Group A

Assignment No: 1

Title of the Assignment: Data Wrangling, I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

Import all the required Python Libraries.

1. Locate open source data from the web (e.g. https://www.kaggle.com).
2. Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into the 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 descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
6. Turn categorical variables into quantitative variables in Python.

Introduction to Data Wrangling

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

With the amount of data and data sources rapidly growing and expanding, it is getting increasingly essential for large amounts of available data to be organized for analysis. This process typically includes manually converting and mapping data from one raw form into another format to allow for more convenient consumption and organization of the data.

The Goals of Data Wrangling

Reveal a "deeper intelligence" by gathering data from multiple sources

Provide accurate, actionable data in the hands of business analysts in a timely matter

Reduce the time spent collecting and organizing unruly data before it can be utilized

Enable data scientists and analysts to focus on the analysis of data, rather than the wrangling

Drive better decision-making skills by senior leaders in an organization

Key Steps to Data Wrangling

Data Acquisition: Identify and obtain access to the data within your sources.

Joining Data: Combine the edited data for further use and analysis.

Data Cleansing: Redesign the data into a usable and functional format and correct/remove any bad data.

Panda functions for Data Formatting and Normalization

The Transforming data stage is about converting the data set into a format that can be analyzed or modelled effectively, and there are several techniques for this process.

a. **Data Formatting:** Ensuring all data formats are correct (e.g. object, text,floating number, integer, etc.) is another part of this initial 'cleaning' process. If you are working with dates in Pandas, they also need to be stored in the exact format to use special date-time functions.

b. **Data normalization:** Mapping all the nominal data values onto a uniform scale (e.g. from 0 to 1) is involved in data normalization. Making the ranges consistent across variables helps with statistical analysis and ensures better comparisons later on..

Algorithm:

Step 1 : Import all the required Python Libraries.

Step 2: Load the weather dataset from Kaggle in dataframe object df

Step 3: Print dataset.

```
 df.head()
df.tail()
```

Step 4.1: Describe function returns the statistical summary of the data frame or series.

```
df.describe()
```

Step 4.2: identify missing data A) convert the "?","blackspace" into NaN(non numerical data) function: replace() The replace () method replaces the specified value with another specified value. The replace() method searches the entire Data Frame and replaces every case of specified value.

```
df.replace("  ", np.nan, inplace = True)
 df.head(9)
```

Step 4.3: Check if missing value are present.

```
missingdata=df.isnull()
df.isnull().sum()
```

Step 4.4: Deal with missing Data a. Drop data Functions:dropna() b. replace data calculate the mean()

```
df.dropna()
```

Step 4.5: Replace Data Calculate the mean and replace the null value by mean by using fillna() function

```
mean_value = df['temperature'].mean()

df['temperature'] = df['temperature'].fillna(mean_value)

mean_v=df['windspeed'].mean()

new_df = df.fillna({'temperature':0,

'windspeed':0,

'event':'no_event'})'event':'no_event'})
```

```
new_df
```

Step 5: Summarize the types of variables by checking the data types

```
df.dtypes

df[["duration"]]=df[["duration"]].astype("float")

df.head()
```

Step 7: Turn categorical variables into quantitative variables

```
df3 = df.copy()

df3 = pd.get_dummies(df3,

                     columns = ['event'])

  display(df3)
```

Conclusion- In this way we have explored the functions of the python library for Data Preprocessing, Data Wrangling Techniques and How to Handle missing values.