Spring 2024: CS5720 Neural Networks & Deep Learning - ICP-1

Assignment-1

NAME: Lakkireddy Sriram Reddy

STUDENT ID:700758340

Github link: https://github.com/sriram7040/Neural-network-and-deep-learning/tree/main

Video link: https://drive.google.com/file/d/1ckeTR-1DKQQDQbiN-nyoXaUySRAlo0Ag/view?usp=drive_link

1. Write a program that takes two stringsfrom the user: first_name, last_name. Passthese variablesto fullname function that should return the (full name).

+ Code + Text

30s

```
def get_fullname(first_name, last_name):
    # Concatenate the first name and last name with a space in between
    return first_name + " " + last_name

def alternate_characters(input_string):
    # Extract every second character from the input string
    return input_string[::2]

# Prompt the user for their first name
first_name = input("Enter your first name: ")

# Prompt the user for their last name
last_name = input("Enter your last name: ")

# Combine the first and last name to form the full name
full_name = get_fullname(first_name, last_name)

# Display the full name to the user
print("Full Name: " + full_name)

# Display the full name with every alternate character
print("Alternate characters in your full name: " + alternate_characters(full_name))
```

```
Enter your first name: Sriram Reddy
Enter your last name: Lakkireddy
Full Name: Sriram Reddy Lakkireddy
Alternate characters in your full name: Sia ed akrdy
```

2. Write a python program to find the wordcount in a file (input.txt) for each line and then print the output. o Finally store the output in output.txt file.

```
def count_words(text):
    # Initialize an empty dictionary to store word frequencies
    word_frequencies = {}

    # Split the input text into individual words
    words = text.split()

    # Count the occurrences of each word
    for word in words:
        if word in word_frequencies:
            word_frequencies[word] += 1
        else:
            word_frequencies[word] = 1

    # Return the dictionary containing word counts
    return word_frequencies

# Read the content of the input file
with open('input.txt', 'r') as file:
    input_text = file.read()

# Compute word counts for the text
word_counts = count_words(input_text)

# Prepare the formatted output
output_text = input_text + "\nWord Count:\n"
output_text += str(word_counts).replace("{", "").replace("}", "")
output_text = output_text.replace("'", "")
```

```
# Prepare the formatted output
output_text = input_text + "\nWord Count:\n"
output_text += str(word_counts).replace("{", "").replace("}", "")
output_text = output_text.replace("'", "")
output_text = output_text.replace(", ", "\n")

# Print the final processed output
print(output_text)

# Write the processed output to a new file
with open('output.txt', 'w') as output_file:
    output_file.write(output_text)
```

```
Python Course
Deep Learning Course
Word Count:
Python: 1
Course: 2
Deep: 1
Learning: 1
```

3. Write a program, which reads heights (inches.) of customersinto a list and convert these heights to centimeters in a separate list using:
   1) Nested Interactive loop.
   2) List comprehensions

```
# Initialize empty lists to store heights in inches and centimeters
heights_in_inches = []
heights_in_cm = []

# Input: Number of customer heights to be entered
num_customers = int(input("Enter the number of customers: "))

# Collect heights in inches using a loop
print("Enter the heights (in inches):")
for _ in range(num_customers):
    height = float(input())
    heights_in_inches.append(height)

# Method 1: Conversion using a nested interactive loop
print("\nUsing Nested Interactive Loop:")
for height in heights_in_inches:
    heights_in_cm.append(round(height * 2.54, 2))
print("Heights in centimeters:", heights_in_cm)

# Method 2: Conversion using list comprehensions
print("\nUsing List Comprehensions:")
converted_heights = [round(height * 2.54, 2) for height in heights_in_inches]
print("Heights in centimeters:", converted_heights)
```

```
Enter the number of customers: 2
Enter the heights (in inches):
12
13

Using Nested Interactive Loop:
Heights in centimeters: [30.48, 33.02]

Using List Comprehensions:
Heights in centimeters: [30.48, 33.02]
```