MEASURE ENERGY CONSUMPTION

NAME: VARSHA

REG NO: 962221106109

PROJECT TITLE: MEASURE ENERGY CONSUMPTION

PHASE 5: PROJECT SUBMISSION AND DOCUMENTATION

TOPIC: IN THIS SECTION WE WILL DOCUMENT THE COMPLETE PROJECT AND PREPARE IT FOR SUBMISSION.

MEASURE ENERGY CONSUMPTION

INTRODUCTION:

Energy consumption is a critical concern in today's world, with the growing need for sustainable energy practices and the increasing demand for efficient energy management. Machine learning offers a powerful tool to monitor, analyze, and optimize energy consumption in various domains, including residential, industrial, and commercial settings. This introduction outlines the fundamental concepts and benefits of using machine learning for measuring energy consumption.

1. WHY MEASURE ENERGY CONSUMPTION:

- Energy is a valuable and limited resource, and its efficient use is essential for environmental and economic reasons.

- **Accurate measurement and analysis of energy consumption enable cost savings, reduced environmental impact, and improved resource allocation.**

2. **CHALLENGES IN MEASURE ENERGY CONSUMPTION:**
    - **Traditional energy monitoring systems may lack granularity and fail to provide insights for optimizing consumption.**
    - **Energy usage patterns can be complex and dynamic, making it challenging to predict and control energy demand.**
3. **ROLE OF MACHINE LEARNING: Machine learning techniques, when applied to energy consumption data, offer several advantages:**
    - **Predictive Analytics: Machine learning models can predict future energy consumption patterns based on historical data, weather conditions, and other relevant factors.**
    - **Anomaly Detection: ML can identify abnormal energy consumption, potentially indicating equipment malfunction or inefficiencies.**
    - **Optimization: ML algorithms can suggest energy-saving strategies, such as adjusting HVAC systems, lighting, or manufacturing processes in real-time.**
    - **Personalized Recommendations: In residential settings, ML can provide users with tailored recommendations to reduce their energy consumption.**
4. **DATA COLLECTION: To apply machine learning to energy consumption, you need high-quality data. This may include historical energy usage data, weather data, occupancy information, and equipment data. Sensors, smart meters, and IoT devices play a crucial role in collecting real-time data.**
5. **TYPES OF MACHINE LEARNING MODEL: There are several ML models suited for energy consumption measurement:**
    - **Regression: For predicting future energy usage based on historical data.**
    - **Time Series Analysis: Essential for analyzing time-dependent energy consumption patterns.**
    - **Clustering: To group similar energy consumption profiles for targeted interventions.**
    - **Deep Learning: Neural networks can handle complex, high-dimensional data and learn intricate patterns.**

6. **DEPLOYMENT AND INTEGRATION:** **Once a machine learning model is trained, it needs to be deployed and integrated into the energy management system. This may involve real-time data processing, automation, and interaction with control systems.**

7. **BENEFITS:**

- **Improved energy efficiency and reduced operational costs.**
- **Reduced carbon footprint and environmental impact.**
- **Enhanced decision-making and resource allocation.**
- **Enhanced user experiences and cost savings in residential settings.**

8. **CHALLENGES:**

- **Data quality and availability are critical and can be a bottleneck.**
- **Model accuracy and interpretability are crucial for trust and decision-making.**
- **Security and privacy concerns when dealing with sensitive energy data.**

9.FUTURE DIRECTIONS: **The field of energy consumption measurement using machine learning is rapidly evolving. Future developments may involve more advanced algorithms, increased automation, and greater integration with smart grids and renewable energy sources.**

In conclusion, machine learning is a powerful tool for measuring and optimizing energy consumption. It can provide valuable insights, enhance efficiency, and contribute to a more sustainable and environmentally friendly energy landscape. However, successful implementation requires a combination of data, domain knowledge, and robust machine learning techniques.

DATASETLINK: **(**https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption**)**

GIVEN DATASET:

S.NO	DATE TIME	AEP MW
1	31-12-2004 01:00	13478
2	31-12-2004 02:00	12865
3	31-12-2004 03:00	12577
4	31-12-2004 04:00	12517
5	31-12-2004 05:00	12670
6	31-12-2004 06:00	13038
7	31-12-2004 07:00	13692
8	31-12-2004 08:00	14297
9	31-12-2004 09:00	14719
10	31-12-2004 10:00	14941
11	31-12-2004 11:00	15184
12	31-12-2004 12:00	15009

13	31-12-2004 12:00	15670

1.DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT:

DESIGN THINKING:

Empathize:

- **Understand the needs, concerns, and behaviors of energy consumers. Conduct surveys, interviews, and observe energy consumption patterns.**

Define:

- **Clearly define the problem statement and the goals of the project. Create user personas to represent different types of energy consumers.**

Ideate:

- **Brainstorm solution to address optimization. Consider technologies like IoT devices, machine learning algorithms, and smart meters.**

Prototype:

Develop a prototype of the energy optimization system. Test it in a controlled environment or with a small group of users.

PYTHON PROGRAMMING:

```
import random
import datetime


# Define the devices and their power ratings (in Watts)
devices = {
   "Device1": 1000,
   "Device2": 500,
   "Device3": 750,
}


# Simulation parameters
start_time = datetime.datetime(2023, 1, 1, 0, 0)
end_time = datetime.datetime(2023, 1, 2, 0, 0)
time_interval = datetime.timedelta(hours=1)
```

```
# Initialize energy consumption variables
total_energy_consumed = {device: 0 for device in devices}

# Simulate energy consumption
current_time = start_time
while current_time < end_time:
    for device, power_rating in devices.items():
        # Simulate power consumption for each device (in Watt-hours)
        energy_consumed = power_rating * random.uniform(0.8, 1.2) * time_interval.total_seconds() / 3600
        total_energy_consumed[device] += energy_consumed

    current_time += time_interval

# Print energy consumption results
for device, energy in total_energy_consumed.items():
    print(f"{device} - Total Energy Consumed: {energy:.2f} Watt-hours")

# Calculate and print the total energy consumption for all devices
total_consumption = sum(total_energy_consumed.values())
print(f"Total Energy Consumed for All Devices: {total_consumption:.2f} Watt-hours")
```

OUTPUT:

	site_id	timestamp	air_temperature	cloud_coverage	dew_temperature	precip_depth_1_hr	sea_level_pressure	wind_direction	w
0	0	2017-01-01 00:00:00	17.796875	4.0	11.703125	NaN	1021.5	100.0	3
1	0	2017-01-01 01:00:00	17.796875	2.0	12.796875	0.0	1022.0	130.0	3
2	0	2017-01-01 02:00:00	16.093750	0.0	12.796875	0.0	1022.0	140.0	3
3	0	2017-01-01 03:00:00	17.203125	0.0	13.296875	0.0	1022.0	140.0	3
4	0	2017-01-01 04:00:00	16.703125	2.0	13.296875	0.0	1022.5	130.0	2

2.DESING AND INNOVATION

1.Data Collection:

- Gather historical energy consumption data.
- This data should include information about the time and date of measurements, energy usage, and potentially other relevant factors like temperature, holidays, or special events.
- Dataset link is below here:

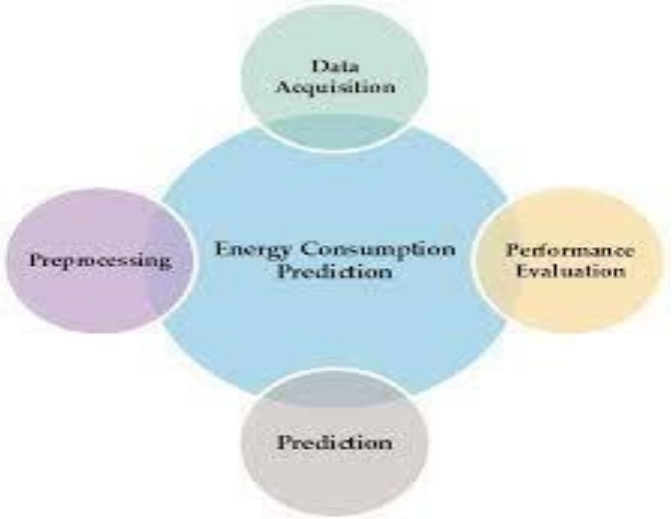
https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

2. Data Preprocessing:

- Data preprocessing is an important step before applying machine learning methods for energy or load prediction.
- It improves accuracy and reliability.
- There are four types of data processing

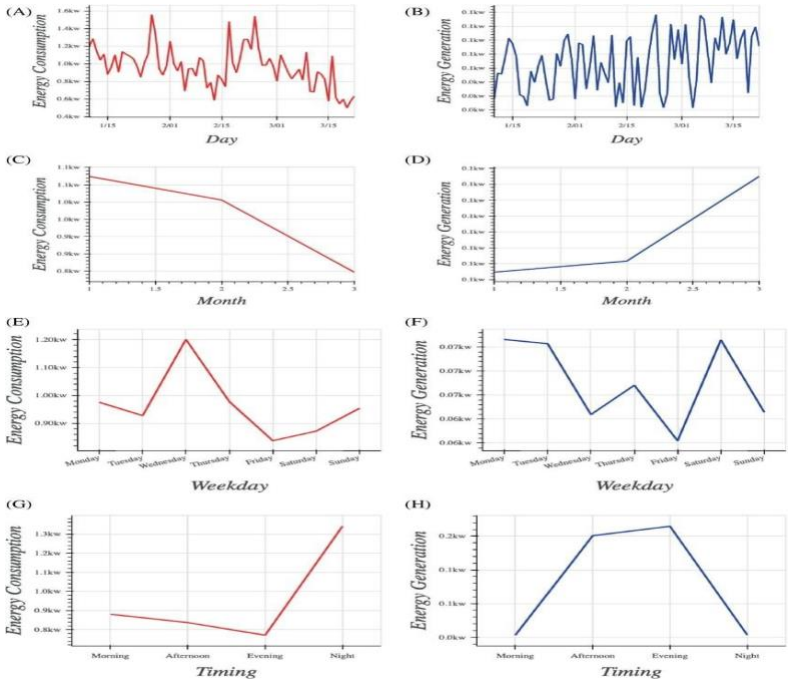
1. Data cleaning

2. **Data integration**
3. **Data transformation**
4. **Data reduction**



3.Exploratory Data Analysis (EDA):

- **Perform EDA to gain insights into the data.**
- **Learn everything you need to know about exploratory data analysis, a method used to analyze and summarize data sets.**
- **Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.**
- **Visualize trends, seasonality, and correlations between energy consumption and other variables.**



PROGRAM:

Source code:

Int[1]:

```
Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import matplotlib.dates as mdates

%matplotlib inline

Import seaborn as sns

Import warnings

Warnings.filterwarnings(“ignore”)

From pandas.plotting import lag_plot

From pylab import rcParams
```

From statsmodels.tsa.seasonal import seasonal_decompose

From pandas import DataFrame

From pandas import concat

Int[2]:

Df=pd.read_csv("../input/hourly-energy-consumptio n/AEP_hourly.csv",index_col='Datetime',parse_date s=True)
Df.head() Out[2]:

	AEP_MW
Datetime	
2004-12-31 01:00:00	13478.0
2004-12-31 02:00:00	12865.0
2004-12-31 03:00:00	12577.0
2004-12-31 04:00:00	12517.0
2004-12-31 05:00:00	12670.0

Int[3]:

df.sort_values(by='Datetime', inplace=True)

print(df) Int[4]:

df.shape Out[4]:

(121273, 1)

Int[5]: df.info()

Out[5]:

<class 'pandas.core.frame.DataFrame'>

DatetimeIndex: 121273 entries, 2004-10-01 01:00:00 to 2018-08-03 00:00:00

Data columns (total 1 columns):

Column Non-Null Count Dtype

--- ------ -------------- -----

0 AEP_MW 121273 non-null float64

dtypes: float64(1)

memory usage: 1.9 MB

Int[6]:

df.describe() Out[6]:

	AEP_MW
count	121273.000000
mean	15499.513717
std	2591.399065
min	9581.000000

25%	13630.000000
50%	15310.000000
75%	17200.000000
100%	25695.000000

Int[7]:

df.index = pd.to_datetime(df.index)

Int[8]:

Extract all Data Like Year MOnth Day Time etc

df["Month"] = df.index.month df["Year"] = df.index.year

df["Date"] = df.index.date df["Hour"] = df.index.hour

df["Week"] = df.index.week df["Day"] =

df.index.day_name()

df.head() Out[8]:

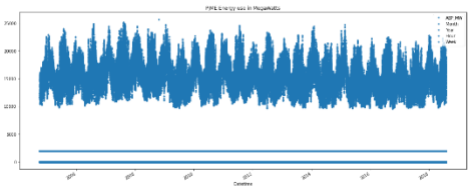
	AEP_MW	Month	Year	Date	Hour	Week	Day
Datetime							

2004-10-01 01:00:00	12379.0	10	2004	2004-10-01 1	1	4	Friday
2004-10-01 02:00:00	11935.0	10	2004	2004-10-01 1	2	4	Friday
2004-10-01 03:00:00	11692.0	10	2004	2004-10-01 1	3	4	Friday
2004-10-01 04:00:00	11597.0	10	2004	2004-10-01 1	4	4	Friday
2004-10-01 05:00:00	05:00:00	10	2004	2004-10-01 1	5	4	Friday

Int[9]:

```
df.plot(title="PJME Energy use in MegaWatts",
        figsize=(20, 8), style=".",
        color=sns.color_palette()[0])
```

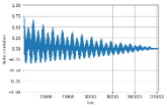
plt.show() Out[9]:



Int[10]:

from pandas.plotting import autocorrelation_plot

autocorrelation_plot(df['AEP_MW']) plt.show() Out[10]:



Int[11]:

#Train Arima Model train_arima =

train_data['AEP_MW'] test_arima =

test_data['AEP_MW']

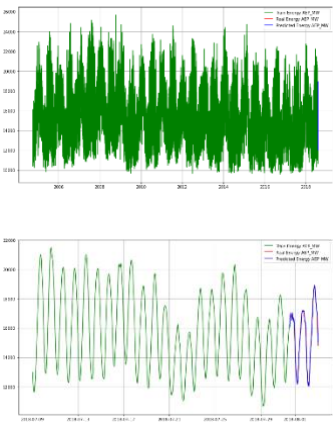
```
history = [x for x in train_arima] y =
test_arima # make first prediction
predictions = list()
model = sm.tsa.arima.ARIMA(history,
order=(5,1,0)) model_fit = model.fit() yhat =
model_fit.forecast()[0]
predictions.append(yhat) history.append(y[0])
# rolling forecasts for i in range(1, len(y)):
    # predict
    model = sm.tsa.arima.ARIMA(history,
order=(5,1,0))
    model_fit = model.fit() yhat =
    model_fit.forecast()[0] # invert
    transformed prediction
    predictions.append(yhat)
    # observation obs = y[i]
    history.append(obs)

plt.figure(figsize=(14,8))
plt.plot(df.index, df['AEP_MW'], color='green', label = 'Train
Energy AEP_MW')
```

```
plt.plot(test_data.index, y, color = 'red', label = 'Real
Energy AEP_MW')
plt.plot(test_data.index, predictions, color = 'blue', label =
'Predicted Energy AEP_MW')
plt.legend() plt.grid(True)
plt.show()

plt.figure(figsize=(14,8))
plt.plot(df.index[-600:], df['AEP_MW'].tail(600), color='green',
label = 'Train Energy AEP_MW') plt.plot(test_data.index, y,
color = 'red', label = 'Real
Energy AEP_MW')
plt.plot(test_data.index, predictions, color = 'blue', label =
'Predicted Energy AEP_MW')
plt.legend() plt.grid(True)
plt.show()

print('MSE: '+str(mean_squared_error(y, predictions)))
print('MAE: '+str(mean_absolute_error(y, predictions)))
print('RMSE: '+str(sqrt(mean_squared_error(y,
predictions)))) Out[11]:
```



4.Feature Engineering: **Create additional features that could impact energy consumption, such as holidays, weather data, day of the week, and time of day.**

- **This includes the use of electricity, gas, diesel, oil, and biomass. The concept of energy consumption is directly related to energy efficiency since higher consumption results in lower energy efficiency.**
- **TEE includes three core components**

1.Resting metabolic rate, or resting energy expendicture
2.The thermic effect of food (TEF)
3. Diet-induced thermogenesis DIT)

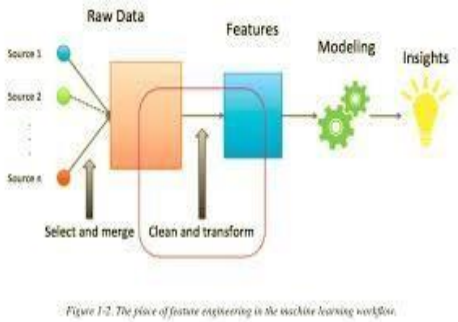


Figure 1-2. The place of feature engineering in the machine learning workflow.

5.Machine Learning Models: A)

Regression Models:

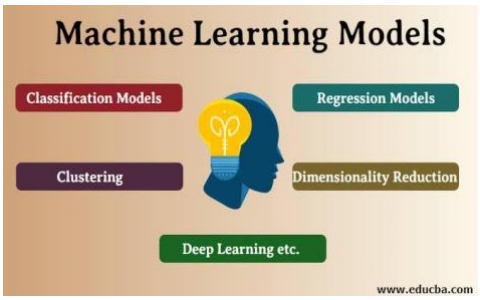
- **Utilize linear regression, decision trees, or random forests to build models that predict energy consumption based on relevant features.**

B) Time Series Forecasting:

- **Implement specialized time series forecasting models like ARIMA, Prophet, or LSTM (Long Short-Term Memory) neural networks.**

C)Ensemble Methods:

- **Combine multiple models to improve prediction accuracy.**



3.BUILD LOADING AND PREPROCESSING THE DATASET

1. DATA COLLECTION AND SOURCES:

Identify reliable data sources: **Ensure you have access to accurate and comprehensive energy consumption data from various sources like sensors, smart meters, utility companies, or government agencies.**

Data formats: **Understand the format of the data, whether it's structured (e.g., CSV, Excel) or unstructured (e.g., text logs), and be prepared to handle different formats.**

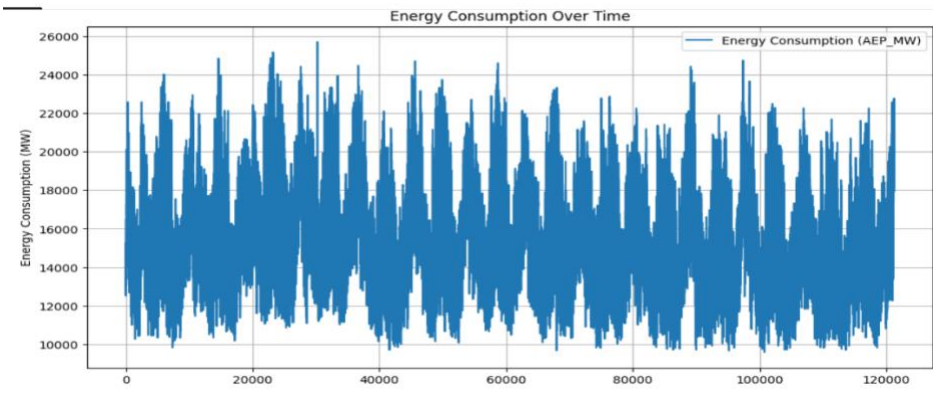
2. DATA CLEANING:

Handle missing data: **Energy data can often have missing values. Decide on a strategy to deal with missing data, whether it's through imputation, interpolation, or removal.**

Outlier detection: **Identify and handle outliers in the data that might be caused by measurement errors or anomalies.**

PROGRAM:

```
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)
```

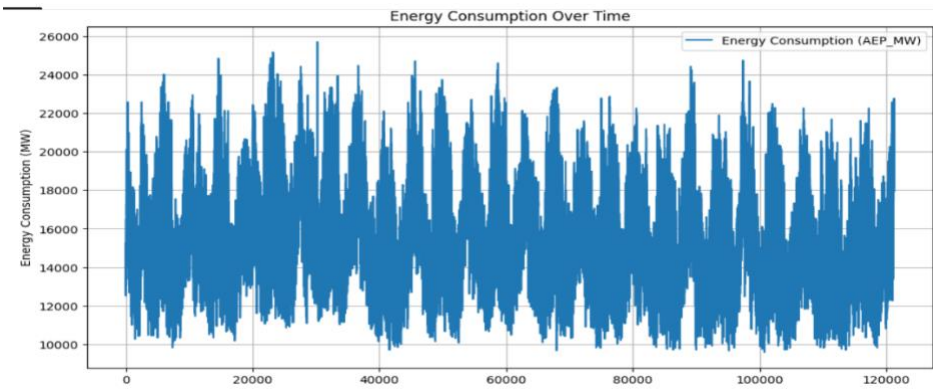


3. DATA ANALYSIS:

Calculate the cost of energy consumption by incorporating tariff rates, taxes, and any other associated costs. This helps in budgeting and cost optimization.

PROGRAM:

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



4.DATA PREPROSESSING:

First we import some basic python libraries like pandas numpy and matplotlib in our project and then we initialized our dataset in our project file by using the read csv command in the pandas as our project is in the csv file.

And then we started our processing of data in the project by dropping nan values and unnecessary columns in the data for the prediction . Thus this is the preprocessing that has been done in our project. let me attach the snipset of the data processing using pandas.

PROGRAM:

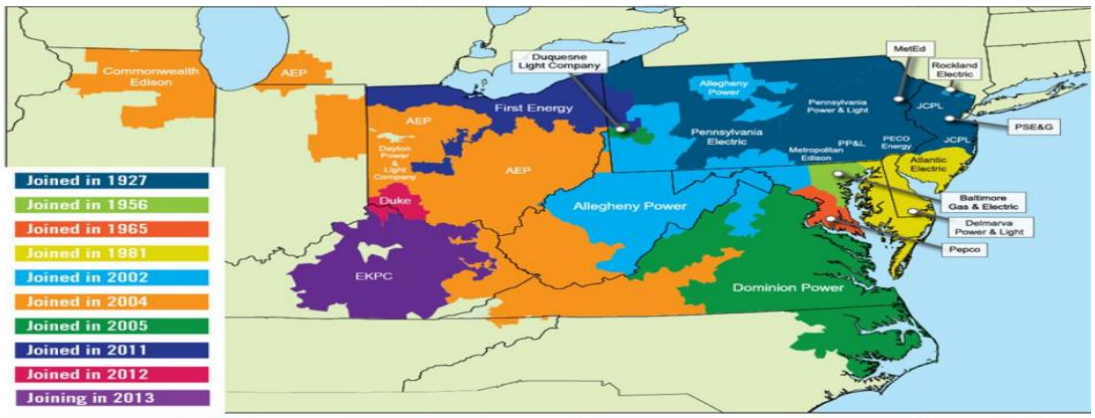
```
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
plt.style.use('ggplot') # Make it pretty
```

```
# Data is saved in parquet format so schema is preserved.
df = pd.read_parquet('../input/est_hourly.paruqet')
```

Data index is the date/hour, columns are for different regions within PJM.

Regions joined at different times, so not all have data for all dates. Regions also split (PJM_Load split to East and West)

```
#Show PJM Regions
from IPython.display import Image
Image(url= "http://slideplayer.com/4238181/14/images/4/PJM+Evolution.jpg")
```




```
df.head()
```

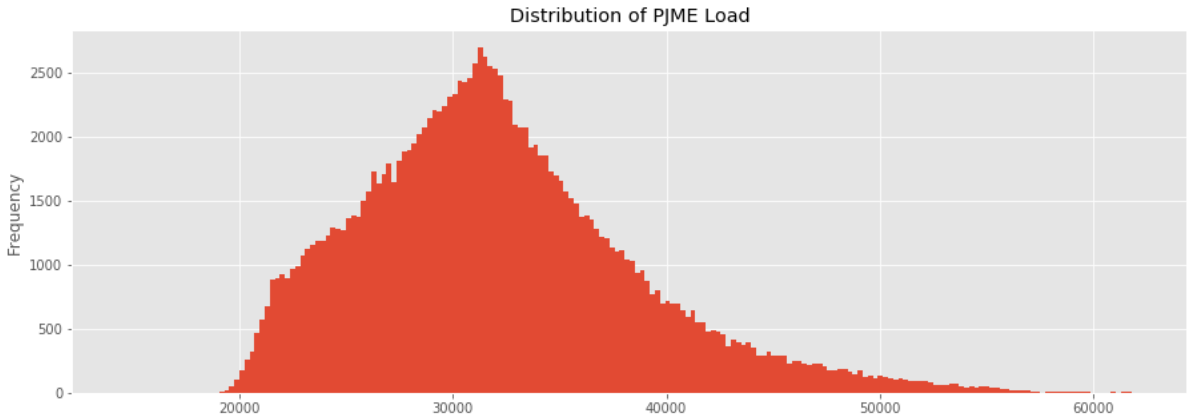
OUTPUT:

	AEP	COMED	DAYTON	DEOK	DOM	DUQ	EKPC	FE	NI	PJME	PJMW	PJM_Load
Datetime												
1998-12-31 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	29309.0
1998-12-31 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28236.0
1998-12-31 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27692.0
1998-12-31 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27596.0
1998-12-31 05:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27888.0

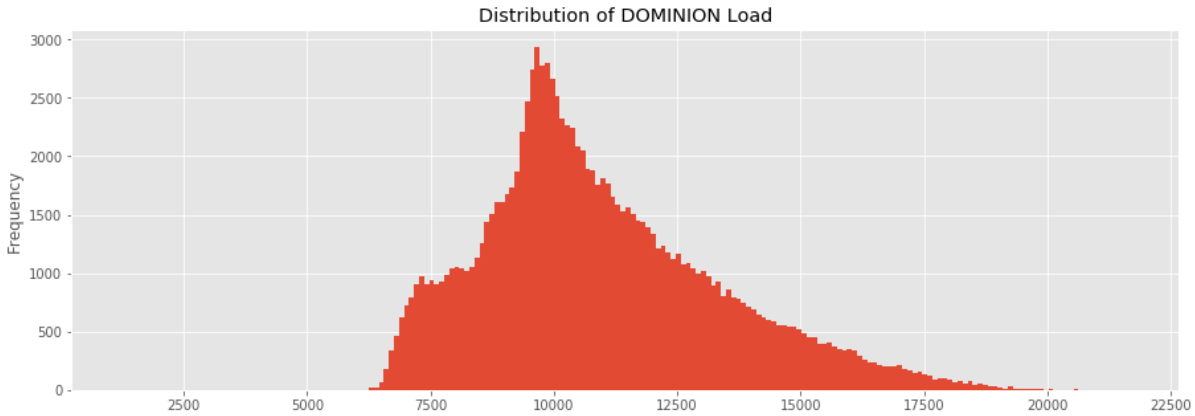
```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
AEP	121273.0	15499.513717	2591.399065	9581.0	13630.0	15310.0	17200.00	25695.0
COMED	66497.0	11420.152112	2304.139517	7237.0	9780.0	11152.0	12510.00	23753.0
DAYTON	121275.0	2037.851140	393.403153	982.0	1749.0	2009.0	2279.00	3746.0
DEOK	57739.0	3105.096486	599.859026	907.0	2687.0	3013.0	3449.00	5445.0
DOM	116189.0	10949.203625	2413.946569	1253.0	9322.0	10501.0	12378.00	21651.0
DUQ	119068.0	1658.820296	301.740640	1014.0	1444.0	1630.0	1819.00	3054.0
EKPC	45334.0	1464.218423	378.868404	514.0	1185.0	1386.0	1699.00	3490.0
FE	62874.0	7792.159064	1331.268006	0.0	6807.0	7700.0	8556.00	14032.0
NI	58450.0	11701.682943	2371.498701	7003.0	9954.0	11521.0	12896.75	23631.0
PJME	145366.0	32080.222831	6464.012166	14544.0	27573.0	31421.0	35650.00	62009.0
PJMW	143206.0	5602.375089	979.142872	487.0	4907.0	5530.0	6252.00	9594.0
PJM_Load	32896.0	29766.427408	5849.769954	17461.0	25473.0	29655.0	33073.25	54030.0

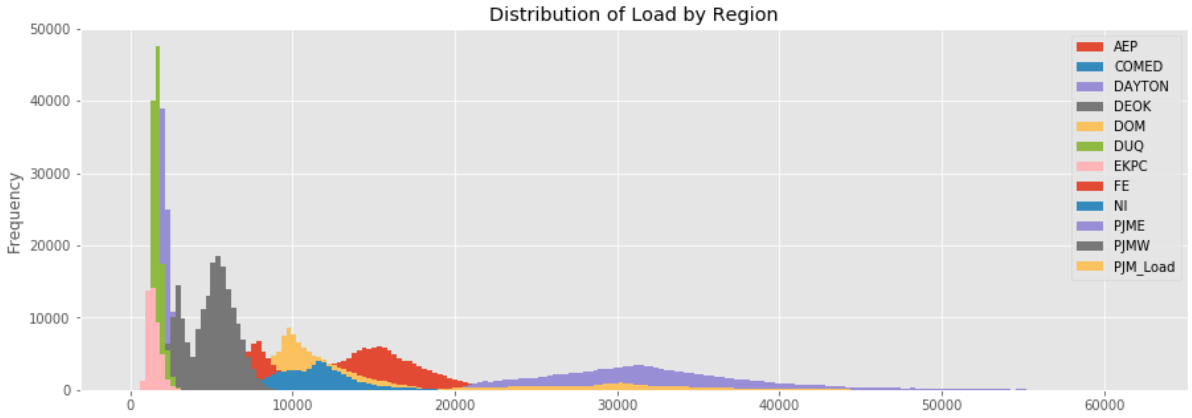
```
_ = df['PJME'].plot.hist(figsize=(15, 5), bins=200, title='Distribution of PJME Load')
```




```
_ = df['DOM'].plot.hist(figsize=(15, 5), bins=200, title='Distribution of DOMINION Load')
```




```
_ = df.plot.hist(figsize=(15, 5), bins=200, title='Distribution of Load by Region')
```



5.Feature Engineering:

Create new features to transform existing one to capture additional information that may measure energy consumption.

6.Data Encoding:

Convert categorical variable into numerical format using techniques like one hot encoding.

4.PERFORMING DIFFERENT ACTIVITIES LIKE FEATURE ENGINEERING, MODEL,TRAINING,EVALUATION etc..,

Necessary step to follow:

1. Import libraries: start by importing the necessary libraries.

Program:

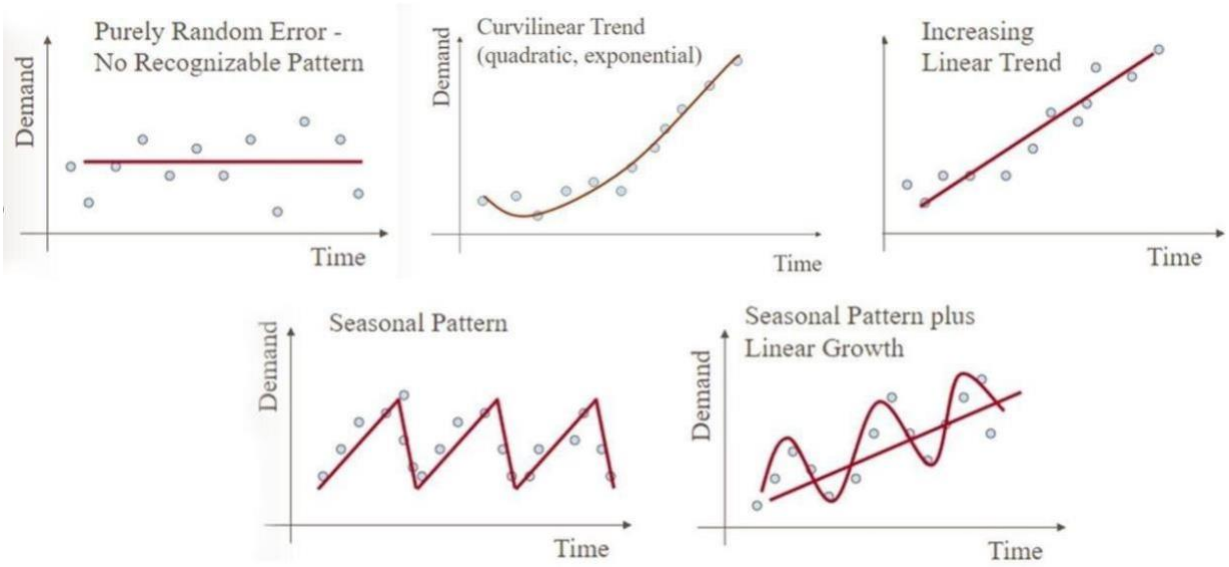
```
import numpy
import pandas as pd
import matplotlib.pyplot as plt
importseaborn as sns
```

import pandas as pd import xgboost as xgb

Time series data by using mean_square_red_error:

import xgboost as xgb

from sklearn.metrics import mean_squared_errorcolor_pal = sns.color_palette() plt.style.use('fivethirtyeight')



The figure shows that the different types of time series dataforecasting with machine learning.

Data Processing:

Data processing involves tasks like data cleaning, handling missing values, and encoding categorical variables to prepare the data foranalysis.

```
df = pd.read_csv('../input/hourly-energy-
consumption/PJME_hourly.csv')

df = df.set_index('Datetime') df.index =

pd.to_datetime(df.index)df.plot(style='.',

        figsize=(15, 5),

        color=color_pal[0],

       title='PJME Energy Use in MW')

plt.show()
```

Dataset for Date Time PJMW MW:

0	2002-12-31 01:00:00	5077.0
1	2002-12-31 02:00:00	4939.0
2	2002-12-31 03:00:00	4885.0
3	2002-12-31 04:00:00	4857.0
4	2002-12-31 05:00:00	4930.0

Training and testing of data split:

Splitting data into training and testing sets is crucial for measuring energy consumption using time series in machine learning. It helps evaluate the model's performance on unseen data.

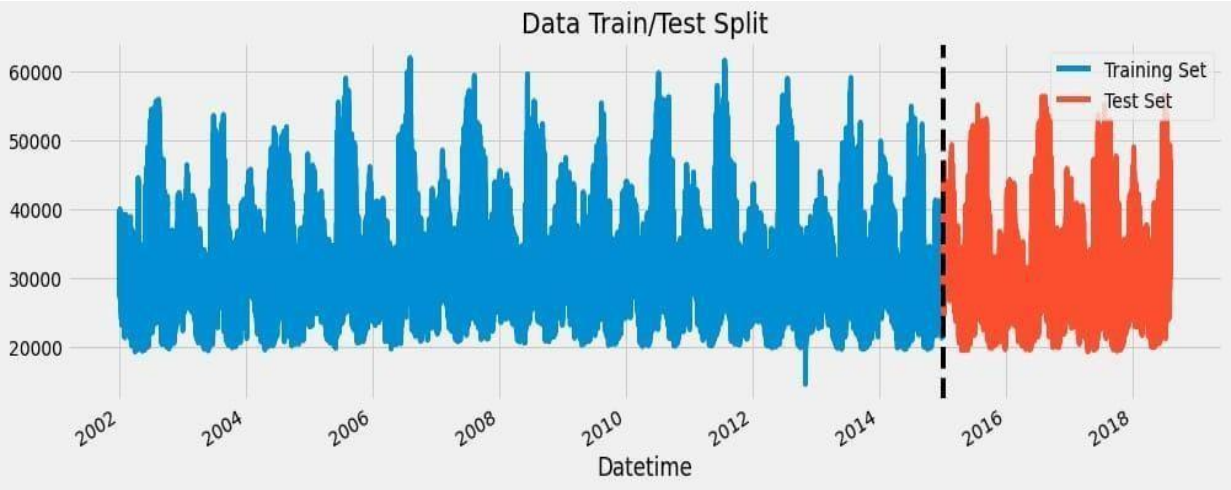
```
fig, ax = plt.subplots(figsize=(15, 5))

train.plot(ax=ax, label='Training Set', title='Data Train/TestSplit')

test.plot(ax=ax, label='Test Set') ax.axvline('01-01-

2015', color='black', ls='--')ax.legend(['Training Set',

'Test Set']) plt.show()
```



Feature creation:

It involves creating new features from the existing data to improve the accuracy of the energy consumption prediction.

In order to measure energy consumption and forecast timeseries data, plays a vital role of feature engineering.

```
def create_features(df):
df = df.copy()
    df['hour'] = df.index.hour df['dayofweek'] =
    df.index.dayofweekdf['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().weekreturn df

df = create_features(df)
      //creation of time series features based on timeseries index.
```

Visualization of feature and target relationship:

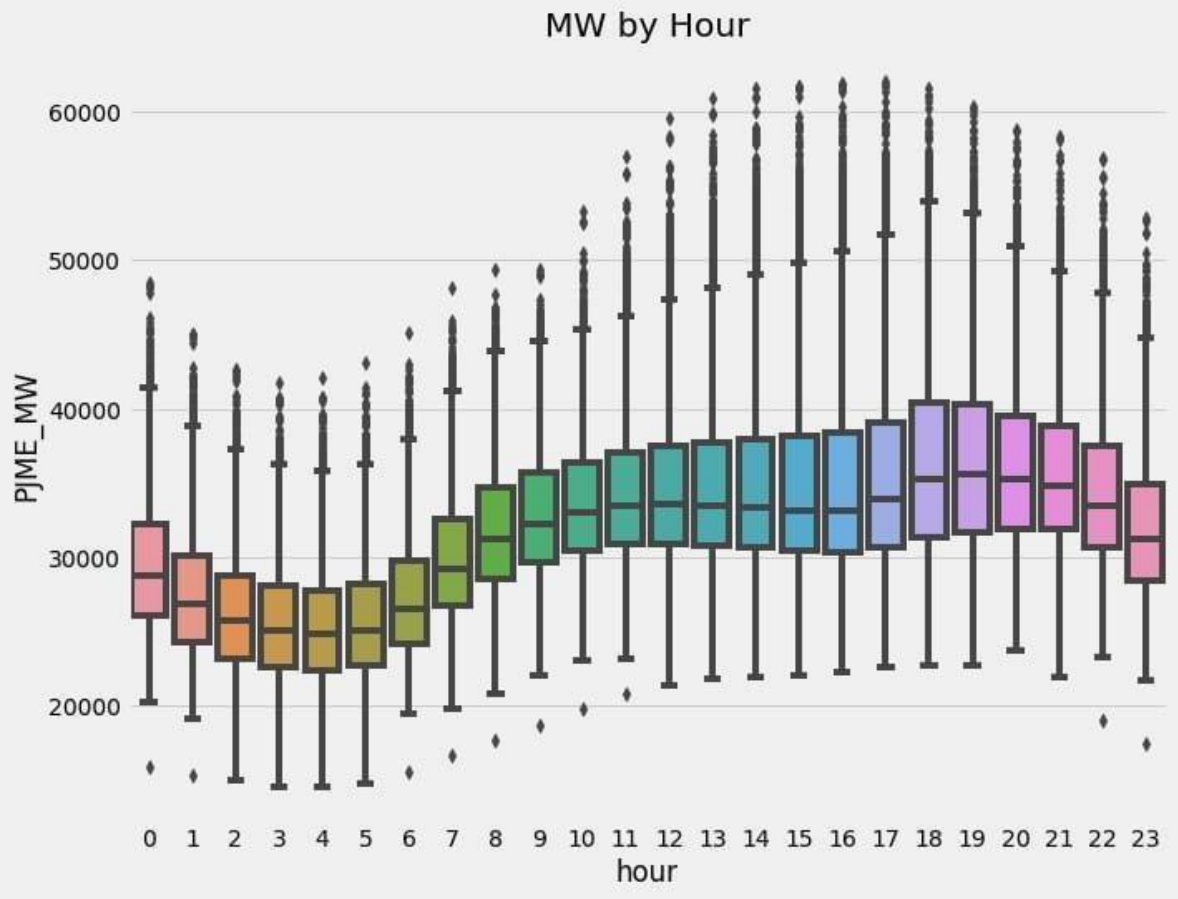
It helps to understand pattern and correlations.

It also allows to identify the features of impact in energy consumption.

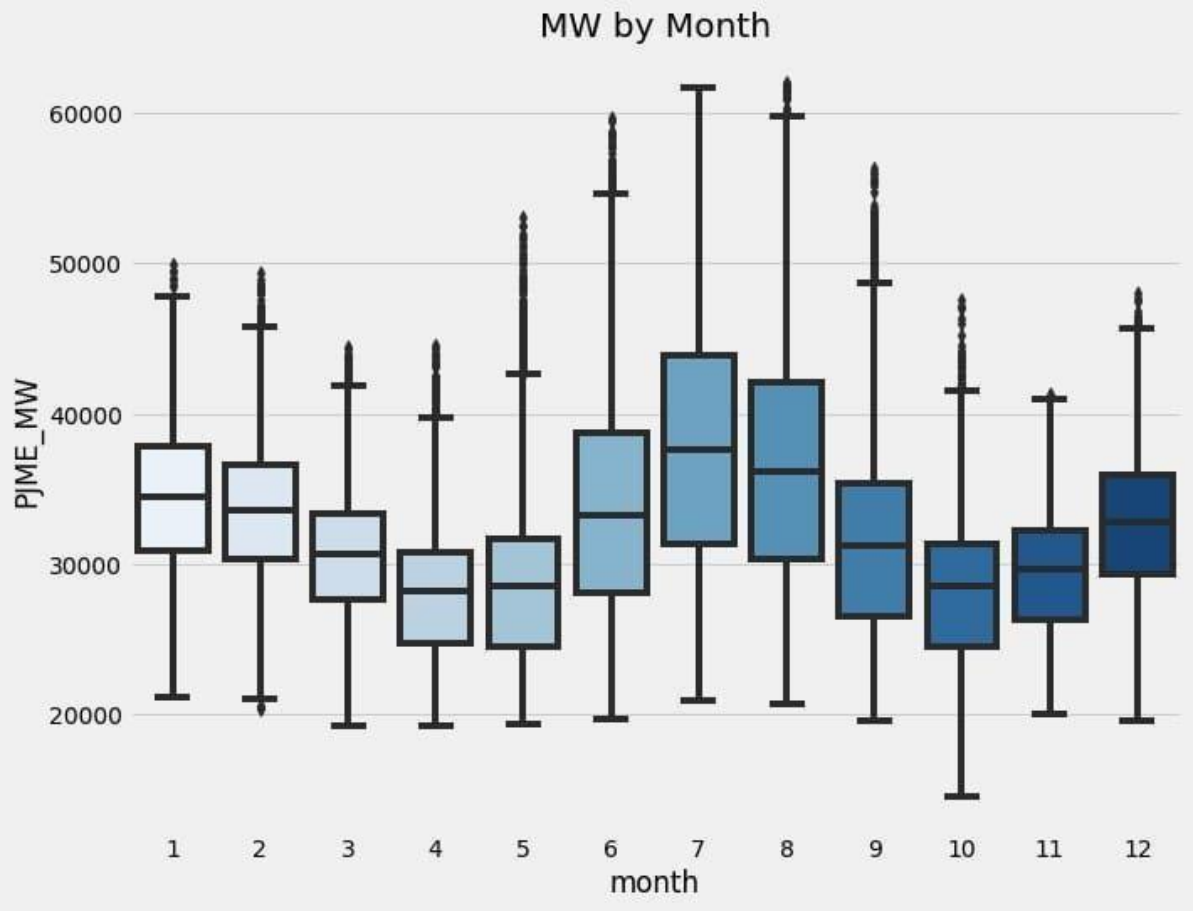
fig, ax = plt.subplots(figsize=(10, 8)) sns.boxplot(data=df,

x='hour', y='PJME_MW')ax.set_title('MW by Hour')

plt.show()



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



Creation of models and training:

Creation of models is choosing an appropriate machine learning algorithm that matches the problem at hand, and fine-tuning hyper parameters for optimal performance.

Training the selected model on the preprocessed data to learn patterns and relationships within the data.

```
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]
```

Program by using regression function:

```
X_test = test[FEATURES]y_test
= test[TARGET]
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)
```

Output:

```
XGBRegressor(base_score=0.5, booster='gbtree',
callbacks=None,
             colsample_bylevel=1, colsample_bynode=1,colsample_bytree=1,
             early_stopping_rounds=50, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1,grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.01, max_bin=256,
max_cat_to_onehot=4,
             max_delta_step=0, max_depth=3, max_leaves=0,min_child_weight=1,
             missing=nan, monotone_constraints='()',
n_estimators=1000,
             n_jobs=0, num_parallel_tree=1, objective='reg:linear',predictor='auto',
             random_state=0, reg_alpha=0, ...)
```

Feature importance of model:

Feature importance refers to determining the contribution of each feature in predicting energy consumption.

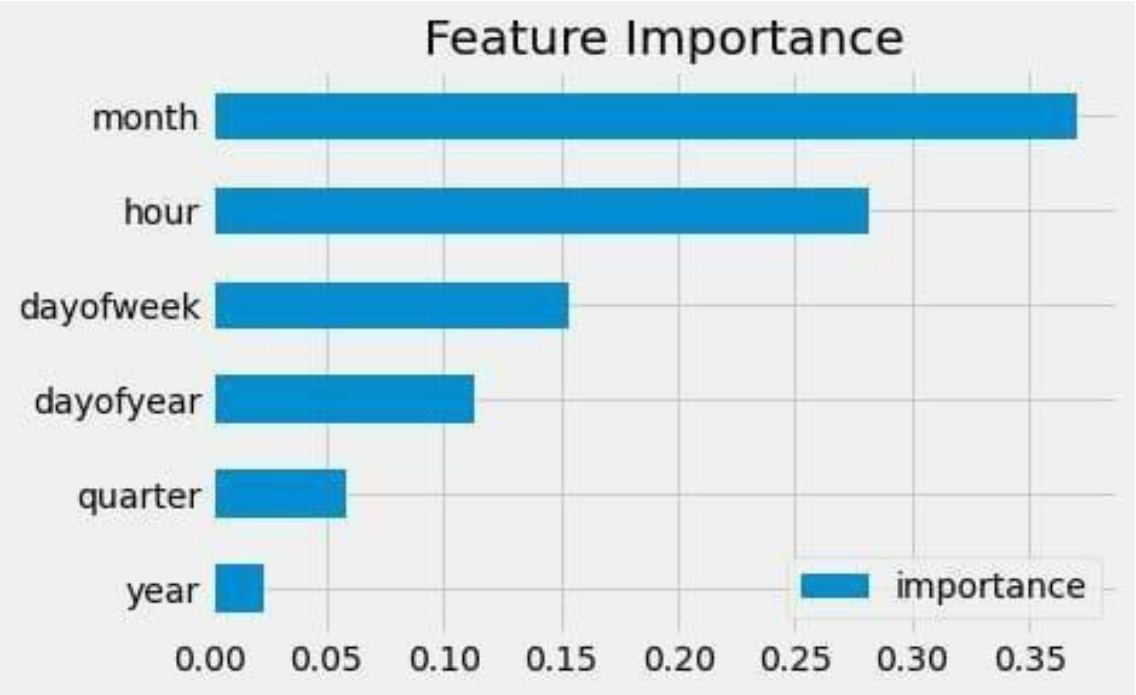
It helps identify which factors have the most influence on energy consumption patterns. This information can guide decision- making and help prioritize actions to optimize energy usage.

Various techniques, such as permutation importance or feature importance from tree-based models, can be used to calculate feature importance.

```
fi = pd.DataFrame(data=reg.feature_importances_,
             index=reg.feature_names_in_,
             columns=['importance'])
fi.sort_values('importance').plot(kind='barh', title='FeatureImportance')
```

```
plt.show()
```

```
fi = pd.DataFrame(data=reg.feature_importances_,
                  index=reg.feature_names_in_,
                  columns=['importance'])
fi.sort_values('importance').plot(kind='barh', title='FeatureImportance')
plt.show()
```



Forecasting on data and test prediction:

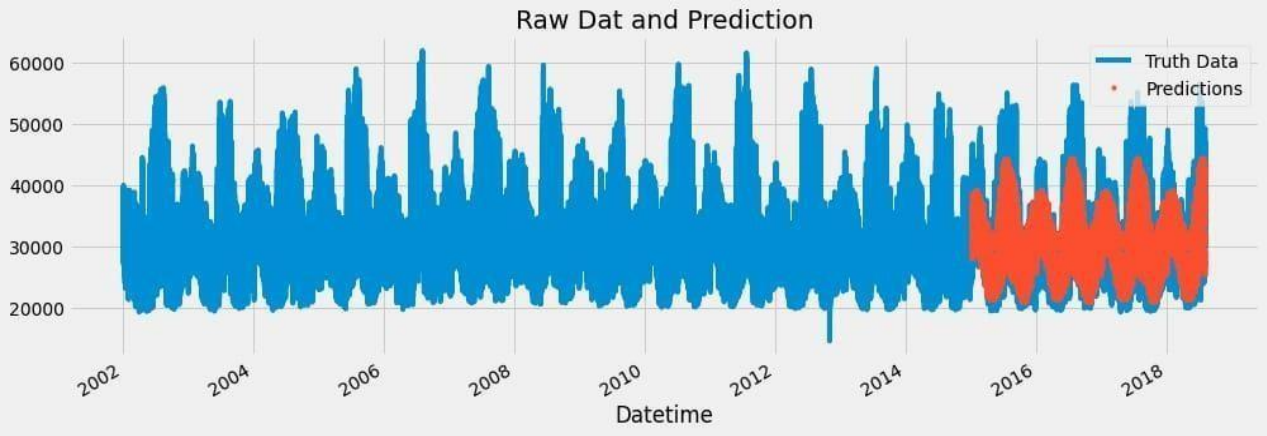
This allows you to evaluate the model's performance and accuracy in predicting energy consumption.

Forecasting a data is also used for model on historical data and then test its predicting energy consumption.

By comparing the model's predictions with the actual energy consumption values, you can assess how well the model is able toforecast future energy consumption patterns.

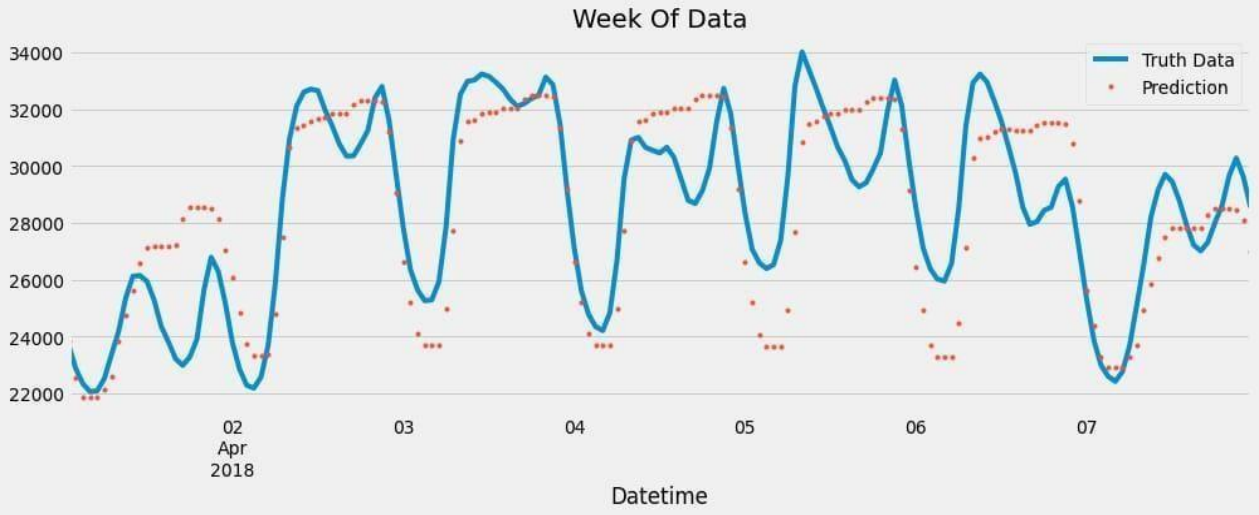
It's an effective way to validate and fine-tune the machine learning model for accurate energy consumption predictions.

```
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()
```




```
ax = df.loc[(df.index > '04-01-2018') & (df.index < '04-08-2018')]['PJME_MW'] \
    .plot(figsize=(15, 5), title='Week Of Data')
df.loc[(df.index > '04-01-2018') & (df.index < '04-08-2018')]['prediction'] \
    .plot(style='.')
plt.legend(['Truth Data','Prediction'])
```

plt.show()



Evaluate and Calculating Errors:

To evaluate and calculate the error in energy consumption prediction,

It performs a various metrics like absolute error (MAE), root mean squared error (RMSE), or mean absolute percentage error (MAPE).

These metrics help quantify the difference between the predicted energy consumption values and the actual values.

By calculating the error, it also assess the accuracy and performance of the prediction model.

```
test['error'] = np.abs(test[TARGET] - test['prediction'])

test['date'] = test.index.date

test.groupby(['date'])['error'].mean().sort_values(ascending=False).head(10)
```

Output:

date

2016-08-13	**12839.595459**
2016-08-14	**12780.209554**
2016-09-10	**11356.302002**
2015-02-20	**10965.976237**
2016-09-09	**10864.953451**
2018-01-06	**10506.844889**
2016-08-12	**10124.050618**
2015-02-21	**9881.798503**
2015-02-16	**9781.549805**
2018-01-07	**9739.143555**

Name: error, dtype: float64

ADVANTAGES:

1. **Improved Accuracy:** Machine learning models can provide highly accurate predictions and analyses of energy consumption patterns. They can take into account a wide range of variables and data sources to make more precise estimates, which is crucial for effective energy management.
2. **Real-time Monitoring:** Machine learning can enable real-time monitoring of energy consumption, allowing for immediate responses to anomalies, failures, or unusual patterns. This can help reduce energy waste and optimize energy usage.
3. **Predictive Analytics:** Machine learning models can predict future energy consumption patterns based on historical data and external factors such as

weather conditions, occupancy, and equipment usage. This helps in better planning and resource allocation.

4. **Anomaly Detection:** Machine learning can identify abnormal energy consumption patterns or unexpected deviations, which can be indicative of equipment malfunctions or energy theft. Early detection of anomalies can lead to faster response times and reduced downtime.
5. **Optimization:** Machine learning can optimize energy consumption by recommending changes in settings, schedules, or equipment operation to minimize energy waste. This can result in significant cost savings and a reduced environmental footprint.
6. **Demand Forecasting:** Machine learning can forecast energy demand, helping utility companies better allocate resources and reduce the need for expensive peak-load generation. This can stabilize the grid and reduce overall energy costs.
7. **Personalized Recommendations:** In residential and commercial settings, machine learning can provide personalized recommendations to users on how to reduce their energy consumption, fostering energy-efficient behaviors.
8. **Integration with IoT Devices:** Machine learning can work seamlessly with Internet of Things (IoT) devices and sensors to collect and analyze data from various sources, allowing for a comprehensive view of energy consumption.
9. **Scalability:** Machine learning solutions can be scaled to handle data from a single building to an entire city or region, making them suitable for a wide range of applications, from smart homes to smart grids.
10. **Cost Reduction**: By optimizing energy consumption and reducing waste, machine learning can lead to cost savings for both individuals and organizations. It can also help identify opportunities for energy efficiency improvements and upgrades.
11. **Environmental Impact:** Reduced energy consumption has a positive impact on the environment by lowering greenhouse gas emissions and reducing the overall carbon footprint.
12. **Data-Driven Decision-Making:** Machine learning enables data-driven decision-making in energy management, providing insights that can drive policy changes, infrastructure improvements, and sustainability initiatives.
13. **Adaptability:** Machine learning models can adapt to changing conditions and new data, making them suitable for dynamic environments where energy consumption patterns may evolve over time.

DISADVANTAGES:

1. **Data Requirements:**

- **Machine learning models require large amounts of data to train effectively. Gathering and managing high-quality energy consumption data can be expensive and time-consuming.**

2. **Data Quality:**
   - **Inaccurate or incomplete data can lead to unreliable predictions and inefficient energy management. Data cleaning and preprocessing are crucial but can be challenging, particularly for older infrastructure with legacy monitoring systems.**
3. **Model Complexity:**
   - **Machine learning models can be complex and difficult to interpret, making it challenging to understand the reasons behind energy consumption changes. This can hinder decision-making and troubleshooting efforts.**
4. **Overfitting:**
   - **Machine learning models may overfit the training data, resulting in models that do not generalize well to new or unseen data. This can lead to inaccurate predictions and decisions.**
5. **Hardware Costs:**
   - **Implementing machine learning for energy consumption monitoring may require investments in hardware, such as sensors, IoT devices, and advanced monitoring equipment, which can be expensive.**
6. **Expertise:**
   - **Developing and deploying machine learning models for energy consumption analysis requires specialized knowledge and skills. Organizations may need to hire or train data scientists and engineers to manage and maintain the system.**
7. **Model Maintenance:**
   - **Machine learning models require ongoing maintenance and updates to remain effective. Changes in the building or infrastructure, equipment upgrades, or shifts in operational patterns can lead to model degradation.**
8. **Privacy Concerns:**
   - **Collecting detailed energy consumption data can raise privacy concerns, especially in residential and commercial settings. Ensuring the security and privacy of this data is crucial and can be challenging.**
9. **Energy Consumption Variability:**

- **Energy consumption can be influenced by various external factors such as weather, occupancy, and economic conditions. Machine learning models may struggle to account for all these variables accurately.**

10.Scalability:

- **Implementing machine learning solutions for energy consumption analysis in a large-scale environment can be complex. Scaling up the infrastructure and models to handle a vast amount of data and devices can pose logistical challenges.**

11.Energy Consumption Behavior Changes:

- **Sometimes, occupants or users may adapt their behavior when they know energy consumption is being monitored, which can result in changes in consumption patterns and potentially undermine the accuracy of the model.**

12.Regulatory Compliance:

- **In some cases, regulations and standards may restrict the use of certain data collection methods or require specific reporting methods. Complying with these regulations while using machine learning can be complex.**

BENEFITS:

1. **Environmental Impact: a. Energy Efficiency: Monitoring energy consumption allows organizations to optimize their machine learning models and algorithms to be more energy-efficient, reducing their carbon footprint and energy bills. b. Sustainable AI: By understanding the energy costs associated with AI and machine learning applications, companies can make more environmentally conscious decisions and contribute to sustainability efforts.**
2. **Cost Savings: a. Reduced Operational Costs: Energy-efficient machine learning models can result in lower operational costs, especially in large-scale applications where energy usage is a significant expense. b. Improved Resource Allocation: Monitoring energy consumption helps organizations allocate resources more effectively, ensuring they are not overspending on computational resources.**
3. **Model Optimization: a. Performance-Per-Watt: Evaluating models based on energy consumption alongside accuracy metrics encourages the development of more efficient and practical solutions. b. Hyperparameter Tuning: Energy consumption measurements can guide hyperparameter tuning, helping to strike a balance between model performance and energy efficiency.**
4. **Hardware Selection: a. Hardware Efficiency: Understanding the energy usage of different hardware accelerators (e.g., GPUs, TPUs) helps in selecting the most**

energy-efficient solutions for machine learning tasks. b. Hardware Upgrades: Organizations can make informed decisions about hardware upgrades, considering energy efficiency as a key factor.

5. **Scalability:** a. Scalability Planning: Energy consumption data can inform decisions about scaling machine learning applications. It allows organizations to predict and manage the energy costs of scaling up infrastructure. b. Real-time Monitoring: Real-time monitoring of energy consumption can detect anomalies and inefficiencies in deployed machine learning systems, enabling prompt action to rectify issues.
6. **Compliance and Regulation**: a. Regulatory Compliance: In some regions, regulations may require organizations to report or limit energy consumption. Measuring and optimizing energy usage helps companies comply with such regulations. b. Sustainability Reporting: Businesses focused on sustainability reporting can include energy efficiency and conservation efforts in their reports.
7. **Model Lifespan:** a. Prolonged Hardware Lifespan: Reducing energy consumption can extend the lifespan of hardware, reducing the frequency of replacements and e-waste generation.
8. **Reputation and Marketing**: a. Green Initiatives: Demonstrating a commitment to energy-efficient machine learning practices can enhance a company's reputation, attract environmentally conscious customers, and contribute to positive branding.
9. **Research and Innovation**: a. Research Focus: Energy consumption measurement fosters research into more efficient machine learning algorithms, hardware, and infrastructure, driving innovation in the field. b. Cross-Domain Insights: Energy data can lead to cross-domain insights and collaborations, benefiting both machine learning and energy management sectors.

CONCLUSION:

In conclusion, using machine learning to measure energy consumption has the potential to revolutionize the way we manage and optimize our energy resources. Through the analysis of data, machine learning models can provide valuable insights and predictions that can lead to more efficient and sustainable energy usage. Here are some key points to consider:

1. **Improved Accuracy:** Machine learning models can provide a higher level of accuracy in predicting and measuring energy consumption compared to traditional methods. This can lead to better decision-making and resource allocation.
2. **Real-time Monitoring:** Machine learning can enable real-time monitoring of energy usage, allowing for quick response to fluctuations and abnormalities. This is

especially important in critical applications, such as in industrial settings or during peak demand periods.

3. **Pattern Recognition:** Machine learning can identify usage patterns and anomalies in energy consumption, helping to uncover hidden inefficiencies or areas for improvement.
4. **Energy Efficiency:** By understanding how energy is used, machine learning can help organizations and individuals optimize their consumption, leading to cost savings and reduced environmental impact.
5. **Predictive Maintenance:** Machine learning can be used to predict when equipment or systems are likely to fail, helping to prevent unexpected downtime and reduce energy waste associated with inefficient or malfunctioning equipment.
6. **Environmental Impact:** Utilizing machine learning to measure and manage energy consumption can contribute to a reduction in greenhouse gas emissions and a more sustainable future.
7. **Data-Driven Decision Making:** Machine learning enables data-driven decision-making, allowing organizations to fine-tune their energy strategies based on actual consumption patterns and trends.