MEASURE ENERGY CONSUMPTION

Phase 4

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings("ignore", category=UserWarning)

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score

RED = "\033[91m"
GREEN = "\033[92m"
YELLOW = "\033[93m"
BLUE = "\033[94m"
RESET = "\033[0m"

df = pd.read_csv("/kaggle/input/hourly-energy-consumption/AEP_hourly.csv")
```

```
df["Datetime"] = pd.to_datetime(df["Datetime"])
```

DATA CLEANING

```
print(BLUE + "\nDATA CLEANING" + RESET)
missing_values = df.isnull().sum()
print(GREEN + "Missing Values : " + RESET)
print(missing_values)
df.dropna(inplace=True)
duplicate_values = df.duplicated().sum()
print(GREEN + "Duplicate Values : " + RESET)
print(duplicate_values)
df.drop_duplicates(inplace=True)
```

DATA ANALYSIS

```
print(BLUE + "\nDATA ANALYSIS" + RESET)
summary_stats = df.describe()
print(GREEN + "Summary Statistics : " + RESET)
print(summary_stats)


print(BLUE + "\nMODELLING" + RESET)
df = df.sample(frac=0.2, random_state=42)
X = df[["Datetime"]]
y = df["AEP_MW"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
X_train["DayOfYear"] = X_train["Datetime"].dt.dayofyear
X_test["DayOfYear"] = X_test["Datetime"].dt.dayofyear
X_train = X_train["DayOfYear"].values.reshape(-1, 1)
X_test = X_test["DayOfYear"].values.reshape(-1, 1)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
svr = SVR(kernel="linear", C=1.0)
svr.fit(X_train_scaled, y_train)
y_pred = svr.predict(X_test_scaled)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color="b", label="Actual")
plt.scatter(X_test, y_pred, color="r", label="Predicted")
plt.xlabel("Day of the Year")
plt.ylabel("Energy Consumption (MW)")
plt.title("SVR Model: Actual vs. Predicted")
plt.legend()
plt.grid()
plt.show()
```

DATA VISUALIZATION

```
print(BLUE + "\nDATA VISUALIZATION" + RESET)
print(GREEN + "LinePlot : " + RESET)
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x="Datetime", y="AEP_MW")
plt.xlabel("Datetime")
plt.ylabel("Energy Consumption (MW)")
plt.title("Energy Consumption Over Year")
plt.grid()
plt.show()
print(GREEN + "Histogram : " + RESET)
plt.figure(figsize=(10, 6))
plt.hist(
    df["AEP_MW"],
    bins=100,
    histtype="barstacked",
    edgecolor="white",
)
plt.xlabel("AEPMW")
plt.ylabel("Frequency")
plt.title("Histogram of MEGAWATT USAGE")
plt.show()

df.to_csv("/kaggle/working/cleaned_AEP_hourly.csv", index=False)
print(BLUE + "\nDATA ANALYSIS" + RESET)
print(GREEN + "Data Cleaned and Saved !" + RESET)
```

OUTPUT:

DATA CLEANING

```
Missing Values :

Datetime        0

AEP_MW          0

dtype: int64

Duplicate Values :

0
```

DATA ANALYSIS

```
Summary Statistics :

                                   Datetime          AEP_MW

count                                121273   121273.000000

mean    2011-09-02 03:17:01.553025024     15499.513717

min                2004-10-01 01:00:00     9581.000000

25%                2008-03-17 15:00:00    13630.000000

50%                2011-09-02 04:00:00    15310.000000

75%                2015-02-16 17:00:00    17200.000000

max                2018-08-03 00:00:00    25695.000000

std                                   NaN     2591.399065
```

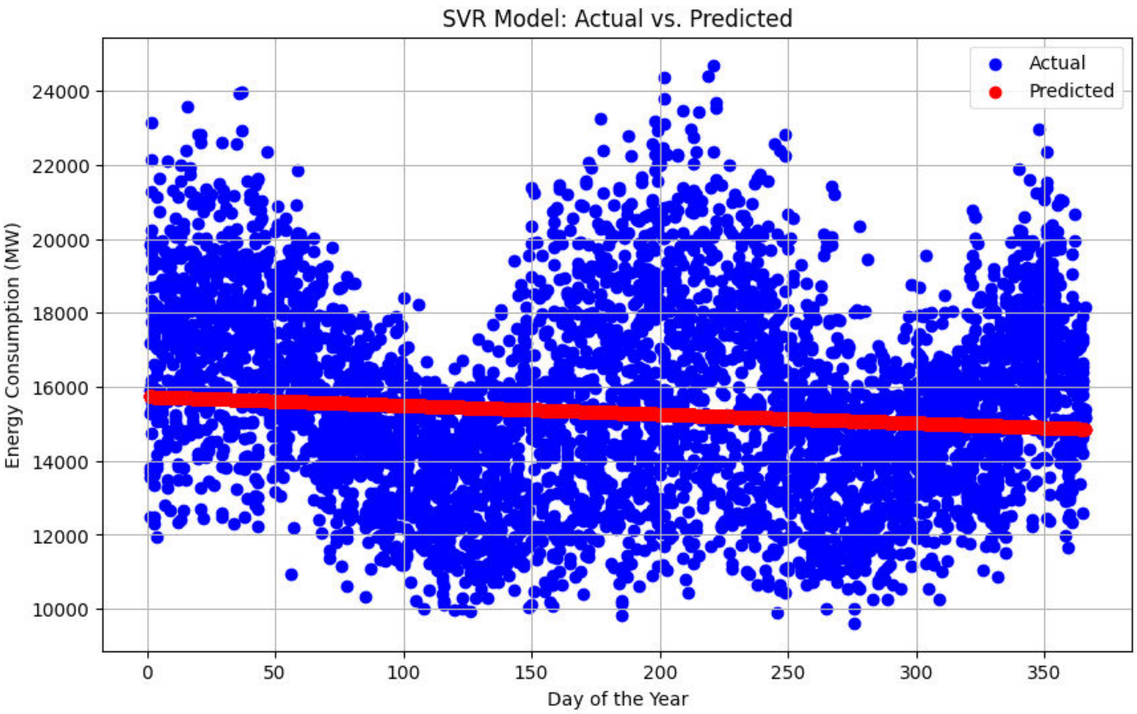
MODELLING

```
Mean Squared Error: 6758395.805638685

R-squared: 0.00270160624748228
```

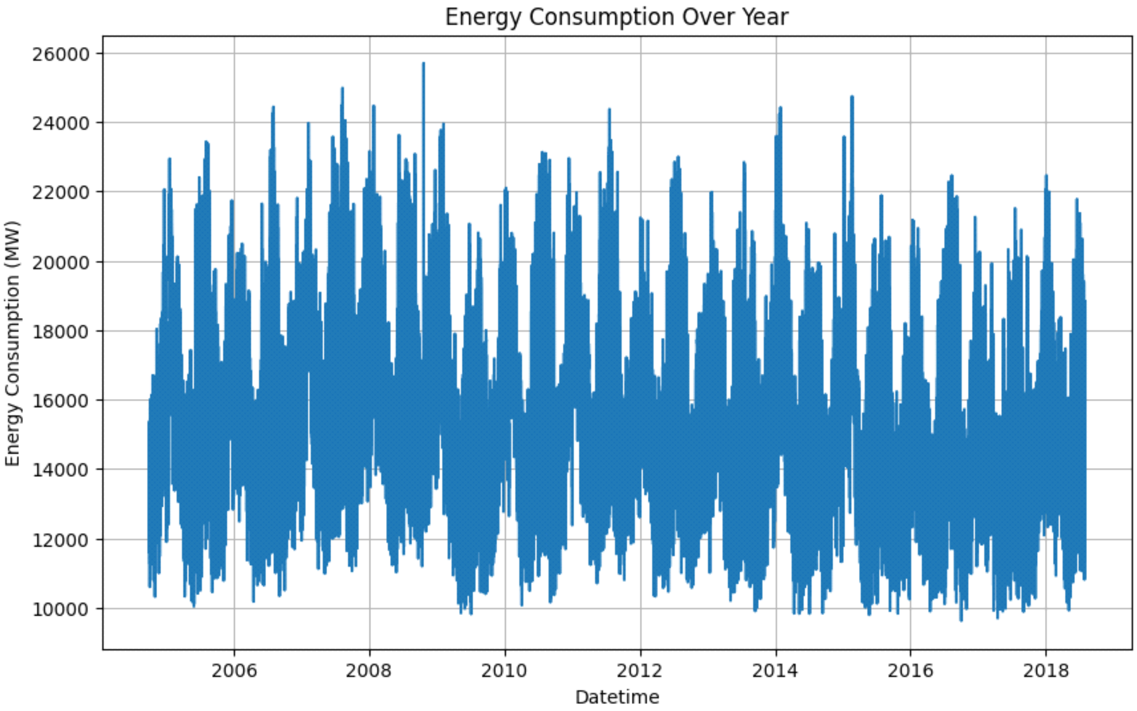
DATA VISUALIZATION

LinePlot :

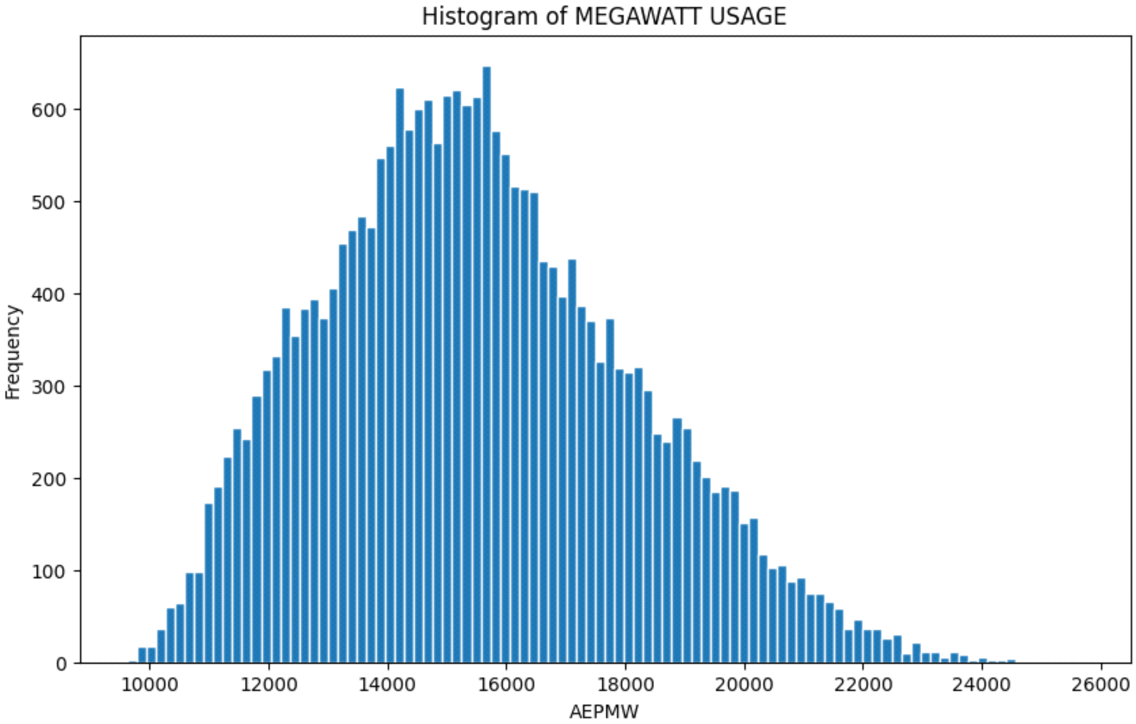


DATA VISUALIZATION

LinePlot :



HISTOGRAM


Histogram of MEGAWATT USAGE
Frequency
0
100
200
300
400
500
600
10000
12000
14000
16000
18000
20000
22000
24000
26000
AEPMW