PAPER 2:

Applied Artificial Intelligence



**CERTIFICATE**

**THE KELKAR EDUCATION TRUST’S**

**VINAYAK GANESH VAZE COLLEGE**

**OF ART, SCIENCE AND COMMERCE (AUT0ONOMOUS)**

Mithagar Road, MULUND(E) – 400 081

This is to certify that Mr./Ms. of M.Sc. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Information Technology) – (Part II) class, Seat No.\_\_\_\_\_\_\_\_\_\_\_\_\_\_has successfully completed the practical for Semester III Paper II **(Applied Artificial Intelligence)** in the InfoTech Laboratory(II) of KET’s V. G. VAZE College during the academic year 2022-2023.

Teacher in-charge Head of Department

Date:

(External Examiner)

|  |  |
| --- | --- |
| INDEX  **List of Practical:** | |
| 1 | Implement Bayes Theorem using Python |
| 2 | Implement Conditional and joining Probability using Python. |
| 3 | Design a Fuzzy based Application using Python. |
| 4 | Write an application to implement clustering algorithm |
| 5 | Write an application to implement support vector machine algorithm. |
| 6 | Design an AI application to implement intelligent agents |
| 7 | Design an application to simulate semantic web |

Q1. Implement Bayes Theorem using Python

Q2. Implement Conditional and joining Probability using Python.

Q3. Design a Fuzzy based Application using Python.

Q4. Write an application to implement clustering algorithm

Q5. Write an application to implement support vector machine algorithm.

Q6. Design an AI application to implement intelligent agents.

Q7. Design an application to simulate semantic web.

Q1. Implement Bayes Theorem using Python

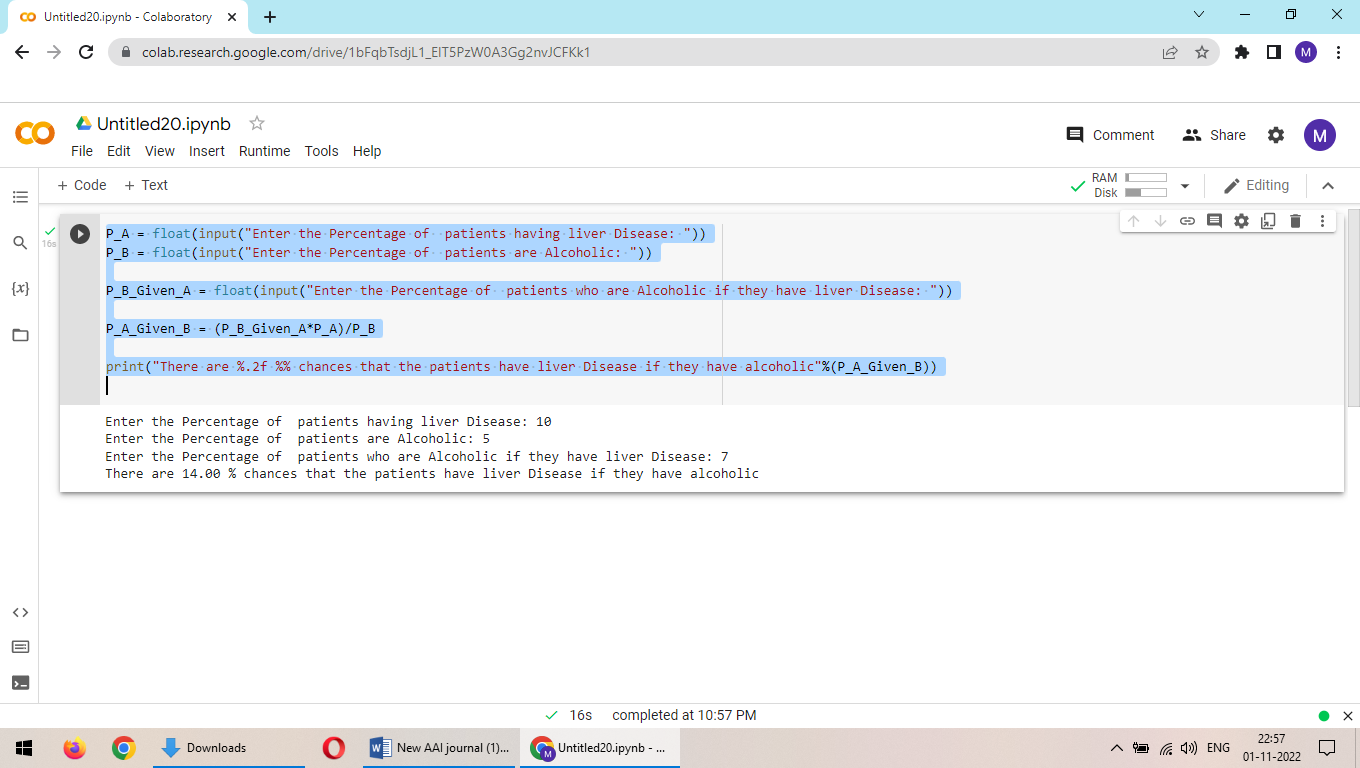
P\_A = float(input("Enter the Percentage of  patients having liver Disease: "))

P\_B = float(input("Enter the Percentage of  patients are Alcoholic: "))

P\_B\_Given\_A = float(input("Enter the Percentage of  patients who are Alcoholic if they have liver Disease: "))

P\_A\_Given\_B = (P\_B\_Given\_A\*P\_A)/P\_B

print("There are %.2f %% chances that the patients have liver Disease if they have alcoholic"%(P\_A\_Given\_B))



Q2. Implement Conditional and joining Probability using Python.

import pandas as pd

import numpy as np

Conditional:

#create pandas DataFrame with raw data

df = pd.DataFrame({'gender': np.repeat(np.array(['Male', 'Female']), 150),

                   'sport': np.repeat(np.array(['Baseball', 'Basketball', 'Football',

                                                'Soccer', 'Baseball', 'Basketball',

                                                'Football', 'Soccer']),

                                    (34, 40, 58, 18, 34, 52, 20, 44))})

#produce contingency table to summarize raw data

survey\_data = pd.crosstab(index=df['gender'], columns=df['sport'], margins=True)

#view contingency table

print(survey\_data)

#extract value in second row and first column

print(survey\_data.iloc[1, 0])

#calculate probability of being male, given that individual prefers baseball

print(survey\_data.iloc[1, 0] / survey\_data.iloc[2, 0])

#calculate probability of preferring basketball, given that individual is female

print(survey\_data.iloc[0, 1] / survey\_data.iloc[0, 4])

Joint Probability:

Card\_Colour= input('Enter the Colour of the card: ')

Card\_Number= input('Enter the Number of the card: ')

# P(A) is the Probability of drawing a card with entered color

P\_A = 26/52

# P(B) is the Probability of drawing a card with entered Number

P\_B = 4/52

print("Probability of drawing a ",Card\_Colour,' card is ',round(P\_A,2))

print("Probability of drawing a card with number ",Card\_Colour,' is ',round(P\_B,2))

P\_A\_AND\_B = round(P\_A\*P\_B,2)

print("Probability of drawing ", Card\_Colour, ' card with the number ', Card\_Number, ' from a normal deck of 52 playing card is ', P\_A\_AND\_B)

Q3. Desing a Fuzzy based Application using Python.

#pip3 install scikit-fuzzy

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

quality = ctrl.Antecedent(np.arange(0, 11, 1), 'quality')

service = ctrl.Antecedent(np.arange(0, 11, 1), 'service')

tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')

quality.automf(3)

service.automf(3)

tip['low'] = fuzz.trimf(tip.universe, [0, 0, 13])

tip['medium'] = fuzz.trimf(tip.universe, [0, 13, 25])

tip['high'] = fuzz.trimf(tip.universe, [13, 25, 25])

quality['average'].view()

service.view()

tip.view()

rule1 = ctrl.Rule(quality['poor'] | service['poor'], tip['low'])

rule2 = ctrl.Rule(service['average'], tip['medium'])

rule3 = ctrl.Rule(service['good'] | quality['good'], tip['high'])

rule1.view()

tipping\_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])

tipping = ctrl.ControlSystemSimulation(tipping\_ctrl)

# Pass inputs to the ControlSystem using Antecedent labels with Pythonic API

tipping.input['quality'] = 6.5

tipping.input['service'] = 9.8

# Crunch the numbers

tipping.compute()

print (tipping.output['tip'])

tip.view(sim=tipping)

Q4. Write an application to implement clustering algorithm

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

######

from sklearn.cluster import KMeans

data = list(zip(x, y))

inertias = []

for i in range(1,11):

    kmeans = KMeans(n\_clusters=i)

    kmeans.fit(data)

    inertias.append(kmeans.inertia\_)

plt.plot(range(1,11), inertias, marker='o')

plt.title('Elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('Inertia')

plt.show()

#####

data = list(zip(x, y))

kmeans = KMeans(n\_clusters=2)

kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels\_)

plt.show()

Q5. Write an application to implement support vector machine algorithm.

import random

def display(room):

    print(room)

room = [

    [1, 1, 1, 1],

    [1, 1, 1, 1],

    [1, 1, 1, 1],

    [1, 1, 1, 1],

]

print("All the rooom are dirty")

display(room)

x =0

y= 0

while x < 4:

    while y < 4:

        room[x][y] = random.choice([0,1])

        y+=1

    x+=1

    y=0

print("Before cleaning the room I detect all of these random dirts")

display(room)

#starting location

x =0

y= 0

z=0 #number of rooms cleaned

#agent code

while x < 4:

    while y < 4:

        if room[x][y] == 1:

            print("Vaccum in this location now,",x, y)

            room[x][y] = 0

            print("cleaned", x, y)

            z+=1

        y+=1

    x+=1

    y=0

pro= (100-((z/16)\*100))

print("Room is clean now")

display(room)

print('performance=',pro,'%')

Q6. Design an AI application to implement intelligent agents.

Q7. Design an application to simulate semantic web.

Save this file as **myfoaf.rdf**

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xmlns:foaf="http://xmlns.com/foaf/0.1/"

xmlns:admin="http://webns.net/mvcb/">

<foaf:Person rdf:nodeID="me">

<foaf:name>Mithilesh</foaf:name>

<foaf:knows>

<foaf:Person>

<foaf:name>Bhumika</foaf:name>

</foaf:Person>

</foaf:knows>

<foaf:knows>

<foaf:Person>

<foaf:name>Tanush</foaf:name>

</foaf:Person>

</foaf:knows>

</foaf:Person>

</rdf:RDF>

**#semanticweb.py :**

Fetching results about a specific subject using RDFLib

import rdflib

mygraph = rdflib.Graph()

mygraph.parse("myfoaf.rdf")

qres = mygraph.query(

"""SELECT DISTINCT ?fname ?lname

WHERE {

?a foaf:knows ?b .

?a foaf:name ?fname .

?b foaf:name ?lname .

}""")

for myrow in qres:

print("%s knows %s" % myrow)

PAPER 3:

Machine Learning



**CERTIFICATE**

**THE KELKAR EDUCATION TRUST’S**

**VINAYAK GANESH VAZE COLLEGE**

**OF ART, SCIENCE AND COMMERCE (AUT0ONOMOUS)**

Mithagar Road, MULUND(E) – 400 081

This is to certify that Mr./Ms. of M.Sc. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Information Technology) – (Part II) class, Seat No.\_\_\_\_\_\_\_\_\_\_\_\_\_\_has successfully completed the practical for Semester III Paper III **(Machine**

**Learning)** in the InfoTech Laboratory(II) of KET’s V. G. VAZE College during the academic year 2022-2023.

Teacher in-charge Head of Department

Date:

(External Examiner)

|  |  |
| --- | --- |
| **List of Practical:** | |
| 2 | 1. Perform Data Loading, Feature selection (Principal Component analysis) and Feature Scoring and Ranking. |
| 3. | B. Write a program to implement Decision Tree |
| 4. | 1. For a given set of training data examples stored in a .CSV file implement Least Square Regression algorithm. 2. For a given set of training data examples stored in a .CSV file implement Linear regression algorithm. |
| 6. | 1. Implement the classification model using clustering for the following techniques with K means clustering with Prediction, Test Score and Confusion Matrix |
| 7. | 1. Implement the classification model using clustering for the following techniques with hierarchical clustering with Prediction, Test Score and Confusion Matrix |

INDEX

Practical 2A:

Perform Data Loading, Feature selection.

Theory:

Feature selection is a process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested.

Having irrelevant features in your data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression.

Three benefits of performing feature selection before modeling your data are:

* Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
* Improves Accuracy: Less misleading data means modeling accuracy improves.
* Reduces Training Time: Less data means that algorithms train faster.

Program:

from pandas import read\_csv

from sklearn.ensemble import ExtraTreesClassifier

# load data

url = "pima-indians-diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

dataframe = read\_csv(url, names=names)

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]

# feature extraction

model = ExtraTreesClassifier(n\_estimators=10)

model.fit(X, Y)

print(names)

print(model.feature\_importances\_)

Output:

['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

[0.10725276 0.21109673 0.10056895 0.08541 0.07307367 0.14485006

0.11504959 0.16269824]

Practical 2B:

Perform Principal Component analysis

Theory:

Principal Component Analysis (or PCA) uses linear algebra to transform the dataset into a compressed form.

Generally, this is called a data reduction technique. A property of PCA is that you can choose the number of dimensions or principal component in the transformed result.

Program:

import numpy

from pandas import read\_csv

from sklearn.decomposition import PCA

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

# load data

url = "pima-indians-diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

dataframe = read\_csv(url, names=names)

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]

# feature extraction

pca = PCA(n\_components=3)

fit = pca.fit(X)

# summarize components

print("Explained Variance: %s" % fit.explained\_variance\_ratio\_)

print(fit.components\_)

Output:

Explained Variance: [0.88854663 0.06159078 0.02579012]

[[-2.02176587e-03 9.78115765e-02 1.60930503e-02 6.07566861e-02

9.93110844e-01 1.40108085e-02 5.37167919e-04 -3.56474430e-03]

[-2.26488861e-02 -9.72210040e-01 -1.41909330e-01 5.78614699e-02

9.46266913e-02 -4.69729766e-02 -8.16804621e-04 -1.40168181e-01]

[-2.24649003e-02 1.43428710e-01 -9.22467192e-01 -3.07013055e-01

2.09773019e-02 -1.32444542e-01 -6.39983017e-04 -1.25454310e-01]]

Practical 3:

Write a program to implement Decision Tree

Theory:

A Decision Tree is a supervised algorithm used in machine learning. It is using a binary tree graph (each node has two children) to assign for each data sample a target value. The target values are presented in the tree leaves. To reach to the leaf, the sample is propagated through nodes, starting at the root node. In each node a decision is made, to which descendant node it should go. A decision is made based on the selected sample’s feature. Decision Tree learning is a process of finding the optimal rules in each internal tree node according to the selected metric.

The decision trees can be divided, with respect to the target values, into:

Classification trees used to classify samples, assign to a limited set of values - classes. In scikit-learn it is DecisionTreeClassifier.

Regression trees used to assign samples into numerical values within the range. In scikit-learn it is DecisionTreeRegressor.

Program:

from matplotlib import pyplot as plt

from sklearn import datasets

from sklearn.tree import DecisionTreeClassifier

from sklearn import tree

# Prepare the data data

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# Fit the classifier with default hyper-parameters

clf = DecisionTreeClassifier(random\_state=1234)

model = clf.fit(X, y)

fig = plt.figure()

\_ = tree.plot\_tree(clf,

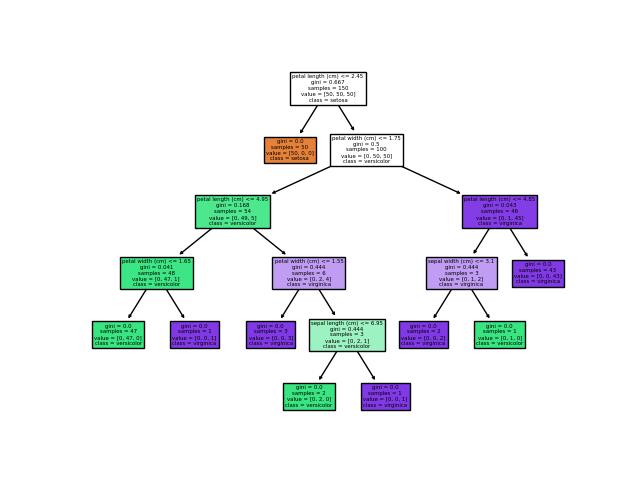
                   feature\_names=iris.feature\_names,

                   class\_names=iris.target\_names,

                   filled=True)

plt.show()

Output:



Practical 4.a:

For a given set of training data examples stored in a .CSV file implement Least Square Regression algorithm.

Theory:

In statistics, Linear Regression is a linear approach to model the relationship between a scalar response (or dependent variable), say Y, and one or more explanatory variables (or independent variables), say X.

Regression Line: If our data shows a linear relationship between X and Y, then the straight line which best describes the relationship is the regression line. It is the straight line that covers the maximum points in the graph.

A regression line is given as Y = a + b\*X where the formula of b and a are given as:

b = (nΣ(xiyi) – Σ(xi)Σ(yi)) ÷ (nΣ(xi2)-Σ(xi)2)

a = ȳ – b.x̄

where x̄ and ȳ are mean of x and y respectively.

* To find regression line, we need to find a and b.
* Calculate a, which is given by a = (\sum yi)/n - b \* (\sum xi)/n
* Calculate b, which is given by
* b = (n\*\sum(xi\*yi) - \sum (xi)\* \sum (yi))/(n\*\sum (xi)^{2}-(\sum xi)^{2})
* Put value of a and b in the equation of regression line.

Program:

"""

To find regression line, we need to find a and b.

Calculate a, which is given by a = (\sum yi)/n - b \* (\sum xi)/n

Calculate b, which is given by

b = (n\*\sum(xi\*yi) - \sum (xi)\* \sum (yi))/(n\*\sum (xi)^{2}-(\sum xi)^{2})

Put value of a and b in the equation of regression line.

"""

# Function to calculate b

def calculateB(x, y, n):

  # sum of array x

  sx = sum(x)

  # sum of array y

  sy = sum(y)

  # for sum of product of x and y

  sxsy = 0

  # sum of square of x

  sx2 = 0

  for i in range(n):

    sxsy += x[i] \* y[i]

    sx2 += x[i] \* x[i]

  b = (n \* sxsy - sx \* sy)/(n \* sx2 - sx \* sx)

  return b

# Function to find the

# least regression line

def leastRegLine(X,Y,n):

  # Finding b

  b = calculateB(X, Y, n)

  meanX = int(sum(X)/n)

  meanY = int(sum(Y)/n)

  # Calculating a

  a = meanY - b \* meanX

  # Printing regression line

  print("Regression line:")

  print("Y = ", '%.3f'%a, " + ", '%.3f'%b, "\*X", sep="")

# Driver code

# Statistical data

import pandas as pd

# Step 1 :Import libraries and dataset

datas = pd.read\_csv('data.csv')

print(datas )

X = datas.TEMPERATURE

Y = datas.PRESSURE

n = len(X)

leastRegLine(X, Y, n)

Output:

SNO TEMPERATURE PRESSURE

0 1 0 0.0002

1 2 20 0.0012

2 3 40 0.0060

3 4 60 0.0300

4 5 80 0.0900

5 6 100 0.2700

Regression line:

Y = -0.117 + 0.002\*X

Practical 4b.:

For a given set of training data examples stored in a .CSV file implement Linear Regression algorithm.

Theory:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

**Hypothesis function for Linear Regression:**



While training the model we are given :  
**x:** input training data (univariate – one input variable(parameter))  
**y:** labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.  
**θ1:** intercept  
**θ2:** coefficient of x

Program:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Step 1 :Import libraries and dataset

datas = pd.read\_csv('data.csv')

print(datas )

#Step 2: Dividing the dataset into 2 components

X = datas.iloc[:, 1:2].values

y = datas.iloc[:, 2].values

#Step 3: Fitting Linear Regression to the dataset

from sklearn.linear\_model import LinearRegression

lin = LinearRegression()

lin.fit(X, y)

plt.scatter(X, y, color = 'blue')

plt.plot(X, lin.predict(X), color = 'red')

plt.title('Linear Regression')

plt.xlabel('Temperature')

plt.ylabel('Pressure')

plt.show()

Output:

====================

SNO TEMPERATURE PRESSURE

0 1 0 0.0002

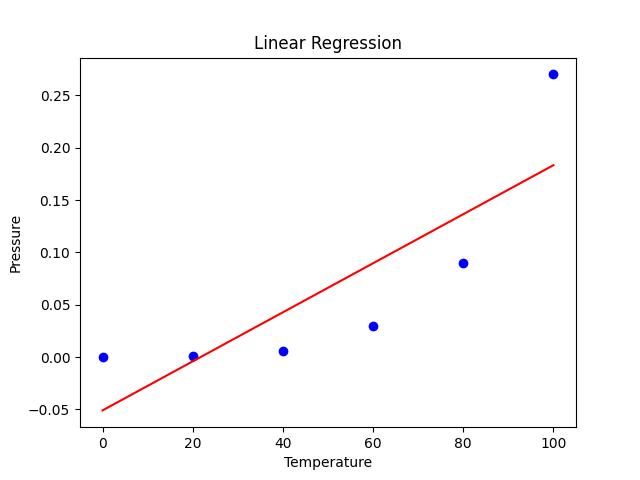
1 2 20 0.0012

2 3 40 0.0060

3 4 60 0.0300

4 5 80 0.0900

5 6 100 0.2700



Practical 6.:

Implement the classification model using clustering for the following techniques with K means clustering with Prediction, Test Score and Confusion Matrix.

Theory:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

Program:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Step 1 :Import libraries and dataset

datas = pd.read\_csv('data.csv')

print(datas )

#Step 2: Dividing the dataset into 2 components

X = datas.iloc[:, 1:2].values

y = datas.iloc[:, 2].values

#Step 3: Fitting Linear Regression to the dataset

from sklearn.linear\_model import LinearRegression

lin = LinearRegression()

lin.fit(X, y)

plt.scatter(X, y, color = 'blue')

plt.plot(X, lin.predict(X), color = 'red')

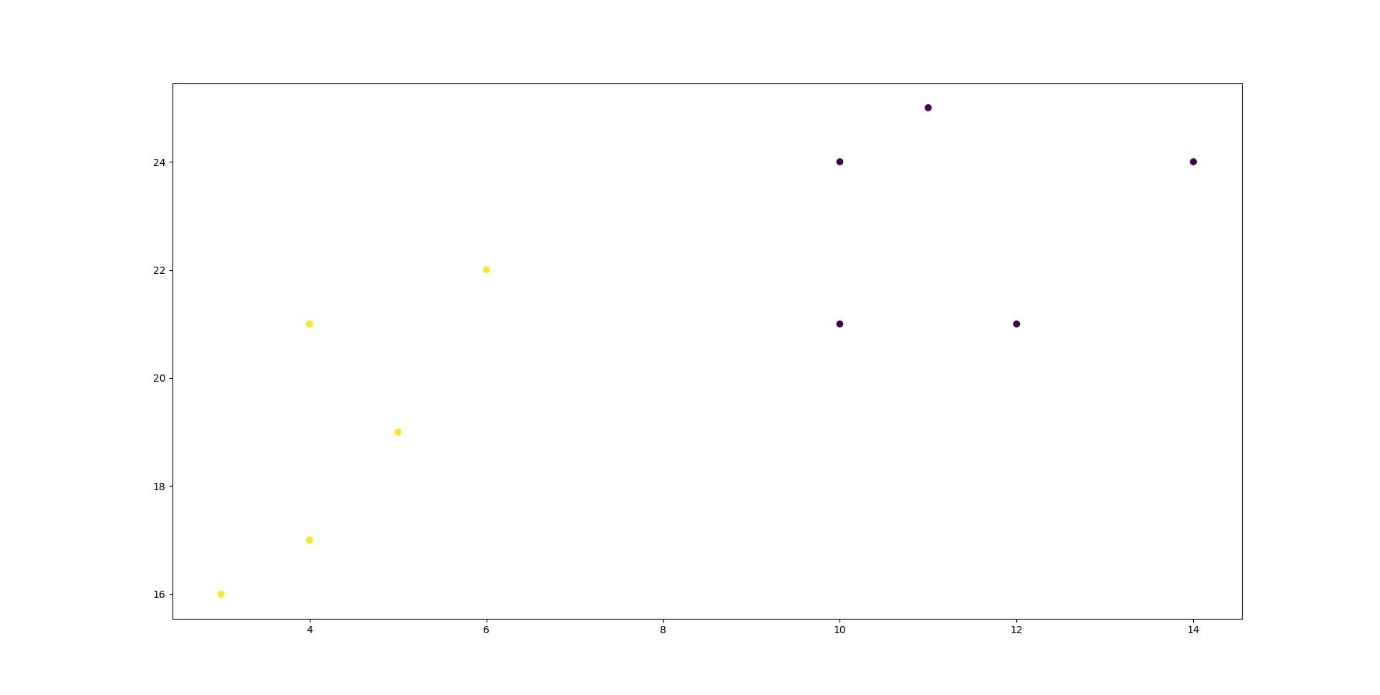
plt.title('Linear Regression')

plt.xlabel('Temperature')

plt.ylabel('Pressure')

plt.show()

Output:



Practical 7:

Implement the classification model using clustering for the following techniques with hierarchical clustering with Prediction, Test Score and Confusion Matrix.

Theory:

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

The hierarchical clustering technique has two approaches:

Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

Program:

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering

from scipy.cluster.hierarchy import dendrogram, linkage

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

data = list(zip(x, y))

hierarchical\_cluster = AgglomerativeClustering(n\_clusters=3, affinity='euclidean', linkage='ward')

labels = hierarchical\_cluster.fit\_predict(data)

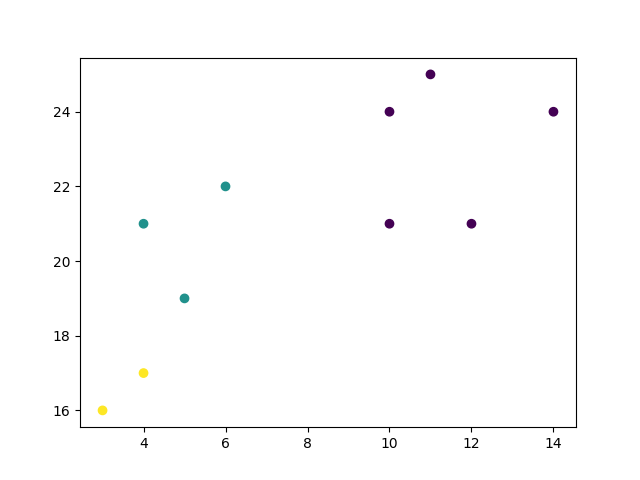
print(labels)

plt.scatter(x, y, c=labels)

plt.show()

Output:

[1 1 0 2 2 0 0 1 0 0]



PAPER 4:

ROBOTIC PROCESS AUTOMATION



**CERTIFICATE**

**THE KELKAR EDUCATION TRUST’S**

**VINAYAK GANESH VAZE COLLEGE**

**OF ART, SCIENCE AND COMMERCE (AUT0ONOMOUS)**

Mithagar Road, MULUND(E) – 400 081

This is to certify that Mr./Ms. of M.Sc. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Information Technology) – (Part II) class, Seat No.\_\_\_\_\_\_\_\_\_\_\_\_\_\_has successfully completed the practical for Semester III Paper IV **(Robotic Process Automation)** in the InfoTech Laboratory(II) of KET’s V. G. VAZE College during the academic year 2022-2023.

Teacher in-charge Head of Department

Date:

(External Examiner)

|  |  |
| --- | --- |
| INDEX  **List of Practicals:** | |
| 1 | Create a simple sequence-based project |
| 2 | 1. Automate UiPath Number Calculation (Subtraction, Multiplication, Division of numbers). 2. Create an automation UiPath project using different types of variables (array, data table) |
| 3 | Create an automation UiPath Project using decision statements. |
| 4 | Automate any process using web/basic recording. |
| 5 | Consider an array of names. We have to find out how many of them start with the letter "a". Create an automation where the number of names starting with "a" is counted and the result is displayed. |
| 6 | 1. Create an application automating the read, write and append operation on excel file. 2. Automate the process to extract data from an data table into an excel file. |

Practical 1

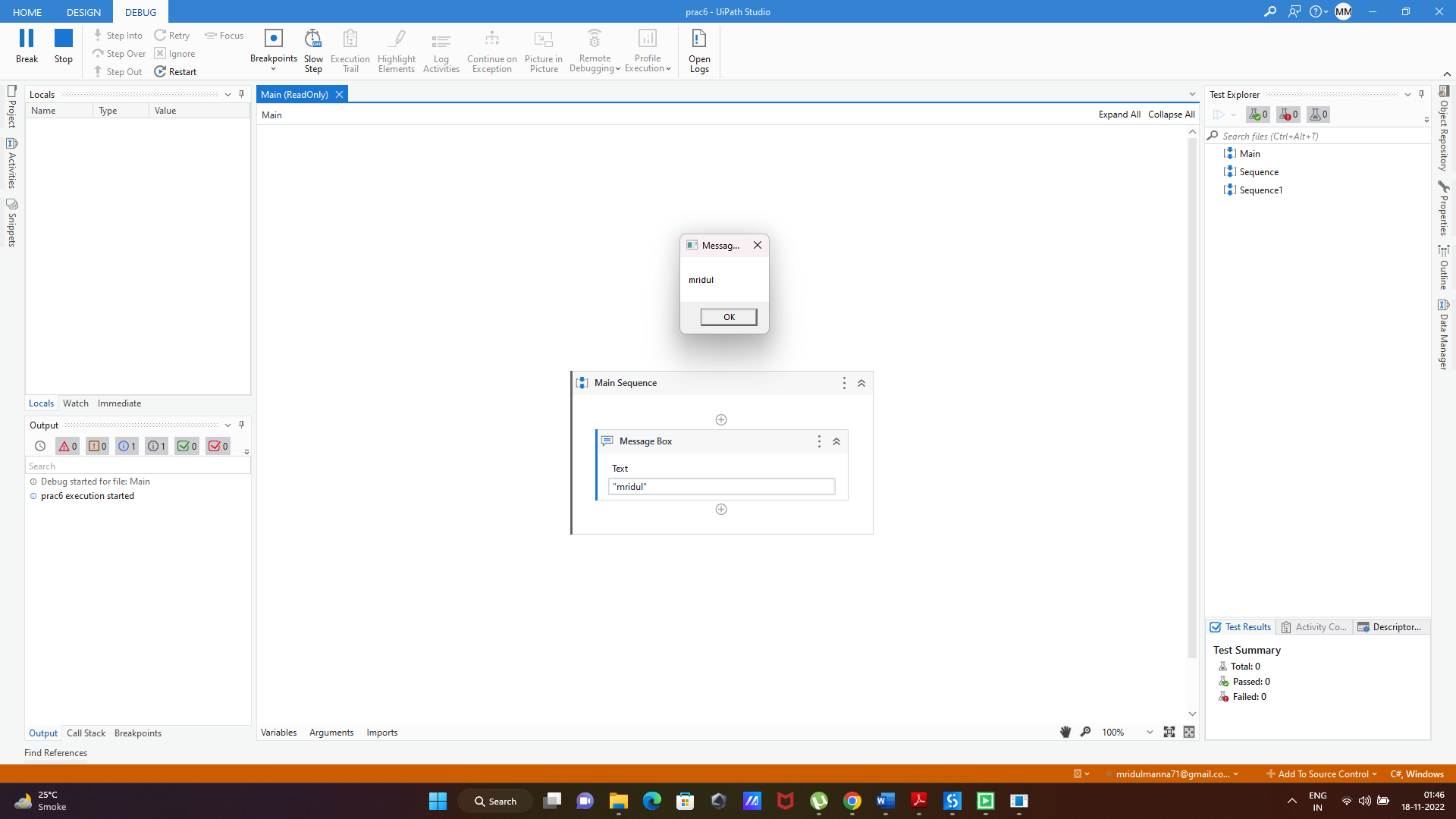
Create a simple sequence-based project

Theory:

A sequence is a basic automation process in RPA. It is used when the process is divided into steps and sequential. It is simpler and easier to use.

Steps:

1. Open UI path studio.
2. In uipath studio start page, under new project click on process
3. A dialog box appears, give a name to the process and select the language as C# and click on create.
4. Once a process is created, go to activity panel and add new sequence
5. Add name to the sequence and click on create. A sequence is created.
6. Once the sequence is created, add a message box.
7. Add the text you want to display and click on debug at the top ribbon.
8. This is how we create a simple sequence.

Output:

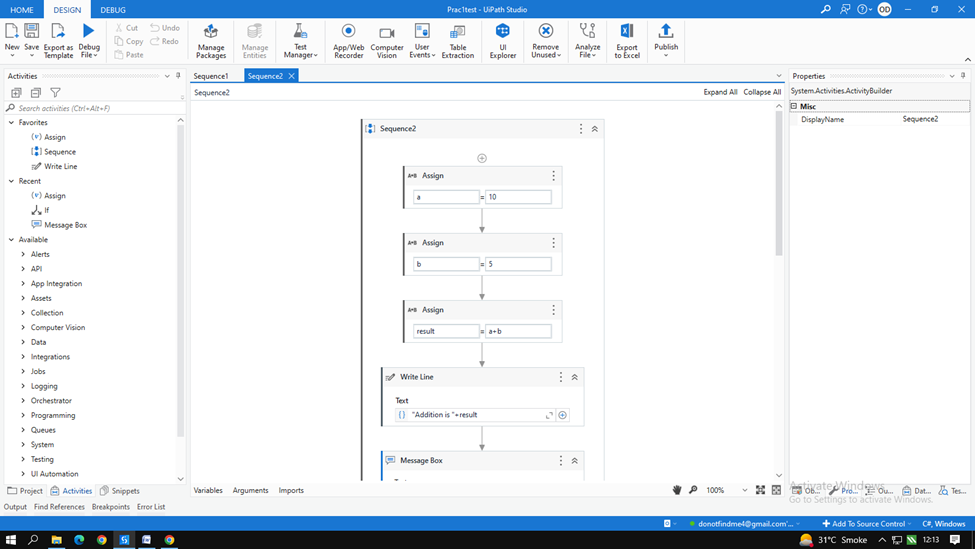
Practical 2a

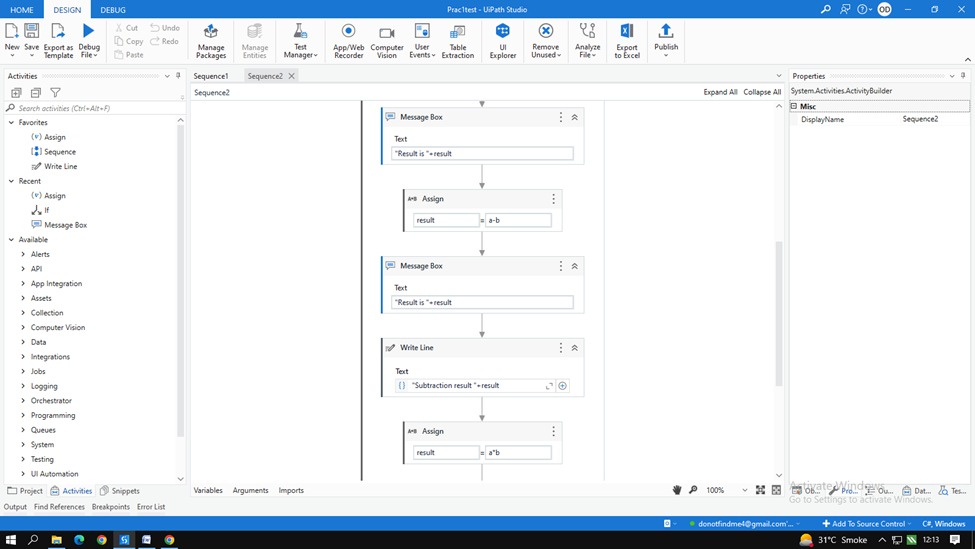
Automate UiPath Number Calculation (Subtraction, Multiplication, Division of numbers).

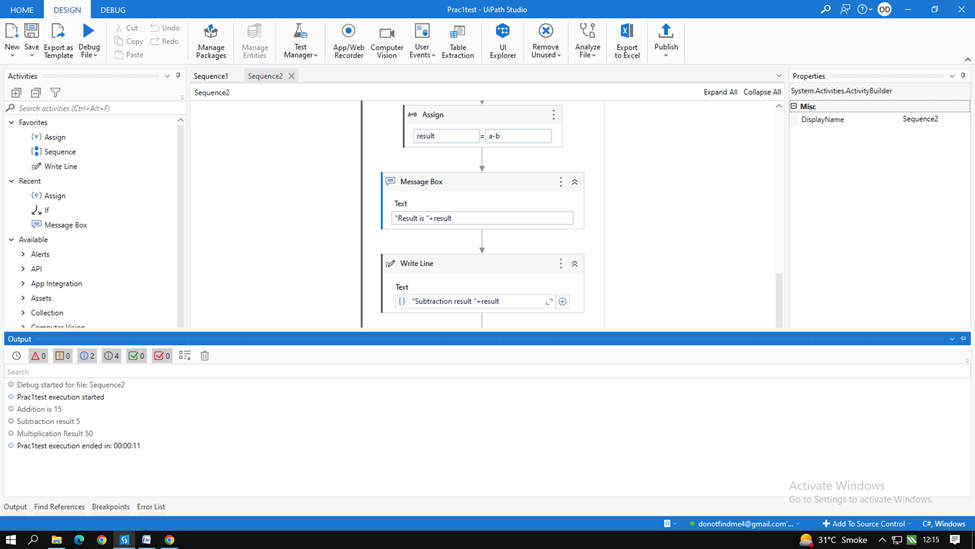
Steps:

1. Create a new sequence.
2. In the sequence, drag and drop an assign box to assign a variable. Assign a variable “a” with value 10.
3. On the variables tab, create the variable and assign the variable type as Int32.
4. Repeat steps 2 and 3 to create another variable “b” of integer type and give it the value 15.
5. Add a third assign box and create a variable “result” that will store the value of the operation between a and b. (a+b). This variable will also be integer.
6. Once done, drag and drop a write line box and add: “Addition is” + result since the output will be in string format.
7. Repeat steps from 5 and 6 thrice and change the operation as c=a-b, c=a\*b, c=a/b.
8. Run the automation.

Output:







Practical 2b

Create an automation UiPath project using different types of variables (array, data table)

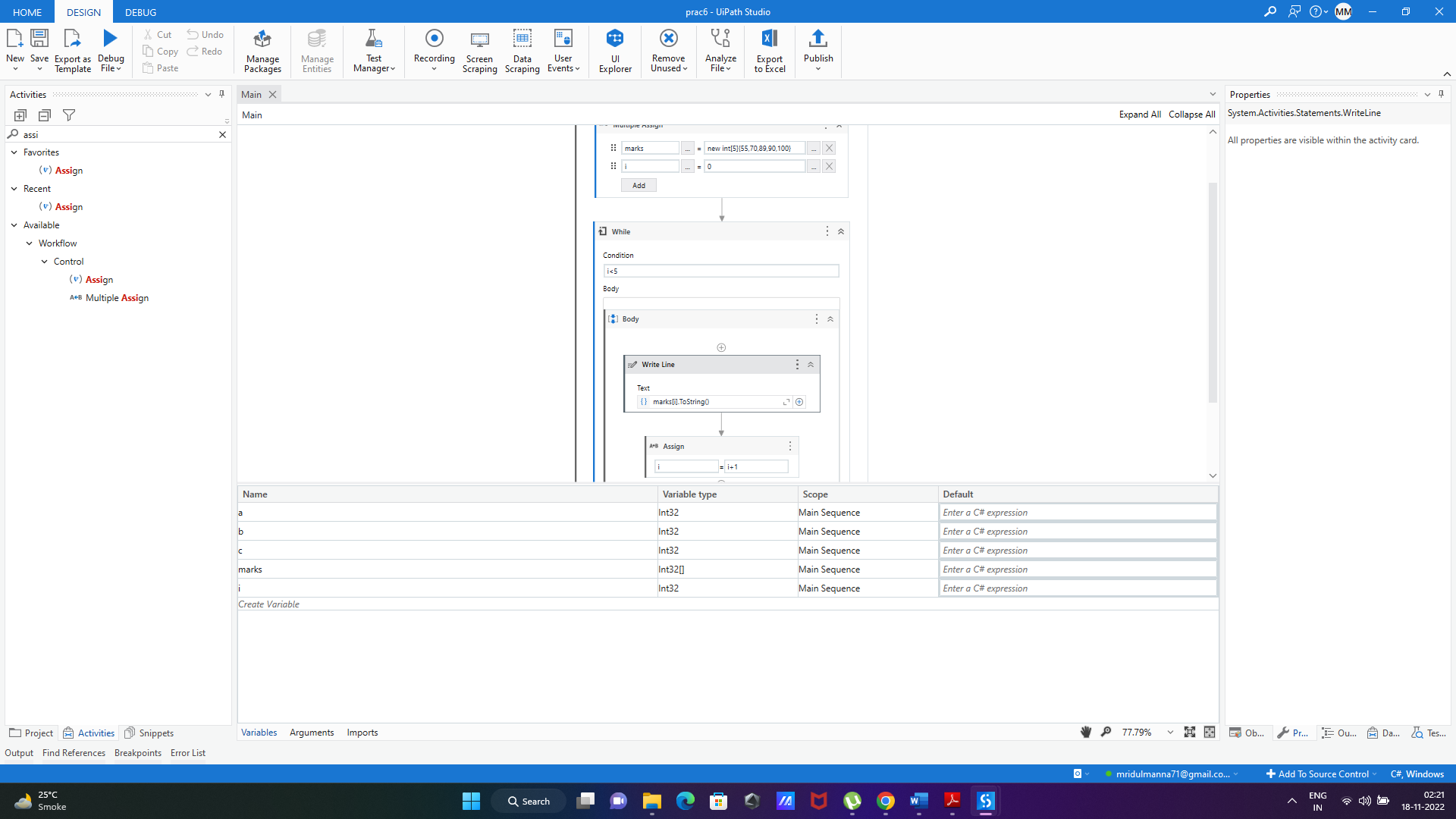
Steps: To create array

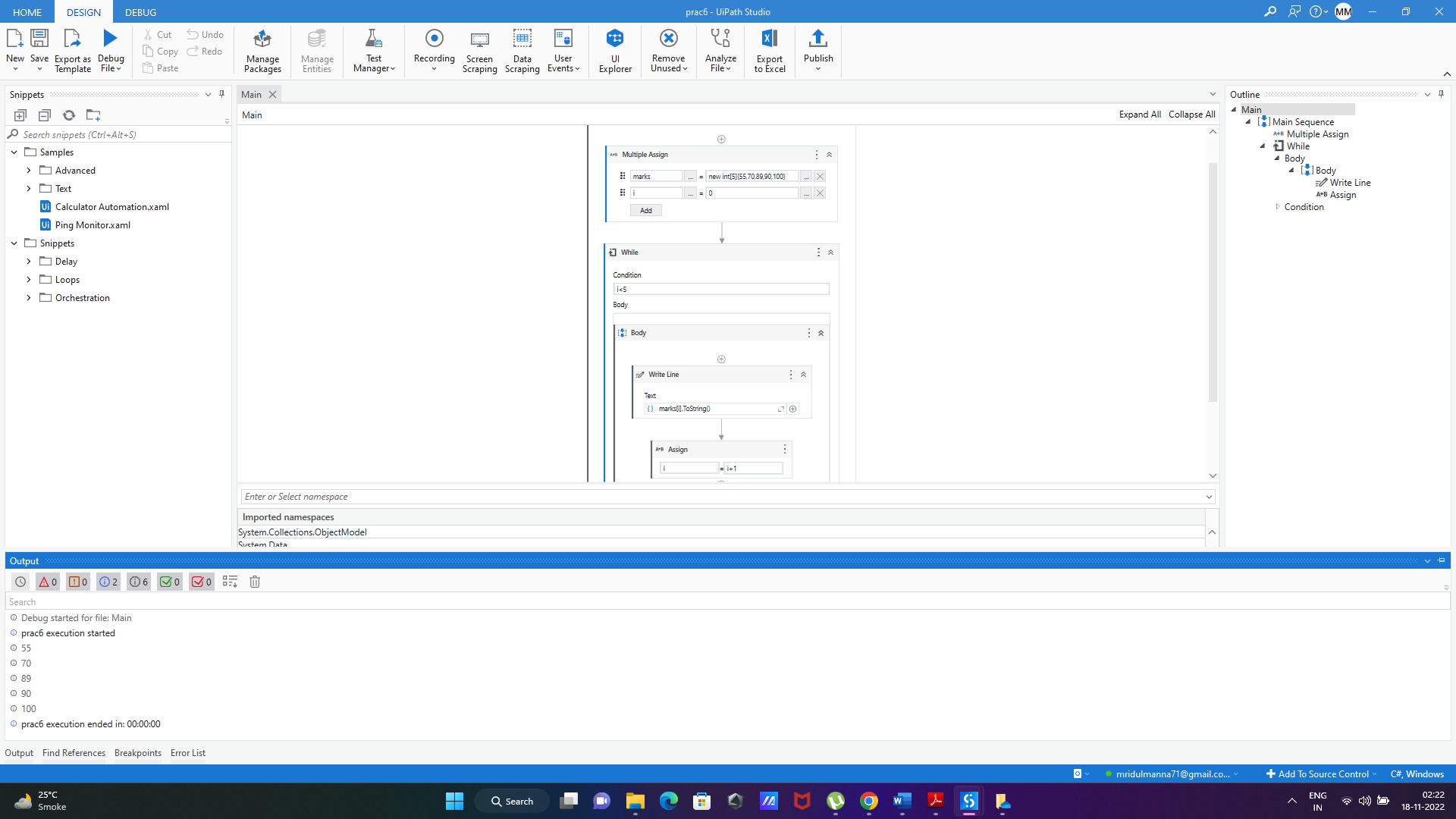
1. Create a new sequence.
2. Drag and drop multiple assign activity.
3. Create an array variable a where,

* marks = new int[5]{99,100,50,22,98}

1. Once done, go to variable option and change the data type of the array to array of [T]. Click on the option and select the array type as int 32.
2. Create another variable i=0 where I will be the iteration.
3. Drag and drop while activity. In the condition part add i<5
4. In the body drag and drop writeline and add marks[i].ToString() to display the array
5. Finally drag and drop a new assign and add i=i+1 to increase the iteration by 1. This will display all the members of the array.
6. Run the automation and check the output.

Output:





Practical 2b

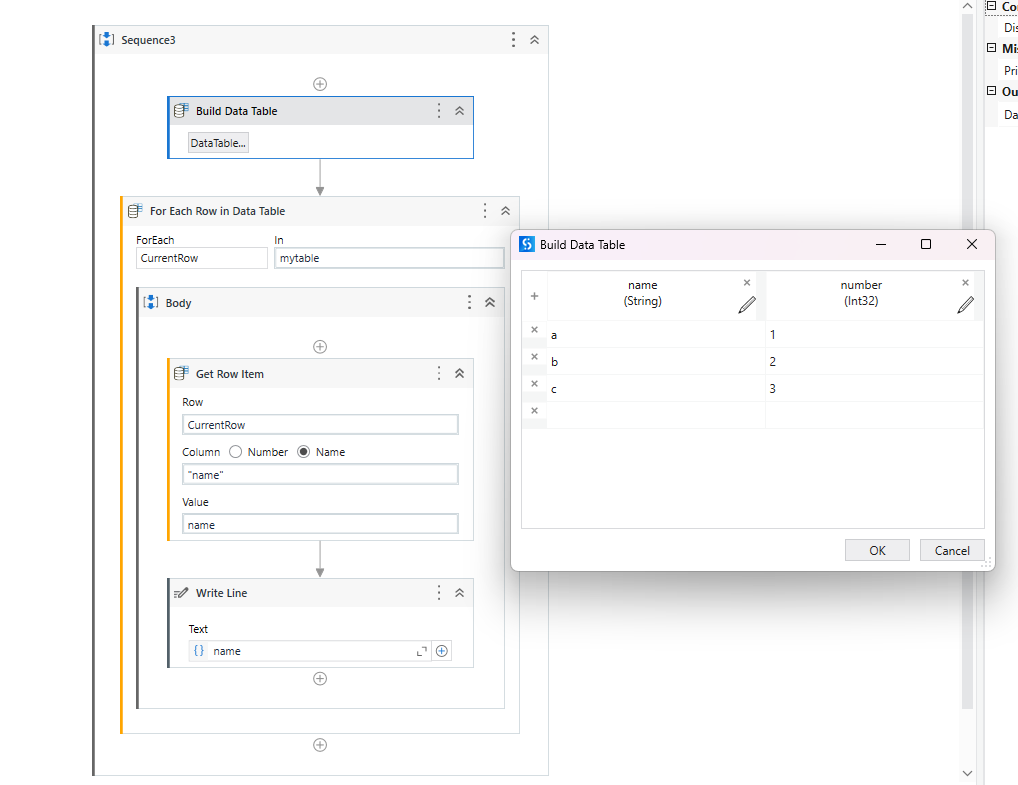
Create an automation UiPath project using different types of variables (array, data table)

Steps: To create datatable.

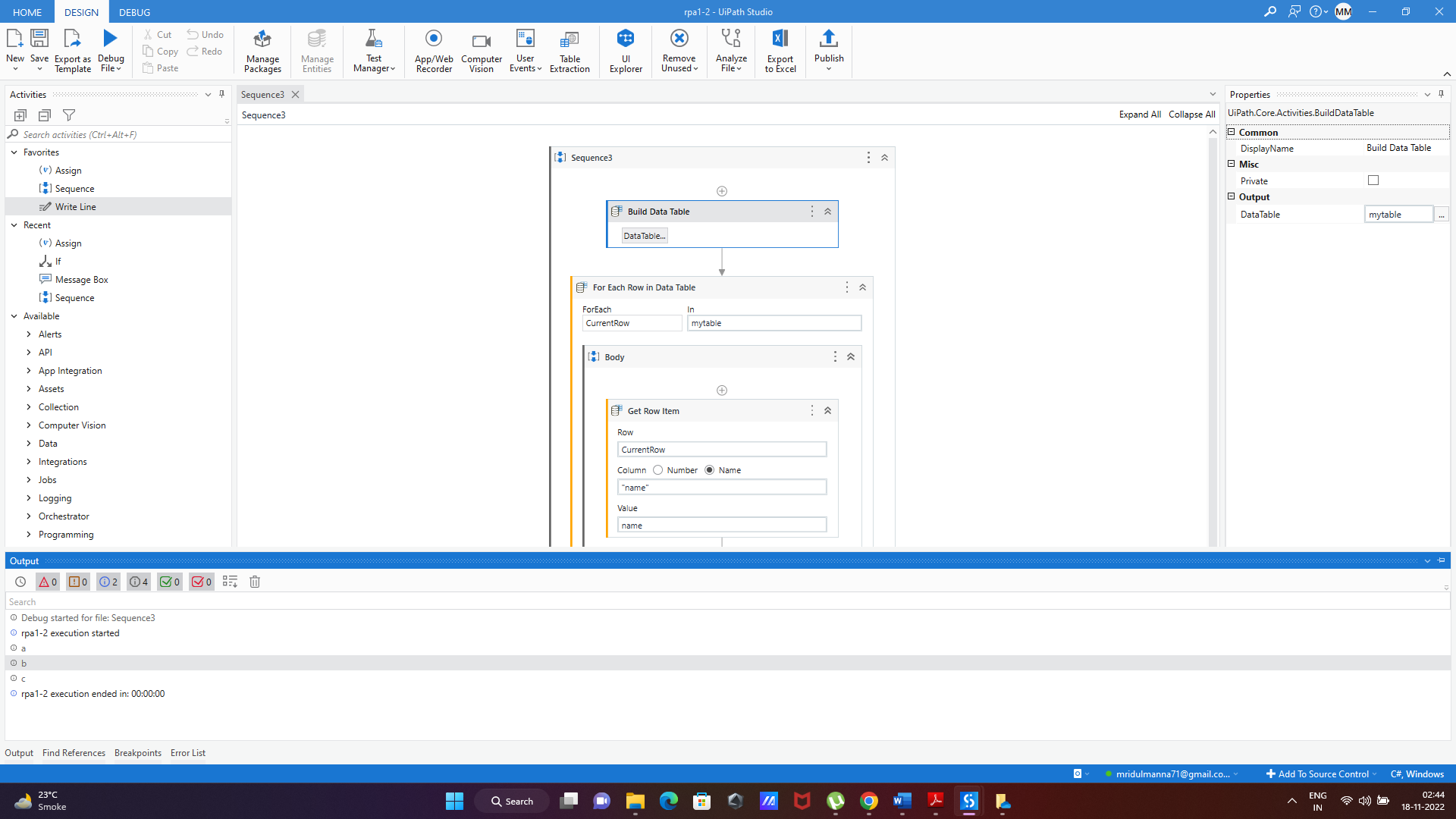
1. Create a new sequence.
2. Drag and drop build data table activity from activity panel.
3. In the build data table acclivity, click on data table button and add the data you want to add in the data table.
4. Once done, on the right side in the properties tab, right click on the output option and click on create variable and add the variable name as mytable.
5. Drag and drop the for each row in data table activity in the sequence.
6. ForEach= CurrentRow and In=mytable. This will take the value in each row in the data table.
7. Once done, in the body of the for each activity, drag and drop the “get row item activity” and add the following:

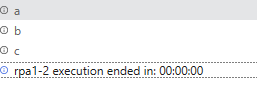
* Row= CurrentRow
* Choose column as Name
* Add column name as “name”
* Add value = name

1. On the right-hand side in the properties panel, right click on output and select create variable
2. Create the variable as name.
3. Once done, drag and drop a writeline activity and add “name” as the column name.
4. Run the automation and check the output.



Output: This will display the data in the name column of the datatable.





**Practical 3A:**

**Create an automation UiPath Project using decision statements.**

Theory:

The If activity contains a statement and two conditions. The first condition (the activity in the Then section) is executed if the statement is true, while the second one (the activity in the optional Else section) is executed if the statement is false.

If activities can be useful to make decisions based on the value of variables.

Steps:

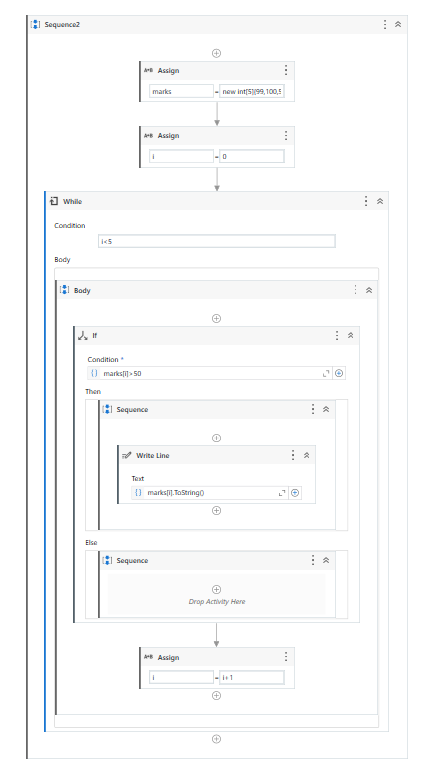
1. Create a new sequence.
2. Drag and drop multiple assign activity.
3. Create an array variable a where,

marks = new int[5]{99,100,50,22,98}

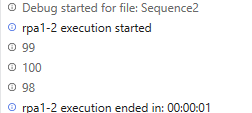
1. Once done, go to variable option and change the data type of the array to array of [T]. Click on the option and select the array type as int 32.
2. Create another variable i=0 where I will be the iteration.
3. Drag and drop while activity. In the condition part add i<5
4. In the body of the while activity, drag and drop the if activity and add the condition:

Marks[i]>50

1. In the “then” panel drag and drop writeline activity and add marks[i].ToString() to display the array
2. Finally drag and drop a new assign outside the if activity and add i=i+1 to increase the iteration by 1. This will display all the members of the array.
3. Run the automation and check the output.



Output: This automation will print the numbers in the array greater than 50..



**Practical 3A:**

**Create an automation UiPath Project using looping statements**

Theory:

Loops are structures used to automate repetitive tasks.

In flowcharts, the simplest types of loops can be created by connecting a certain point in our workflow to an earlier execution one.

In sequences, there are special activities (or containers) that repeat the action that is inside the body section.

While loop:

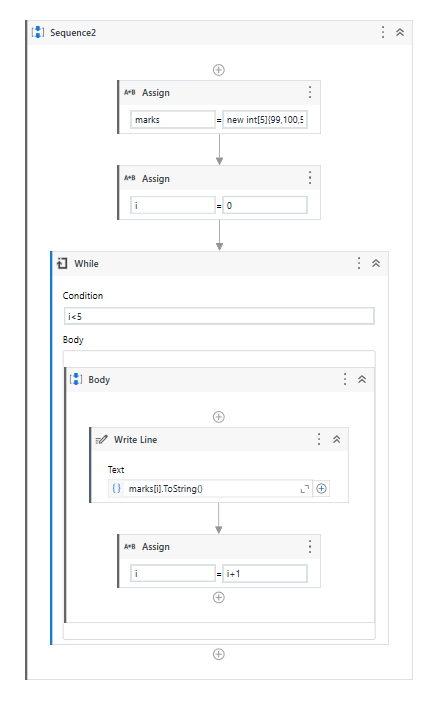
If the condition is met, the set of actions in the body are executed. If the conditions are false, then the loop breaks and the next step is executed. Unless the condition is true, the loop will not start to execute in while loop.

For- each activity:

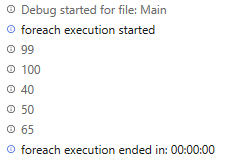
The For Each loop works by iterating through a list of items, one item at a time, and executing whatever actions are in the body of the action.

Steps: While activity

1. Create a new sequence.
2. Drag and drop multiple assign activity.
3. Create an array variable a where,
   1. marks = new int[5]{99,100,50,22,98}
4. Once done, go to variable option and change the data type of the array to array of [T]. Click on the option and select the array type as int 32.
5. Create another variable i=0 where I will be the iteration.
6. Drag and drop while activity. In the condition part add i<5
7. In the body drag and drop writeline and add marks[i].ToString() to display the array
8. Finally drag and drop a new assign and add i=i+1 to increase the iteration by 1. This will display all the members of the array.
9. Run the automation and check the output.



Output:

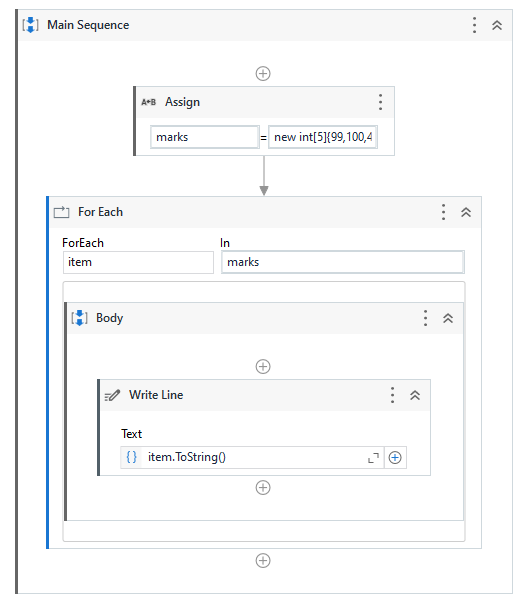


Steps: For-each activity

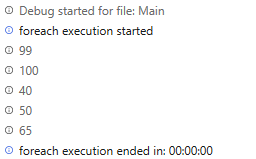
1. Create a new sequence.
2. Drag and drop assign activity.
3. Create an array variable marks where,
   1. marks = new int[5]{99,100,50,22,98}
4. Once done, go to variable option and change the data type of the array to array of [T]. Click on the option and select the array type as int 32.
5. Drag and drop for each activity.
6. In the ForEach activity add the following statements:

* In for each field add item
* Add marks in “In” field

1. In the body drag and drop writeline and add item.ToString() to display the array
2. Run the automation and check the output.



Output: This automation will print an array using for each activity.



**Practical 4:**

**Automate any process using web/basic recording.**

Theory:

Recording is an important part of UiPath Studio, that can help you save a lot of time when automating your business processes. This functionality enables you to easily capture a user’s actions on the screen and translates them into sequences.

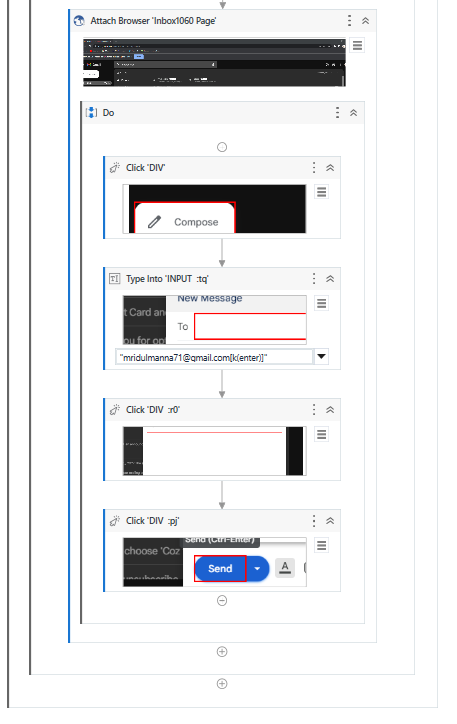
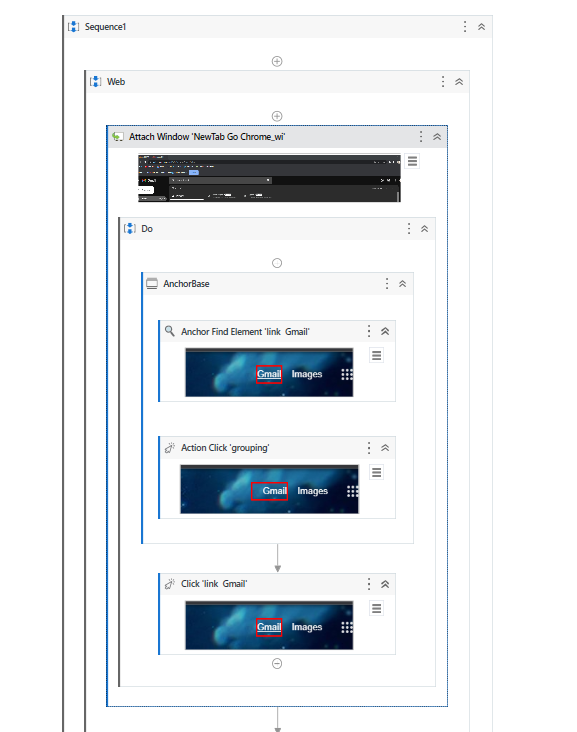
These projects can be modified and parameterized so that you can easily replay and reuse them in as many other processes as you need.

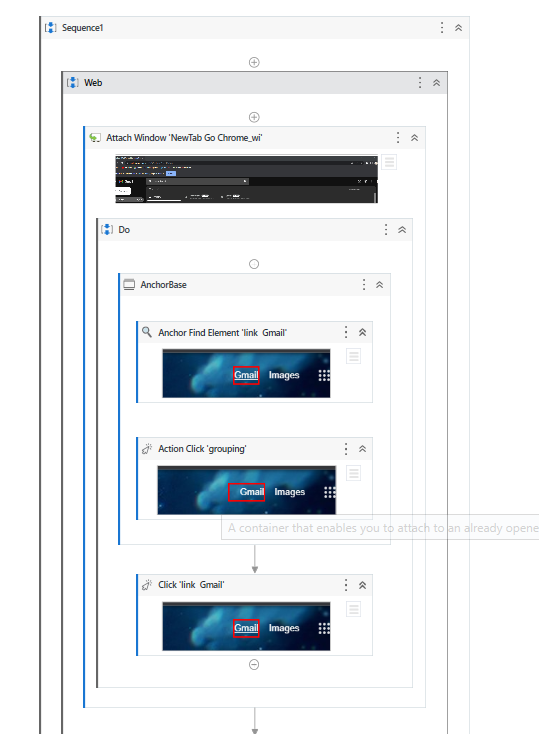
All user interface elements are highlighted while you record, as you can see in the following screenshot, so that you can be sure the correct buttons, fields or menus are selected.

Steps:

1. Create a new sequence.
2. On the top in the ribbon, click on the recording option. A drop down will appear.
3. Select web recording option from the dropdown.
4. Open the Google chrome window.
5. Start recording
6. On the chrome window, click on gmail link
7. After clicking on gmail link, the gmail webpage opens. Click on compose.
8. After compose, a box appears to enter the mail.
9. Click on the To: option and enter the email address you want the mail to send to
10. After adding the receiver’s email address, press enter
11. Click on the body of the email and add any text and click on send
12. Press esc and click on save and exit
13. Run the automation and check the output

Output:





**Practical 5:**

**Consider an array of names. We have to find out how many of them start with the letter "a". Create an automation where the number of names starting with "a" is counted and the result is displayed.**

Steps:

1. Create a new sequence.
2. Drag and drop multiple assign activity
3. Create a string array and add the names you wish to add

Names=New String[5]{“Aman”,”Ajay”,”Omkar”,”Ashish”,”Tejas”}

1. Once the array is created, go to the variable tab and change the data type of the array to Array of [T] and select the datatype as string.
2. Once array is created, create a new variable i=0 for iteration and count=0 and select the datatype as Int32 for both the variables.
3. Once done, Drag and drop a while activity from activities panel.
4. Add condition as i<5
5. In the body panel of the while activity, drag and drop an if activity
6. Add the following condition in the if activity:

Names[i].StartsWith(“A”)

1. Drag and drop an assign activity in the “then” panel of the if activity and add

Count=count+1(This will increase the counter by 1 if the condition is true)

1. After the else panel in the if activity, drag and drop an assign activity and add

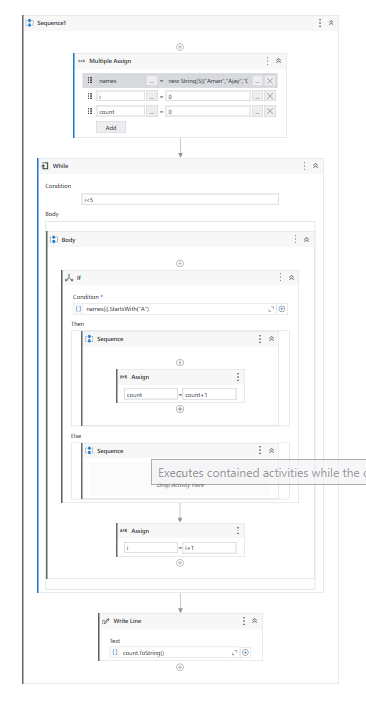
i=i+1

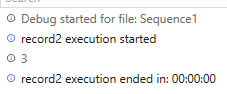
1. Outside the while loop in the main sequence, drag and drop writeline activity and add:

Count.ToString().

1. Run the automation and check the output.

Output:

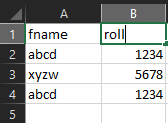


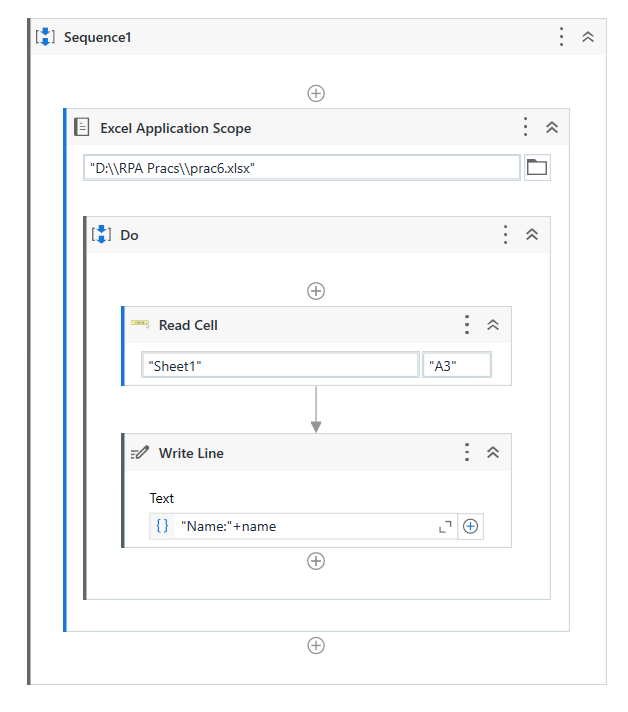


**Practical 6A:**

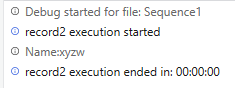
**Create an application automating the read, write and append operation on excel file.**

Steps: Read operation

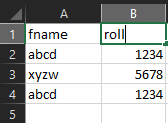
1. Create a new sequence.
2. Create a excel file and add 2 columns of data
3. Drag and drop Excel application scope from the activities panel
4. In the application scope, browse the excel file in the system and add the path of the file.
5. In the DO panel of the excel application scope, Drag and drop read cell activity.
6. It will automatically take the sheet as “Sheet1” and cell as “A1” as the initial point .
7. Right click on the “Sheet1” and select create variable and add a variable as name and add any cell number you want to print.
8. Drag and drop a Writeline activity and add : “Name:”+name
9. Run the automation and check the output.

****

Output: The system will print the data available in cell A3



Steps: Write operation using user input

1. Create a new sequence.
2. Create a excel file and add 2 columns of data
3. Drag and drop Excel application scope from the activities panel
4. In the application scope, browse the excel file in the system and add the path of the file.
5. Drag and drop an input dialog in the DO activity of the excel application scope.
6. Add the following:

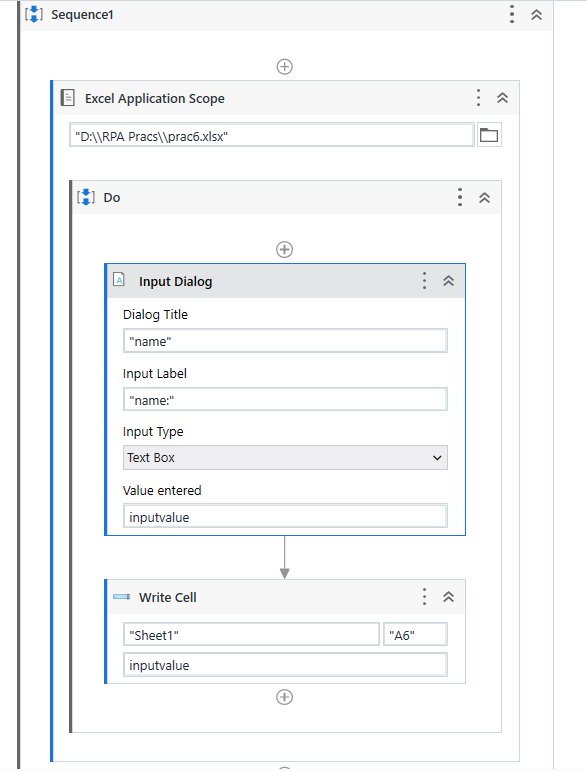
Dialog Title = "name"

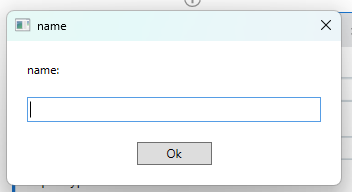
InputLabel = "name:"

Select the Input type as TextBox

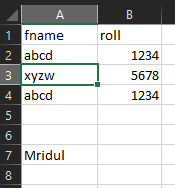
Right click on the Value Entered panel and click on create variable and add a variable “inputvalue”

1. In the DO panel of the excel application scope, Drag and drop write cell activity.
2. It will automatically take the sheet as “Sheet1” and cell as “A1” as the initial point.
3. It will also give a text box to enter the data you wish to write in the excel file. Add the variable name as inputvalue
4. Run the automation and check the output.





Output: This automation will add “Mridul” in cell A7 of excel file by taking user input

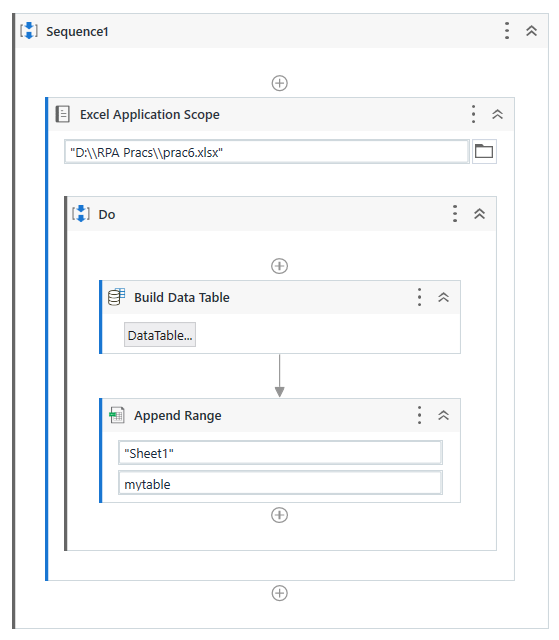
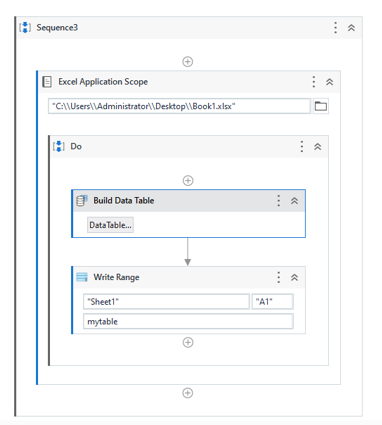


**Practical 6B:**

**Automate the process to extract data from an data table into an excel file**

Steps: Write operation using user input

1. Create a new sequence.
2. Create a excel file and keep the file empty
3. Drag and drop Excel application scope from the activities panel
4. In the application scope, browse the excel file in the system and add the path of the file.
5. Drag and drop Build data table activity and add a datatable and click on create option
6. On the right side of the page, right click on the output option and create a variable “mytable”
7. In the DO panel of the excel application scope, Drag and drop append range / write range activity.
8. It will automatically take the sheet as “Sheet1” and cell as “A1” as the initial point if you use write range.
9. It will also give a text box to enter the data you wish to write in the excel file. Add the variable name as mytable.
10. Run the automation and check the output.

****

Output: This automation will write the data of the data table in the excel file and append the data the number of timer you run the automation.

