Online Assessment (5%)

Name : Rinashini A/P Arunasalam Sukormaru

Matric ID : WQD170077

1. Importing the dataset and printing the first 5 rows of the dataset

df = pd.read\_csv("dataset.csv")  
print(df.head())

Results:

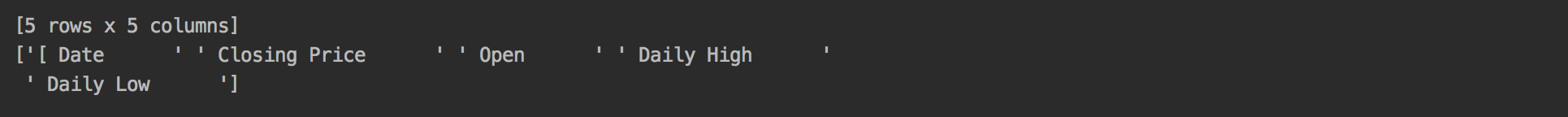
A screenshot of a cell phone

Description automatically generated

1. Cleaning the column names of the dataset
2. Printing the column names

print(df.columns.values)

Results: As we can observe, the “Date” column has a square bracket “ [ “ before it, hence, we will remove the “ [ “ and also the blank spaces after each column names



1. Removing the “ [ “ and the blank spaces before and after the column names

df.columns = df.columns.str.strip().str.replace('[', '').str.lstrip()  
print(df.columns.values)

Results:



1. Determining the number of missing values in each column
2. Determining the size of the dataset

print(df.shape)

Results:



1. Showing example of rows with blank values

print(df[52:58])

Results:

A close up of a screen

Description automatically generated

1. Replacing the blank values with “NaN”

df = df.replace(r'^\s\*$', pd.np.NaN, regex=True)  
print(df[52:58])

Results: Showing that the previous rows with blank values have now been replaced with “NaN”

A screen shot of a computer

Description automatically generated

1. Showing the number of missing values in each column

col\_missing = df.isnull().sum()  
print(col\_missing)

Results: The results show that only “Open”, “Daily High” and “Daily Low” columns have missing values

A close up of a logo

Description automatically generated

1. Data imputation of “NaN” with the column mean values of the dataset
2. Converting the values in each column to numeric form to enable the computation of each columns’ mean value

df['Date'] = pd.to\_datetime(df['Date'])  
df['Closing Price'] = pd.to\_numeric(df['Closing Price'])  
df['Open'] = pd.to\_numeric(df['Open'])  
df['Daily High'] = pd.to\_numeric(df['Daily High'])  
df['Daily Low'] = pd.to\_numeric(df['Daily Low'])  
print(df.dtypes)

Results: The format type of the “Date” column is datetime64 while the format type of the “Closing Price”, “Open”, “Daily High” and “Daily Low” are float64.

A close up of a logo

Description automatically generated

1. Obtaining the mean values of each column and imputing the “NaN” values with the mean values of its own column

df\_NaN = df.dropna()  
Open\_mean\_value = df\_NaN['Open'].mean().round(2)  
df['Open'] = df['Open'].fillna(Open\_mean\_value)  
  
Daily\_High\_mean\_value = df\_NaN['Daily High'].mean().round(2)  
df['Daily High'] = df['Daily High'].fillna(Daily\_High\_mean\_value)  
  
Daily\_Low\_mean\_value = df\_NaN['Daily Low'].mean().round(2)  
df['Daily Low'] = df['Daily Low'].fillna(Daily\_Low\_mean\_value)  
  
print(df[52:58])

Results: Showing that the previous rows with “NaN” values have now been replaced with the mean values of its own column

A close up of a screen

Description automatically generated

1. Exporting the dataset into a csv file

df.to\_csv(r'dataset1.csv', index=False)

Results:

A screenshot of a cell phone

Description automatically generated