import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

RED = "\033[91m"

GREEN = "\033[92m"

YELLOW = "\033[93m"

BLUE = "\033[94m"

RESET = "\033[0m"

df = pd.read\_csv("C:/Users/PMCTECH/Documents/AEP\_hourly.csv")

df["Datetime"] = pd.to\_datetime(df["Datetime"])

# DATA CLEANING

print(BLUE + "\nDATA CLEANING" + RESET)

# --- Check for missing values

missing\_values = df.isnull().sum()

print(GREEN + "Missing Values : " + RESET)

print(missing\_values)

# --- Handle missing values

df.dropna(inplace=True)

# --- Check for duplicate values

duplicate\_values = df.duplicated().sum()

print(GREEN + "Duplicate Values : " + RESET)

print(duplicate\_values)

# --- Drop duplicate values

df.drop\_duplicates(inplace=True)

# DATA ANALYSIS

print(BLUE + "\nDATA ANALYSIS" + RESET)

# --- Summary Statistics

summary\_stats = df.describe()

print(GREEN + "Summary Statistics : " + RESET)

print(summary\_stats)

# Data Visualization

# Line plot for energy consumption over time

plt.figure(figsize=(12, 6))

plt.plot(df.index, df["AEP\_MW"], label="Energy Consumption (AEP\_MW)")

plt.xlabel("Datetime")

plt.ylabel("Energy Consumption (MW)")

plt.title("Energy Consumption Over Time")

plt.grid()

plt.legend()

plt.show()

# SAVING THE FILE

df.to\_csv("C:/Users/PMCTECH/Documents/AEP\_hourly.csv", index=False)

print(BLUE + "\nDATA ANALYSIS" + RESET)

print(GREEN + "Data Cleaned and Saved !" + RESET)

import psutil

# Measure energy consumption before feature engineering

before\_energy = psutil.cpu\_percent()

# Your feature engineering code here

# Measure energy consumption after feature engineering

after\_energy = psutil.cpu\_percent()

energy\_consumption\_feature\_engineering = after\_energy - before\_energy

print(f"Energy consumption during feature engineering: {energy\_consumption\_feature\_engineering}%")

import psutil

# Measure energy consumption before model training

before\_energy = psutil.cpu\_percent()

# Your model training code here

# Measure energy consumption after model training

after\_energy = psutil.cpu\_percent()

energy\_consumption\_training = after\_energy - before\_energy

print(f"Energy consumption during model training: {energy\_consumption\_training}%")

import psutil

# Measure energy consumption before model evaluation

before\_energy = psutil.cpu\_percent()

# Your model evaluation code here

# Measure energy consumption after model evaluation

after\_energy = psutil.cpu\_percent()

energy\_consumption\_evaluation = after\_energy - before\_energy

print(f"Energy consumption during model evaluation: {energy\_consumption\_evaluation}%")