Data Pre-Processing

In this milestone, we will be pre-processing the dataset that is collected.

Pre-processing includes

1. Handling the null values.
2. Handling the categorical values if any.
3. Normalize the data if required.
4. Identifying the dependent and independent variables.
5. Split the dataset into train and test sets.

Import Required Libraries

Importing the libraries

Step 1 - Launch Jupyter notebook through anaconda navigator or anaconda prompt.

Step 2 - Create a new notebook by clicking on "new" button on the top right corner of the page.

The libraries can be imported using the import keyword. Insert commands as shown below.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix,accuracy_score
```

Read The Dataset

Reading the dataset

The dataset is read as a **data frame** by using pandas library. Insert the commands as shown below

(Here ds is referred as **data frame & pd** is the alias name given to pandas library).

```
#Import Dataset
ds= pd.read_csv("dataset_website.csv")
ds.head()
```

Sample output of the dataset rows is shown below.

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain
0	1	-1	1	1	1	-1	-1	-1
1	2	1	1	1	1	1	-1	0
2	3	1	0	1	1	1	-1	-1
3	4	1	0	1	1	1	-1	-1
4	5	1	0	-1	1	1	-1	1

5 rows × 32 columns

Handling Null Values

Checking for Null values in a dataset and handling if any

In this activity, we will check if there are any null values in a dataset and fill/handle them.

To know if there are any null values present in a dataset **isnull()** method can be used.

Input the commands as shown below to check for **null** values.

```
#Analysing the data using pandas and Checking if the dataset contains any Null values.
ds.info()
ds.isnull().any() #no nullvalues
```

Output: the command **ds.isnull().any()** returns true if null values are present.

Here, the dataset which we have used doesn't have any null values.

Splitting The Data

Splitting data into independent and dependent variables

Identifying Independent & dependent variables:

In this activity, the dependent and independent variables are to be identified. The last column (Result) in the dataset is the dependent variable which is dependent on the 30 different factors. The independent columns are considered as x and the dependent column as y.

Input the commands as shown

```
#Splitting data as independent and dependent
#removing index column in independent dataset
x=ds.iloc[:,1:31].values
y=ds.iloc[:,-1].values
print(x,y)
```

Splitting the data:

After identifying the dependent and independent variables, the dataset now has to be split into two sets, one set is used for training the model and the second set is used for testing how good the model is built. The split ratio we consider is 80% for training and 20% for testing.

Input the commands as shown

```
#Splitting data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```