

Neural Network Deep Learning

Assignment – 1

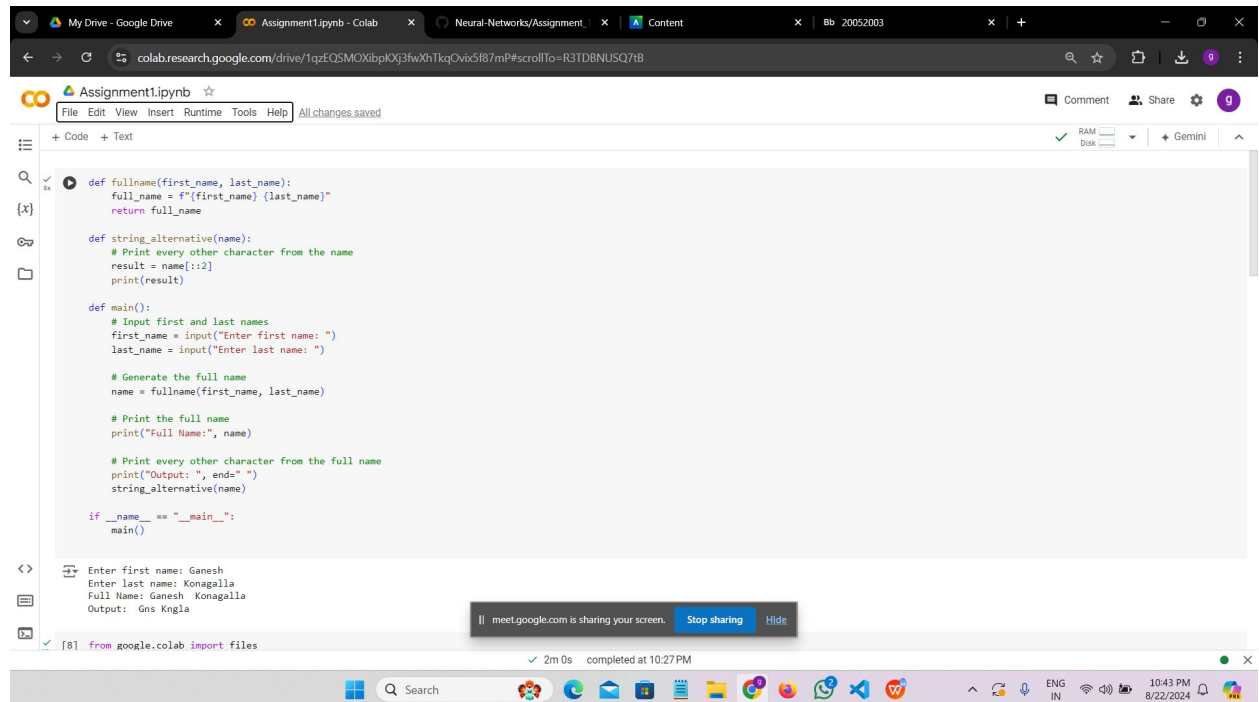
Name: Ganesh konagalla

Student ID: 700756412

Github Link :https://github.com/ganeshkonagalla123/Neural-Networks/tree/main/Assignment_1

1. Write a program that takes two strings from the user: `first_name`, `last_name`. Pass these variables to `fullname` function that should return the (full name).
 - o For example: ▪ `First_name` = “your first name”, `last_name` = “your last name” ▪ `Full_name` = “your full name”

Note: You need to create a function named “`string_alternative`” for this program and call it from main function.



```
def fullname(first_name, last_name):
    full_name = f"{first_name} {last_name}"
    return full_name

def string_alternative(name):
    # Print every other character from the name
    result = name[::2]
    print(result)

def main():
    # Input first and last names
    first_name = input("Enter first name: ")
    last_name = input("Enter last name: ")

    # Generate the full name
    name = fullname(first_name, last_name)

    # Print the full name
    print("Full Name:", name)

    # Print every other character from the full name
    print("Output: ", end=" ")
    string_alternative(name)

if __name__ == "__main__":
    main()
```

Enter first name: Ganesh
Enter last name: Konagalla
Full Name: Ganesh Konagalla
Output: Gns Knglla

meet.google.com is sharing your screen. Stop sharing Hide

[8] from google.colab import files

2m 0s completed at 10:27 PM

10:43 PM 8/22/2024

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.

Example: Input: a file includes two lines: Python Course Deep Learning Course

Output: Python Course Deep Learning Course

Word_Count:

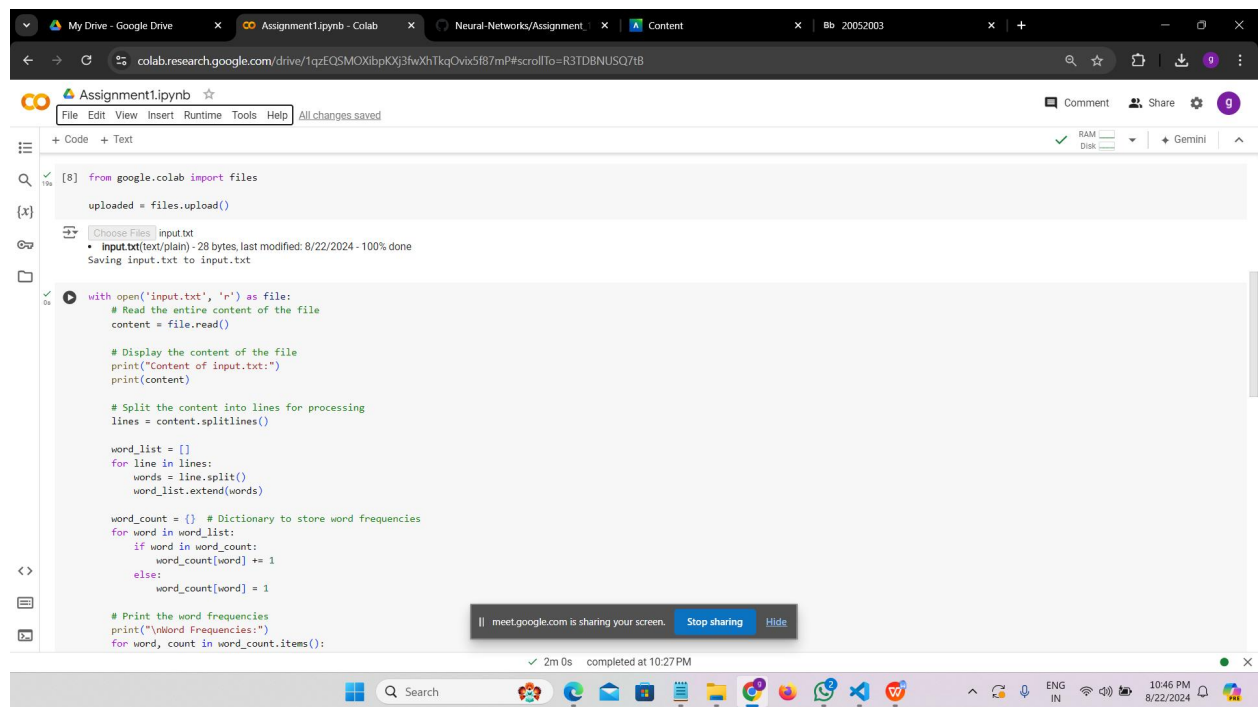
Python: 1

Course: 2

Deep: 1

Learning: 1

In the below Code snippet, we are replacing python with python3



The screenshot shows a Google Colab notebook titled 'Assignment1.ipynb'. The notebook contains a Python script that reads a file named 'input.txt', splits its content into lines, and then counts the frequency of each word. The script uses a dictionary to store the word frequencies. The output of the script is printed to the console.

```
[8] from google.colab import files

uploaded = files.upload()

Choose Files input.txt
• input.txt(text/plain) - 28 bytes, last modified: 8/22/2024 - 100% done
Saving input.txt to input.txt

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

    # Display the content of the file
    print("Content of input.txt:")
    print(content)

    # Split the content into lines for processing
    lines = content.splitlines()

    word_list = []
    for line in lines:
        words = line.split()
        word_list.extend(words)

    word_count = {} # Dictionary to store word frequencies
    for word in word_list:
        if word in word_count:
            word_count[word] += 1
        else:
            word_count[word] = 1

    # Print the word frequencies
    print("\nWord Frequencies:")
    for word, count in word_count.items():
```

The notebook interface shows the code is executed successfully, with a status bar indicating '2m 0s completed at 10:27 PM'. The bottom of the screen shows a Windows taskbar with various application icons and the system clock displaying 10:46 PM on 8/22/2024.

My Drive - Google Drive x Assignment1.ipynb - Colab x Neural-Networks/Assignment. x Content x 20052003 x +

colab.research.google.com/drive/1qzEQSMOXibpKXj3fwXhTkqOvi5f87mP#scrollTo=R3TDBNUSQ7tB

Assignment1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk + Gemini

```
[9] with open('input.txt', 'r') as file:
    # Read the entire content of the file
    content = file.read()

    # Display the content of the file
    print("Content of input.txt:")
    print(content)

    # Split the content into lines for processing
    lines = content.splitlines()

    word_list = []
    for line in lines:
        words = line.split()
        word_list.extend(words)

    word_count = {} # Dictionary to store word frequencies
    for word in word_list:
        if word in word_count:
            word_count[word] += 1
        else:
            word_count[word] = 1

    # Print the word frequencies
    print("\nWord Frequencies:")
    for word, count in word_count.items():
        print(f"{word}: {count}")
```

<> Content of input.txt:
Deep Learning
Python course

Word Frequencies:
Deep: 1
Learning: 1
Python: 1

meet.google.com is sharing your screen. Stop sharing Hide

2m 0s completed at 10:27 PM

Search

ENG IN 10:46 PM 8/22/2024

My Drive - Google DriveAssignment1.ipynb - ColabNeural-Networks/AssignmentContent20052003

colab.research.google.com/drive/1qzEQSMOXibpKXj3fwXhTkqOvi5f87mP#scrollTo=R3TDBNUSQ7tB

Assignment1.ipynb

File Edit View Insert Runtime Tools HelpAll changes saved

150%ResetCommentShare

+ Code+ Text

RAMDisk

Gemini

0s

```
word_count[word] = 1

# Print the word frequencies
print("\nWord Frequencies:")
for word, count in word_count.items():
    print(f"{word}: {count}")
```

Content of input.txt:
Deep Learning
Python course

Word Frequencies:
Deep: 1
Learning: 1
Python: 1
course: 1

2m

```
def cm(inches):
```

meet.google.com is sharing your screen. Stop sharingHide

2m 0s completed at 10:27 PM

Search

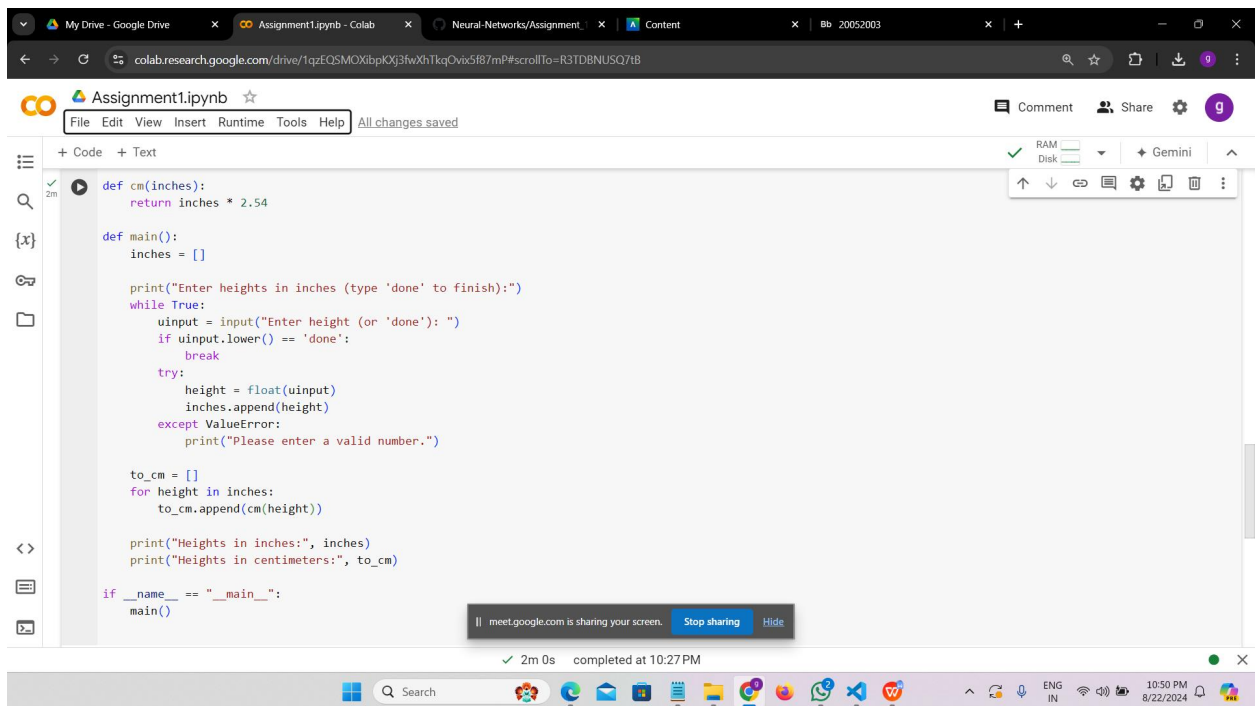
ENG IN

10:47 PM 8/22/2024

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

In the below Code snippet, we are calculating the height of the customer into centimeters using two approaches.

Nested Interactive Loop:

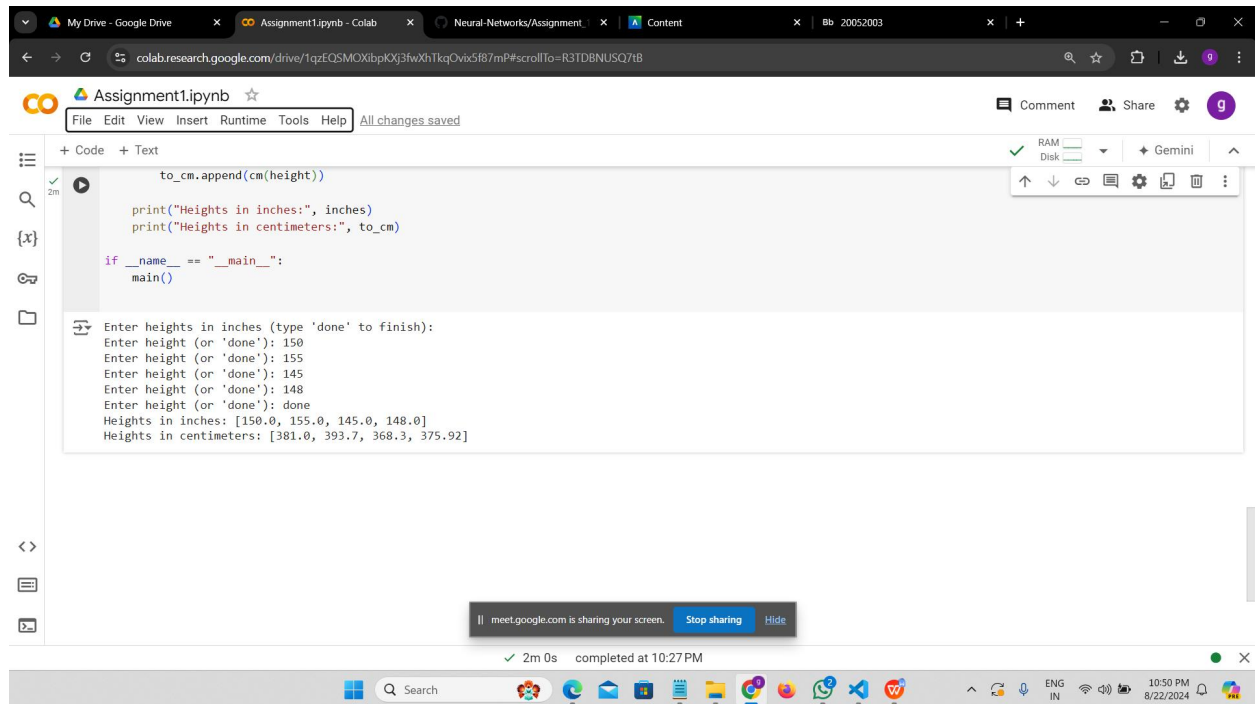


The screenshot shows a Google Colab notebook titled "Assignment1.ipynb". The code defines a function `cm(inches)` that returns `inches * 2.54`. The `main()` function prompts the user to enter heights in inches until they type "done". It then converts these heights to centimeters using a nested loop and a list comprehension. The output shows the heights in inches and the corresponding heights in centimeters.

```
def cm(inches):  
    return inches * 2.54  
  
def main():  
    inches = []  
  
    print("Enter heights in inches (type 'done' to finish):")  
    while True:  
        uinput = input("Enter height (or 'done'): ")  
        if uinput.lower() == 'done':  
            break  
        try:  
            height = float(uinput)  
            inches.append(height)  
        except ValueError:  
            print("Please enter a valid number.")  
  
    to_cm = []  
    for height in inches:  
        to_cm.append(cm(height))  
  
    print("Heights in inches:", inches)  
    print("Heights in centimeters:", to_cm)  
  
if __name__ == "__main__":  
    main()
```

2m 0s completed at 10:27 PM

Output:



The screenshot displays a Google Colab notebook titled "Assignment1.ipynb". The code cell contains the following Python code:

```
to_cm.append(cm(height))

print("Heights in inches:", inches)
print("Heights in centimeters:", to_cm)

if __name__ == "__main__":
    main()
```

The output of the code execution is as follows:

```
Enter heights in inches (type 'done' to finish):
Enter height (or 'done'): 150
Enter height (or 'done'): 155
Enter height (or 'done'): 145
Enter height (or 'done'): 148
Enter height (or 'done'): done
Heights in inches: [150.0, 155.0, 145.0, 148.0]
Heights in centimeters: [381.0, 393.7, 368.3, 375.92]
```

The interface also shows a status bar at the bottom indicating "2m 0s completed at 10:27 PM".