

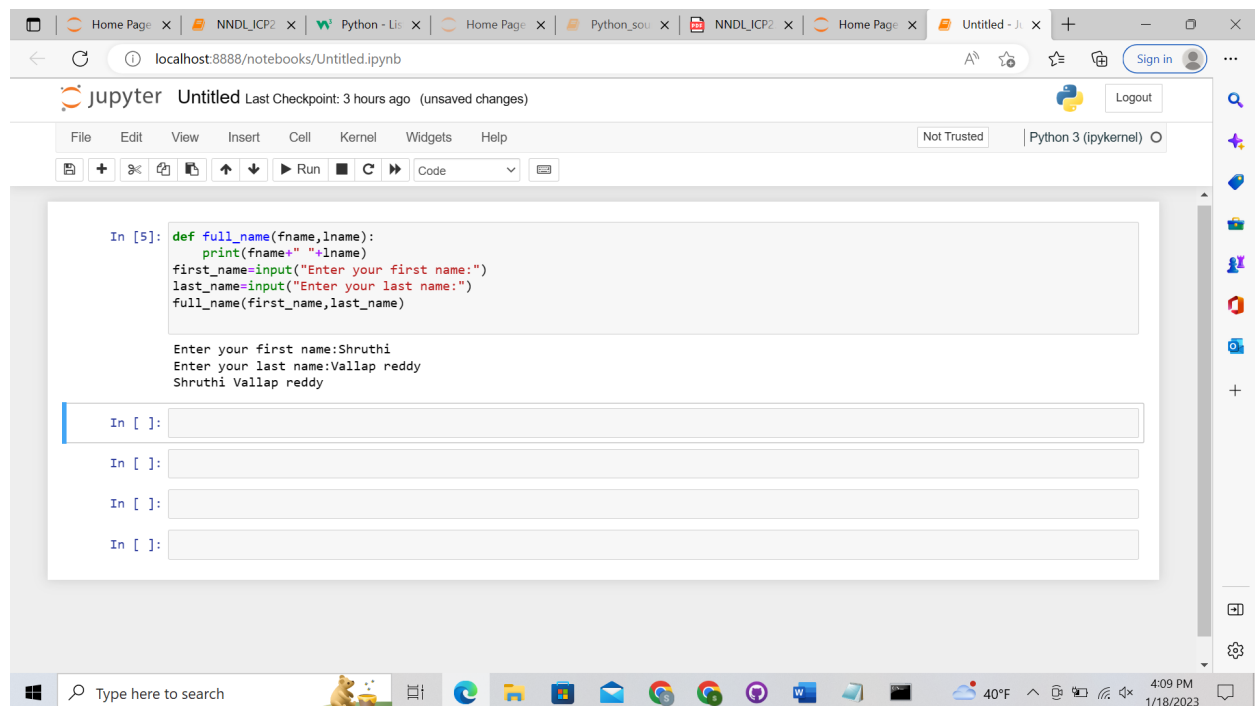
Spring 2023: CS5520
Neural Networks and Deep Learning - ICP-2

1) a) Write a program that takes two strings from the user: first_name, last_name. Pass these variables to the 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"

Ans :



The screenshot shows a Jupyter Notebook running in a web browser at localhost:8888. The notebook has a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, undo, redo, and running code. The main area contains a code cell with the following Python code:

```
In [5]: def full_name(fname,lname):  
        print(fname+" "+lname)  
        first_name=input("Enter your first name:")  
        last_name=input("Enter your last name:")  
        full_name(first_name,last_name)  
  
        Enter your first name:Shruthi  
        Enter your last name:Vallap reddy  
        Shruthi Vallap reddy
```

Below the code cell, there are four empty input fields for the next steps.

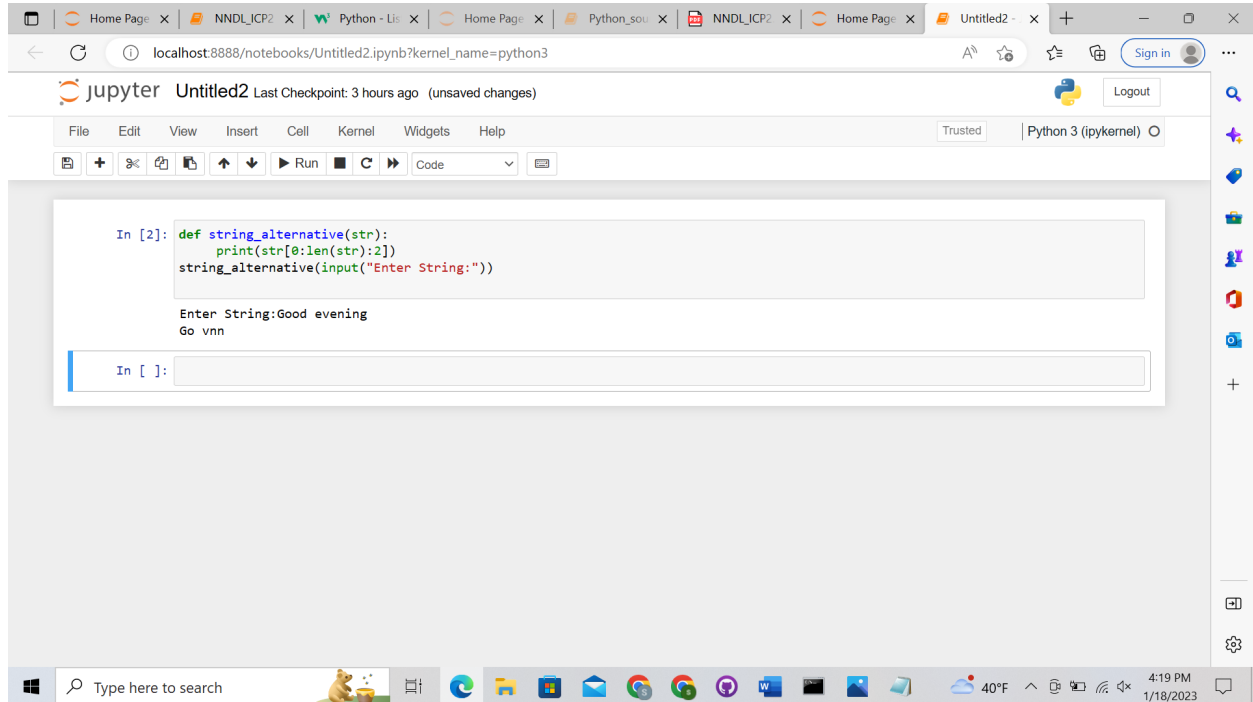
1. Created the function full_name() which accepts two parameters fname and lname.
2. The variable first_name reads the first name entered by the user
3. The variable last_name reads the last name entered by the user
4. The full_name() function is called in the main method and first_name and last_name parameters are passed to the full_name() function.
5. The full_name() functions concatenates the parameters passed to it and prints the output.

- Sample input
Enter your first name : Shruthi
Enter your last name : Vallap reddy
- Sample output
Shruthi Vallap Reddy

b) Write a function named “string_alternative” that returns every other char in the full_name string.

- Str = “Good evening”
- Output: Go vnn

Ans



The screenshot shows a Jupyter Notebook titled 'Untitled2' running on a local host. The notebook contains a Python function named 'string_alternative' that takes a string 'str' as input, prints every other character using slicing, and then calls itself with user input. The input 'Good evening' is entered, and the output 'Go vnn' is displayed. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar at the bottom showing system information like temperature and time.

```
In [2]: def string_alternative(str):  
        print(str[0:len(str):2])  
        string_alternative(input("Enter String:"))  
  
Enter String:Good evening  
Go vnn  
  
In [ ]:
```

- 1) Created the function string_alternative() which accepts the parameter string str
- 2) The string_alternative() function is called in the main method and the string from the input console is passed to the function
- 3) The string_alternative() function will skip every other alternative character in the string and print the output.

- Sample Input

Enter String : Shruthi Vallapreddy

- Sample Output

SrtiVlardy

2. Write a python program to find the word count in a file (input.txt) for each line and then print the output.

o Finally store the output in the output.txt file.

- Example:

Input: a file includes two lines:

Python Course

Deep Learning Course

Output:

Python Course

Deep Learning Course

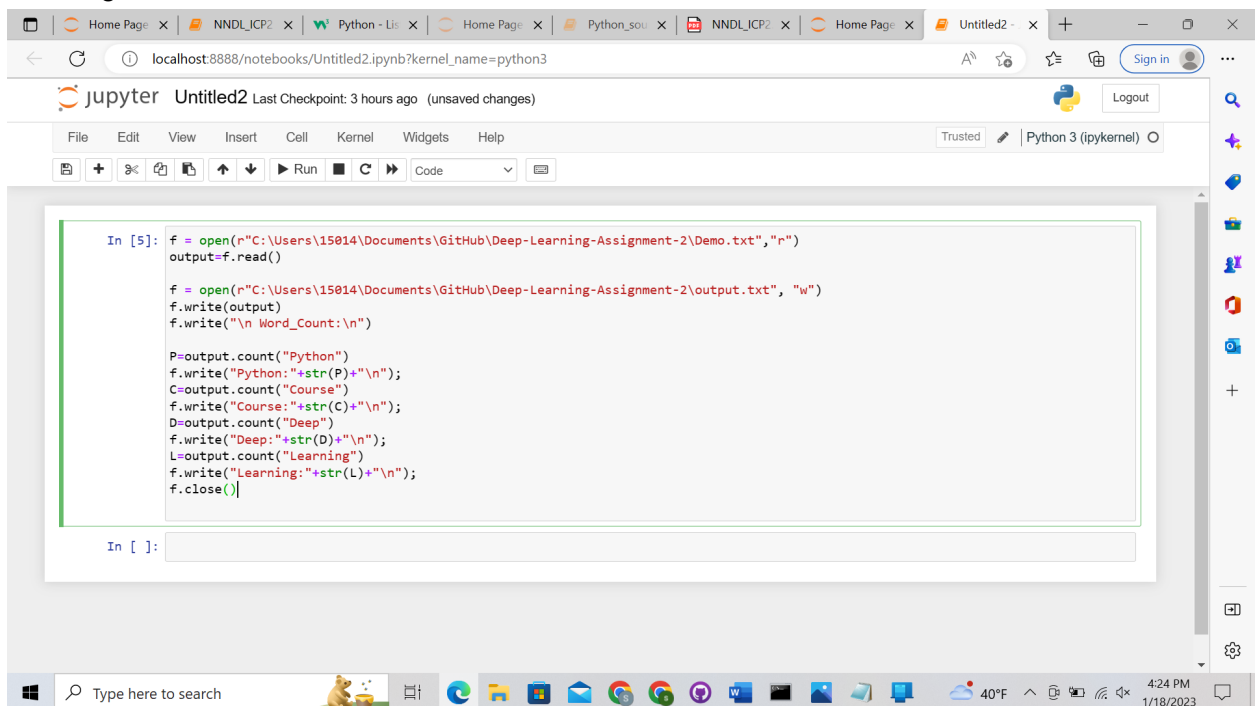
Word_Count:

Python: 1

Course: 2

Deep: 1

Learning: 1



The screenshot shows a Jupyter Notebook titled 'Untitled2' running on a local host. The notebook contains a single code cell with the following Python code:

```
In [5]: f = open(r"C:\Users\15014\Documents\GitHub\Deep-Learning-Assignment-2\Demo.txt", "r")
        output=f.read()

        f = open(r"C:\Users\15014\Documents\GitHub\Deep-Learning-Assignment-2\output.txt", "w")
        f.write(output)
        f.write("\n Word_Count:\n")

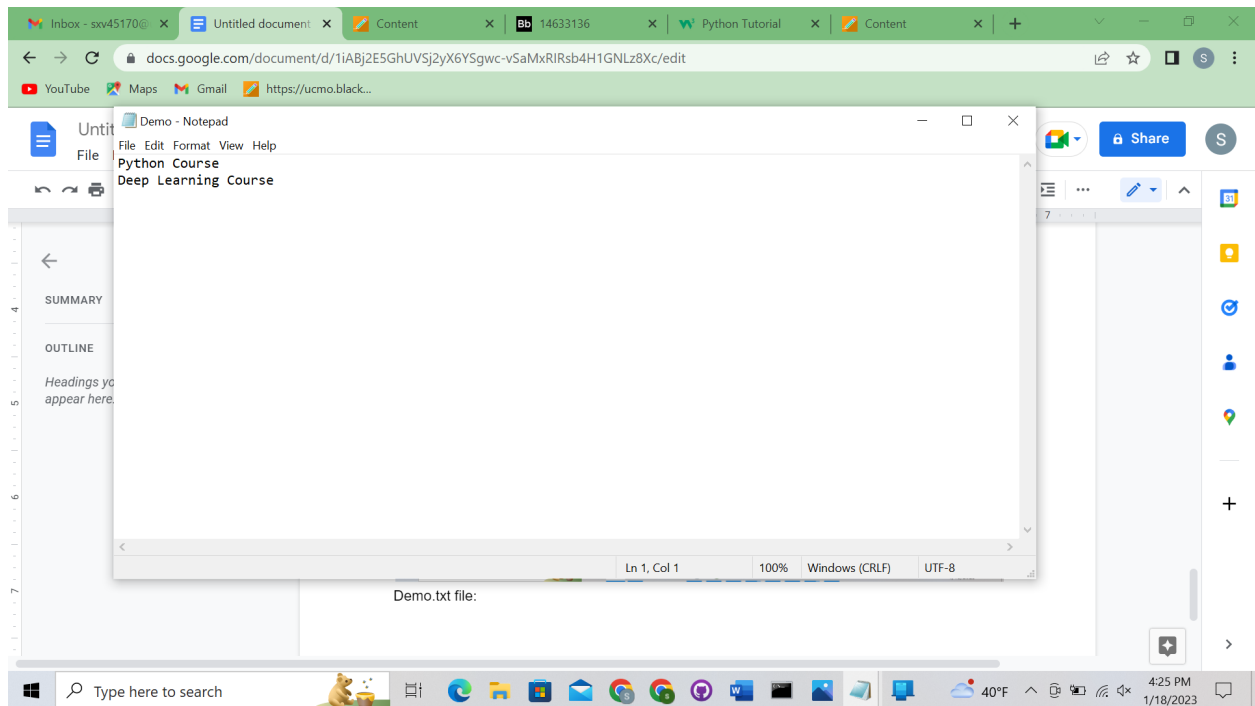
        P=output.count("Python")
        f.write("Python:"+str(P)+"\n");
        C=output.count("Course")
        f.write("Course:"+str(C)+"\n");
        D=output.count("Deep")
        f.write("Deep:"+str(D)+"\n");
        L=output.count("Learning")
        f.write("Learning:"+str(L)+"\n");
        f.close()
```

The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar at the bottom showing the system clock and temperature.

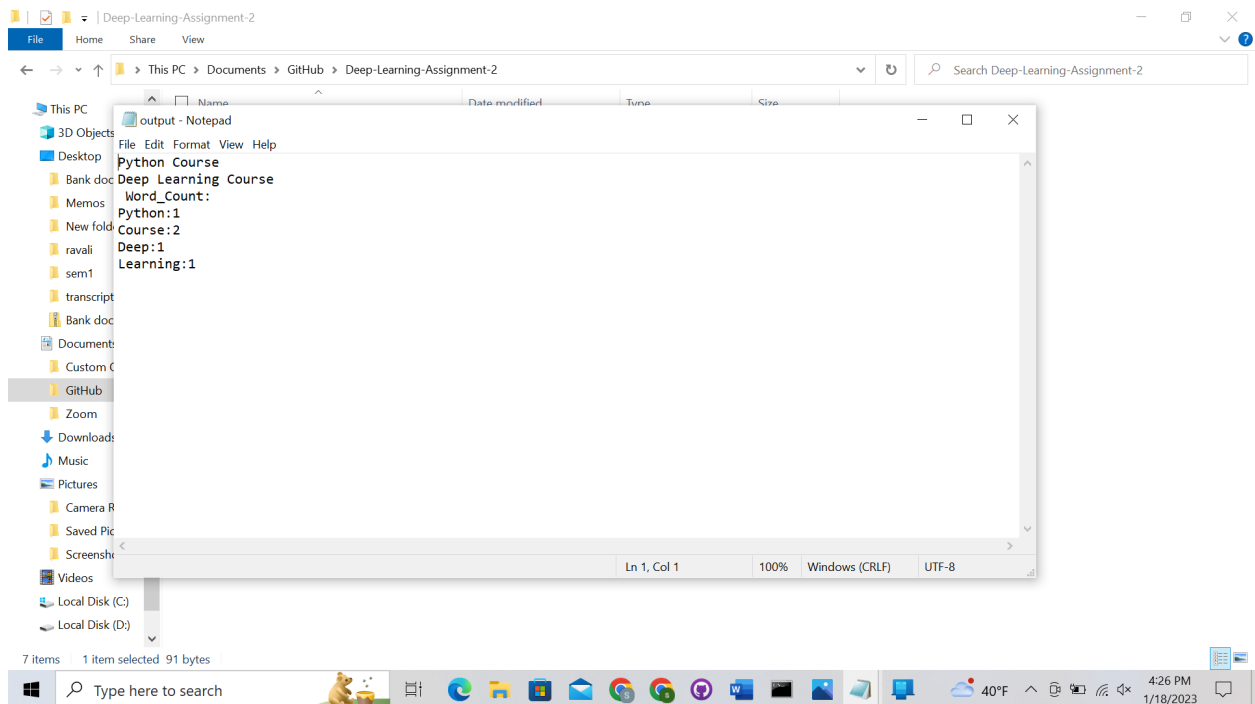
- 1) There is a Demo.txt file in location "C:\Users\15014\Documents\GitHub\Deep-Learning-Assignment-2\Demo.txt" which has the input text for the program.
- 2) The output variable reads the text inside the Demo.txt file and stores it.
- 3) Then we need to write the count of each word of the text inside Demo.txt into the new Output.txt file.
- 4) So we need to open the output.txt file at the location "C:\Users\15014\Documents\GitHub\Deep-Learning-Assignment-2\output.txt" in the write mode.
- 5) So here we used the count method of type string which returns occurrences of substring in string, count(substring, start=..., end=..) to print the word count

- 6) The Demo.txt file contains the input and the Output.txt file contains the output as shown below.

Demo.txt file:



Output.txt



3. Write a program, which reads heights (inches.) customers into a list and convert these heights to centimeters in a separate list using:

1) Nested Interactive loop.

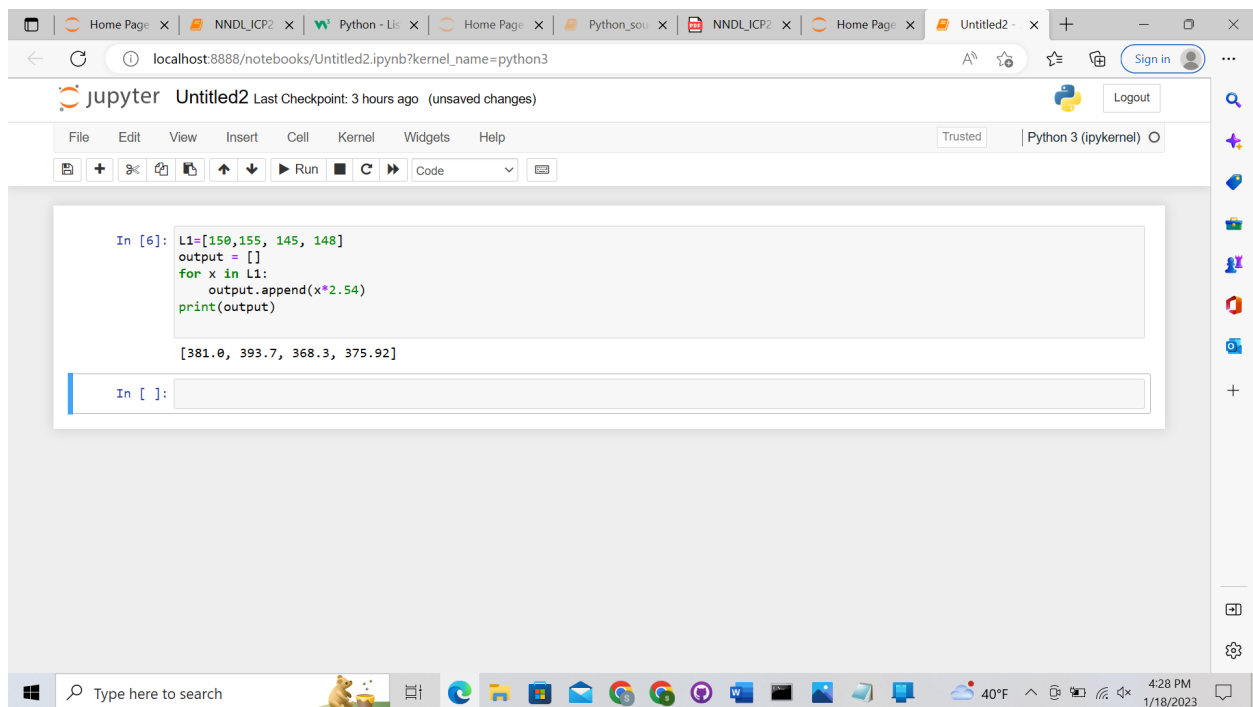
2) List comprehensions

Example:

L1: [150,155, 145, 148]

Output: [68.03, 70.3, 65.77, 67.13]

1) Nested Interactive loop



```
In [6]: L1=[150,155, 145, 148]
output = []
for x in L1:
    output.append(x*2.54)
print(output)

[381.0, 393.7, 368.3, 375.92]
```

- 1) Have initialized the List named Li with the input as [150,155,145,148]
- 2) And also created the empty list with the name output.
- 3) Here we have used the for loop to convert each and every element in the List Li from inches to centimeters.
- 4) To convert inches to centimeters we need to multiply the inches with 2.54
1 inch = 2.54 centimeter
- 5) Each and every element in the Li list is multiplied with 2.54 and appended to List output simultaneously.
- 6) Finally the output is printed to the console using print function.

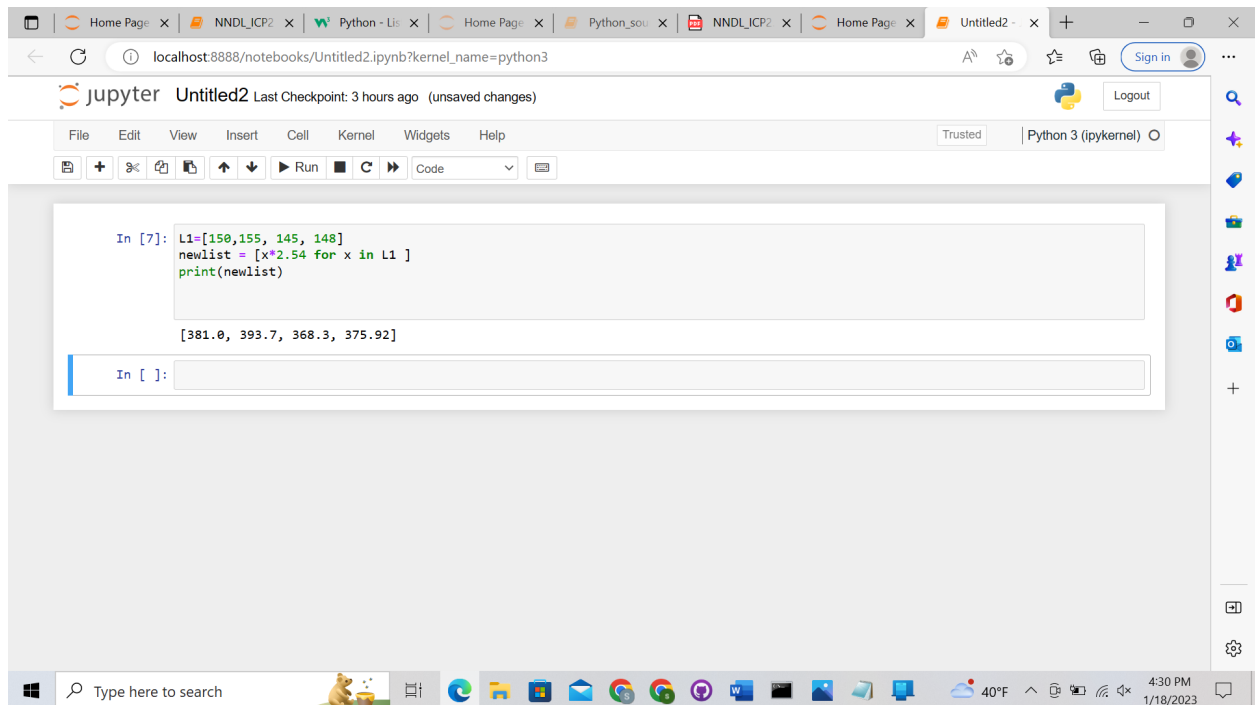
- Sample Input

L1: [150,155, 145, 148]

- Sample Output

[381.0, 393.7, 368.3, 375.92]

2) List comprehensions



The screenshot shows a Jupyter Notebook titled 'Untitled2' running on a local host. The notebook contains a single code cell with the following Python code:

```
In [7]: L1=[150,155, 145, 148]
        newlist = [x*2.54 for x in L1 ]
        print(newlist)
```

The output of the code cell is displayed below the code:

```
[381.0, 393.7, 368.3, 375.92]
```

The Jupyter Notebook interface includes a top toolbar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The bottom status bar shows the system clock as 4:30 PM on 1/18/2023.

- 1) Have initialized the List named L1 with the input as [150,155,145,148]
- 2) Here we have used List comprehensions which offer a shorter syntax when you want to create a new list based on the values of an existing list.
- 3) So the new values get assigned to the newlist [].
- 4) Finally the print function prints the output to the console

- Sample Input

L1: [150,155, 145, 148]

- Sample Output

[381.0, 393.7, 368.3, 375.92]

Git repo link : <https://github.com/ShruthiVallapReddy/NNDL-Assignment-2.git>