MEASURE ENERGY CONSUMPTION

Phase5-Submission

Team Member:au952721104021-SHAJITHA BARVEEN.A

Project title:Measure Energy Consumption

Topic:In this section you will document the complete project and prepare it for submission

INTRODUCTION:

Energy consumption is the amount of energy used by an individual, household, business, or sector over a period of time. It is typically measured in kilowatt-hours (kWh), which is the equivalent of one kilowatt of power used for one hour.

There are a number of ways to measure energy consumption. The most common method is to use a smart meter, which is a type of energy meter that can record energy usage in real time. Smart meters are often installed by utility companies, but they can also be purchased and installed independently.Another way to measure energy consumption is to use a portable energy meter. Portable energy meters can be plugged into individual appliances or outlets to measure their energy usage over a period of time.Once energy consumption has been measured, it can be analyzed to identify areas where energy is being wasted.

For example, if a household is using a lot of energy at night, it may be a sign that they are leaving lights or appliances on unnecessarily.

Measuring energy consumption is an important first step towards reducing energy use and saving money. By tracking their energy usage, individuals and organizations can identify areas where energy is being wasted and take steps to make changes.

1.Data Collection and Preprocessing:

Preprocessing a dataset for measuring energy consumption typically involves cleaning and transforming the data to make it suitable for analysis.

Collect the energy consumption data from your source, whether it's from utility companies, sensors, or other sources.Identify the source of your energy consumption data. This could be from sensors, utility companies, smart meters, or other sources.

2.Data Analysis:

Data analysis for measuring energy consumption can be used to identify patterns and trends in energy usage, as well as to identify areas where energy is being wasted.

Descriptive statistics: This involves calculating basic statistics such as the mean, median, mode, and standard deviation of the energy consumption data. This can help to identify the average energy consumption, as well as the variability in energy consumption over time.

Time series analysis: This involves analyzing the energy consumption data over time to identify patterns and trends. This can be used to predict future energy consumption and to identify seasonal variations in energy usage.

Benchmarking: This involves comparing the energy consumption of a building or system to similar buildings or systems. This can help to identify areas where energy consumption is high and where there is potential for improvement.

Pareto analysis: This involves identifying the 20% of factors that account for 80% of the energy consumption. This can help to identify the most important areas to focus on for energy reduction efforts.

3.Feature Engineering:

Feature engineering for measuring energy consumption is the process of creating new features from existing data that can be used to improve the

accuracy of energy consumption predictions. This can be done by combining features, transforming features, or creating new features from scratch.

Creating features that represent the time of day and day of the week. This can be useful for capturing seasonal variations in energy consumption.

Creating features that represent the weather conditions. This can be useful for capturing the impact of weather on energy consumption.

Creating features that represent the occupancy of a building or system. This can be useful for capturing the impact of occupancy on energy consumption.

4.Machine Learning Model:

Machine learning models can be used to measure energy consumption by predicting the energy consumption of a device, system, or process based on historical data. This can be done by training a machine learning model on a dataset of historical energy consumption data and other relevant features, such as the time of day, weather conditions, and occupancy.

Once the machine learning model is trained, it can be used to predict the energy consumption of a device, system, or process under different operating conditions. This information can be used to develop strategies to reduce energy consumption and improve energy efficiency.

5.Deep Learning:

Deep learning is a type of machine learning that uses artificial neural networks to learn complex patterns in data. Deep learning algorithms have been shown to be very effective for measuring energy consumption.

One of the main advantages of deep learning for measuring energy consumption is that deep learning algorithms can learn complex relationships between different factors that affect energy consumption. For example, a deep learning algorithm can learn the relationship between the time of day, weather conditions, occupancy, and energy consumption. This allows deep learning algorithms to make more accurate predictions of energy consumption than traditional machine learning algorithms.

6.Visualization:

Visualization is a powerful tool for measuring energy consumption. By visualizing energy consumption data, it is possible to identify trends and patterns that may not be obvious from the raw data. This information can then be used to develop strategies to reduce energy consumption and improve energy efficiency.

7.Simulation and Modeling:

Simulation and modeling can be used to develop accurate and detailed models of energy consumption. By using these models, it is possible to identify trends and patterns in energy consumption.

Dataset:

Dataset	Location	Duration	#Houses	#Sensors (per house)	Features	Resolution
AMPds [7]	Greater Vancouver	1 year	1	19	I, V, pf, F, P, Q, S	1 min
BLUED [5]	Pittsburg, PA	8 days	1	0 (manual switch event labels available)	I, V	12 Khz
GREEND	Austria, Italy	1 year (3-6 months completed)	9	9 (appliance level)	P	1 Hz
HES	UK	1 month (255 houses) - 1 year (26 houses)	251	13-51	P	2 min
iAWE [8]	India	73 days	1	33 sensors (10 appliance level)	V, I, f, P, S, E, Φ	1 Hz
IHEPCDS[1]	France	4 years	1	3 circuits	I, V, P, Q	1 min
OCTES[2]	Finland, Iceland, Scotland	4-13 months	33	Aggregated	Aggregated, Energy price	7 secs
REDD [4]	Boston, MA	3 - 19 days	6	9-24	Aggregate: V, P; Submetered: P	15 Khz (aggr.), 3 sec (sub)
Sample dataset[3]	Austin, TX	7 days	10	12	S	1 min
Smart* [9]	Western Massachussets	3 months	1 Sub-metered +2 (Aggregated + Submetered)	25 circuits, 29 appliance monitors	P, S (circuits), P (submetered)	1 Hz
Tracebase [6]	Germany	N/A	Lab env.	158 devices (43 types)	P	1-10 sec
UK-DALE [10]	UK	499 days	4	5 (house 3) - 53 (house 1)	Aggregated P, Sub P, switch-status	16 Khz (aggr.), 6 sec (sub.)

[1] http://tinyurl.com/IHEPCDS
[2] http://octes.oamk.fi/final/
[3] http://www.pecanstreet.org/projects/consortium/

DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT

Design thinking is a human-centered approach to innovation that emphasizes understanding the needs of the user, re-framing problems in new ways, and rapidly prototyping and testing solutions. It can be used to address a wide range of challenges, including measuring energy consumption.

1. Empathize: Understand the needs and challenges of the users who need to measure their energy consumption. This could involve conducting interviews, surveys, or focus groups.

2. Define: Define the problem that needs to be solved in a way that is user-centered and actionable. For example, the problem could be defined as "How can we make it easier for homeowners to track their energy consumption?"
3. Ideate: Generate a wide range of creative solutions to the defined problem. This could involve brainstorming, sketching, or building prototypes.
4. Prototype: Build and test low-fidelity prototypes of the most promising solutions. This could involve developing simple tools or experiments to test the feasibility and usability of the solutions.
5. Test: Test the high-fidelity prototypes with users to get feedback and make necessary improvements. This could involve conducting user testing sessions or field trials.

Once the design thinking process is complete, the results can be used to develop and implement effective solutions for measuring energy consumption.

Here are some examples of how design thinking has been used to measure energy consumption:

- Smart meters: Smart meters use design thinking principles to make it easier for homeowners to track their energy consumption. Smart meters provide real-time energy consumption data that can be visualized and analyzed using user-friendly tools.
- Building energy management systems (BEMS): BEMS use design thinking principles to help building managers reduce energy consumption. BEMS collect and analyze energy consumption data for buildings and provide insights and recommendations to building managers.
- Smart grid energy consumption measurement: Smart grids use design thinking principles to develop new and innovative ways to measure energy consumption. Smart grid energy consumption measurement systems use sensors and communication technologies to collect and analyze energy consumption data across a network of power lines.

DESIGN FOR INNOVATION

Design for innovation in measuring energy consumption is the process of creating new and innovative solutions to measure energy consumption more accurately, efficiently, and cost-effectively. This can involve developing new technologies, new methods, or new business models.

some tips for designing for innovation in measuring energy consumption:

- Start with a deep understanding of the needs of the users. What are their biggest challenges in measuring energy consumption? What information do they need to make informed decisions about their energy use?
- Be creative and open to new ideas. Don't be afraid to challenge the status quo and think outside the box.
- Use a prototype-test-iterate approach. Develop and test prototypes of your solutions early and often to get feedback from users and make necessary improvements.
- Collaborate with others. Innovation is often the result of collaboration between different people and organizations. Seek out partners who can help you to develop and implement your ideas.

1.Data collection and preprocessing:

Collect the energy consumption data from your source, whether it's from utility companies, sensors, or other sources.Identify the source of your energy consumption data. This could be from sensors, utility companies, smart meters, or other sources

- Cleaning: This involves removing errors and inconsistencies from the data.
- Imputation: This involves filling in missing values in the data.
- Normalization: This involves scaling the data to a common range.
- Feature engineering: This involves creating new features from the existing data that may be more predictive of energy consumption.

Data analysis and Data visualization:

Data analysis and visualization are essential tools for measuring energy consumption. By analyzing and visualizing energy consumption data, it is possible to identify trends and patterns, evaluate the effectiveness of energy conservation measures, and make informed decisions about energy use.

Data visualization for measuring energy consumption typically involves using charts and graphs to display the data in a way that is easy to understand and interpret.

Data visualization is a powerful technique for representing data graphically to help people understand and make sense of complex information. It involves the use of visual elements such as charts, graphs, and maps to convey data

patterns, trends, and insights. Here's an explanation of the theory behind data visualization:

Visual Encoding:

Visual encoding is the process of mapping data attributes (e.g., values, categories) to visual properties (e.g., position, size, color, shape). For example, in a bar chart, the height of the bars represents data values, and the x-axis or y-axis position represents categories or time.

Data Types and Visualizations:

Nominal Data: Nominal data represents categories or labels with no inherent order, such as colors, names, or types. Nominal data can be visualized using bar charts, pie charts, or color-coded maps.

Ordinal Data:

Ordinal data has categories with a specific order or ranking, like customer satisfaction ratings (e.g., "poor," "average," "excellent"). Ordinal data can be represented using bar charts or dot plots.

Interval Data:

Interval data has a consistent measurement scale with equal intervals between values but no true zero point (e.g., temperature in Celsius). Line charts and histograms are often used for interval data.

Ratio Data:

Ratio data has a meaningful zero point (e.g., age, income, height). Scatter plots, histograms, and bar charts are common choices for visualizing ratio data.

Types of Visualizations:

Bar Charts:

Bar charts represent data using horizontal or vertical bars, with the length or height of each bar corresponding to the data value. They are suitable for comparing values across categories.

Line Charts:

Line charts connect data points with lines, making them ideal for showing trends and changes over time.

Pie Charts:

Pie charts display data as a circle divided into slices, where each slice represents a portion of the whole. They are useful for showing parts of a whole.

Scatter Plots:

Scatter plots display individual data points as dots on a two-dimensional plane. They are used to visualize the relationships between two variables.

Histograms:

Histograms group data into intervals (bins) and display the frequency or density of data points within each bin. They are used to visualize data distribution.

Box Plots:

Box plots show the distribution of data using quartiles, providing insights into the spread and skewness of the data.

Model Development & Evaluation:

Data Acquisition Module:

Implement code to interface with energy meters or sensors to collect real-time energy consumption data. Utilize libraries like pyserial or GPIO (for Raspberry Pi) to establish communication with hardware devices.

Data Processing Module:

Develop algorithms to process raw energy data, including parsing, cleaning, and converting data into usable formats (kWh, Joules, etc.). Apply data filtering techniques to remove noise and anomalies from the acquired data.

Data Storage Module:

Choose a suitable database system (e.g., SQLite, MySQL) to store processed energy consumption data. Implement code to establish a connection with the database and store data securely.

Visualization Module:

Use data visualization libraries such as Matplotlib or Plotly to create interactive charts and graphs representing energy consumption patterns. Display real-time data on a graphical user interface (GUI) for easy interpretation.

User Interface Module:

Design a user-friendly interface using GUI libraries like Tkinter or PyQt. Allow users to view historical data, set energy usage thresholds, and receive notifications when consumption exceeds specified limits.

Energy Analytics Module:

Develop algorithms to analyze energy usage patterns over time. Implement machine learning models for predicting future energy consumption based on historical data (optional).

EVALUATION OF THE PROJECT:

Accuracy:

Evaluate the accuracy of energy consumption measurements by comparing the data collected by your system with a known standard or reference data.

Performance:

Measure the performance of your system in terms of data processing speed, response time for user queries, and real-time data visualization

User Experience:

Gather feedback from users regarding the usability and intuitiveness of the interface. Evaluate whether users find it easy to set thresholds and receive notifications.

Reliability:

Test the reliability of the notification system by simulating various scenarios, including network issues and unexpected system failures.

Scalability:

Evaluate how well the system handles a large volume of data and users. Assess whether the system performance degrades under heavy loads**.**

Security:

Ensure that the data storage and communication channels are secure. Evaluate the system against common security threats and vulnerabilities.

Documentation:

Evaluate the completeness and clarity of project documentation, including user manuals, code comments, and technical guides.

Algorithm Name: Linear Regression for Energy Consumption Prediction

Explanation:

Loading Data:

In this example, the pjm_hourly_est.csv file dataset is loaded. You can replace it with your specific dataset by changing the features and target variables according to your dataset's column names.

Data Preprocessing:

The dataset is split into features (X) and the target variable (y). Then, it's further split into training and testing sets (80% training, 20% testing) using train_test_split.

Model Training:

A Linear Regression model is initialized and trained using the training data (X_train and y_train).

Prediction:

The trained model predicts energy consumption based on the test features (X_test).

Model Evaluation:

Mean Squared Error (MSE) and R-squared value are calculated to evaluate the model's performance. MSE measures the average squared difference between predicted and actual values. R-squared indicates the proportion of the variance in the target variable that is predictable from the features

Program:

In [1]:

```
# import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# customize the style
pd.options.display.float_format = '{:.5f}'.format
pd.options.display.max_rows = 12

# load the data
filepath = '../input/hourly-energy-consumption/PJME_hourly.csv'
df = pd.read_csv(filepath)

print("Now, you're ready for step one")
```

```
Now, you're ready for step one
Step 1: Explore the data
```

To better understand the data, I need to create a graph to see the change in PJM Energy over time.

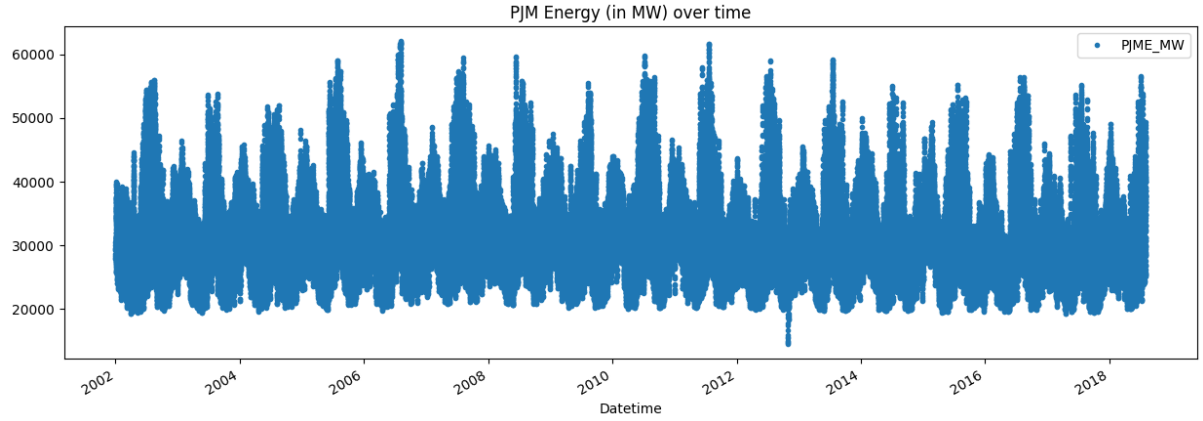
In [2]:

```
# turn data to datetime
df = df.set_index('Datetime')
df.index = pd.to_datetime(df.index)
```

In [3]:

```
# create the plot
```

```
df.plot(style='.',
        figsize=(15, 5),
        title='PJM Energy (in MW) over time')
plt.show()
```



Step 2: Split the data

Everything prior to January 2015 will be our training data and keep our test data as the following dates.

In [4]:

```
# train / test split
train = df.loc[df.index < '01-01-2015']
test = df.loc[df.index >= '01-01-2015']
```

unfold_moreShow hidden code

Step 3: Feature Engineering

We're going to create some time features using the Datetime index. After that, we'll explore the distributions of Hourly and Monthly megawatt usage.

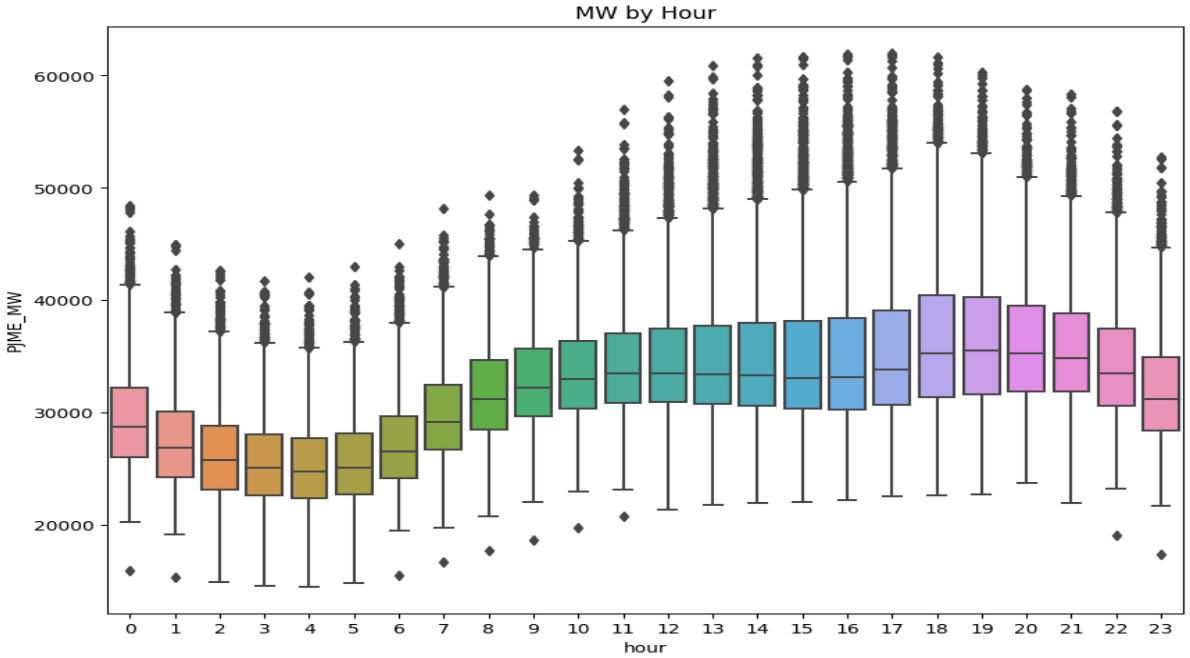
In [6]:

```
# feature creation
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)
```

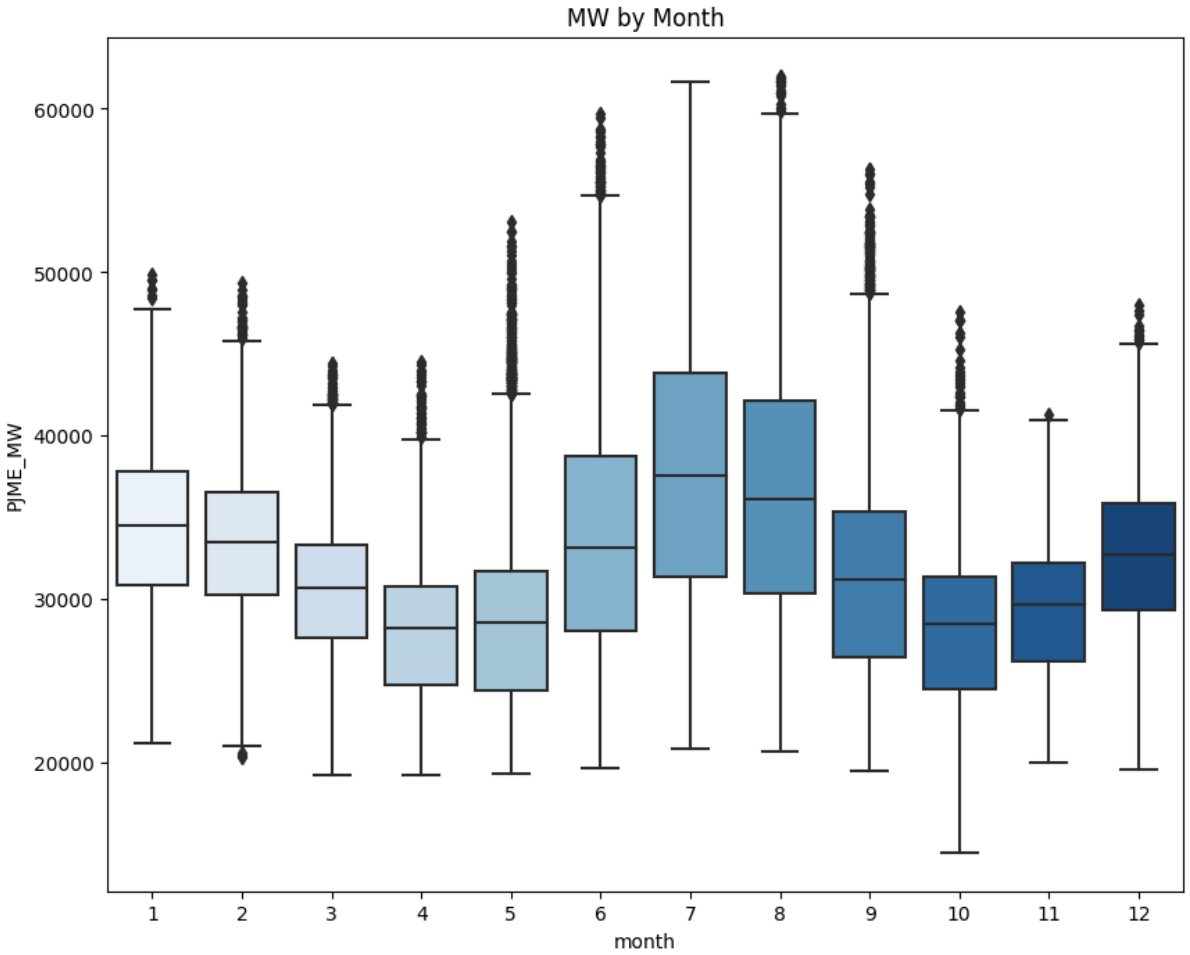
In [7]:

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



In [8]:

```
# viaualize the monthly Megawatt
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()
```



Step 4: Modelling

XGBoost is good and reliable model for regression and time series analysis as well. Also, for the metrics, we'll use mean squared error.

4.1 Prepare the data

In [9]:

```
# 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]
```

4.2 Build the model

In [10]:

```
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)
```

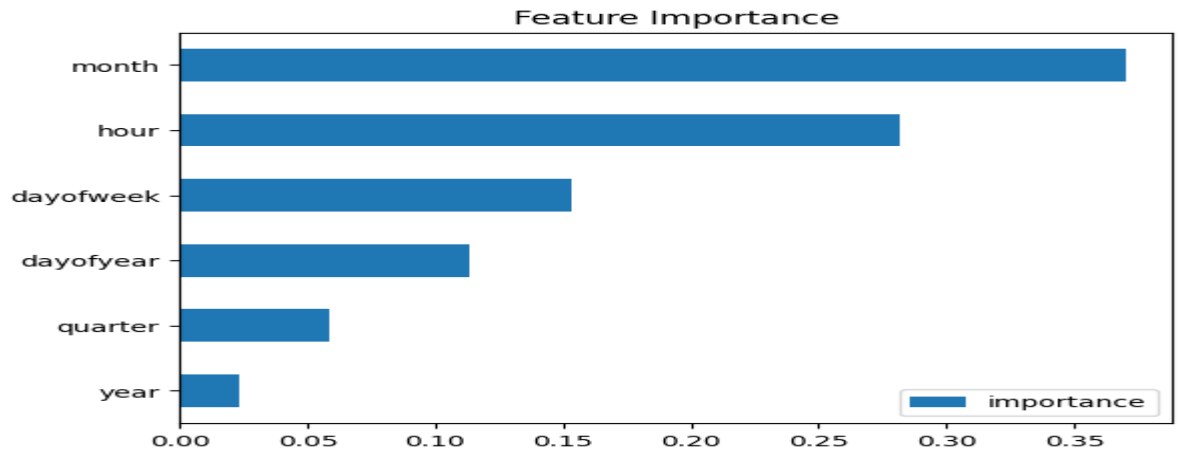
unfold_moreShow hidden output

4.3 Features importance

We need to see how much these features were used in each of the trees built by XGBoost model.

In [11]:

```
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()
```



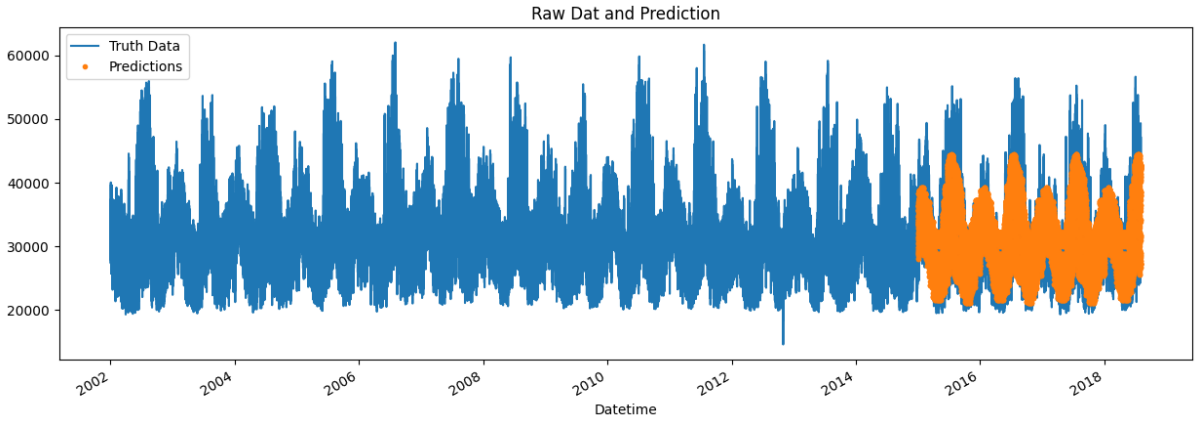
Step 5: Forecasting on test data

compare the prediction with the actual values.

In [12]:

```
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()
```



In [13]:

```
# RMSE Score
score = np.sqrt(mean_squared_error(test['PJME_MW'], test['prediction']))
print(f'RMSE Score on Test set: {score:0.2f}')
```

RMSE Score on Test set: 3721.75

In [14]:

```
# R2 Score
from sklearn.metrics import r2_score

r2 = r2_score(test['PJME_MW'], test['prediction'])
print("R-squared (R2) Score:", r2)\
```

squared (R2) Score: 0.6670230260104328

Our goal in this project is to use the hourly power consumption data to predict the PJME in the future. This is done by developing a model that can be used to predict future values based on the past values of the time series.

```
# import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# customize the style
```

```
pd.options.display.float_format = '{:.5f}'.format
pd.options.display.max_rows = 12

# load the data
filepath = '../input/hourly-energy-consumption/PJME_hourly.csv'
df = pd.read_csv(filepath)

print("Now, you're ready for step one")
```

```
Now, you're ready for step one
```

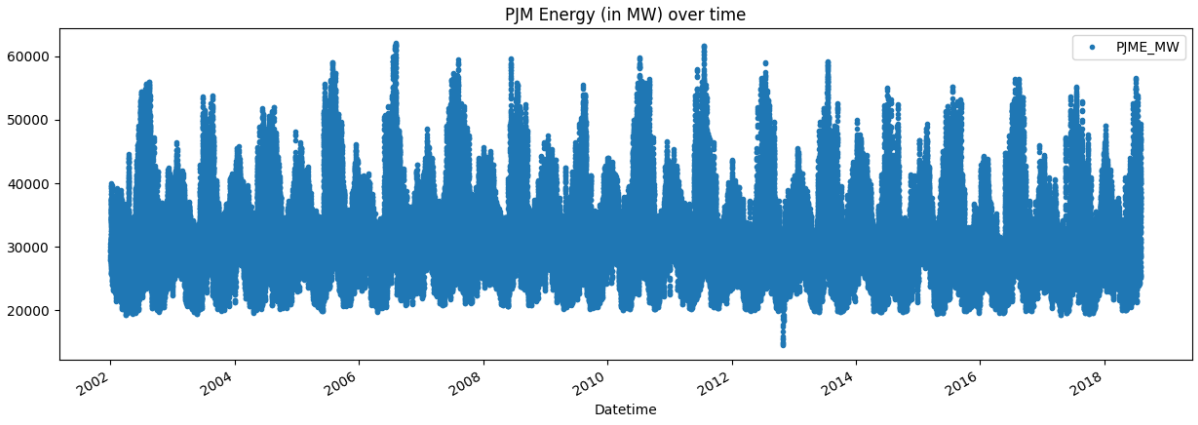
Step 1: Explore the data

In [2]:

```
# turn data to datetime
df = df.set_index('Datetime')
df.index = pd.to_datetime(df.index)
```

In [3]:

```
# create the plot
df.plot(style='.',
        figsize=(15, 5),
        title='PJM Energy (in MW) over time')
plt.show()
```



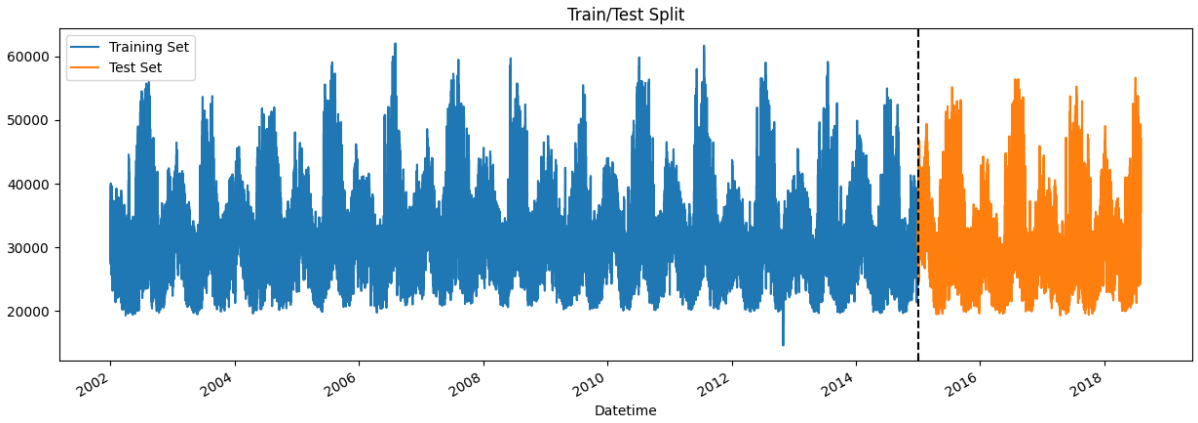
Step 2: Split the data

Everything prior to January 2015 will be our training data and keep our test data as the following dates.

In [4]:

```
# train / test split
train = df.loc[df.index < '01-01-2015']
test = df.loc[df.index >= '01-01-2015']
```

unfold_moreShow hidden code



Step 3: Feature Engineering

We're going to create some time features using the Datetime index. After that, we'll explore the distributions of Hourly and Monthly megawatt usage.

In [6]:

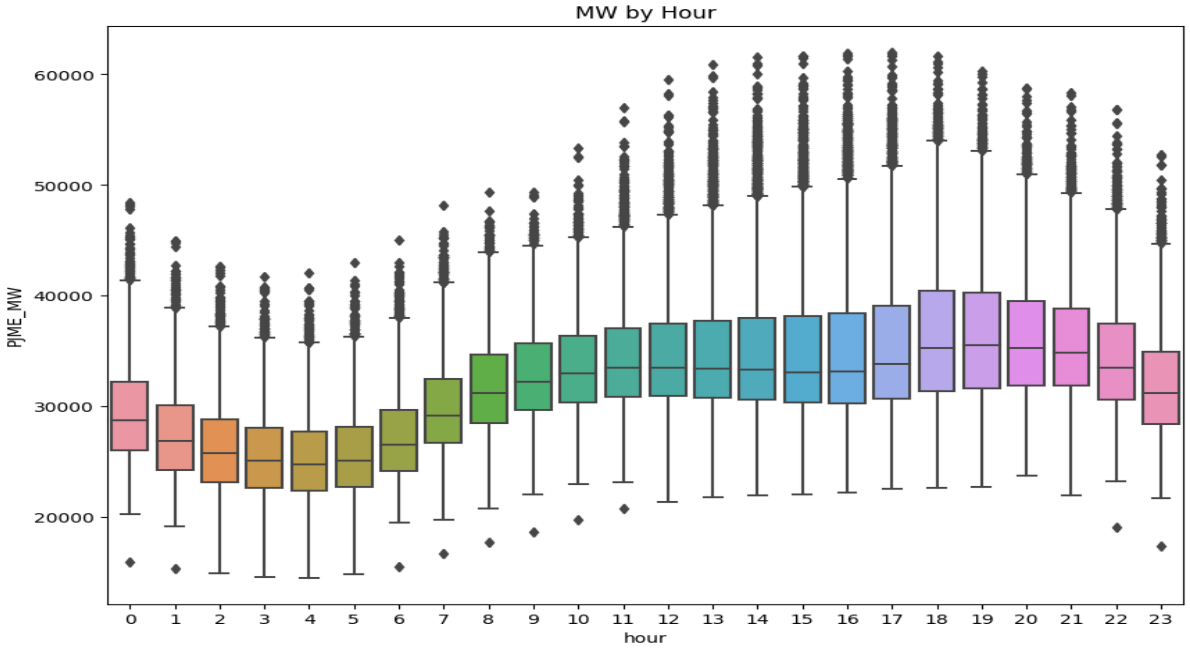
```
# feature creation
def create_features(df):
    df = df.copy()
    df['hour'] = df.index.hour
    df['dayofweek'] = df.index.dayofweek
    df['quarter'] = df.index.quarter
    df['month'] = df.index.month
    df['year'] = df.index.year
    df['dayofyear'] = df.index.dayofyear
    df['dayofmonth']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

df = create_features(df)
```

In [7]:

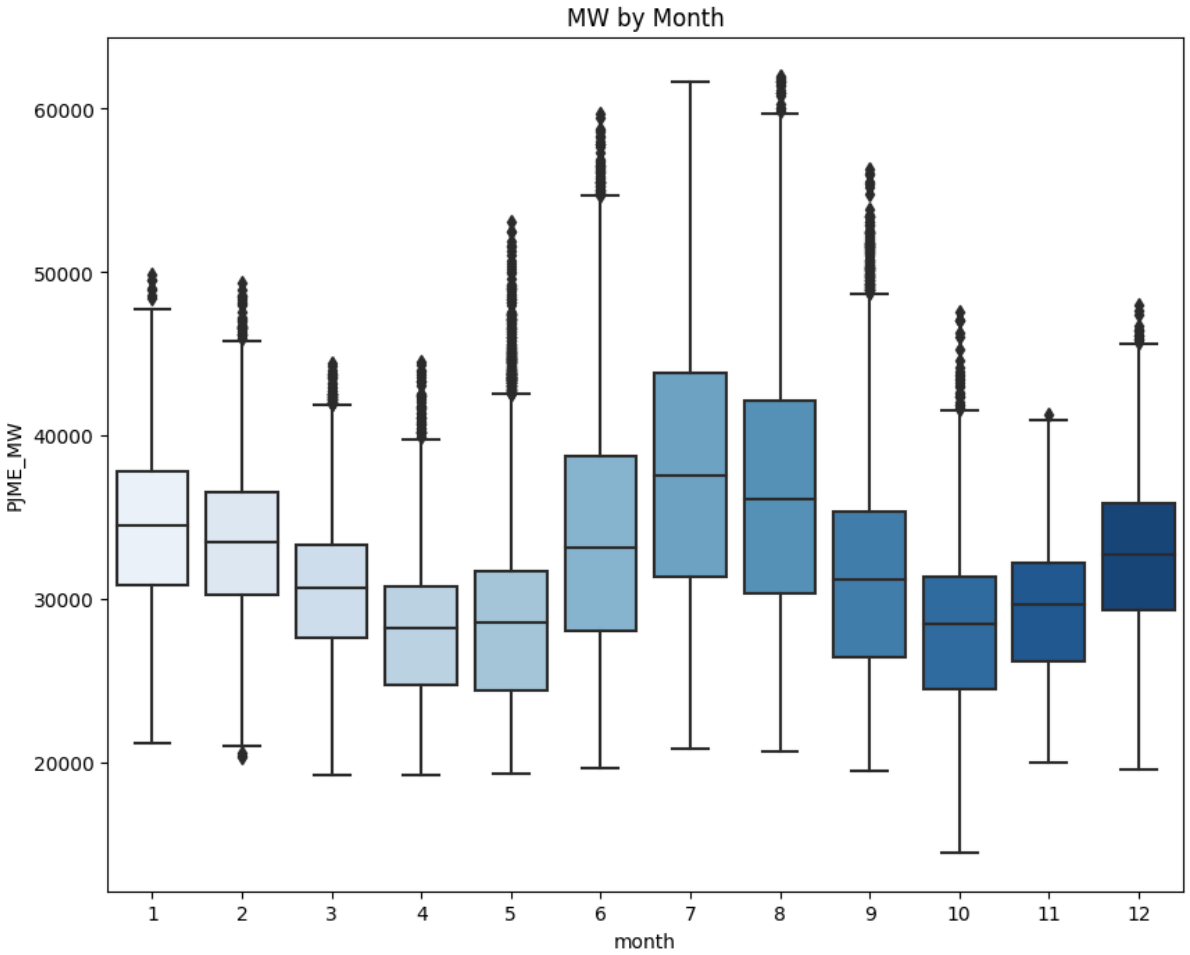
```
# visualize the hourly Megawatt
fig, ax = plt.subplots(figsize=(10, 8))
```

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



In [8]:

```
# viaualize the monthly Megawatt
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()
```



Step 4: Modelling

XGBoost is good and reliable model for regression and time series analysis as well. Also, for the metrics, we'll use mean squared error.

4.1 Prepare the data

In [9]:

```
# 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]
```

4.2 Build the model

In [10]:

```
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)
```

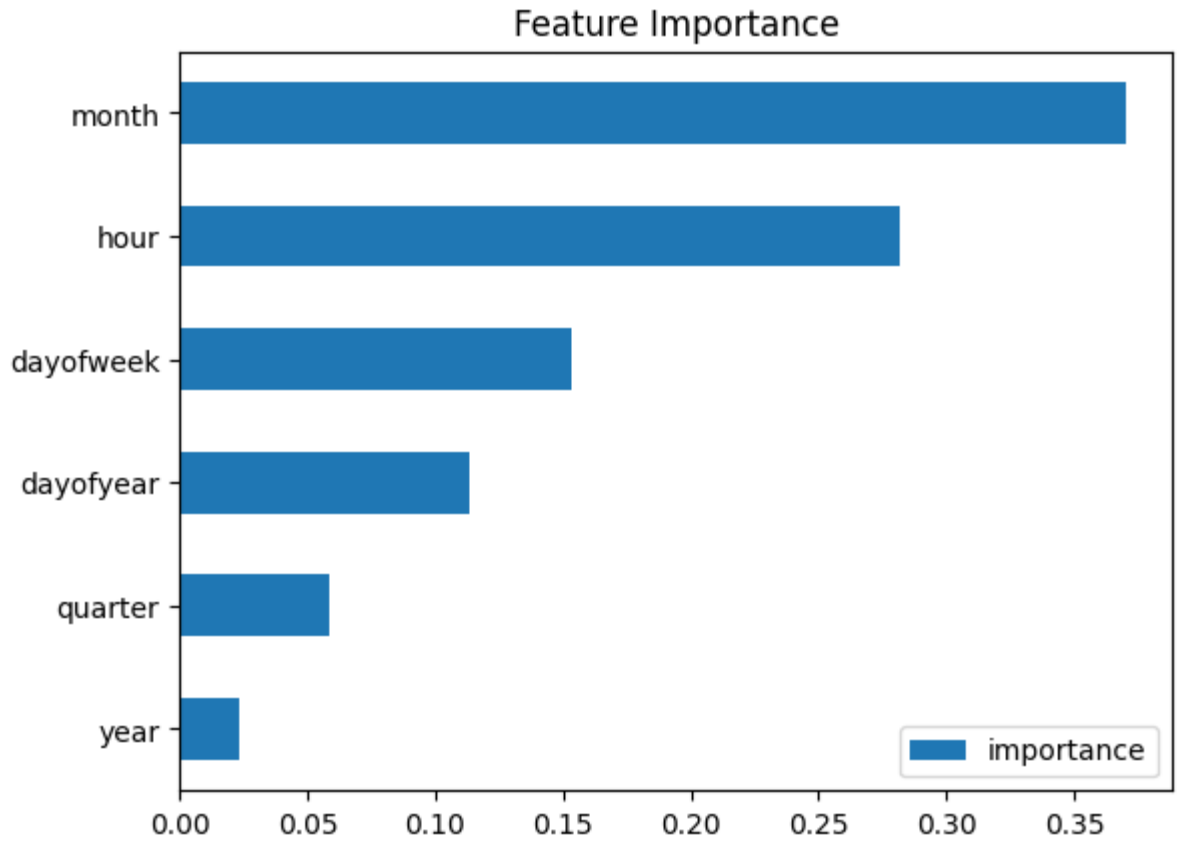
unfold_moreShow hidden output

4.3 Features importance

We need to see how much these features were used in each of the trees built by XGBoost model.

In [11]:

```
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()
```

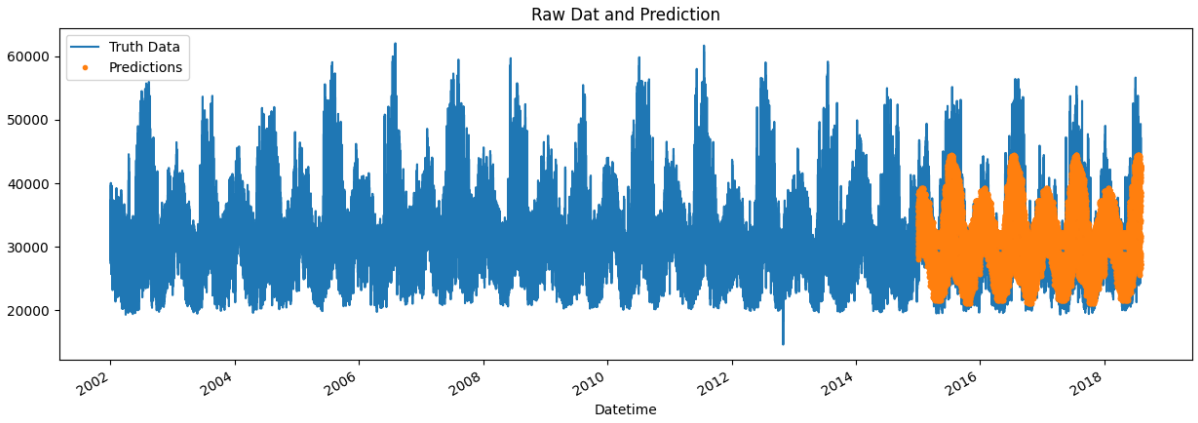


Step 5: Forecasting on test data

compare the prediction with the actual values.

In [12]:

```
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()
```



In [13]:

```
# RMSE Score
score = np.sqrt(mean_squared_error(test['PJME_MW'], test['prediction']))
print(f'RMSE Score on Test set: {score:0.2f}')
```

```
RMSE Score on Test set: 3721.75
```

In [14]:

```
# R2 Score
from sklearn.metrics import r2_score

r2 = r2_score(test['PJME_MW'], test['prediction'])
print("R-squared (R2) Score:", r2)
```

```
R-squared (R2) Score: 0.6670230260104328
```

Conclusion:

In conclusion, measuring energy consumption is a critical step in our journey towards a more sustainable and responsible use of resources. It empowers us to make informed decisions, reduce waste, lower costs, and contribute to a healthier planet. It is a collective responsibility that, when embraced at all levels of society, can lead to a more sustainable and prosperous future.