Problem Definition:

The problem at hand is to create an automated system that measures energy consumption, analyzes the data, and provides visualizations for informed decision-making. This solution aims to enhance efficiency, accuracy, and ease of understanding in managing energy consumption across various sectors.

Design Thinking:

1. Data Source: Identify an available dataset containing energy consumption measurements.
2. Data Preprocessing: Clean, transform, and prepare the dataset for analysis
3. Feature Extraction: Extract relevant features and metrics from the energy consumption data.
4. Model Development: Utilize statistical analysis to uncover trends, patterns, and anomalies in the data
5. Visualization: Develop visualizations (graphs, charts) to present the energy consumption trends and insights

.

6. Automation: Build a script that automates data collection, analysis, and visualization processes.

MEASURE ENERGY CONSUMPTION

Phase 2

Methodology:

1. Identify the Device or Area: Determine whether you want to measure the energy consumption of a specific device (e.g., refrigerator) or an entire area (e.g., your home or office).

2. Choose a Measurement Tool: You can use an energy meter or a smart energy monitoring system. Energy meters are simple devices that you plug into an electrical outlet, and they display real-time energy usage. Smart energy monitoring systems may offer more features and connect to your smartphone or computer for remote monitoring.

3. Install the Device: If you're measuring a single device, plug it into the energy meter. If you're monitoring an entire area, you may need to install a smart energy monitoring system, which usually involves connecting it to your electrical panel.

4. Monitor and Record Data: Start monitoring the energy consumption over a period of time.Note the energy usage in kilowatt-hours (kWh). You can typically view this data on the device's display or through a mobile app or web portal if you're using a smart system.

5. Analyze the Data: Review the data to identify trends, peak usage times, and areas where you can reduce energy consumption.

6. Take Action: Based on your analysis, implement energy-saving measures, such as using energy-

efficient appliances, sealing drafts, or adjusting thermostat settings.

7. Monitor Continuously: Regularly check and record energy consumption to track your progress and make further adjustments as needed.

MEASURE ENERGY CONSUMPTION

USING MACHINE LEARNING

Project title: Measure Energy Consumption

Phase 3: Development part 1

Topic: *Start building the measure energy consumption model by loading and pre-processing the dataset*

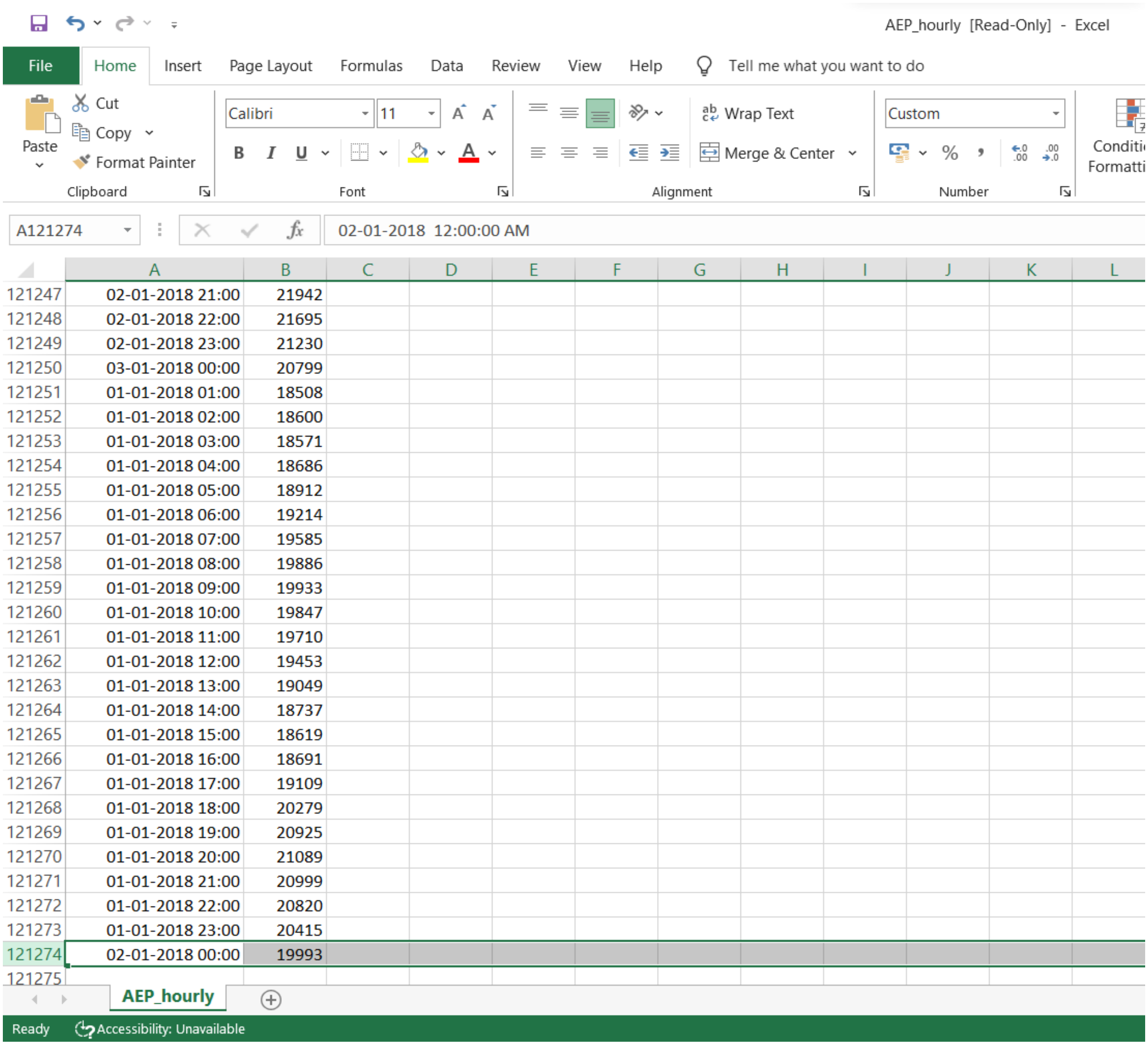
Measure energy consumption

Introduction:

Measuring energy consumption is a critical practice in today's world, as our reliance on energy sources continues to grow, and the environmental impact of our energy usage becomes increasingly apparent. Understanding and monitoring energy consumption is essential for a variety of reasons, including reducing costs, conserving resources, and mitigating the effects of climate change. This introduction will delve into the significance of measuring energy consumption and the various methods and tools used to do so.

Energy consumption measurement plays a pivotal role in our quest for sustainability and efficiency. It provides insights into how we use energy in our homes, businesses, industries, and transportation systems. By quantifying energy usage, we can identify areas where energy is wasted, make informed decisions to reduce consumption, and ultimately lower our carbon footprint.

Given data set:



AEP_hourly [Read-Only] - Excel

A121274 — 02-01-2018 12:00:00 AM

	A	B
121247	02-01-2018 21:00	21942
121248	02-01-2018 22:00	21695
121249	02-01-2018 23:00	21230
121250	03-01-2018 00:00	20799
121251	01-01-2018 01:00	18508
121252	01-01-2018 02:00	18600
121253	01-01-2018 03:00	18571
121254	01-01-2018 04:00	18686
121255	01-01-2018 05:00	18912
121256	01-01-2018 06:00	19214
121257	01-01-2018 07:00	19585
121258	01-01-2018 08:00	19886
121259	01-01-2018 09:00	19933
121260	01-01-2018 10:00	19847
121261	01-01-2018 11:00	19710
121262	01-01-2018 12:00	19453
121263	01-01-2018 13:00	19049
121264	01-01-2018 14:00	18737
121265	01-01-2018 15:00	18619
121266	01-01-2018 16:00	18691
121267	01-01-2018 17:00	19109
121268	01-01-2018 18:00	20279
121269	01-01-2018 19:00	20925
121270	01-01-2018 20:00	21089
121271	01-01-2018 21:00	20999
121272	01-01-2018 22:00	20820
121273	01-01-2018 23:00	20415
121274	02-01-2018 00:00	19993
121275		

AEP_hourly

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries.

Program:

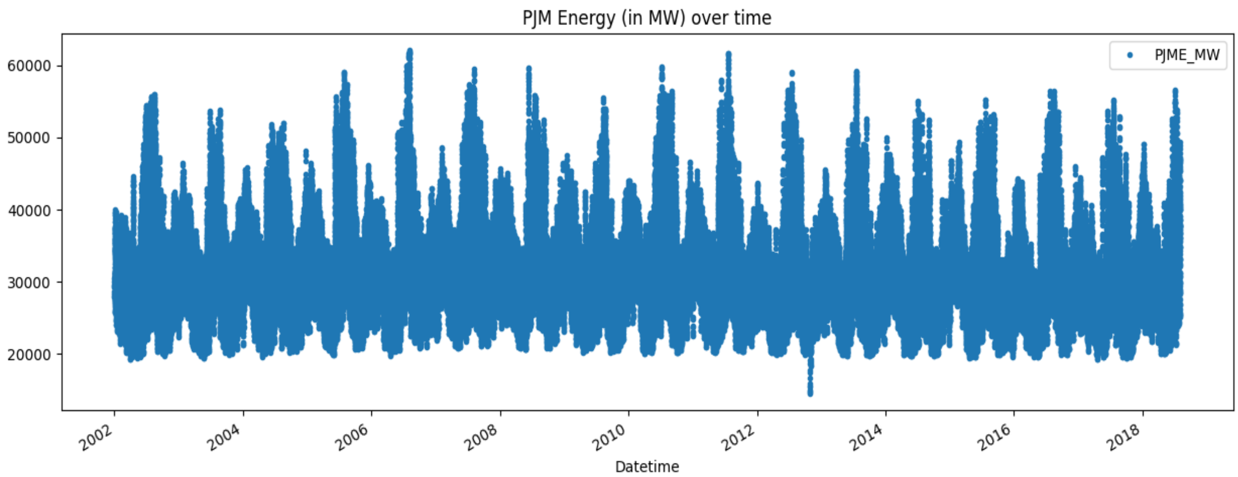
```
#import the libraries

import pandas as pd

import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns
pd.options.display.float_format = '{:.5f}'.format
pd.options.display.max_rows = 12
filepath = '../input/hourly-energy-consumption/PJME_hourly.csv'
df = pd.read_csv(filepath)
print("Now, you're ready for step one")
```

Explore the data:

```
 # turn data to datetime
df = df.set_index('Datetime')
df.index = pd.to_datetime(df.index)
df.plot(style='.',
        figsize=(15, 5),
        title='PJM Energy (in MW) over time')
plt.show()
```

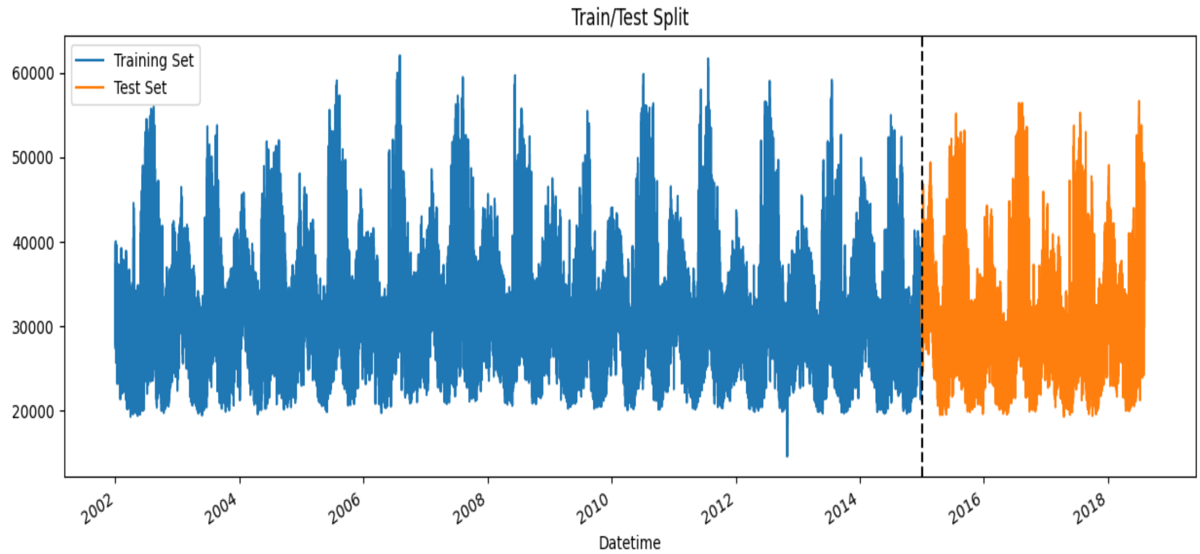
Output:



Split the data:

```
train = df.loc[df.index < '01-01-2015']

test = df.loc[df.index >= '01-01-2015']
```

Output:

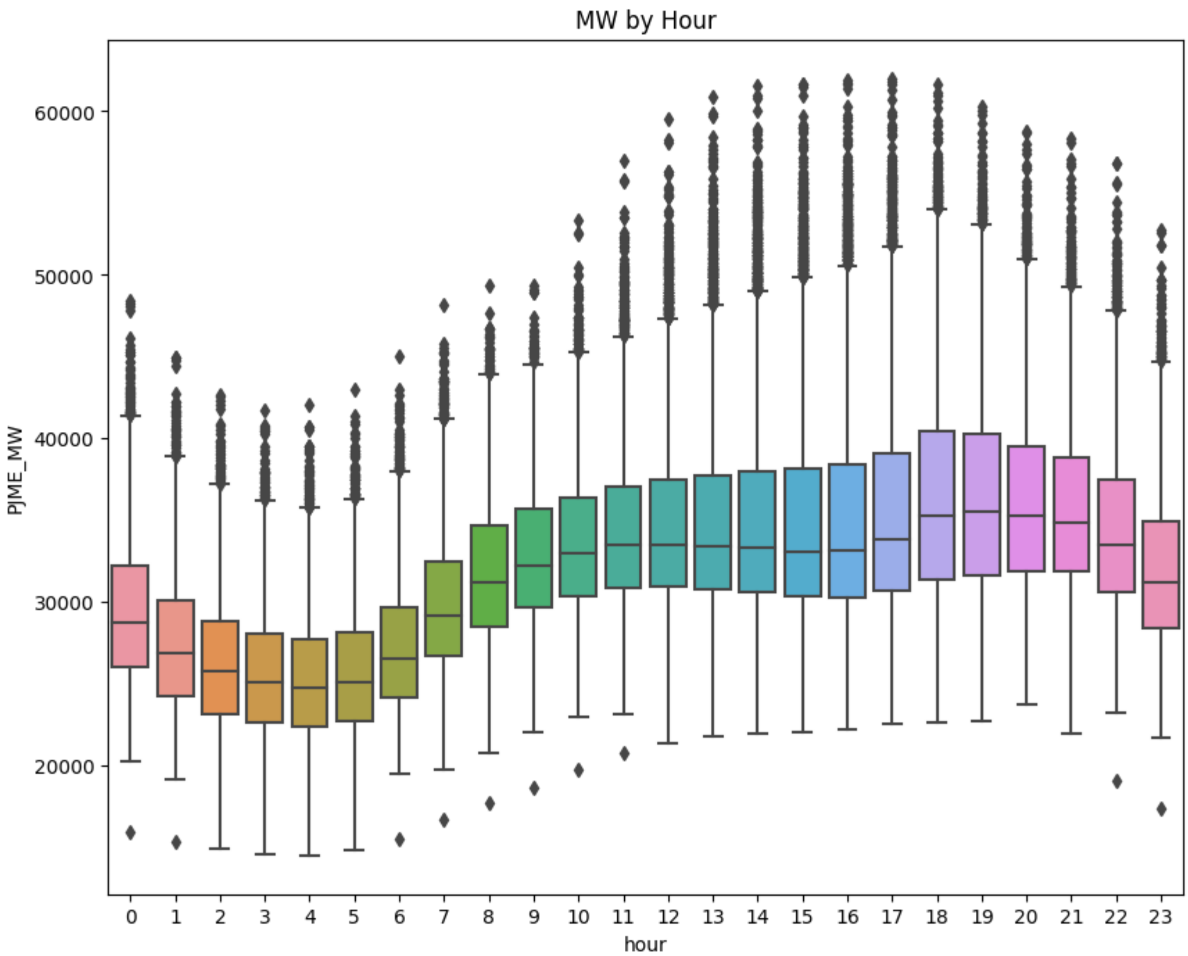


Features Engineering:

```
def create_features(df):
    df = df.copy()
    df['hour'] = df.index.hour
    df['dayofweek'] = df.index.dayofweek
    df['quarter'] = df.index.quarter
    df['month'] = df.index.month
    df['year'] = df.index.year
    df['dayofyear'] = df.index.dayofyear
    df['dayofmonth'] = df.index.day
    df['weekofyear'] = df.index.isocalendar().week
    return df
df = create_features(df)
```

```
fig, ax = plt.subplots(figsize=(10, 8))
sns.boxplot(data=df, x='hour', y='PJME_MW')
ax.set_title('MW by Hour')
plt.show()
```

virtualization:



Modelling and preprocessing :

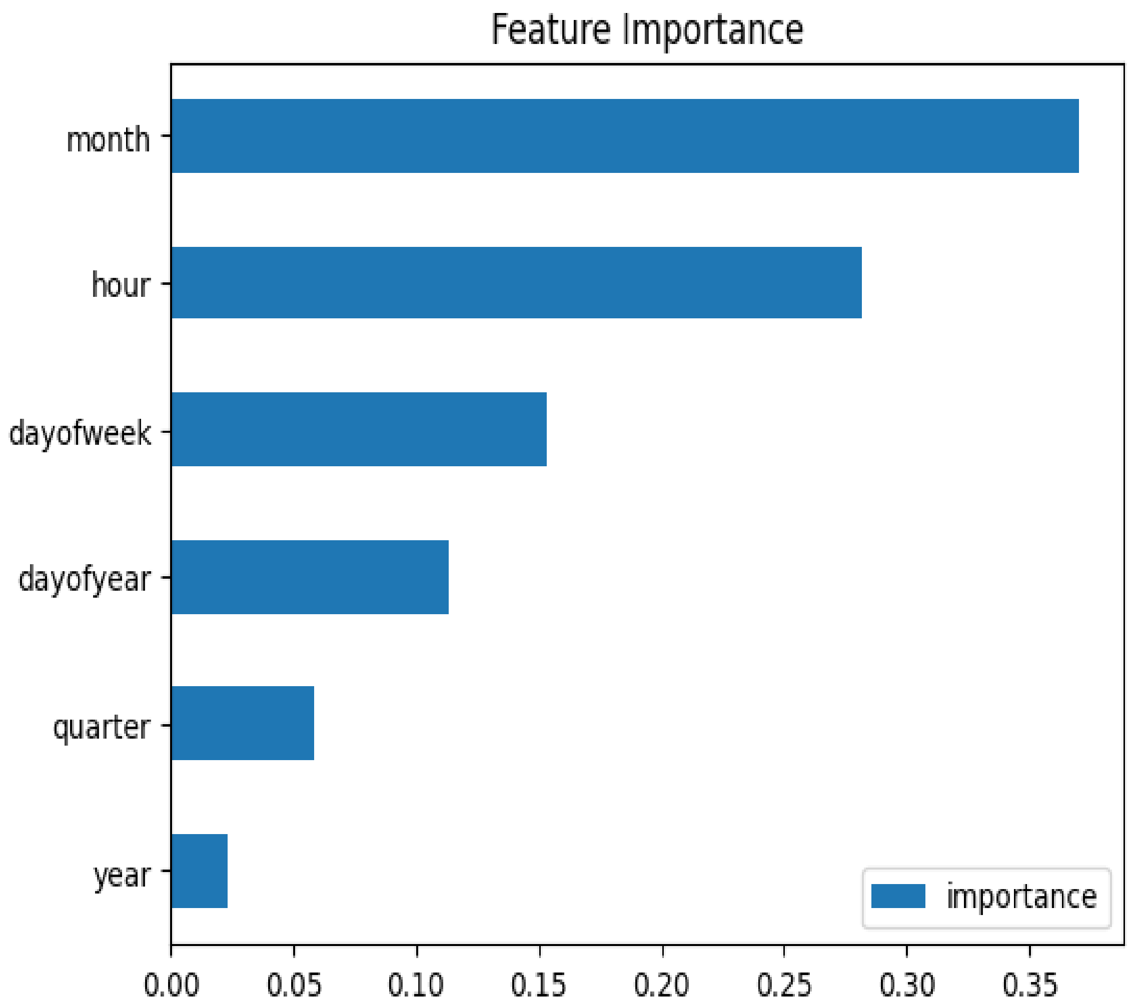
```
# preprocessing
train = create_features(train)
test = create_features(test)
features = ['dayofyear', 'hour', 'dayofweek', 'quarter', 'month', 'year']
target = 'PJME_MW'
```

```
X_train = train[features]
y_train = train[target]
X_test = test[features]
y_test = test[target]
```

Building the model:

```
import xgboost as xgb
from sklearn.metrics import mean_squared_error
# build the regression model
reg = xgb.XGBRegressor(base_score=0.5, booster='gbtree',
                       n_estimators=1000,
                       early_stopping_rounds=50,
                       objective='reg:linear',
                       max_depth=3,
                       learning_rate=0.01)
reg.fit(X_train, y_train,
        eval_set=[(X_train, y_train), (X_test, y_test)],
        verbose=100)
fi = pd.DataFrame(data=reg.feature_importances_,
             index=reg.feature_names_in_,
             columns=['importance'])
fi.sort_values('importance').plot(kind='barh', title='Feature
Importance')
plt.show()
```

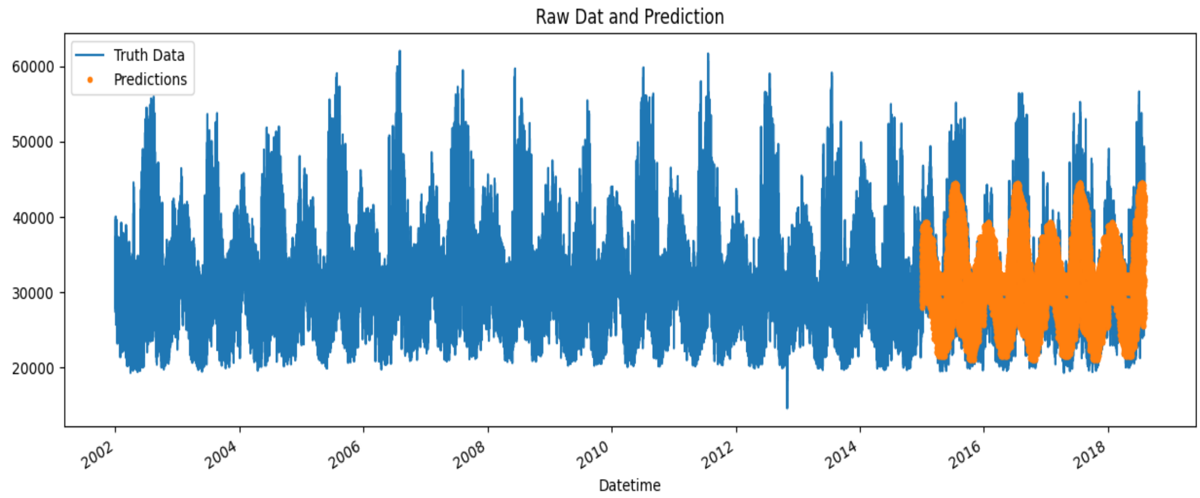
Output:



Forecasting on test data:

```
test['prediction'] = reg.predict(X_test)
df = df.merge(test[['prediction']], how='left', left_index=True,
right_index=True)
ax = df[['PJME_MW']].plot(figsize=(15, 5))
df['prediction'].plot(ax=ax, style='.')
plt.legend(['Truth Data', 'Predictions'])
ax.set_title('Raw Dat and Prediction')
plt.show()
```

Output:



Conclusion:

In conclusion, measuring energy consumption is not just a matter of tracking numbers; it's a pivotal practice that has far-reaching implications for our planet, our wallets, and our overall well-being. Whether at the individual, industrial, or governmental level, the act of quantifying and monitoring energy usage holds immense value.

By carefully measuring energy consumption, we can pinpoint inefficiencies, make informed decisions to reduce energy waste, and ultimately save money while reducing our impact on the environment. The information collected through these measurements empowers us to set and achieve energy-saving goals, contributing to a more sustainable and responsible world.

In the context of global climate change, energy consumption measurements are crucial in the fight against greenhouse gas emissions. They help us track our progress toward reducing our carbon footprint and implementing effective energy-efficient policies.

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

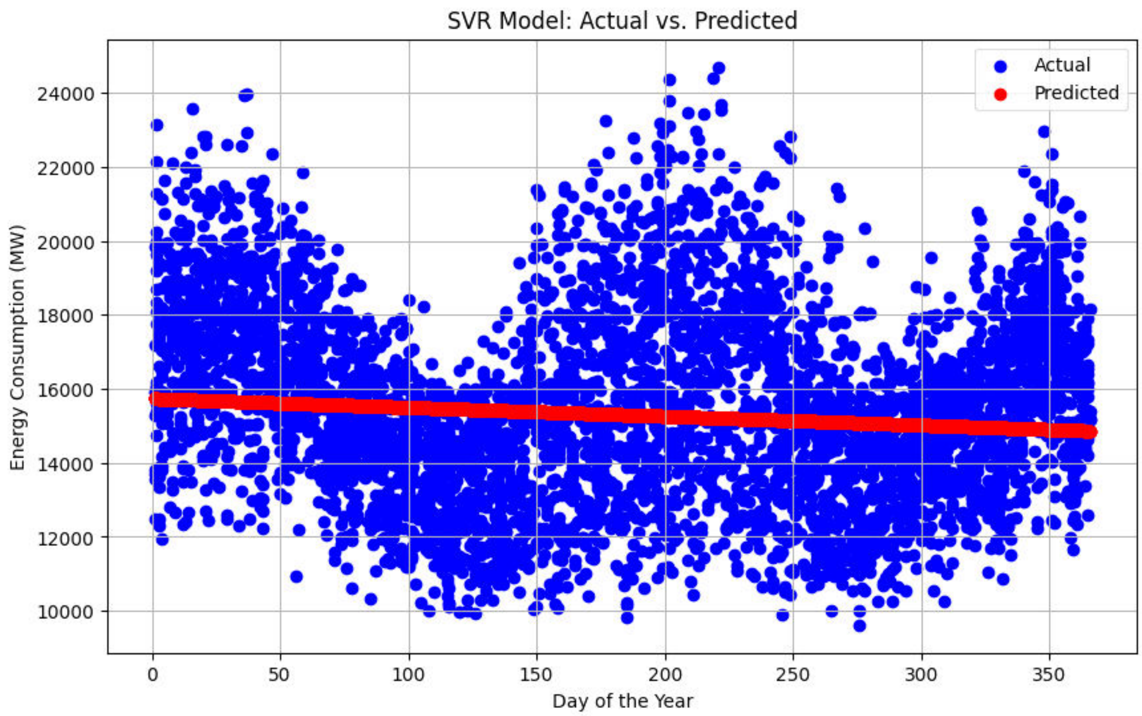
MODELLING

Mean Squared Error: 6758395.805638685

R-squared: 0.00270160624748228

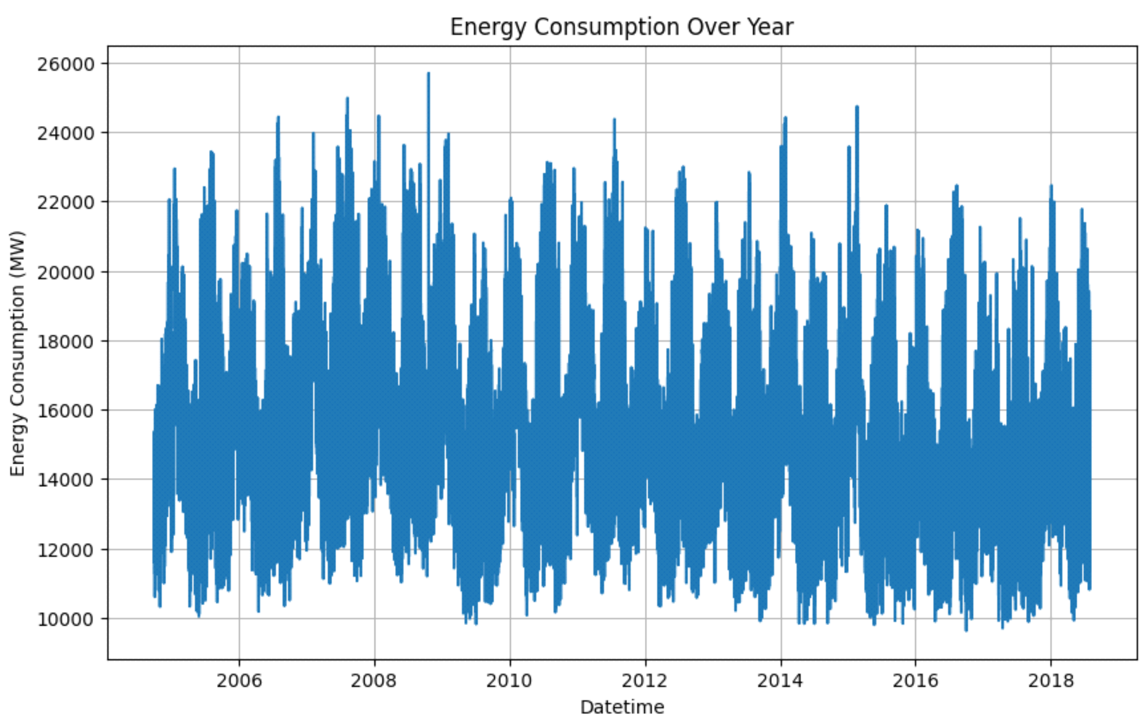
DATA VISUALIZATION

LinePlot :

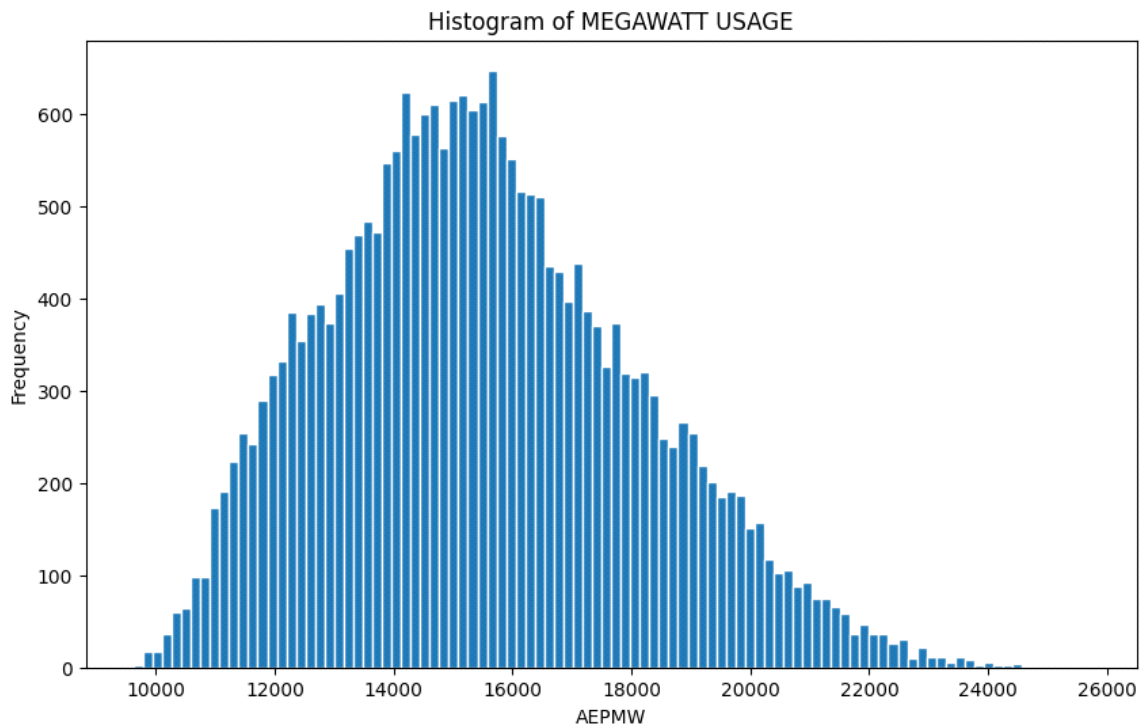


DATA VISUALIZATION

LinePlot :



HISTOGRAM


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