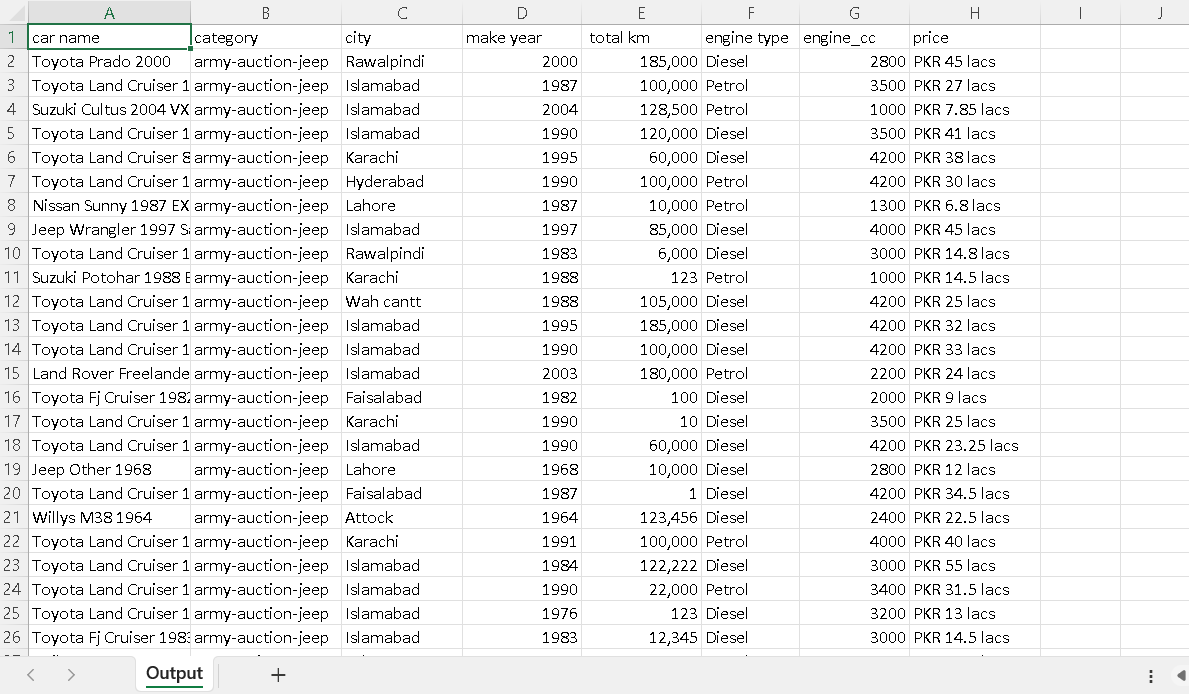
import pandas as pd

df = pd.DataFrame(merged)







****