**Assignment 1**

# Problem Statement:-

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

* 1. read data from different formats (like csv, xls)
  2. indexing and selecting data, sort data,
  3. describe attributes of data, checking data types of each column,
  4. counting unique values of data, format of each column, converting variable data type (e.g. from long to short, vice versa),
  5. identifying missing values and fill in the missing values.

# Library:

1)Software used : VS code

2)Library used : Pandas, Matplotlib.

# Theory:

Data Preparation in Machine Learning

Data Preparation is the process of cleaning and transforming raw data to make predictions accurately through using ML algorithms. Although data preparation is considered the most complicated stage in ML, it reduces process complexity later in real-time projects. Various issues have been reported during the data preparation step in machine learning as follows:

* + - **Missing data:** Missing data or incomplete records is a prevalent issue found in most datasets. Instead of appropriate data, sometimes records contain empty cells, values (e.g., NULL or N/A), or a specific character, such as a question mark, etc.
    - **Outliers or Anomalies:** ML algorithms are sensitive to the range and distribution of values when data comes from unknown sources. These values can spoil the entire machine learning training system and the performance of the model. Hence, it is essential to detect these outliers or anomalies through techniques such as visualization technique.
    - **Unstructured data format:** Data comes from various sources and needs to be extracted into a different format. Hence, before deploying an ML project, always consult with domain experts or import data from known sources.
    - **Limited Features:** Whenever data comes from a single source, it contains limited features, so it is necessary to import data from various sources for feature enrichment or build multiple features in datasets.
    - **Understanding feature engineering:** Features engineering helps develop additional content in the ML models, increasing model performance and accuracy in predictions.

**Methodology:**

a) Read Data from Different Formats:

1. the necessary library for data manipulation, such as Pandas for Python.

2.Use the appropriate function to read data from various formats like CSV, Excel (XLS/XLSX), JSON, etc.

3.Specify the file path or URL of the dataset as the input to the read function.

4.Verify that the data has been successfully loaded into a DataFrame or other suitable data structure.

b) Indexing and Selecting Data, Sorting Data:

1.Once the data is loaded, use indexing and selection techniques to access specific rows, columns, or subsets of the dataset.

2.Use methods like iloc[] or loc[] for integer-based or label-based indexing respectively.

3.Apply sorting operations using the sort\_values() function to sort the data based on one or more columns.

5.Specify the column(s) to sort by and the desired sorting order (ascending or descending).

c) Describe Attributes of Data, Checking Data Types of Each Column:

1.Use the info() method to obtain a concise summary of the DataFrame, including information about the data types of each column, memory usage, and non-null counts.

2.Alternatively, use the dtypes attribute to directly access the data types of each column.

3. descriptive statistics functions like describe () to obtain summary statistics such as mean, median, standard deviation, etc., for numerical columns.

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

1.Use the value\_counts() method to count the unique values of categorical variables or specific columns.

2.Access the data format of each column using the dtypes attribute or the info() method.

3.Convert the data type of a column using functions like astype() to convert to a specific data type (e.g., from object to integer or vice versa).

5.Use string manipulation functions to convert data types from long to short or vice versa if needed.

e) Identifying Missing Values and Fill in the Missing Values:

1.Use the isna() or isnull() methods to identify missing values in the dataset.

2.Determine the appropriate method to handle missing values based on the dataset and the nature of the missingness (e.g., mean imputation, median imputation, mode imputation, or using forward/backward fill).

3.Use the fillna() method to fill in missing values with the chosen strategy.

**Advantages:**

1.It is very easy to used library that’s why it is also famous library.

2.It provided powerful data structure like Series and Data Frame.

3.It comes with wide functionality for data manipulation.

**Disadvantages:**

1.Padas may consume significant memory while working with large dataset.

2. It very much integrated with python eco-system, which may limit its interoperability with other programming languages or environments.

**Conclusion:**

In summary, this assignment provided an introduction to the Pandas library, an essential tool for data manipulation and analysis in Python. We explored its basic functions, such as reading various data formats, organizing and describing data, and handling missing values. Through practical exercises, we gained a better understanding of how Pandas can simplify complex data tasks, making data analysis more accessible and efficient. These foundational skills with Pandas will undoubtedly serve as a solid starting point for tackling more advanced data analysis projects in the future.