PYTHON PROGRAMMING

Lab 18-ANSWERS

HAREESHA H M

AF0364330

1. Suppose you have a dataset containing daily temperature readings for a city, and you want to identify days with extreme temperature conditions. Find days where the temperature either exceeded 35 degrees Celsius (hot day) or dropped below 5 degrees Celsius (cold day).

Input: temperatures = np.array([32.5, 34.2, 36.8, 29.3, 31.0, 38.7, 23.1, 18.5, 22.8, 37.2])

Code:

import numpy as np #importing numpy as np.

temperatures = np.array([32.5, 34.2, 36.8, 29.3, 31.0, 38.7, 23.1, 18.5, 22.8, 37.2]) #inputing temperature.

hot\_days = temperatures > 35 #given condition values to decide hot\_days.

cold\_days = temperatures < 5 #given condition values to decide cold\_days.

extreme\_days = np.logical\_or(hot\_days, cold\_days) # checking the given values whether hot or cold.

print("Days with extreme temperatures:") #printing the extreme temperature.

print(np.where(extreme\_days)[0]) #printing the extreme temperature.

Output:

Days with extreme temperatures:

[2 5 9]

2. Suppose you have a dataset containing monthly sales data for a company, and you want to split this data into quarterly reports for analysis and reporting purposes.

Input: monthly\_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225])

Code:

import numpy as np #importing numpy as np.

monthly\_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225]) # inputing monthly\_sales.

quarterly\_reports = monthly\_sales.reshape(-1, 3).sum(axis=1)

print("Quarterly sales reports:") #printing the final result as quarterly report.

print(quarterly\_reports) #printing the final result as quarterly report.

Output:

Quarterly sales reports:

[403 500 563 633]

3. Suppose you have a dataset containing customer data, and you want to split this data into two groups: one group for customers who made a purchase in the last 30 days and another group for customers who haven't made a purchase in the last 30 days.

 Input: customer\_ids = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110]) last\_purchase\_days\_ago = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 55])

Code:

import numpy as np #importing  umy as py.

first\_array= np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110]) #inserting first\_array list.

second\_array = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 55]) #inserting second\_array list

recent\_customers = customer\_ids[second\_array <= 30] #condition for recent\_customers.

inactive\_customers = customer\_ids[second\_array  > 30] # condtion for inactive\_customers.

print("Customers who made a purchase in the last 30 days:") #printing the recent\_customers.

print(recent\_customers)#printing the recent\_customers.

print("Customers who haven't made a purchase in the last 30 days:")#printing inactive\_customers.

print(inactive\_customers)#printing inactive\_customers.

Output:

Customers who made a purchase in the last 30 days:

[101 102 103 104 105]

Customers who haven't made a purchase in the last 30 days:

[106 107 108 109 110]

4.Suppose you have two sets of employee data—one containing information about full-time employees and another containing information about part-time employees. You want to combine this data to create a comprehensive employee dataset for HR analysis.

Input: # Employee data for full-time employees

 full\_time\_employees = np.array([ [101, 'John Doe', 'Full-Time', 55000], [102, 'Jane Smith', 'Full-Time', 60000], [103, 'Mike Johnson', 'Full-Time', 52000] ])

Code:

import numpy as np # importing numpy as np.

# Employee data for full-time employees.

full\_time\_employees = np.array([[101, 'John Doe', 'Full-Time', 55000],[102, 'Jane Smith', 'Full-Time', 60000],[103, 'Mike Johnson', 'Full-Time', 52000]])

# Employee data for part-time employees.

part\_time\_employees = np.array([[201, 'Alice Brown', 'Part-Time', 25],[202, 'Bob White', 'Part-Time', 30],[203, 'Eva Green', 'Part-Time', 28]])

all\_employees = np.concatenate((full\_time\_employees, part\_time\_employees), axis=1) # Combining full-time and part-time employee data.

print("Combined employee data:") #printing combined employee data.

print(all\_employees) #printing all\_employees.

Output:

Combined employee data:

[['101' 'John Doe' 'Full-Time' '55000' '201' 'Alice Brown' 'Part-Time'

'25']

['102' 'Jane Smith' 'Full-Time' '60000' '202' 'Bob White' 'Part-Time'

'30']

['103' 'Mike Johnson' 'Full-Time' '52000' '203' 'Eva Green' 'Part-Time'

'28']]