Data Wrangling

Data Wrangling is the process of gathering, collecting, and transforming Raw data into another format for better understanding, decision-making, accessing, and analysis in less time.

Data wrangling in Python deals with the below functionalities:

1. **Data exploration:** In this process, the data is studied, analyzed, and understood by visualizing representations of data.
2. **Dealing with missing values:** Most of the datasets having a vast amount of data contain missing values of *NaN, they are needed to be taken* care of by replacing them with mean, mode, the most frequent value of the column, or simply by dropping the row having a *NaN* value.
3. **Reshaping data:** In this process, data is manipulated according to the requirements, where new data can be added or pre-existing data can be modified.
4. **Filtering data:** Some times datasets are comprised of unwanted rows or columns which are required to be removed or filtered
5. **Other:** After dealing with the raw dataset with the above functionalities we get an efficient dataset as per our requirements and then it can be used for a required purpose like data analyzing, machine learning, data visualization, model training etc.

Data exploration in Python

Here in Data exploration, we load the data into a dataframe, and then we visualize the data in a tabular format.

```
# Import pandas package
import pandas as pd

# Assign data
data = {'Name': ['Jai', 'Princi', 'Gaurav',
                 'Anuj', 'Ravi', 'Natasha', 'Riya'],
        'Age': [17, 17, 18, 17, 18, 17, 17],
        'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'],
        'Marks': [90, 76, 'NaN', 74, 65, 'NaN',
71]}

# Convert into DataFrame
df = pd.DataFrame(data)
# Display data
df
```

Output:

	Name	Age	Gender	Marks
0	Jai	17	M	90
1	Princi	17	F	76
2	Gaurav	18	M	NaN
3	Anuj	17	M	74
4	Ravi	18	M	65
5	Natasha	17	F	NaN
6	Riya	17	F	71

Dealing with missing values in Python

As we can see from the previous output, there are *NaN* values present in the *MARKS* column which is a missing value in the dataframe that is going to be taken care of in data wrangling by replacing them with the column mean.

```
# Compute average
c = avg = 0
for ele in df['Marks']:
    if str(ele).isnumeric():
        c += 1
        avg += ele
avg /= c

# Replace missing values
df = df.replace(to_replace="NaN",
                value=avg)

# Display data
df
```

Output:

	Name	Age	Gender	Marks
0	Jai	17	M	90.0
1	Princi	17	F	76.0
2	Gaurav	18	M	75.2
3	Anuj	17	M	74.0
4	Ravi	18	M	65.0
5	Natasha	17	F	75.2
6	Riya	17	F	71.0

replacing Nan values with average

Data Replacing in Data Wrangling

in the *GENDER* column, we can replace the Gender column data by categorizing them into different numbers.

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

# Display data
```

df

Output:

	Name	Age	Gender	Marks
0	Jai	17	0.0	90.0
1	Princi	17	1.0	76.0
2	Gaurav	18	0.0	75.2
3	Anuj	17	0.0	74.0
4	Ravi	18	0.0	65.0
5	Natasha	17	1.0	75.2
6	Riya	17	1.0	71.0

Data encoding for gender variable in data wrangling

Filtering data in Data Wrangling

suppose there is a requirement for the details regarding name, gender, and marks of the top-scoring students. Here we need to remove some using the pandas slicing method in data wrangling from unwanted data.

```
# Filter top scoring students
df = df[df['Marks'] >= 75].copy()

# Remove age column from filtered
DataFrame
df.drop('Age', axis=1, inplace=True)

# Display data
df
```

Output:

	Name	Gender	Marks
0	Jai	0.0	90.0
1	Princi	1.0	76.0
2	Gaurav	0.0	75.2
5	Natasha	1.0	75.2

Dropping column and filtering rows

Data Wrangling Using Merge Operation

Merge operation is used to merge two raw data into the desired format.

Syntax: *pd.merge(data_frame1,data_frame2, on="field ")*

Here the field is the name of the column which is similar in both data-frame.

For example: Suppose that a Teacher has two types of Data, the first type of Data consists of Details of Students and the Second type of Data Consist of Pending Fees Status which is taken from the Account Office. So The Teacher will use the merge operation here in order to merge the data and provide it meaning. So that teacher will analyze it easily and it also reduces the time and effort of the Teacher from Manual Merging.

Creating First Dataframe to Perform Merge Operation using Data Wrangling:

```
# import module
import pandas as pd

# creating DataFrame for Student Details
details = pd.DataFrame({
    'ID': [101, 102, 103, 104, 105, 106,
           107, 108, 109, 110],
    'NAME': ['Jagroop', 'Praveen', 'Harjot',
             'Pooja', 'Rahul', 'Nikita',
             'Saurabh', 'Ayush', 'Dolly', "Mohit"],
    'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE',
'CSE',
               'CSE', 'CSE', 'CSE', 'CSE', 'CSE']})

# printing details
print(details)
```

Output:

	ID	NAME	BRANCH
0	101	Jagroop	CSE
1	102	Praveen	CSE
2	103	Harjot	CSE
3	104	Pooja	CSE
4	105	Rahul	CSE
5	106	Nikita	CSE
6	107	Saurabh	CSE
7	108	Ayush	CSE
8	109	Dolly	CSE
9	110	Mohit	CSE

printing dataframe

Creating Second Dataframe to Perform Merge operation using Data Wrangling:

```
# Import module
import pandas as pd

# Creating Dataframe for Fees_Status
fees_status = pd.DataFrame(
    {'ID': [101, 102, 103, 104, 105,
            106, 107, 108, 109, 110],
     'PENDING': ['5000', '250', 'NIL',
                 '9000', '15000', 'NIL',
                 '4500', '1800', '250', 'NIL']})

# Printing fees_status
print(fees_status)
```

Output:

	ID	PENDING
0	101	5000
1	102	250
2	103	NIL
3	104	9000
4	105	15000
5	106	NIL
6	107	4500
7	108	1800
8	109	250
9	110	NIL

Define second dataframe

Data Wrangling Using Merge Operation:

```
# Import module
import pandas as pd

# Creating Dataframe
details = pd.DataFrame({
   'ID': [101, 102, 103, 104, 105,
         106, 107, 108, 109, 110],
   'NAME': ['Jagroop', 'Praveen', 'Harjot',
          'Pooja', 'Rahul', 'Nikita',
          'Saurabh', 'Ayush', 'Dolly', "Mohit"],
   'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE',
'CSE',
           'CSE', 'CSE', 'CSE', 'CSE', 'CSE']})

# Creating Dataframe
fees_status = pd.DataFrame(
   {'ID': [101, 102, 103, 104, 105,
         106, 107, 108, 109, 110],
    'PENDING': ['5000', '250', 'NIL',
             '9000', '15000', 'NIL',
             '4500', '1800', '250', 'NIL']})

# Merging Dataframe
print(pd.merge(details, fees_status, on='ID'))
```

Output:

	ID	NAME	BRANCH	PENDING
0	101	Jagroop	CSE	5000
1	102	Praveen	CSE	250
2	103	Harjot	CSE	NIL
3	104	Pooja	CSE	9000
4	105	Rahul	CSE	15000
5	106	Nikita	CSE	NIL
6	107	Saurabh	CSE	4500
7	108	Ayush	CSE	1800
8	109	Dolly	CSE	250
9	110	Mohit	CSE	NIL

Merging two dataframes

Data Wrangling Using Grouping Method

The grouping method in Data wrangling is used to provide results in terms of various groups taken out from Large Data. This method of pandas is used to group the outset of data from the large data set.

Example: There is a Car Selling company and this company have different Brands of various Car Manufacturing Company like Maruti, Toyota, Mahindra, Ford, etc., and have data on where different cars are sold in different years. So the Company wants to wrangle only that data where cars are sold during the year 2010. For this problem, we use another data Wrangling technique which is a pandas *groupby()* method.

Creating dataframe to use Grouping methods[Car selling datasets]:

```
# Import module
import pandas as pd

# Creating Data
car_selling_data = {'Brand': ['Maruti', 'Maruti',
'Maruti',
                  'Maruti', 'Hyundai',
'Hyundai',
                  'Toyota', 'Mahindra',
'Mahindra',
                  'Ford', 'Toyota', 'Ford'],
            'Year': [2010, 2011, 2009, 2013,
                  2010, 2011, 2011, 2010,
                  2013, 2010, 2010, 2011],
            'Sold': [6, 7, 9, 8, 3, 5,
                  2, 8, 7, 2, 4, 2]}

# Creating Dataframe of car_selling_data
df = pd.DataFrame(car_selling_data)

# printing Dataframe
print(df)
```

Output:

	Brand	Year	Sold
0	Maruti	2010	6
1	Maruti	2011	7
2	Maruti	2009	9
3	Maruti	2013	8
4	Hyundai	2010	3
5	Hyundai	2011	5
6	Toyota	2011	2
7	Mahindra	2010	8
8	Mahindra	2013	7
9	Ford	2010	2
10	Toyota	2010	4
11	Ford	2011	2

Creating new dataframe

Creating Dataframe to use Grouping methods[DATA OF THE YEAR 2010]:

```
# Import module
import pandas as pd

# Creating Data
```

```
car_selling_data = {'Brand': ['Maruti', 'Maruti',
'Maruti',
                    'Maruti', 'Hyundai',
'Hyundai',
                    'Toyota', 'Mahindra',
'Mahindra',
                    'Ford', 'Toyota', 'Ford'],
                'Year': [2010, 2011, 2009, 2013,
                    2010, 2011, 2011, 2010,
                    2013, 2010, 2010, 2011],
                'Sold': [6, 7, 9, 8, 3, 5,
                    2, 8, 7, 2, 4, 2]}

# Creating Dataframe for Provided Data
df = pd.DataFrame(car_selling_data)

# Group the data when year = 2010
grouped = df.groupby('Year')
print(grouped.get_group(2010))
```

Output:

	Brand	Year	Sold
0	Maruti	2010	6
4	Hyundai	2010	3
7	Mahindra	2010	8
9	Ford	2010	2
10	Toyota	2010	4

Using groupby method on dataframe

Data Wrangling by Removing Duplication

Pandas *duplicates()* method helps us to remove duplicate values from Large Data. An important part of Data Wrangling is removing Duplicate values from the large data set.

Syntax: *DataFrame.duplicated(subset=None, keep='first')*

Here subset is the column value where we want to remove the Duplicate value.

In keeping, we have 3 options :

- *if keep ='first' then the first value is marked as the original rest of all values if occur will be removed as it is considered duplicate.*
- *if keep='last' then the last value is marked as the original rest the above same values will be removed as it is considered duplicate values.*
- *if keep ='false' all the values which occur more than once will be removed as all are considered duplicate values.*

For example, A University will organize the event. In order to participate Students have to fill in their details in the online form so that they will contact them. It may be possible that a student will fill out the form multiple times. It may cause difficulty for the event organizer if a single student will fill in multiple entries. The Data that the organizers will get can be Easily Wrangles by removing duplicate values.

Creating a Student Dataset who want to participate in the event:

```
# Import module
import pandas as pd

# Initializing Data
student_data = {'Name': ['Amit', 'Praveen',
'Jagroop',
                'Rahul', 'Vishal', 'Suraj',
                'Rishab', 'Satyapal', 'Amit',
                'Rahul', 'Praveen', 'Amit'],

            'Roll_no': [23, 54, 29, 36, 59, 38,
                12, 45, 34, 36, 54, 23],

            'Email': ['xxxx@gmail.com',
'xxxxxx@gmail.com',
                'xxxxxx@gmail.com',
'xx@gmail.com',
                'xxxx@gmail.com',
'xxxxx@gmail.com',
                'xxxxx@gmail.com',
'xxxxx@gmail.com',
                'xxxxx@gmail.com',
'xxxxxx@gmail.com',
                'xxxxxxxxxx@gmail.com',
'xxxxxxxxxx@gmail.com']}

# Creating Dataframe of Data
df = pd.DataFrame(student_data)

# Printing Dataframe
print(df)
```

Output:

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com
3	Rahul	36	xx@gmail.com
4	Vishal	59	xxxx@gmail.com
5	Suraj	38	xxxxx@gmail.com
6	Rishab	12	xxxxx@gmail.com
7	Satyapal	45	xxxxx@gmail.com
8	Amit	34	xxxxx@gmail.com
9	Rahul	36	xxxxxx@gmail.com
10	Praveen	54	xxxxxxxxxx@gmail.com
11	Amit	23	xxxxxxxxxx@gmail.com

Student Dataset who want to participate in the event

Removing Duplicate data from the Dataset using Data wrangling:

```
# import module
```

```
import pandas as pd

# initializing Data
student_data = {'Name': ['Amit', 'Praveen',
'Jagroop',
                'Rahul', 'Vishal', 'Suraj',
                'Rishab', 'Satyapal', 'Amit',
                'Rahul', 'Praveen', 'Amit'],

          'Roll_no': [23, 54, 29, 36, 59, 38,
                12, 45, 34, 36, 54, 23],
          'Email': ['xxxx@gmail.com',
'xxxxxx@gmail.com',
                'xxxxxx@gmail.com',
'xx@gmail.com',
                'xxxx@gmail.com',
'xxxxx@gmail.com',
                'xxxxx@gmail.com',
'xxxxx@gmail.com',
                'xxxxx@gmail.com',
'xxxxxx@gmail.com',
                'xxxxxxxxxx@gmail.com',
'xxxxxxxxxx@gmail.com']}

# creating dataframe
df = pd.DataFrame(student_data)

# Here df.duplicated() list duplicate Entries in
ROllno.
# So that ~(NOT) is placed in order to get non
duplicate values.
non_duplicate = df[~df.duplicated('Roll_no')]

# printing non-duplicate values
print(non_duplicate)
```

Output:D

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com
3	Rahul	36	xx@gmail.com
4	Vishal	59	xxxx@gmail.com
5	Suraj	38	xxxxx@gmail.com
6	Rishab	12	xxxxx@gmail.com
7	Satyapal	45	xxxxx@gmail.com
8	Amit	34	xxxxx@gmail.com

Remove – Duplicate data from Dataset using Data wrangling

Creating New Datasets Using the Concatenation of Two Datasets In Data Wrangling.

We can join two dataframe in several ways. For our example in Concanating Two datasets, we use pd.concat() function.

Creating Two Dataframe For Concatenation.

```
# importing pandas module
import pandas as pd

# Define a dictionary containing employee
data
data1 = {'Name':['Jai', 'Princi', 'Gaurav',
'Anuj'],
      'Age':[27, 24, 22, 32],
      'Address':['Nagpur', 'Kanpur', 'Allahabad',
'Kannuaj'],
      'Qualification':['Msc', 'MA', 'MCA',
'Phd'],
      'Mobile No': [97, 91, 58, 76]}

# Define a dictionary containing employee
data
data2 = {'Name':['Gaurav', 'Anuj', 'Dhiraj',
'Hitesh'],
      'Age':[22, 32, 12, 52],
      'Address':['Allahabad', 'Kannuaj',
'Allahabad', 'Kannuaj'],
      'Qualification':['MCA', 'Phd', 'Bcom',
'B.hons'],
      'Salary':[1000, 2000, 3000, 4000]}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data1,index=[0, 1, 2, 3])

# Convert the dictionary into DataFrame
df1 = pd.DataFrame(data2, index=[2, 3, 6, 7])
We will join these two dataframe along axis
0.
res = pd.concat([df, df1])
```

output:

```
  Name  Age  Address  Qualification
Mobile No  Salary
0  Jai   27  Nagpur       Msc
97.0       NaN
1  Princi  24  Kanpur      MA          91.0
NaN
2  Gaurav  22  Allahabad   MCA
58.0       NaN
3  Anuj  32  Kannuaj      Phd          76.0
NaN
4  Gaurav  22  Allahabad   MCA
NaN        1000.0
5  Anuj  32  Kannuaj      Phd          NaN
2000.0
6  Dhiraj  12  Allahabad   Bcom         NaN
3000.0
7  Hitesh  52  Kannuaj     B.hons
NaN        4000.0
```