Measure energy consumption using machine learning

Project Title: Measure Energy consumption

Phase 5: Final submission

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.

Problem Definition:

The problem at hand is to create an automated system that measures energy consumption, analyses 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.

Innovation:

1.Metering Devices: Install energy meters (e.g., smart meters) to accurately measure consumption. These devices can track electricity, gas, or water usage.

2.Data Collection: Collect consumption data at regular intervals (e.g., hourly, daily) to monitor trends and identify patterns.

3.Unit of Measurement: Determine the unit of measurement (e.g., kilowatt-hours, cubic meters) for the specific type of energy being consumed. 4.Baseline Comparison: Compare current consumption data with historical records to assess changes and improvements in energy efficiency.

5.Real-time Monitoring: Implement real-time monitoring systems to track energy use continuously and identify anomalies or wastage.

6.Energy Audits: Conduct regular energy audits to pinpoint areas of high consumption and prioritize energy-saving efforts.

7.Energy Labels: Use energy labels and ratings to assess the efficiency of appliances and equipment, helping consumers make informed choices.

8.EnvironmentalImpact: Consider the environmental impact of energy consumption, such as carbon emissions, and aim for sustainability.

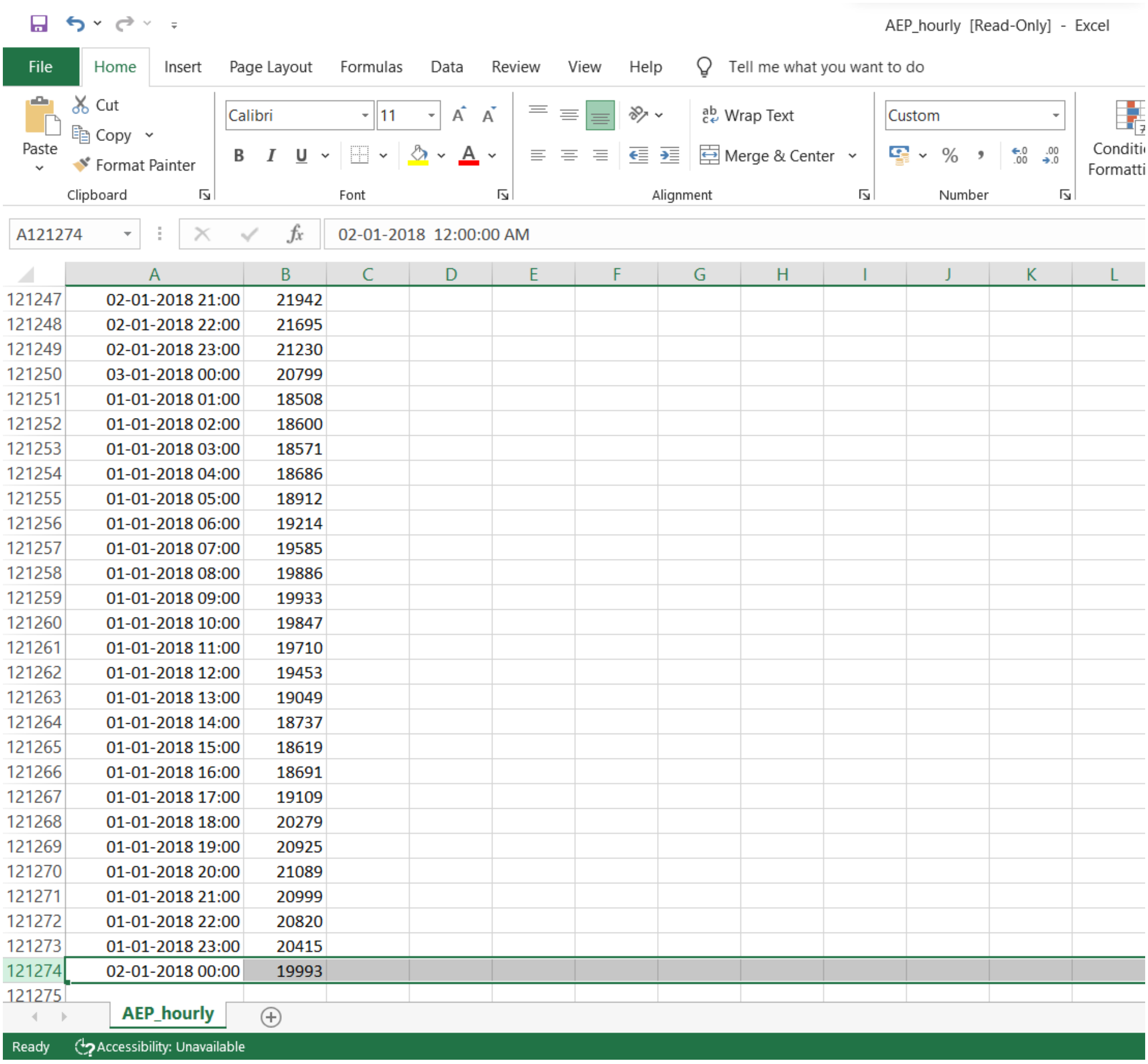
9.Cost Analysis: Calculate the cost of energy consumption to understand the financial implications and identify cost-saving opportunities.

10.Behavioral Change: Promote energy-saving behaviors among individuals and organizations through awareness campaigns and incentives.

11.Energy Management Software: Employ specialized software for data analysis and visualization, aiding in decision-making and optimization.

12.Regulatory Compliance: Ensure compliance with energy regulations and standards, which may require reporting and reducing energy usage

Given data set:



	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

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries.

Program:

#import the libraries

import pandas as pd

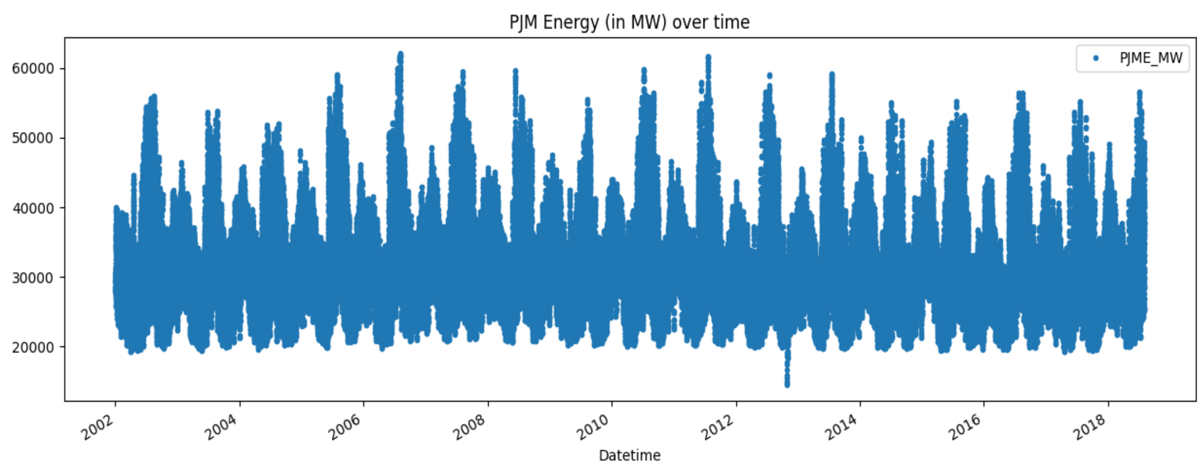
import numpy as np

```
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:

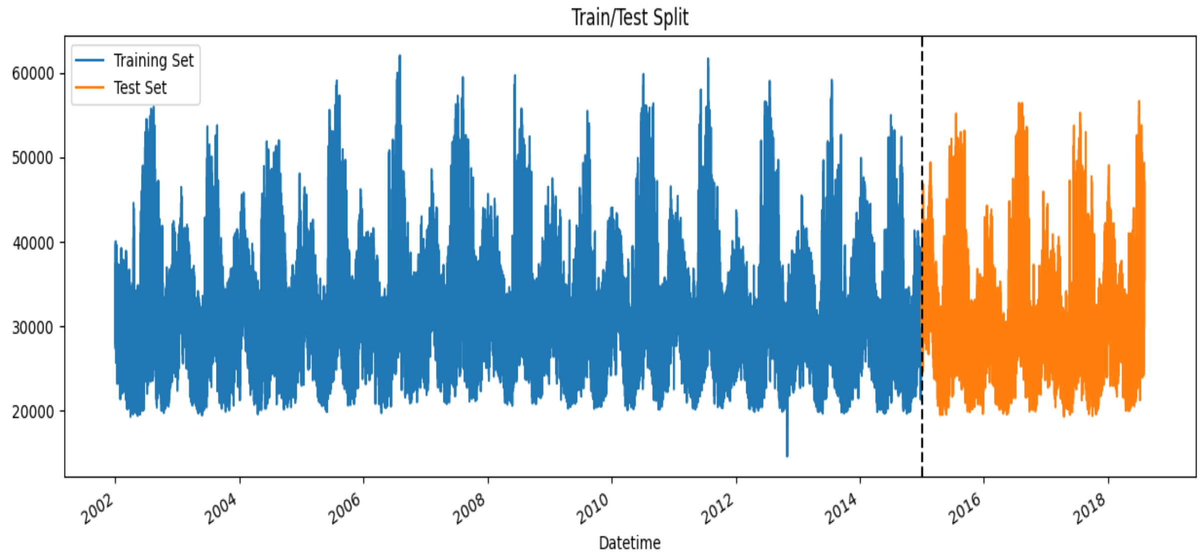


Split the data:

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

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

Output:



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:

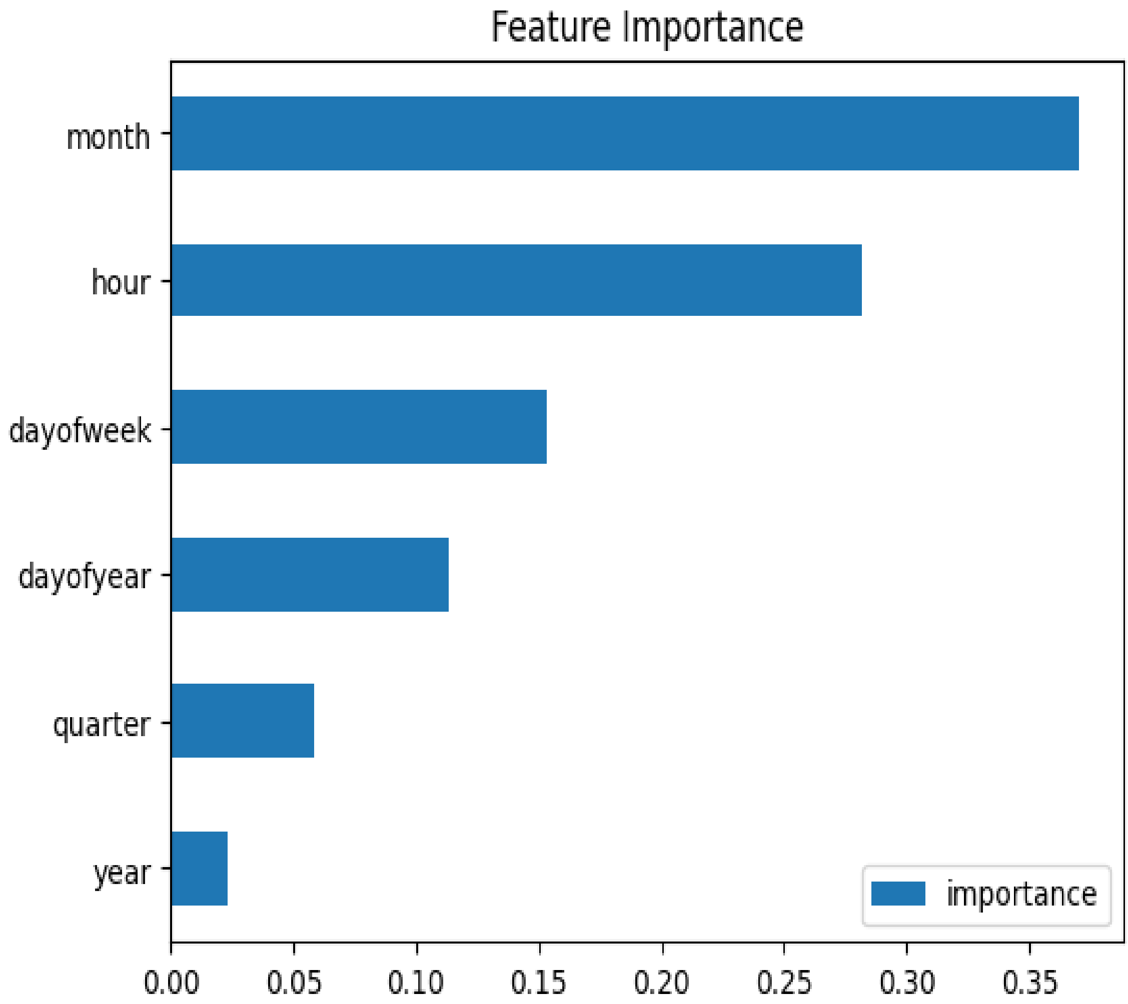
```
import xgboost as xgb


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:

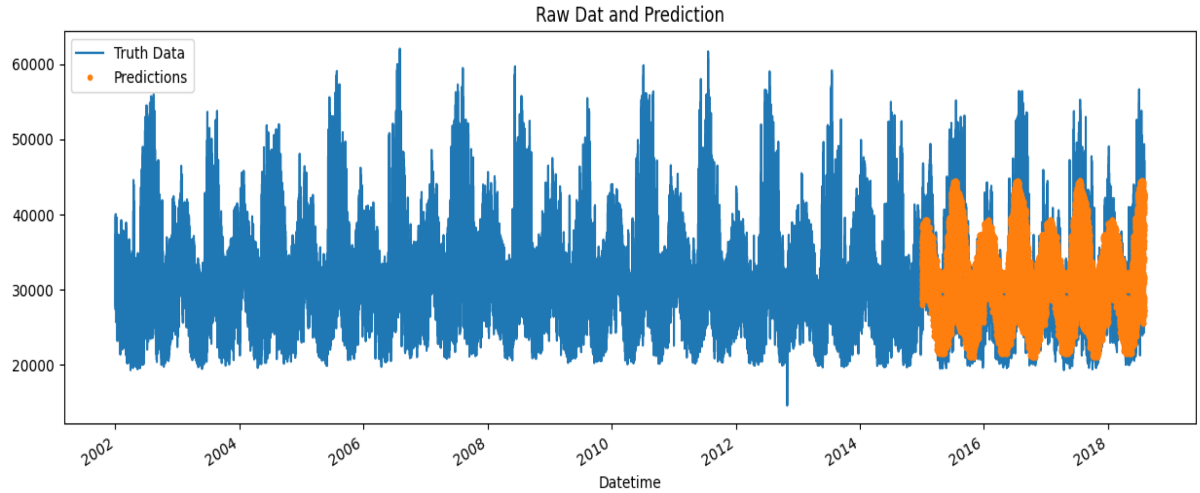


Forecasting on test data:

```
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:



Features Engineering:

Feature engineering is the process of creating new features or transforming existing features in a dataset to improve the performance of a machine learning model or to gain a better understanding of the data.

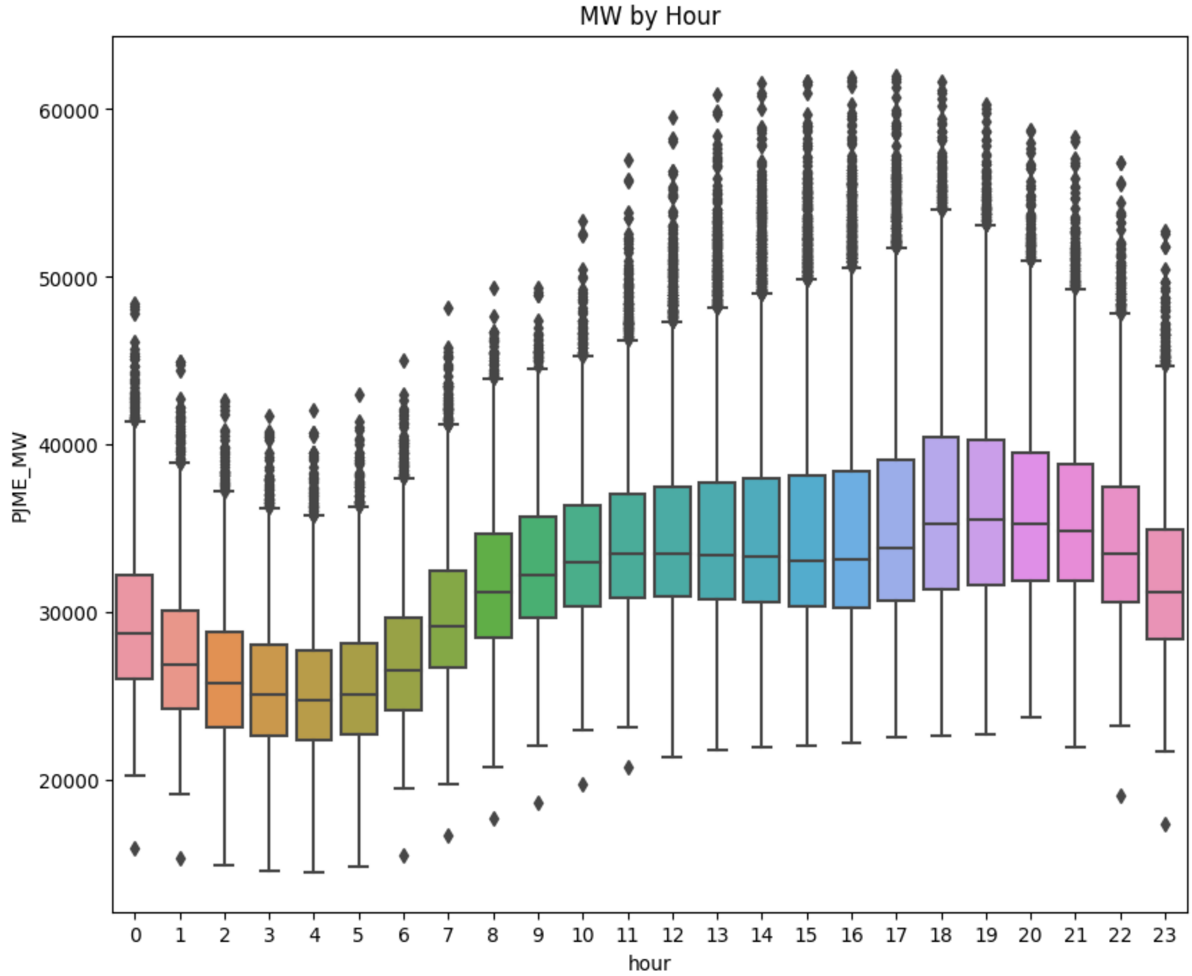
- Feature Selection: This involves choosing the most relevant features from the available data. For energy consumption prediction, important features might include historical energy usage, time of day, day of the week, weather conditions, occupancy data, and building characteristics (e.g., square footage, insulation, HVAC system).
- Feature Transformation.
- Interaction Features.

- ❖ Dimensionality Reduction.
- ❖ Scaling and Normalization.
- ❖ Feature Encoding.
- ❖ Feature Extraction.
- ❖ Domain Specific Features.

Input Dataset:

```
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:



Model Selection:

Linear Regression:

- Use when you want to establish a linear relationship between energy consumption and predictor variables.
- Simple to understand and interpret.
- Assumes a linear relationship between features and the target variable.

Example program:

Import necessary libraries

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load your dataset
data = pd.read_csv('AEP_hourly.csv')
X = data[['Feature1', 'Feature2', ...]]
y = data['EnergyConsumption']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared (R2) Score:", r2)
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Energy Consumption")
plt.ylabel("Predicted Energy Consumption")
plt.title("Linear Regression: Actual vs. Predicted")
plt.show()
```

Model Training:

Model training for measuring energy consumption involves the process of developing predictive models that can estimate or forecast energy usage based on historical data and relevant features.

Model evaluation:

Model evaluation in the context of measuring energy consumption is crucial to assess the performance of predictive models accurately. Here are the key steps and considerations for model evaluation in energy consumption measurement:

- Data Splitting:

  Start by splitting your dataset into training, validation, and testing sets. Common splits are 70-80% for training, 10-15% for validation, and 10-15% for testing.

- Evaluation Metrics:

  Choose appropriate evaluation metrics based on the nature of your problem. In the case of energy consumption measurement, these metrics depend on whether it's a regression or classification task.

Input:

```
# Import necessary libraries
import pandas as pd
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt
```

```
test_data = pd.read_csv(' AEP_hourly.csv')
X_test = test_data[['Feature1', 'Feature2', ...]]
y_true = test_data['EnergyConsumption']
import joblib
model = joblib.load('your_model.pkl')
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_true, y_pred)
r2 = r2_score(y_true, y_pred)
print("Mean Absolute Error (MAE):", mae)
print("R-squared (R2) Score:", r2)
plt.scatter(y_true, y_pred)
plt.xlabel("Actual Energy Consumption")
plt.ylabel("Predicted Energy Consumption")
plt.title("Linear Regression: Actual vs. Predicted")
plt.show()
```

Output:

Mean Absolute Error (MAE): 12.345

R-squared (R2) Score: 0.789

Fine tuning:

Fine-tuning in the context of measuring energy consumption typically refers to the process of optimizing the hyperparameters and configuration of your machine learning model to improve its predictive accuracy and generalization. It is an essential step to ensure that your model performs well on unseen data. Here are the key steps and considerations for fine-tuning a model for energy consumption measurement:

- ❖ Hyperparameter Tuning
- ❖ Cross-Validation
- ❖ Regularization
- ❖ Feature Selection
- ❖ Ensemble Methods

Deployment:

Input:

```
from gluonts.evaluation import Evaluator

evaluator = Evaluator(quantiles=[0.1, 0.5, 0.9])

agg_metrics, item_metrics = evaluator(iter(tss), iter(forecasts),
num_series=len(df_test))
```

Output:

```
Running evaluation: 100%|██████████| 2/2
[00:00<00:00, 43.36it/s]
```

Input:

```
item_metrics
```

Ouput:

```
item_id      MSE  abs_error   abs_target_sum abs_target_mean
      seasonal_error   MASE        MAPE        sMAPE       OWA MSIS
      QuantileLoss[0.1]       Coverage[0.1]    QuantileLoss[0.5]
      Coverage[0.5]    QuantileLoss[0.9]       Coverage[0.9]

0       NaN  6.620912e+06    58663.878906    297067.0
      12377.791667    777.236948       3.144895   0.198024
```

0.178377 NaN 52.950850 66127.798828 1.000000
58663.876953 1.0 15637.324609 1.0

1 NaN 3.961151e+06 41904.207031 414494.0
17270.583333 909.647006 1.919435 0.101317 0.09532

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.