

```
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}%")
```