

# Assignments

By Priyanshu Das | SIC : 20BCSE21 | Branch : CSE

---

## Assignment 1

---

### WAP using the 'r+' mode and 'w+' mode.

- `w+` mode

```
In [1]: # implementing using 'w+'
with open('newfile.txt',mode='w+') as f:
    language = 'Python'
    f.write('Hie whatsup?? \nThis is Priyanshu here!\n')
    f.write(f'Welcome to the Introduction to Machine Learning with {language}')
```

- `r+` mode

```
In [2]: # implementing using 'r+'
with open('newfile.txt',mode='r+') as f:
    data=f.read()
    print(data)

    #Go to the Last Line to append data
    length=len(data)
    f.seek(length)
    f.write('\nThis is Priyanshu here !')
```

```
Hie whatsup??
This is Priyanshu here!
Welcome to the Introduction to Machine Learning with Python
```

```
In [3]: #To check if data is appended
with open('newfile.txt',mode='r+') as f:
```

```
data=f.read()
print(data)
```

Hie whatsup??  
This is Priyanshu here!  
Welcome to the Introduction to Machine Learning with Pyth  
This is Priyanshu here !

---

## Assignment 2

---

WAP to find out the user guess is correct or not from the given list after shuffled

```
In [4]: import numpy as np
        from numpy import random as rnd
        arr = np.arange(1,6,1)

        guess = int(input("Guess the number at 1st position in an array consisting of numbers from 1 to 5:"))
        shuffled_arr = rnd.permutation(arr)

        if shuffled_arr[0]==guess:
            print("Correct guess!")
        else:
            print("Better luck next time :( . The element was ", shuffled_arr[0])
```

Better luck next time :( . The element was 5

---

## Assignment 3

---

Reverse a string using lambda expression

```
In [5]: # Reverse the string using Lambda expression

        rev=lambda x : x[::-1]
```

```
inp = 'Machine Learning'  
out = rev(inp)  
print(out)
```

gninrael enihcaM

---

## Assignment 4

---

Handle the exception thrown by the code below by using try-except block & use finally block to print "Good Job!"

Code :

```
for i in ['x','y','z']:  
    print(i**2)
```

In [6]:

```
try:  
    for i in ['x','y','z']:  
        print(i**2)  
except SyntaxError:  
    print('Syntax Error')  
except TypeError:  
    print('Type Error')  
except OSError:  
    print('OS Error')  
finally:  
    print('Good job!')
```

Type Error  
Good job!

---

## Assignment 5

---

- **Standardization vs Normalization**

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1 or between -1 and 1. It is also known as Min-Max scaling.

Here,  $X_{max}$  and  $X_{min}$  are the maximum and the minimum values of the feature respectively.

1. When the value of  $X$  is the minimum value in the column, the numerator will be 0, and hence  $X'$  is 0
2. On the other hand, when the value of  $X$  is the maximum value in the column, the numerator is equal to the denominator and thus the value of  $X'$  is 1
3. If the value of  $X$  is between the minimum and the maximum value, then the value of  $X'$  is between 0 and 1

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation

However, at the end of the day, the choice of using normalization or standardization will depend on your problem and the machine learning algorithm you are using. There is no hard and fast rule to tell you when to normalize or standardize your data. \*\*You can always start by fitting your model to raw, normalized and standardized data and compare the performance for best results.\*\*

It is a good practice to fit the scaler on the training data and then use it to transform the testing data. \*This would avoid any data leakage during the model testing process. Also, the scaling of target values is generally not required.\*

</p>

---

- **What do you mean by the result `pandas.core.frame.DataFrame` in `type(dataset)` ?**

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A DataFrame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

---

- **What is `nd` here ?**

The most important object defined in NumPy is an N-dimensional array type called ndarray. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index.

- **What is the meaning of dtype = object ?**

Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called dtype). The memory taken by the array now is filled with pointers to Python objects which are being stored elsewhere in memory. Since various datatypes are being used in the table, dtype=object instead of int or float.

---

- **What is the significance of random\_state=1 ?**

The random\_state parameter is used for initializing the internal random number generator, which will decide the splitting of data into train and test indices in your case. If random\_state is None or np.random, then a randomly-initialized RandomState object is returned. If random\_state is an integer, then it is used to seed a new RandomState object. This is to check and validate the data when running the code multiple times. Setting random\_state a fixed value will guarantee that the same sequence of random numbers is generated each time you run the code.

---