Machine Learning - Assignment II

Video Link: https://drive.google.com/file/d/1G4Xx4Fo8b9BsMVo2A6x1PBJaO6aPgQZ1/view?usp=sharing

GitHub Link: https://github.com/Kalyansai6/ML---Assignment-2

**1. Use a python code to display the following star pattern using the for loop**

**#Code\_Below**

**# No. of rows**

rows = 5

for i in range(0, rows):

**# Nested loop for each column**

for j in range(0, i + 1):

**# Printing stars**

print("\*", end=' ')

**# New line after each row**

print("\r")

rows = 4

for i in range(rows + 1, 0, -1):

**# Reversing the nested loop**

for j in range(0, i - 1):

**# Printing stars**

print("\*", end=' ')

print(" ")

**Description:** In the above code firstly I’ve taken 5 rows and by using the nested loop for each column I’ve added stars to it starting from row 1 to row 5 in an increasing order. And after each row I’ve adding a new line; then reversed the pattern by reversing the nested loop, starting from row 4 in a decreasing order to row 1.

Text, whiteboard

Description automatically generated

**2. Use looping to output the elements from a provided list present at odd indexes.**

**my\_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]**

**#Code\_Below**

my\_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

**#printing the above list**

print("The list is :", my\_list)

print("The elements in odd positions are : ")

**# Using loop start from index 1( i.e., 1, 3, 5, 7, 9)**

for i in range(1, len(my\_list), 2):

print(my\_list[i])

**Description**: In the above source code, I’ve taken random values. Then using the for loop starting from the index 0, I picked the odd positions and printed the values (i.e., 20, 40, 60, 80, 100).

Text

Description automatically generated

**3. Write a code that appends the type of elements from a given list.**

**Input**

**x = [23, ‘Python’, 23.98]**

**Expected output.**

**[23, 'Python', 23.98]**

**[<class 'int'>, <class 'str'>, <class 'float'>]**

**#Code\_Below**

x = [23, 'Python', 23.98]

**# Printing x list**

print(x)

**# Printing all the type of the elements from the given x list**

print([type(i) for i in x])

**Description**: In the above source code, I’ve taken different datatype elements and stored them in x list and printed them. I’ve used type() function to define the element datatype and printed the datatype.

**Graphical user interface, text, application

Description automatically generated**

**4. Write a function that takes a list and returns a new list with unique items of the first list.**

**Sample List: [1,2,3,3,3,3,4,5]  
Unique List: [1, 2, 3, 4, 5]**

**#Code\_Below**

**# Defining Unique\_list()**

def unique\_list(l):

x = []

for a in l:

**# If a not present in x then append a to x**

if a not in x:

x.append(a)

**# Return x elements**

return x

**# Printing Sample list**

sample\_list=[1,2,3,3,3,3,4,5]

print("Sample List :", sample\_list)

**# Printing Unique list of values from Sample list**

print("Unique List :",unique\_list(sample\_list))

**Description:** In the above source code, first I’ve defined unique\_list() which will be used later. Then I’ve taken some random values as a Sample list. Using for loop from a in I and checking if a is not in x. If so then appending a to x and then used the unique\_list() which I have defined at beginning to print the unique values from the sample list.

Text

Description automatically generated

**5. Write a function that accepts a string and calculate the number of upper-case letters and lower-case letters.**

**Input String: 'The quick Brow Fox'**

**Expected Output:**

**No. of Upper-case characters: 3**

**No. of Lower-case Characters: 12**

**#Code\_Below**

**# Defining String\_test()**

def string\_test(a):

b={"UPPER\_CASE":0, "LOWER\_CASE":0}

for c in a:

**# If c is Uppercase then count**

if c.isupper():

b["UPPER\_CASE"]+=1

**# Else if c is Lowercase then count otherwise pass**

elif c.islower():

b["LOWER\_CASE"]+=1

else:

pass

**# Print the input string**

print ("Input String : ", a)

**# Print the count of Uppercase**

print ("No. of Upper-case characters : ", b["UPPER\_CASE"])

**# Print the count of lowercase**

print ("No. of Lower-case Characters : ", b["LOWER\_CASE"])

string\_test('The quick Brow Fox')

**Description**: In the above source code, I’ve defined string\_test() which will be used later. Using for loop from c in a and checking if c is uppercase or lowercase, otherwise just pass it. Post that I’ve used string\_test() which I have defined at beginning to print count of uppercase and lowercase from the given string.

Text

Description automatically generated