Data Wrangling I :

Data Preprocessing , Normalization using Python

Problem Statement :

Perform the following operations using Python or any open source dataset

1. Import all the required python libraries
2. Locate an open source data from the web.
3. Load the dataset into pandas data frame
4. Data preprocessing check for missing values in the data using pandas, insult() , describe() function to get some initial statistics . Provide variable description.
5. Data formatting and Data normalization : Summarize the types of variables by checking data types of the variables in the data set
6. Turn categorical variables into quantitative variables in python

Introduction :

Data preprocessing is the process of transforming raw data into an understandable format . It is also an important step in data mining as we cannot work with raw data .

Data normalization is a technique often aplied as part of data preparation for machine learning . The goal of normalization is to change the values of numeric columns in the dataset to a comman scale, without distorting differences in the range of values.

Theory :

Data Wrangling : Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis. Data wrangling involves processing the data in various formats and analyze and get them to be used with another set of data and bringing them together into variable insights.

Wrangling the data is crucial, yet it is considered as a backbone to the entire analysis part. The main purpose of data wrangling is to make raw data usuable. In other words, getting data into a shape. Pandas is the main library which is used to perform data wrangling operation.

DataFrame : DataFrame is two dimensional size mutable, potentially heterogeneous tabular data structure with labeled axes (row and columns). In DataFrame data is aligned in a tabular fashion in rows and columns.

isnull() function : isnull() function detect missing values in the given Series object. It return a boolean same-sized object indicating if the values are NA. Missing values gets mapped to True and non-missing values gets mapped to false.

Algorithm / Steps :

Step 1 : Data Exploration - Here the visualization of data is done in a tabular format

```
import pandas as pd
data = {'Name' : ['Jay', 'Kunal', 'Mohit', 'Yash'],
        'Age' : [20, 18, 20, 22],
        'Gender' : ['M', 'M', 'M', 'M'],
        'Marks' : [90, 'NaN', 'NaN', 71]}
df = pd.DataFrame(data)
df
```

Step 2 : Dealing With Missing Values - Here the null values present in the data in the marks column are removed and replaced with the mean value.

```
c = avg = 0
for ele in df['Marks']:
    if str(ele).isnumeric():
        c += 1
        avg += ele
avg /= c
df = df.replace(to_replace = "NaN", value = avg)
df
```

Step 3 : Reshaping the Data - The categorical values can be represented by a numerical value. As the data contain categorical values in the gender column, it can be reshaped by

categorizing them into numbers

```
df['Gender'] = df['Gender'].map({'M': 0, 'F': 1}).
                astype(float)
df
```

Step 4 : Filtering - Here the data is restructured to the specific format by removing the unwanted data in a table.

```
df = df[df['Marks'] >= 75]
df = df.drop(['Age'], axis = 1)
df
```