Name : Surya Krishna Bharadwaj Kunapuli

Registration Number: BL.EN.U4AIE22154

Course Code: 22AIE213

Course: Machine Learning

Submitted to: Mr. Marimganti Srinivas (CSE)

ASSIGNMENT-1

Question 1:

Write a program to count the number of vowels and consonants present in an input string.

Code:

def count\_vowels\_consonants(input\_str):

vowels = "aeiouAEIOU"

count\_vowels = sum(1 for char in input\_str if char in vowels)

count\_consonants = sum(1 for char in input\_str if char.isalpha() and char not in vowels)

return count\_vowels, count\_consonants

input\_string = input("Enter a string: ")

result = count\_vowels\_consonants(input\_string)

print(f"Number of vowels: {result[0]}")

print(f"Number of consonants: {result[1]}")

Algorithm:

1. Initialize counters for vowels and consonants.
2. Iterate through each character in the input string.
3. Check if the character is a vowel or a consonant.
4. Update the respective counter accordingly.
5. Return the counts of vowels and consonants.

PseudoCode:

count\_vowels\_consonants(input\_str):

vowels = "aeiouAEIOU"

count\_vowels = 0

count\_consonants = 0

for char in input\_str:

if char is a vowel:

increment count\_vowels

else if char is an alphabet:

increment count\_consonants

return count\_vowels, count\_consonants

Question 2:

Write a program that accepts two matrices A and B as input and returns their product AB. Check if A & B are multipliable; if not, return error message.

Code:

def matrix\_multiply(A, B):

if len(A[0]) != len(B):

return "Error: Matrices A and B are not multipliable."

result = [[0 for \_ in range(len(B[0]))] for \_ in range(len(A))]

for i in range(len(A)):

for j in range(len(B[0])):

for k in range(len(B)):

result[i][j] += A[i][k] \* B[k][j]

return result

# Example matrices A and B

matrix\_A = [[1, 2, 3], [4, 5, 6]]

matrix\_B = [[7, 8], [9, 10], [11, 12]]

result\_matrix = matrix\_multiply(matrix\_A, matrix\_B)

print(result\_matrix)

Algorithm:

1. Check if the number of columns in matrix A is equal to the number of rows in matrix B.
2. If not, return an error message.
3. Initialize a result matrix with dimensions (rows of A) x (columns of B).
4. Perform matrix multiplication using three nested loops.
5. Return the resulting matrix.

Pseudocode:

def matrix\_multiply(A, B):

if len(A[0]) != len(B):

return "Error: Matrices A and B are not multipliable."

result = [[0 for \_ in range(len(B[0]))] for \_ in range(len(A))]

for i in range(len(A)):

for j in range(len(B[0])):

for k in range(len(B)):

result[i][j] += A[i][k] \* B[k][j]

return result

Question 3:

Write a program to find the number of common elements between two lists. The lists contain integers.

Code:

def common\_elements(list1, list2):

return len(set(list1) & set(list2))

# Example lists

list\_a = [1, 2, 3, 4, 5]

list\_b = [3, 4, 5, 6, 7]

common\_count = common\_elements(list\_a, list\_b)

print(f"Number of common elements: {common\_count}")

Algorithm:

1. Convert both lists to sets to easily find the intersection (common elements).
2. Find the intersection of the two sets.
3. Return the count of common elements.

Pseudocode:

common\_elements(list1, list2):

set1 = convert list1 to set

set2 = convert list2 to set

common\_set = intersection of set1 and set2

return size of common\_set

Question 4:

Write a program that accepts a matrix as input and returns its transpose.

Code:

def transpose\_matrix(matrix):

return [[matrix[j][i] for j in range(len(matrix))] for i in range(len(matrix[0]))]

# Example matrix

example\_matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

transposed\_matrix = transpose\_matrix(example\_matrix)

print(transposed\_matrix)

Algorithm:

1. Initialize a new matrix with dimensions (columns of original) x (rows of original).
2. Use nested loops to iterate through each element in the original matrix.
3. Transpose the elements and place them in the corresponding position in the new matrix.
4. Return the transposed matrix.

Pseudocode:

transpose\_matrix(matrix):

new\_matrix = create matrix with dimensions (columns of original) x (rows of original)

for i from 0 to rows of original matrix:

for j from 0 to columns of original matrix:

new\_matrix[j][i] = original\_matrix[i][j]

return new\_matrix