MEASURING ENERGY CONSUMPTION

# PHASE 4

# CODE:

# Importing necessary libraries

import os

import pandas as pd

#import plotly.express as px

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

def load\_and\_process\_data\_in\_folder(folder\_path, energy\_column\_index=1):

# Get a list of all CSV files in the specified folder

csv\_files = [f for f in os.listdir(folder\_path) if f.endswith('.csv')]

# Initialize an empty list to store DataFrames

dataframes = []

for file\_name in csv\_files:

# Read the CSV file into a pandas DataFrame

file\_path = os.path.join(folder\_path, file\_name)

df = pd.read\_csv(file\_path)

# Clean and preprocess the data

df = df.drop\_duplicates()

df = df.dropna()

dataframes.append(df) #Appending the processed data files

return dataframes

def calculate\_energy\_consumption(dataframes, energy\_column\_index=1): #energy\_coulumn\_index indicates the 2nd column that has the energy consumption data

for df in dataframes:

if df is not None and len(df.columns) > energy\_column\_index:

df['Datetime'] = pd.to\_datetime(df['Datetime']) # Convert the "Datetime" column to datetime data type

# Extract date and time components as new features

df['Year'] = df['Datetime'].dt.year

df['Month'] = df['Datetime'].dt.month

df['Day'] = df['Datetime'].dt.day

df['Hour'] = df['Datetime'].dt.hour

df['Minute'] = df['Datetime'].dt.minute

df.drop(['Datetime'], axis=1, inplace=True) # Drop the original "Datetime" column

energy\_column = df.columns[energy\_column\_index]

# Split the data into features (X) and the target variable (y)

X = df.drop([energy\_column], axis=1) # Features

y = df[energy\_column] # Target variable

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Train the Model

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error (MSE) : {mse}")

print(f"R-squared (R2): {r2}")

Main Program:

folder\_path = “D:\\NM” # Replace with your folder path

dfs = load\_and\_process\_data\_in\_folder(folder\_path)

energy\_consumption = calculate\_energy\_consumption(dfs)

# OUTPUT:

