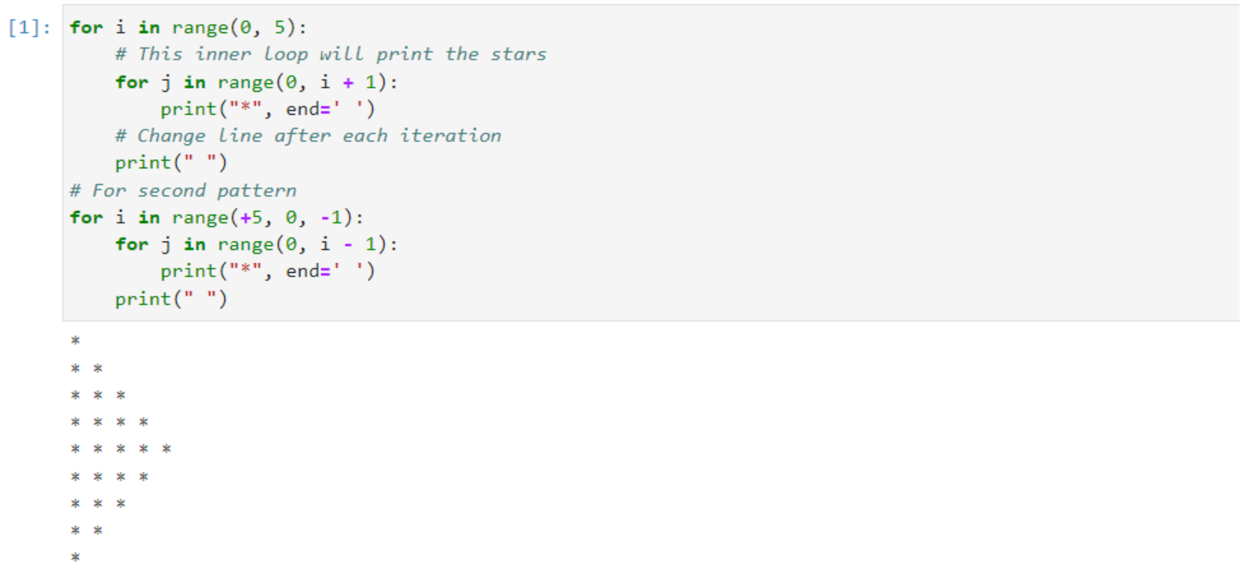
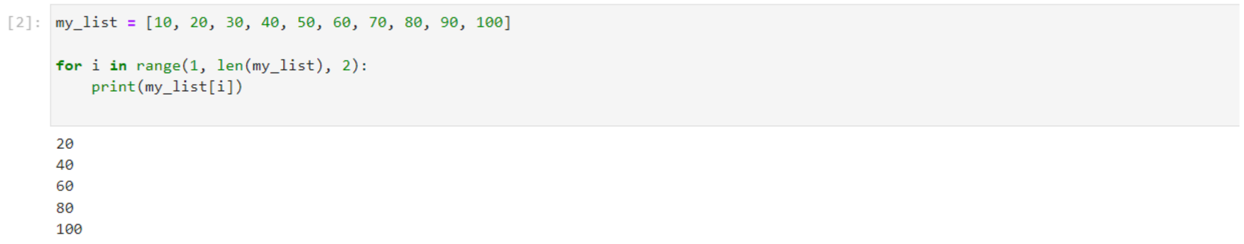
Machine Learning (Assignment # 2)



```
[1]: for i in range(0, 5):
         # This inner loop will print the stars
         for j in range(0, i + 1):
             print("*", end=' ')
         # Change line after each iteration
         print(" ")
     # For second pattern
     for i in range(+5, 0, -1):
         for j in range(0, i - 1):
             print("*", end=' ')
         print(" ")
```

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

This code will print two pyramid patterns of asterisks, where the first one will print 5 rows of asterisks, increasing from 1 to 5, and the second one will print 5 rows of asterisks, decreasing from 5 to 1.



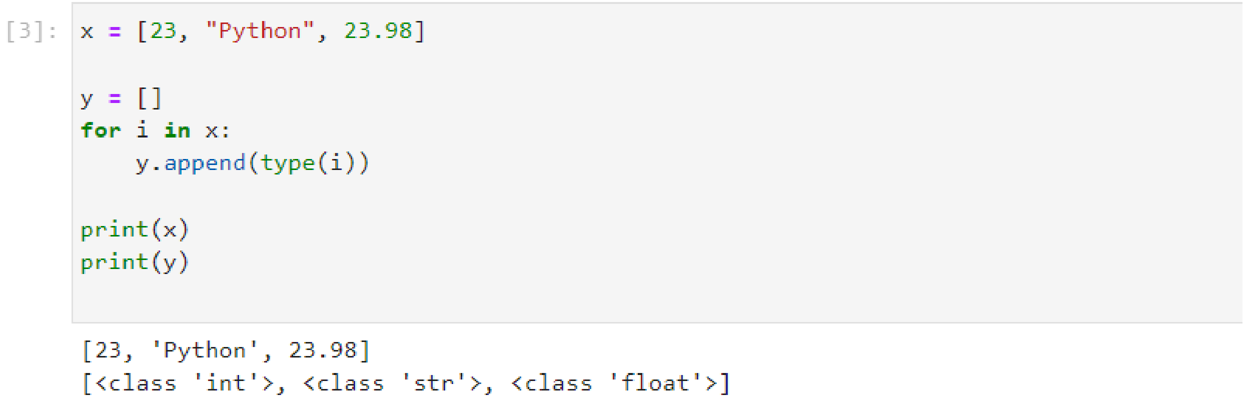
```
[2]: my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

     for i in range(1, len(my_list), 2):
         print(my_list[i])
```

```
20
40
60
80
100
```

This code will output the elements of the ‘my_list’ that are located at odd index positions. Here I used a ‘for’ loop with the ‘range’ function to iterate over the list, and the ‘print’ statement to output the elements.

Machine Learning (Assignment # 2)



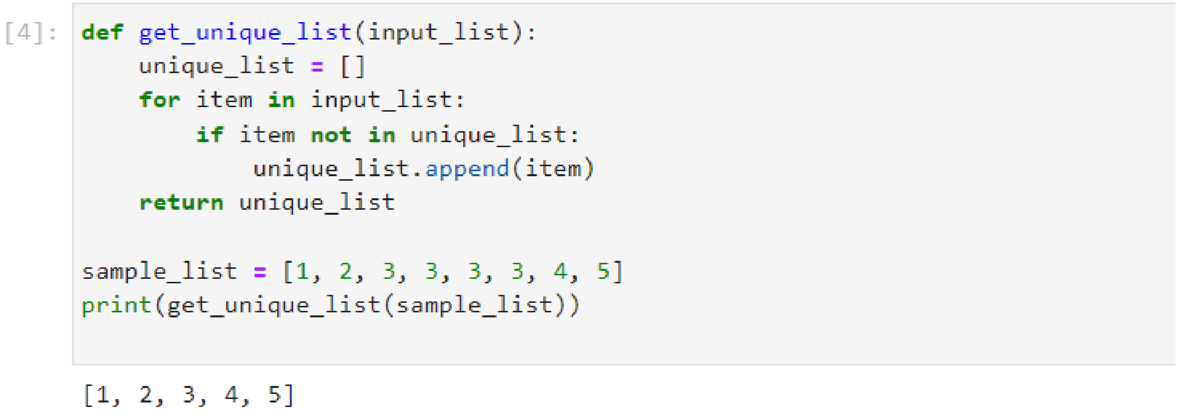
```
[3]: x = [23, "Python", 23.98]

     y = []
     for i in x:
         y.append(type(i))

     print(x)
     print(y)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

Here I created a new list ‘y’ that holds the type of elements from the original list ‘x’ and used a for loop to iterate over the elements of the list ‘x’ and the append method to add the type of each element to the new list ‘y’. Finally, the code prints both the original list ‘x’ and the new list ‘y’ that contains the type of elements.



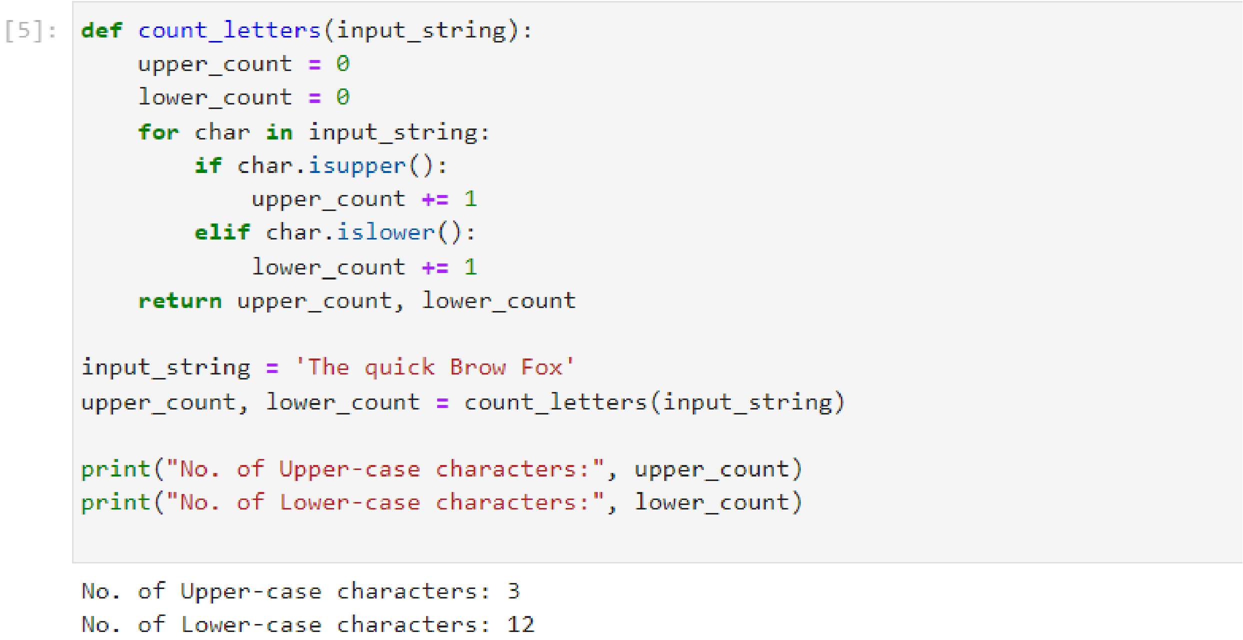
```
[4]: def get_unique_list(input_list):
         unique_list = []
         for item in input_list:
             if item not in unique_list:
                 unique_list.append(item)
         return unique_list

     sample_list = [1, 2, 3, 3, 3, 3, 4, 5]
     print(get_unique_list(sample_list))
```

```
[1, 2, 3, 4, 5]
```

Here I defined a function ‘get_unique_list’ that takes a list as input and returns a new list with unique elements from the original list. I used a ‘for’ loop to iterate over the elements of the input list, and the ‘if’ statement to check if the current item is not in the ‘unique_list’. If the current item is not in the ‘unique_list’, it adds it to the list using the append method. Finally, the function returns the ‘unique_list’ that contains the unique elements from the input list.

Machine Learning (Assignment # 2)



```
[5]: def count_letters(input_string):
         upper_count = 0
         lower_count = 0
         for char in input_string:
             if char.isupper():
                 upper_count += 1
             elif char.islower():
                 lower_count += 1
         return upper_count, lower_count

     input_string = 'The quick Brow Fox'
     upper_count, lower_count = count_letters(input_string)

     print("No. of Upper-case characters:", upper_count)
     print("No. of Lower-case characters:", lower_count)
```

```
No. of Upper-case characters: 3
No. of Lower-case characters: 12
```

In this code I defined a function "count_letters" which takes an input string as an argument and returns the number of upper-case and lower-case characters in the string. The function uses a for loop to iterate over each character in the input string and checks if it is an upper-case or lower-case character using the "isupper" and "islower" methods. If a character is an upper-case character, the "upper_count" variable is incremented by 1. If a character is a lower-case character, the "lower_count" variable is incremented by 1. The function then returns the "upper_count" and "lower_count" variables as a tuple. The input string "The quick Brow Fox" is then passed as an argument to the "count_letters" function and the number of upper-case and lower-case characters in the string is printed.

GitHub URL:

https://github.com/SaiPriyankaNarra/ML_700741613_Assignment-2.git

Video URL:

https://drive.google.com/file/d/1su70fFgdQmeGPzVckNGNE9_HZfUTB5nT/view?usp=share_link