Assignment No. 01

Performance	Understanding	Regularity	Total	Dated Sign of Subject Teacher
03	01	01	05	

Date of Performance: .... **Date of Completion**: ....

Title Perform the various operations using Python on the Facebook metrics data sets

Objectives:

To understand and apply the Analytical concept of Big data using Python.

Problem Statement:

Perform the following operations using Python on the Facebook metrics data sets

c. Create data subsets
d. Merge Data
e. Sort Data
f. Transposing Data
g. Shape and reshape Data

Outcomes:

Students will be able to,

1. Apply the Analytical concept of Big data using Python.

Software and Hardware requirements:

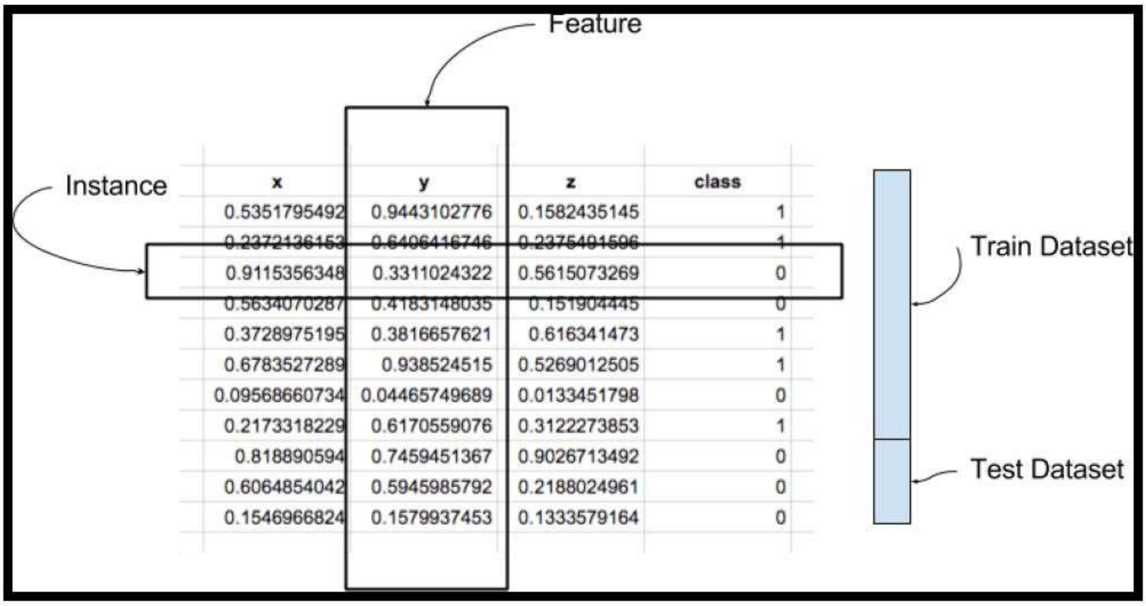
1. Software: Ubuntu OS, Anaconda, Jupyter Notebook
2. Hardware: Processor, Ethernet Connection or WiFi, RAM 1GB, HDD, Sound Card, camera, microphone (depending upon website selection)

Theory:

1. Introduction to Dataset
2. Python Libraries for Data Science
3. Panda Dataframe functions for load the dataset
4. Panda functions for create data subsets
5. Panda functions for Merge datasets
6. Panda Functions for Sort the datasets
7. Panda Functions for Transpose the datasets
8. Panda Functions for Shape and reshape the data.

1. Introduction to Dataset

A dataset is a collection of records, similar to a relational database table. Records are similar to table rows, but the columns can contain not only strings or numbers, but also nested data structures such as lists, maps, and other records.



x	y	z	class
0.5351795492	0.9443102776	0.1582435145	1
0.2372136153	0.6406416746	0.2375491596	1
0.9115356348	0.3311024322	0.5615073269	0
0.5634070287	0.4183148035	0.151904445	0
0.3728975195	0.3816657621	0.616341473	1
0.6783527289	0.938524515	0.5269012505	1
0.09568660734	0.04465749689	0.0133451798	0
0.2173318229	0.6170559076	0.3122273853	1
0.818890594	0.7459451367	0.9026713492	0
0.6064854042	0.5945985792	0.2188024961	0
0.1546966824	0.1579937453	0.1333579164	0

Instance: A single row of data is called an instance. It is an observation from the domain. **Feature:** A single column of data is called a feature. It is a component of an observation and is also called an attribute of a data instance. Some features may be inputs to a model (the predictors) and others may be outputs or the features to be predicted.

Data Type: Features have a data type. They may be real or integer-valued or may have a categorical or ordinal value. You can have strings, dates, times, and more complex types, but typically they are reduced to real or categorical values when working with traditional machine learning methods.

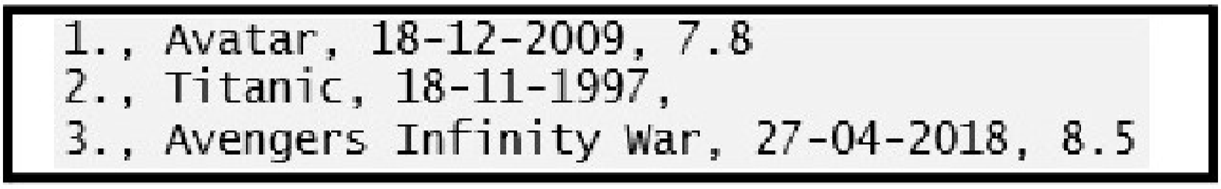
Datasets: A collection of instances is a dataset and when working with machine learning methods we typically need a few datasets for different purposes.

Training Dataset: A dataset that we feed into our machine learning algorithm to train our model.

Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.

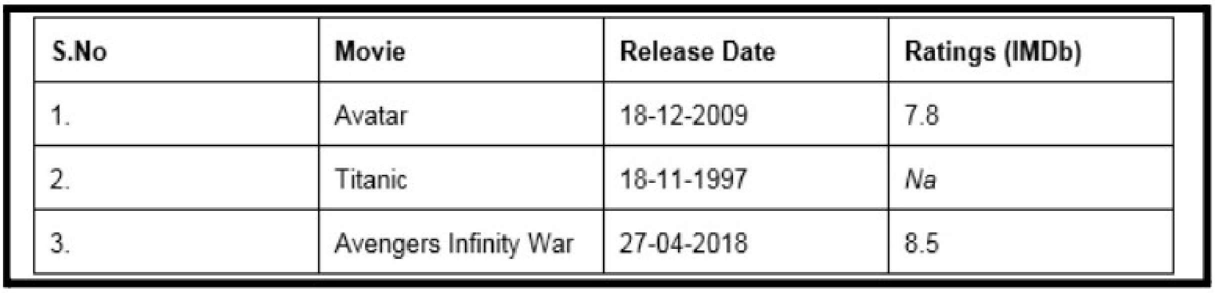
Data Represented in a Table:

Data should be arranged in a two-dimensional space made up of rows and columns. This type of data structure makes it easy to understand the data and pinpoint any problems. An example of some raw data stored as a CSV (comma separated values).



```
1., Avatar, 18-12-2009, 7.8
2., Titanic, 18-11-1997,
3., Avengers Infinity War, 27-04-2018, 8.5
```

The representation of the same data in a table is as follows:



S.No	Movie	Release Date	Ratings (IMDb)
1.	Avatar	18-12-2009	7.8
2.	Titanic	18-11-1997	*Na*
3.	Avengers Infinity War	27-04-2018	8.5

Pandas Data Types:

A data type is essentially an internal construct that a programming language uses to understand how to store and manipulate data.

A possible confusing point about pandas data types is that there is some overlap between pandas, python and numpy. This table summarizes the key points:

Pandas dtype	Python type	NumPy type	Usage
object	str or mixed	string_, unicode_, mixed types	Text or mixed numeric and non-numeric values
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	datetime64[ns]	Date and time values
timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

2. Python Libraries for Data Science:

a. Pandas:

Pandas are an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python programming language.

What can you do with Pandas?

1. Indexing, manipulating, renaming, sorting, merging data frame
2. Update, Add, Delete columns from a data frame
3. Impute missing files, handle missing data or NANs
4. Plot data with histogram or box plot

b. NumPy:

One of the most fundamental packages in Python, NumPy is a general-purpose array processing package. It provides high-performance multidimensional array objects and tools to work with the arrays. NumPy is an efficient container of generic multi-dimensional data.

NumPy’s main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same dat atype, indexed by a tuple of positive integers. In NumPy, dimensions are called axes and the number of axes is called rank. NumPy’s array class is called ndarray aka array

What can you do with NumPy?

1. Basic array operations: add, multiply, slice, flatten, reshape, index arrays
2. Advanced array operations: stack arrays, split into sections, broadcast arrays
3. Work with DateTime or Linear Algebra
4. Basic Slicing and Advanced Indexing in NumPy Python

There are more libraries like **seaborn**, **matplotlib** and **scikitlearn** which are discuss in next assignments.

3. Pandas Dataframe functions for loading Dataset:

1. The dataset is downloads from UCI repository

   csv_url = ‘https://archive.ics.uci.edu/ml/datasets/Facebook+metrics'

2. Now Read CSV File as a Dataframe in Python from from path where you saved the same The facebook data set is stored in .csv format. **‘.csv’** stands for comma separated values. It is easier to load .csv files in Pandas data frame and perform various analytical operations on it.

Load ‘Facebook_matrics.csv’ into a Pandas data frame

Syntax:

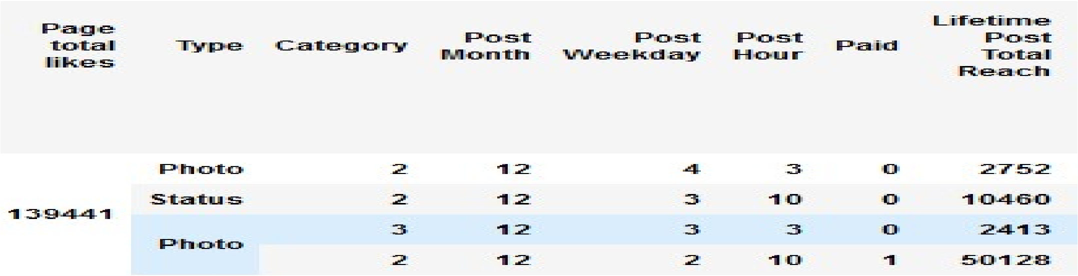
df=pd.read_csv(csv_url, header=none)

3. The csv file at the UCI repository does not contain the variable/column names. They are located in a separate file.

col_names = ['Type’, ‘Category’, ‘Post Month’, ‘Post Weekday’, ‘Post Hour’, ‘Paid’]

4. read in the dataset from the UCI Machine Learning Repository link and specify column names to use

df=pd.read_csv(csv_url, names=col_names)

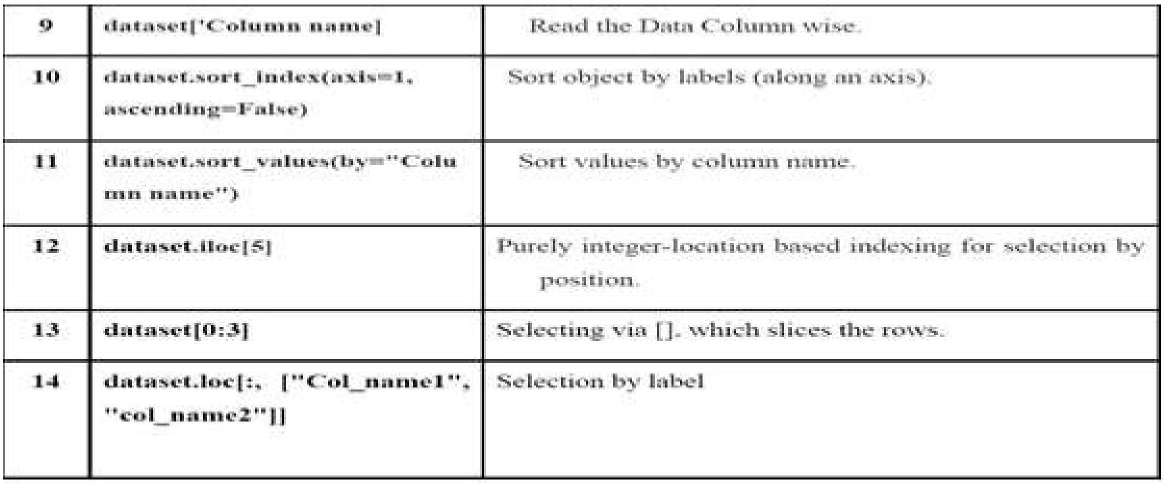


Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach
139441	Photo	2	12	4	3	0	2752
	Status	2	12	3	10	0	10460
	Photo	3	12	3	3	0	2413
		2	12	2	10	1	50128

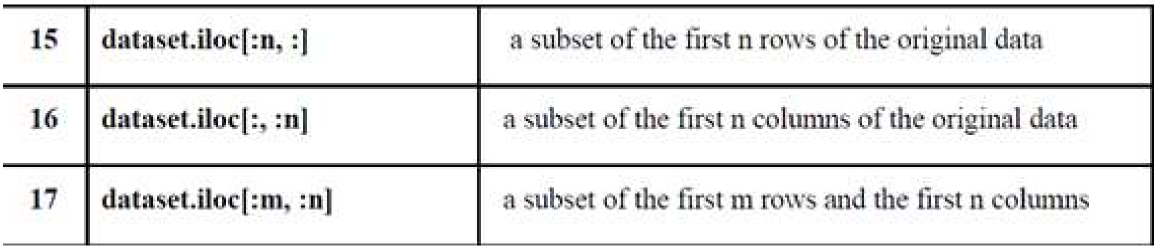
5. Panda Data-frame functions for Data Preprocessing:

Data-frame Operations:

Sr. No	Data Frame Function	Description
1	**dataset.head(n=5)**	**Return the first n rows.**
2	**dataset.tail(n=5)**	**Return the last n rows.**
3	**dataset.index**	The index (row labels) of the Dataset.
4	**dataset.columns**	The column labels of the Dataset.
5	**dataset.shape**	Return a tuple representing the dimensionality of the Dataset.
6	**dataset.dtypes**	Return the dtypes in the Dataset.
		This returns a Series with the data type of each column. The result's index is the original Dataset's columns. Columns with mixed types are stored with the object dtype.
7	**dataset.columns.values**	Return the columns values in the Dataset in array format
8	**dataset.describe(include='all')**	Generate descriptive statistics. to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. Analyzes both numeric and object series, as well as Dataset column sets of mixed data types.



9	dataset['Column name]	Read the Data Column wise.
10	dataset.sort_index(axis=1, ascending=False)	Sort object by labels (along an axis).
11	dataset.sort_values(by="Column name")	Sort values by column name.
12	dataset.iloc[5]	Purely integer-location based indexing for selection by position.
13	dataset[0:3]	Selecting via []. which slices the rows.
14	dataset.loc[:, ["Col_name1", "col_name2"]]	Selection by label



15	dataset.iloc[:n, :]	a subset of the first n rows of the original data
16	dataset.iloc[:, :n]	a subset of the first n columns of the original data
17	dataset.iloc[:m, :n]	a subset of the first m rows and the first n columns

4. Panda functions for create data subsets:

There are three ways to create dataset subsets:

1. **Create a subset of a Python dataframe using the loc() function**

Python loc() function enables us to form a subset of a data frame according to a specific row or column or a combination of both.

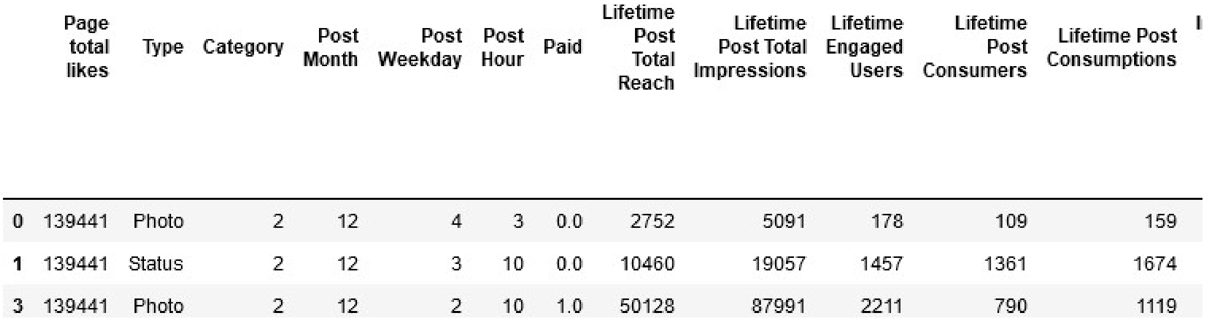
The loc() function works on the basis of labels i.e. we need to provide it with the label of the row/column to choose and create the customized subset.

Syntax: `dataframe.loc[]`

Example 1: Extract data of specific rows of a dataframe

```
dataframe.loc[[0,1,3]]
```

As seen below, we have created a subset which includes all the data of row 0, 1, and 3.

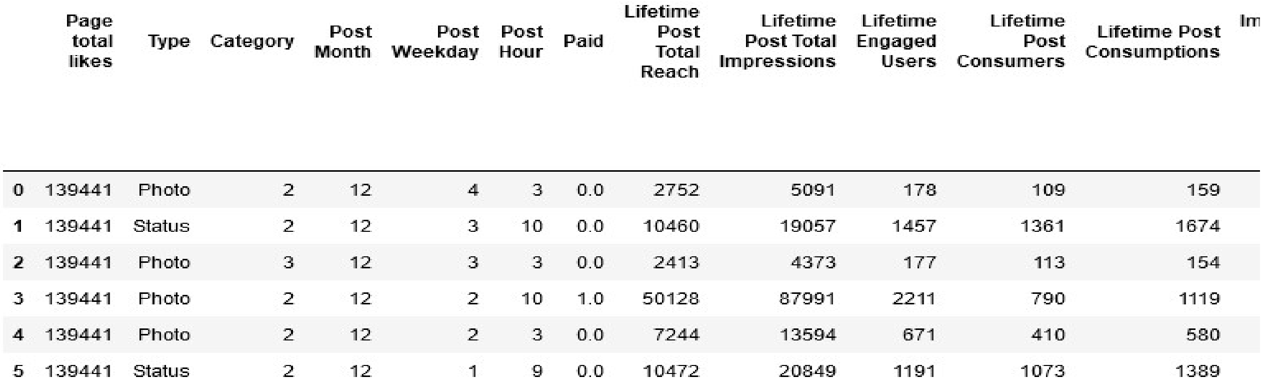


	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	I
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	109	159	
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	1361	1674	
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	790	1119	

Example 2: Create a subset of rows using slicing

```
dataframe.loc[0:5]
```

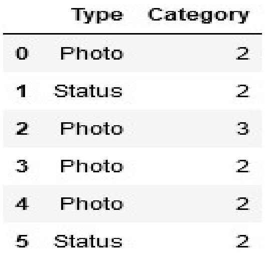
Here, we have extracted the data of all the rows from index 0 to index 3 using slicing operator with loc() function.



	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Im
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	109	159	
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	1361	1674	
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	113	154	
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	790	1119	
4	139441	Photo	2	12	2	3	0.0	7244	13594	671	410	580	
5	139441	Status	2	12	1	9	0.0	10472	20849	1191	1073	1389	

Example 3: Create a subset of particular columns using labels

```
dataframe.loc[0:5,['Type','Category']]
```



	Type	Category
0	Photo	2
1	Status	2
2	Photo	3
3	Photo	2
4	Photo	2
5	Status	2

Here, we have created a subset which includes data from rows 0 to 5, but includes that of only some specific columns i.e. 'Type' and 'Category'.

2. Using Python iloc() function to create a subset of a dataframe

Python iloc() function enables us to create subset choosing specific values from rows and columns based on indexes. That is, unlike loc() function which works on labels, iloc() function works on index values. We can choose and create a subset of a Python dataframe from the data providing the index numbers of the rows and columns.

Syntax: `dataframe.iloc[]`

Example: `df.iloc[0:5,[0,2]]`

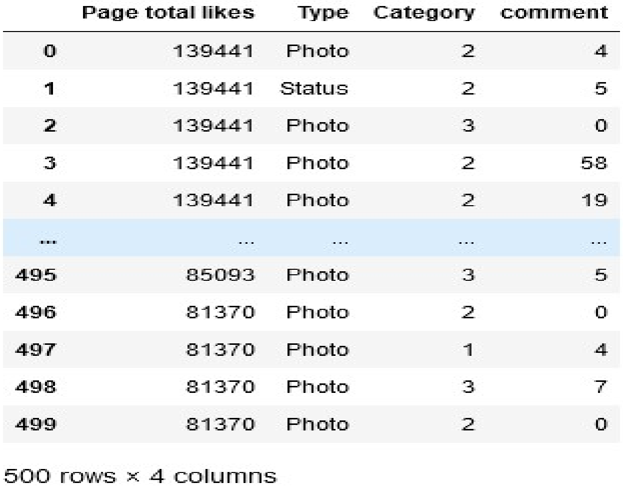
Here, we have created a subset which includes the data of the rows 0 to 5 as well as column number 0 and 2 i.e. 'Page total likes' and 'Category'.

3. Indexing operator to create a subset of a dataframe

In a simple manner, we can make use of an indexing operator i.e. square brackets to create a subset of the data.

Syntax: `dataframe`[['col1','col2','colN']]

Example: `df[['Page total likes','Type','Category','comment']]`



	Page total likes	Type	Category	comment
0	139441	Photo	2	4
1	139441	Status	2	5
2	139441	Photo	3	0
3	139441	Photo	2	58
4	139441	Photo	2	19
...	...	...	...	...
495	85093	Photo	3	5
496	81370	Photo	2	0
497	81370	Photo	1	4
498	81370	Photo	3	7
499	81370	Photo	2	0

500 rows × 4 columns

5. Pandas function for merge datasets:

To combine datasets together, the `concat` function of pandas can be utilized

We have two divided dataset as follows:

```
df1=df[['Page total likes','Type','Category','comment']]

df2=df[['Post Month','Post Weekday','Post Hour','Paid']]
```

These pieces dataframe df1 and df2 can be combined using the `concat()` function:

```
pd.concat([df1,df2])
```

6. Pandas function for Sort datasets:

In order to sort the data frame in pandas, function sort_values() is used. Pandas sort_values() can sort the data frame in Ascending or Descending order.

Example 1: Sorting the Data frame in ascending order

```
df.sort_values(by=['Category'])
```

Example 2: Sorting the Data frame in descending order

```
df.sort_values(by=['Category'], ascending=False)
```

Example 3: Sorting Pandas Data frame by putting missing values first

```
df.sort_values(by=['Category'], na_position='first')
```

Example 4: Sorting Data frames by multiple columns but different order

```
df.sort_values(by=['Type', 'Category'],ascending=[False, True])
```

7. Pandas function for Transpose Functions:

The `transpose()` function is used to transpose index and columns. Reflect the DataFrame over its main diagonal by writing rows as columns and vice-versa.

```
result = df.transpose()
```

8. Pandas function for Shape and reshape Data

Shape:

The DataFrame. shape attribute in Pandas enables us to obtain the shape of a DataFrame.

For example, if a DataFrame has a shape of (80, 10), this implies that the DataFrame is made up of 80 rows and 10 columns of data.

Syntax: `dataframe.shape      or  dataframe.shape(5)`

Reshape:

Pandas has two methods that aid in reshaping the data into a desired format. Pandas has two methods namely, **`melt() and pivot()`**, to reshape the data.

melt()

Melt in pandas reshape dataframe from wide format to long format. It uses the "id_vars['col_names']" for melt the dataframe by column names.

```
df_melt = df.melt(id_vars='Category')
df_melt
```

pivot()

This method does the reverse of what melt() did. It transforms the key-value pairs into columns. Reshape data (produce a "pivot" table) based on column values. Uses unique values from specified index / columns to form axes of the resulting DataFrame. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns. See the User Guide for more on reshaping.

syntax: `pandas.pivot_table(data, values=None, index=None, columns=None)`

Example: `pivot_tab=pd.pivot_table(df, values='likes', index=['Type','Category'])`

Conclusion: Hence, students have performed about various operations using python on the facebook metrics datasets.

FAQ's:

1) What is datasets? Define Features and instances in datasets?

2) What are different data types in Pandas?

3) Explain Python libraries which used in Data Science?

4) What can you do with Pandas?

5) What can you do with NumPy?

6) What are the difference between 'loc()' and 'iloc()' functions.

7) Which methods are used for reshape data?

Source Code with output: