Assignment No 1

Aim:

Perform the following operations using R/Python on suitable data sets:

a) read data from different formats (like csv, xls)

b) Find Shape of Data

c) Find Missing Values

d) Find data type of each column

e) Finding out Zero's

f) Indexing and selecting data, sort data,

g) Describe attributes of data, checking data types of each column,

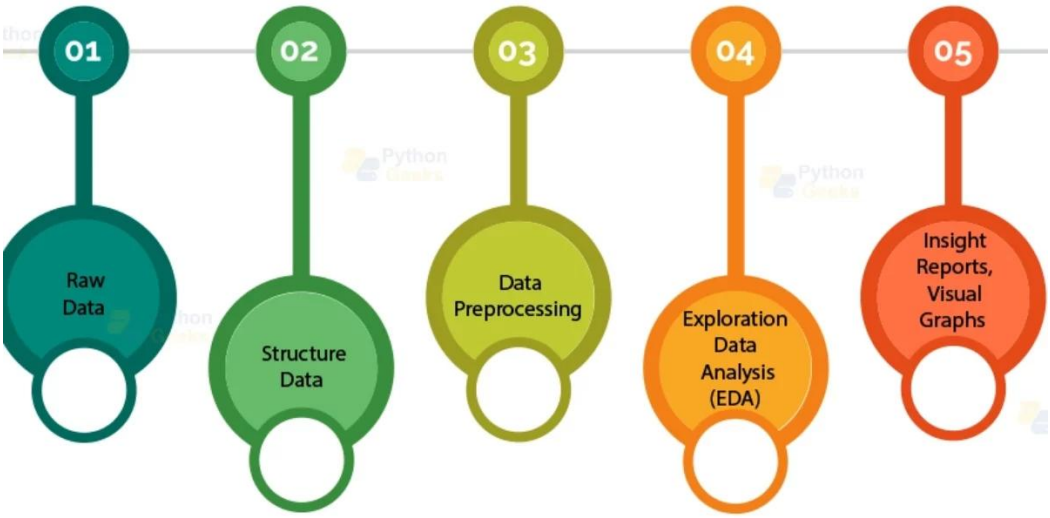
h) counting unique values of data, format of each column, converting variable

Theory:

What is Data Preprocessing?

Data preprocessing is a crucial step in the data analysis pipeline. It includes cleaning, transforming, and organizing raw data into a usable format. Without proper preprocessing, data analysis or machine learning models may yield inaccurate results.

Data is usually collected from multiple sources and stored in formats like CSV, XLS/XLSX, JSON, etc. Before conducting any statistical or machine learning task, we need to inspect, clean, and understand the data through exploratory techniques.



Methods and Explanations of Operations

a) Reading Data from Different Formats (CSV, XLS)

- - CSV (Comma-Separated Values): A plain text file format that stores tabular data.
- - XLS/XLSX: Excel spreadsheet files. Reading these formats requires additional libraries in Python like pandas with openpyxl or xlrd, and readxl in R.
- - Why Important: Helps bring external data into your working environment for processing and analysis.

b) Find Shape of Data

- - Represents the number of rows and columns in a dataset.
- - Syntax: (rows, columns)
- - Why Important: Understanding the shape is the first step in knowing what you're working with.

c) Find Missing Values

- - Real-world datasets often have missing entries.
- - Missing values are typically represented as NaN, NA, or blanks.
- - Why Important: They can skew analysis, so you need to either fill, impute, or drop them.

d) Find Data Type of Each Column

- - Helps in understanding how each column is interpreted: numerical, string, boolean, datetime, etc.
- - Why Important: Many operations are data-type sensitive. For instance, you can't compute the mean of a text field.

e) Finding Out Zeros

- - Zero values may or may not be significant.
- - Sometimes zeros indicate missing values or errors (especially in medical or survey data).
- - Why Important: Differentiating actual values from placeholders helps with data cleaning.

f) Indexing and Selecting Data, Sorting Data

- - Indexing: Refers to accessing specific rows or columns.
- - Selection: Filter data based on condition.
- - Sorting: Reorder data based on column values (ascending/descending).
- - Why Important: Gives flexibility in analyzing data slices or ordering it for better visibility.

g) Describe Attributes of Data, Checking Data Types of Each Column

- - Includes mean, median, min, max, standard deviation, quartiles.
- - Helps to get a statistical summary of each column.
- - Why Important: Quickly provides distribution and spread of numerical data, helping to identify outliers and trends.

h) Counting Unique Values, Format of Each Column, Converting Variable Data Type

- - Unique Counts: Useful for categorical variables to know diversity.
- - Format: Consistency check (e.g., dates, strings).
- - Conversion: Sometimes needed for modeling (e.g., converting float to integer, or string to category).
- - Why Important: Improves model performance and ensures compatibility between tools and libraries.

Outputs:

jupyter ML_Assignment_1 Last Checkpoint: 2 months ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab

Python 3 (ipykernel)

```
# Perform the following operations using Python on suitable data sets:
# a) read data from different formats (like csv, xls)
# b) find Shape of Data
# c) find Missing values
# d) find data type of each column
# e) finding out Zero's
# f) Indexing and selecting data, sort data,
# g) describe attributes of data, checking data types of each column,
# h) counting unique values of data, format of each column, converting variable
# data type (e.g. from long to short, vice versa)
```

```
import numpy as np
import pandas as pd
```

```
def d_():
    print("="*30)
```

```
dataset_path = "C:\\Users\\sonam\\Desktop\\ML_Jupyter\\datasets\\admission.csv"
df = pd.read_csv(dataset_path)
d_()
print("Head of the df: ")
print(df.head())
d_()
```

```
==============================
Head of the df:
   Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  \
0           1        337          118                  4  4.5   4.5  9.65
1           2        324          107                  4  4.0   4.5  8.87
2           3        316          104                  3  3.0   3.5  8.00
3           4        322          110                  3  3.5   2.5  8.67
4           5        314          103                  2  2.0   3.0  8.21

   Research  Chance of Admit
0         1             0.92
1         1             0.76
2         1             0.72
3         1             0.80
4         0             0.65
==============================
```

```
d_()
print("Shape of Dataframe")
print(df.shape)
d_()
```

```
==============================
Shape of Dataframe
(500, 9)
==============================
```

```
d_()
print("Missing values of the dataset:")
print(df.isnull())
d_()
print("Sum of Missing values of the dataset:")
print(df.isnull().sum())
d_()
```

```
==============================
Missing values of the dataset:
     Serial No.  GRE Score  TOEFL Score  University Rating    SOP   LOR   \
0         False      False        False             False  False  False
1         False      False        False             False  False  False
2         False      False        False             False  False  False
3         False      False        False             False  False  False
4         False      False        False             False  False  False
..          ...        ...          ...               ...    ...    ...
495       False      False        False             False  False  False
496       False      False        False             False  False  False
497       False      False        False             False  False  False
498       False      False        False             False  False  False
499       False      False        False             False  False  False

     CGPA  Research  Chance of Admit
0   False     False            False
1   False     False            False
2   False     False            False
3   False     False            False
4   False     False            False
..    ...       ...              ...
495 False     False            False
496 False     False            False
497 False     False            False
498 False     False            False
499 False     False            False

[500 rows x 9 columns]
==============================
Sum of Missing values of the dataset:
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
==============================
```

```
d_()
print("Datatype of Each Column: ")
print(df.info())
d_()
```

```
==============================
Datatype of Each Column:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Serial No.         500 non-null    int64
 1   GRE Score          500 non-null    int64
 2   TOEFL Score        500 non-null    int64
 3   University Rating  500 non-null    int64
 4   SOP                500 non-null    float64
 5   LOR                500 non-null    float64
 6   CGPA               500 non-null    float64
 7   Research           500 non-null    int64
 8   Chance of Admit    500 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 35.3 KB
None
==============================
```

```
print("Zeros in Each Column: ")
print((df==0).sum())
d_()
```

```
==============================
Zeros in Each Column:
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research           220
Chance of Admit      0
dtype: int64
==============================
```

```
print("Sorting df on basis of column: ")
print(df.sort_values(by='GRE Score', ascending=True))
d_()
```

```
==============================
Sorting df on basis of column:
     Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  \
377         378        290          100                  1  1.5   2.0  7.56
117         118        290          104                  4  2.0   2.5  7.46
108         109        293           97                  2  2.0   4.0  7.88
79           80        294           93                  1  1.5   2.0  7.36
272         273        294           95                  1  1.5   1.5  7.64
..          ...        ...          ...                ...  ...   ...   ...
81           82        340          120                  4  5.0   5.0  9.50
84           85        340          115                  5  4.5   4.5  9.45
143         144        340          120                  4  4.5   4.0  9.92
304         305        340          113                  4  5.0   5.0  9.74
429         430        340          115                  5  5.0   4.5  9.86

     Research  Chance of Admit
377         0             0.47
117         0             0.45
108         1             0.64
79          0             0.46
272         0             0.49
..        ...              ...
81          1             0.96
84          1             0.94
143         1             0.97
304         1             0.96
429         1             0.95

[500 rows x 9 columns]
==============================
```

```
d_()
print("Description of Dataframe: ")
print(df.describe())
d_()
```

```
==============================
Description of Dataframe:
       Serial No.  GRE Score  TOEFL Score  University Rating         SOP  \
count  500.000000  500.000000   500.000000         500.000000  500.000000
mean   250.500000  316.472000   107.192000           3.114000    3.374000
std    144.481833   11.295148     6.081868           1.143512    0.991004
min      1.000000  290.000000    92.000000           1.000000    1.000000
25%    125.750000  308.000000   103.000000           2.000000    2.500000
50%    250.500000  317.000000   107.000000           3.000000    3.500000
75%    375.250000  325.000000   112.000000           4.000000    4.000000
max    500.000000  340.000000   120.000000           5.000000    5.000000

             LOR        CGPA    Research  Chance of Admit
count  500.00000  500.000000  500.000000       500.00000
mean     3.48400    8.576440    0.560000         0.72174
std      0.92545    0.604813    0.496884         0.14114
min      1.00000    6.800000    0.000000         0.34000
25%      3.00000    8.127500    0.000000         0.63000
50%      3.50000    8.560000    1.000000         0.72000
75%      4.00000    9.040000    1.000000         0.82000
max      5.00000    9.920000    1.000000         0.97000
==============================
```

```
unique_counts = df.nunique()
print("Unique Counts:\n", unique_counts)
d_()
```

```
==============================
Unique Counts:
 Serial No.           500
GRE Score             49
TOEFL Score           29
University Rating      5
SOP                    9
LOR                    9
CGPA                 184
Research               2
Chance of Admit       61
dtype: int64
==============================
```

```
d_()
df['SOP'] = df['SOP'].astype(float)
print("Updated Data Types after Conversion:\n", df.dtypes)
d_()
```

```
==============================
Updated Data Types after Conversion:
```

Conclusion

We have successfully performed all essential data preprocessing tasks using R/Python as part of this assignment. We started with importing data from various formats and explored different techniques such as identifying missing values, checking and converting data types, indexing and sorting, detecting zeros, and summarizing attributes.Through this assignment, I have learned how crucial preprocessing is to prepare raw datasets for analysis or modeling. These skills are fundamental for ensuring the accuracy and reliability of results in any data-driven project.