VELAMMAL INSTITUTE OF TECHNOLOGY

Approved by AICTE - New Delhi
Affiliated to Anna University - Chennai
Accredited by NBA & NAAC

NAAN MUDHALVAN PROJECT(IBM)

IBM AI 101 ARTIFICIAL INTELLIGENCE-GROUP 1

Title : Measure Energy Consumption

Team name: Proj_224826_Team_1

Phase : Phase 5

Team members: SIVASUBRAMANI C (reg no 1133211036090)

THIRUMURUGAN S(reg no 113321106107)

SANTHOSH R(reg no 113321106083)

SRINIVASA PRABHU(reg no 113321106095)

SAISHANKAR(reg 113321106302)

Problem Statement:

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. To maintain reliability and effectiveness, the system includes ongoing support and continuous improvement. This ensures that it remains a valuable asset for optimizing energy usage and promoting sustainability.

Project Description: This project aims to develop an end-to-end solution for Targeted area :Home power consumption by collecting energy consumption data, performing real-time analysis, and providing insightful visualizations to support informed decision-making. The system will enhance efficiency, accuracy, and ease of understanding in managing energy consumption across various sectors.

Project Components:

1. **Data Collection and Integration**:
   - **Data Sources**:
     - Smart Meters
     - IoT Sensors
     - Building Management Systems (BMS)
     - Weather Data
     - Energy Management Systems (EMS)
   - **Data Collection**:
     - Implement data collection modules to retrieve data from various sources.
     - Integrate data collection processes to ensure data from different sources is consolidated.

Four main steps to be follow

--> Data Sources

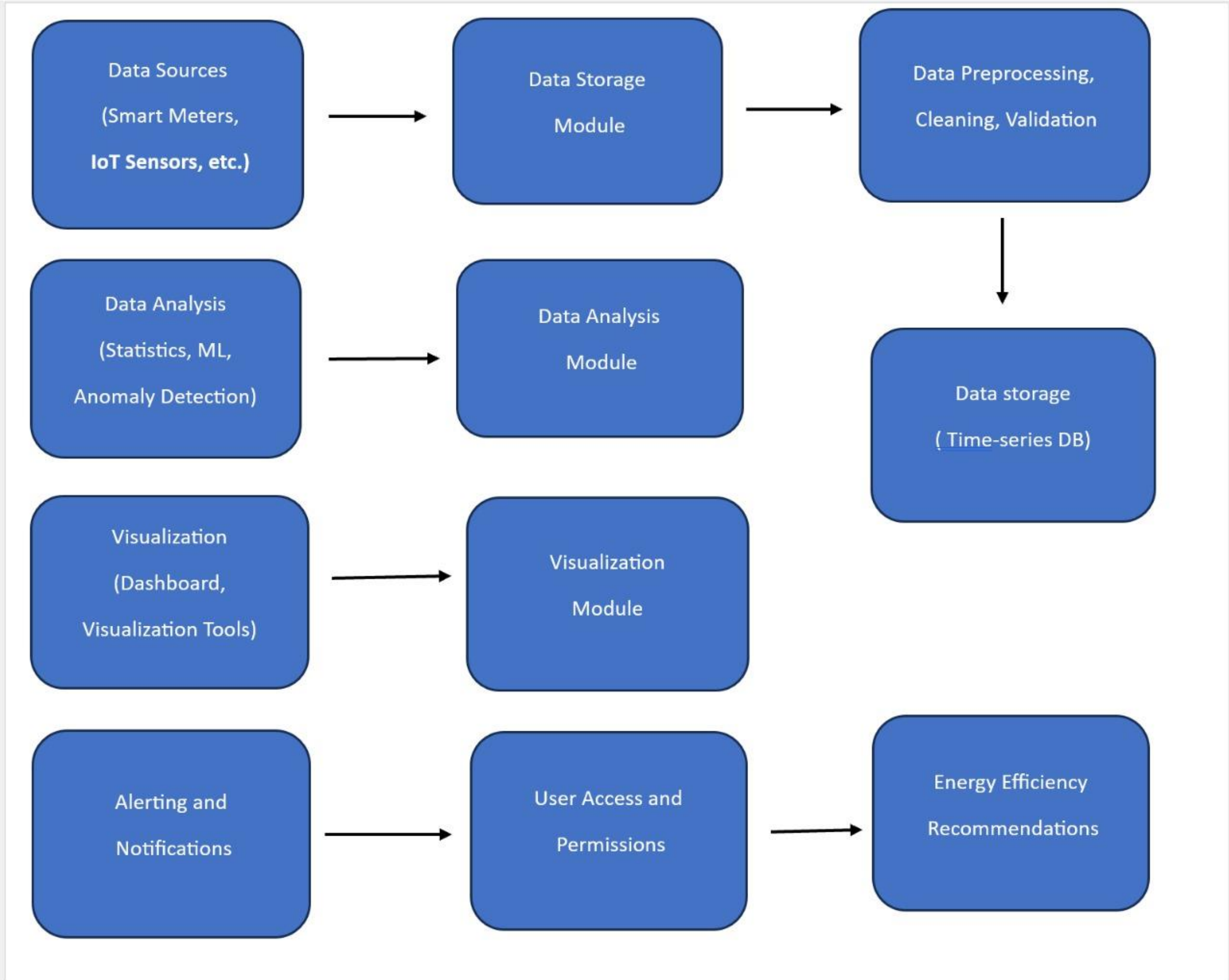
--> Data Collection

--> Data Storage

--> Data Preprocessing

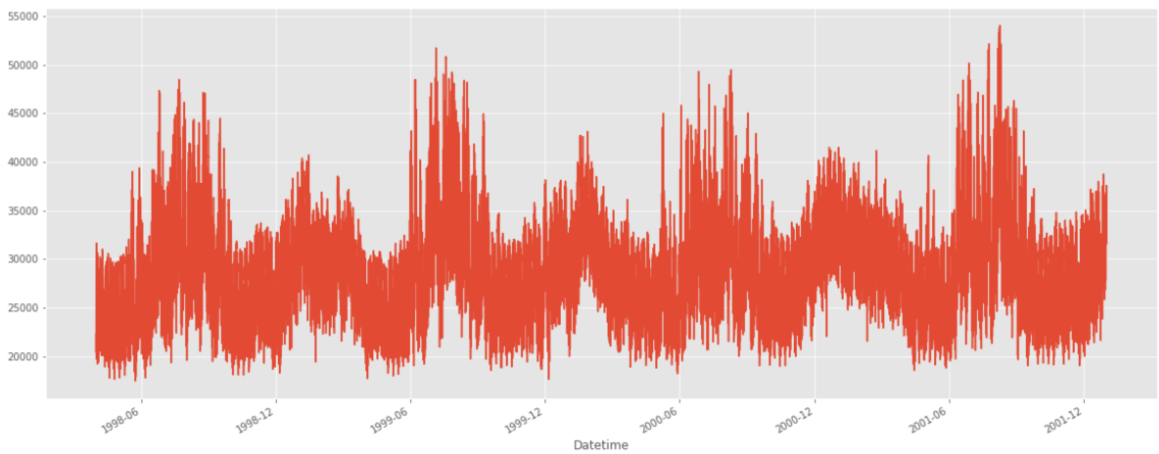
Block Diagram:

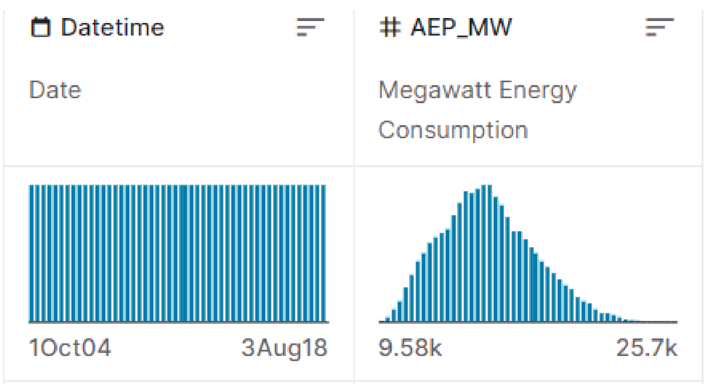
A block diagram represents the major components and their relationships in the system. In this case, the block diagram would depict the flow of data and operations within the system. Here's a textual representation of the block diagram:



Data set Analysis Techniques:

Use techniques such as regression analysis, time series analysis, clustering, or machine learning to extract valuable insights from the data. For your energy consumption data, you might want to explore seasonality, trends, and potential anomalies.





Machine Learning Algorithm:

Random Forest Regression Algorithm

Input:

• Training dataset with n samples and m features: (X, y)

Parameters:

Nestimators: Number of decision trees in the forest.

Mfeatures: Number of features considered for each split.

D: Maximum depth of each decision tree.

N samples split: Minimum number of samples required to split an internal node.

N samples leaf: Minimum number of samples required to be at a leaf node.

Output:

Ensemble of Nestimators regression trees.

1. Initialization:

• For i in 1 to Nestimators:

• Create a bootstrap sample (Xi, yi) from (X, y)

• Build a decision tree T_{i} using (X_{i}, y_{i}) with the following

Parameters:

• Maximum depth D.

Minimum samples required to split nsamples_split.

• Minimum samples required at a leaf node samples_leaf-

Randomly select Mfeatures features for each split.

2. Prediction:

• For a new sample x, predict the output ŷ as follows:

$\hat{Y}$=RFregessor(x)

Where

ŷ is the predicted output (target variable) for the new sample x.

RFregressor is the trained Random Forest Regressor model.

x represents the input features of the new sample for which you want to make a prediction.

3. Ensemble Learning:

• Combine the predictions from multiple trees by averaging (for regression tasks).

4. Mean Squared Error (MSE):

• Calculate the MSE to evaluate the performance of the model:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

5. Feature Importance:

• Analyse feature importance based on metrics like Gin impurity or mean decrease in impurity.

6. Final Output:

The final output is an ensemble of Nestimators decision trees, which collectively predict the target variable.

Algorithm for the proposed method:

Importing Libraries: The script begins by importing the necessary Python libraries:

numpy (np) is used for generating random data.

matplotlib.pyplot (plt) is used for creating data visualizations.

sklearn.model_selection.train_test_split is used to split the dataset into training and testing sets.

sklearn.ensemble.RandomForestRegressor is used to create and train a Random Forest Regressor model.

sklearn.metrics.mean_squared_error is used to calculate the Mean Squared Error (MSE) as an evaluation metric.

Generating a Sample Dataset: In this script, a synthetic dataset is generated for demonstration purposes. The dataset consists of random feature (X) values and target (y) values representing energy consumption. The target values (y) are generated with some random noise to simulate real-world data.

Splitting the Data: The dataset is split into training and testing sets using the train_test_split function from scikit-learn. The training set (X_train, y_train) is used to train the model, while the testing set (X_test, y_test) is used to evaluate the model's performance.

Initializing and Training the Random Forest Regressor: A Random Forest Regressor model is initialized with 100 trees using the RandomForestRegressor class from scikit-learn. The n_estimators parameter specifies the number of trees in the forest. The model is then trained using the training data (X_train, y_train) by calling the fit method.

Making Predictions: After the model is trained, it is used to make predictions on the test data (X_test) using the predict method. The predicted values are stored in the y_pred variable.

Calculating Mean Squared Error (MSE): The script calculates the Mean Squared Error (MSE) as an evaluation metric. MSE measures the average squared difference between the actual and predicted values. It is calculated using the mean_squared_error function from scikit-learn and is stored in the mse variable.

Visualizing the Results: The script creates three types of data visualizations to help understand the model's performance:

Scatter Plot: This plot compares the actual and predicted energy consumption values using different colors for clarity.

Bar Chart: This chart displays the total energy consumption for both actual and predicted values, helping to see how close the model's predictions are to the actual values.

Pie Chart: The pie chart shows the distribution of energy consumption between actual and predicted values. It provides a visual representation of the proportions of actual and predicted energy consumption.

Python Coding:

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
```

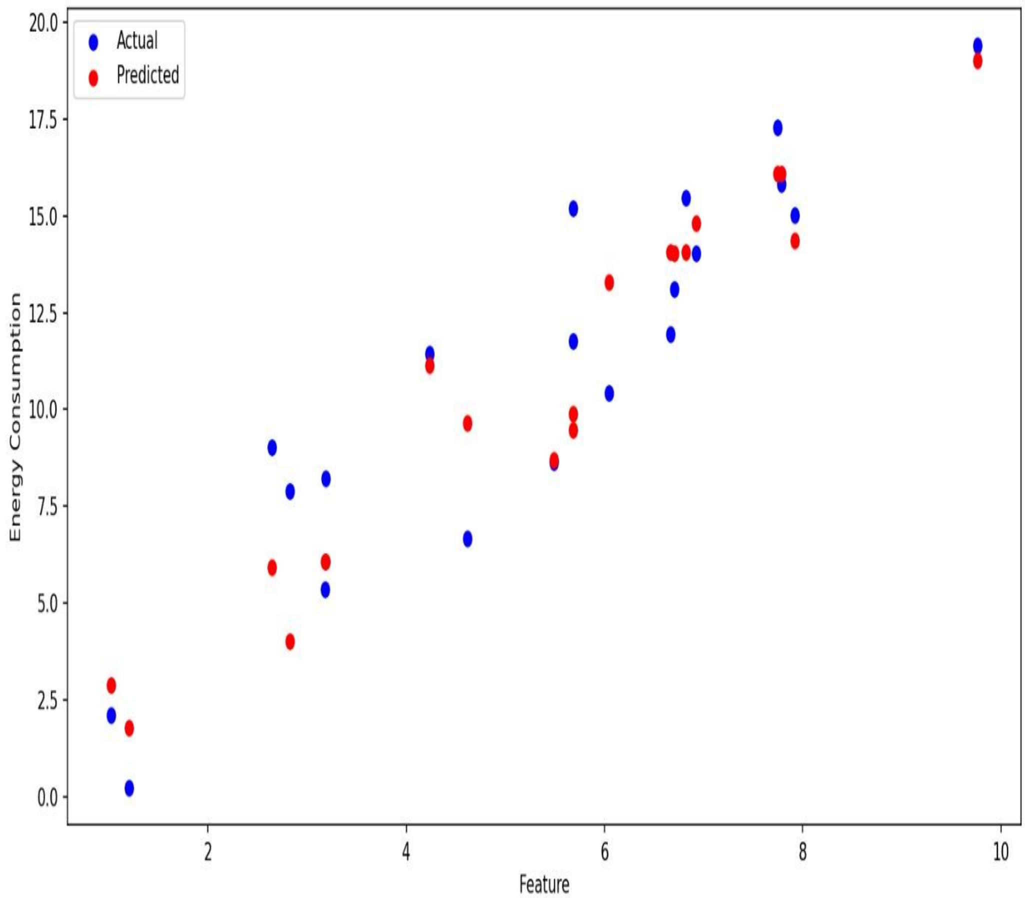
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# Generate a sample dataset (replace this with your actual data)
np.random.seed(0)
X = np.random.rand(100, 1) * 10
y = 2 * X.squeeze() + np.random.randn(100) * 2  # Simulated energy
consumption

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and train the Random Forest Regressor
rf_regressor = RandomForestRegressor(n_estimators=100,
random_state=42)
rf_regressor.fit(X_train, y_train)

# Predict energy consumption for test set
y_pred = rf_regressor.predict(X_test)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')

# Visualize the results
```

```
plt.figure(figsize=(12, 4))

# Scatter Plot
plt.subplot(1, 3, 1)
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.scatter(X_test, y_pred, color='red', label='Predicted')
plt.xlabel('Feature')
plt.ylabel('Energy Consumption')
plt.title('Actual vs. Predicted')
plt.legend()

# Bar Chart
plt.subplot(1, 3, 2)
labels = ['Actual', 'Predicted']
values = [np.sum(y_test), np.sum(y_pred)]
plt.bar(labels, values, color=['blue', 'red'])
plt.xlabel('Category')
plt.ylabel('Total Energy Consumption')
plt.title('Total Energy Consumption')

# Pie Chart
plt.subplot(1, 3, 3)
sizes = [np.sum(y_test), np.sum(y_pred)]
colors = ['blue', 'red']
```

```
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Energy Consumption Distribution')

plt.tight_layout()
plt.show()
```

Output:

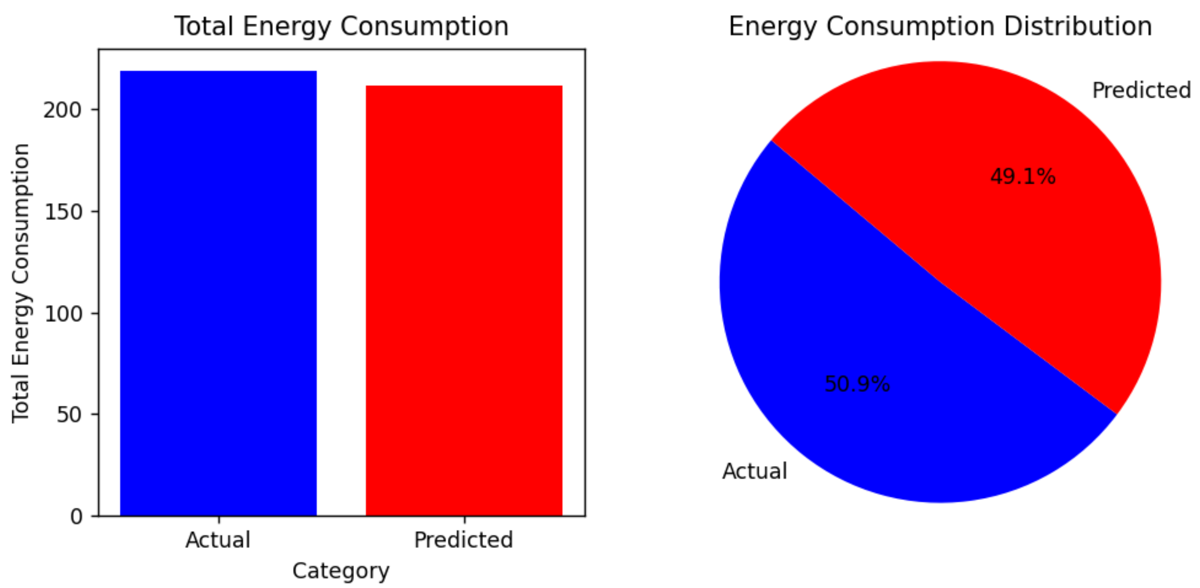
1) A regression line analysis of actual and predicted data.



2) Mean Squared Error (MSE):

```
*IDLE Shell 3.10.2*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: D:/Subbu/vit 5th sem/ML algorithm.py ================
Traceback (most recent call last):
  File "D:/Subbu/vit 5th sem/ML algorithm.py", line 3, in <module>
    from sklearn.model_selection import train_test_split
ModuleNotFoundError: No module named 'sklearn'
>>>
================= RESTART: D:/Subbu/vit 5th sem/ML algorithm.py ================
Mean Squared Error: 4.65
```

3)Bar chart and pie chart



Conclusion

Our Project Measure Energy Consumption use of a Random Forest Regressor, a robust machine learning algorithm, to predict energy consumption. It efficiently splits data into training and testing sets, calculates the Mean Squared Error (MSE) for evaluation, and visualizes results. While the code showcases a synthetic dataset, it provides a foundation for building accurate energy consumption prediction models, which are invaluable for optimizing resource utilization and driving sustainability."