**NAAN MUDHALVAN PROJECT 4 PHASE 2**

**Project Name:** Measure Energy Consumption

**Phase 2:** In this we have to transform our idea into innovation. Here are the methods and the program for data cleaning and analysis

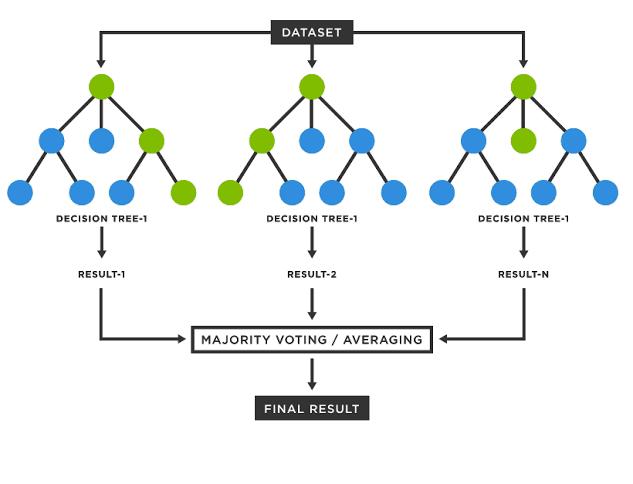
To improve the accuracy and robustness of an energy consumption prediction system, you can explore innovative techniques like ensemble methods and deep learning architectures. They are

**Ensemble Methods:**

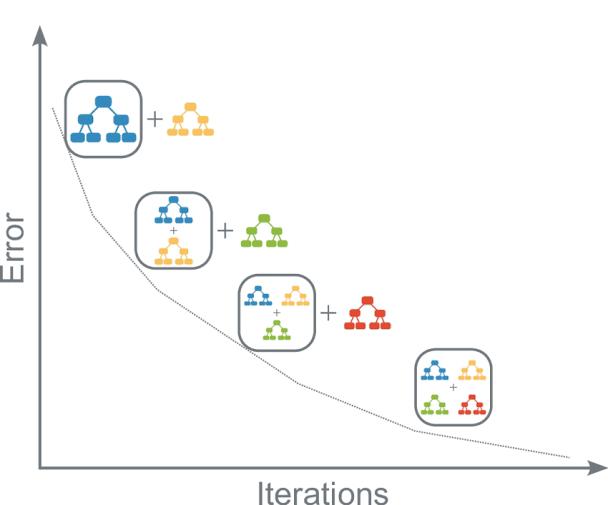
Ensemble methods combine multiple machine learning models to enhance prediction accuracy and robustness. Here are some ensemble techniques to consider:

**1. Random Forests:**

* Random Forests are an ensemble of decision trees. They are highly effective for regression tasks like energy consumption prediction.
* Random Forests can capture complex relationships and provide feature importance analysis, helping you understand which factors influence consumption the most.

**2. Gradient Boosting Algorithms:**

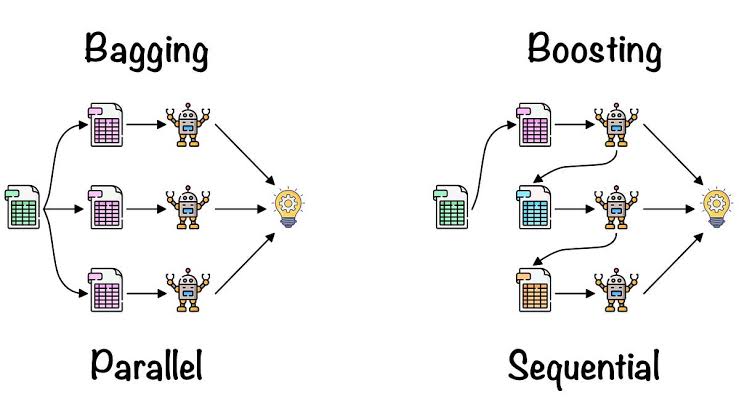
* Algorithms like XGBoost, LightGBM, and CatBoost are powerful for improving predictive accuracy.
* They work by sequentially training weak learners (typically decision trees) and focusing on samples that are misclassified or have higher residuals.

**3. Stacking:**

* Stacking involves training multiple diverse models and then combining their predictions to make the final prediction.
* You can experiment with different base models and meta-learners to optimize performance.

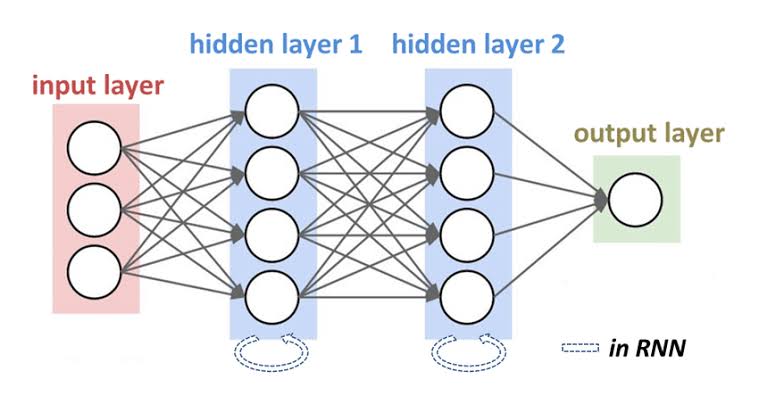
**4. Bagging (Bootstrap Aggregating):**

* Bagging generates multiple subsets of the training data through bootstrapping and trains models on each subset.
* The final prediction is an average or majority vote of predictions from individual models, reducing overfitting.

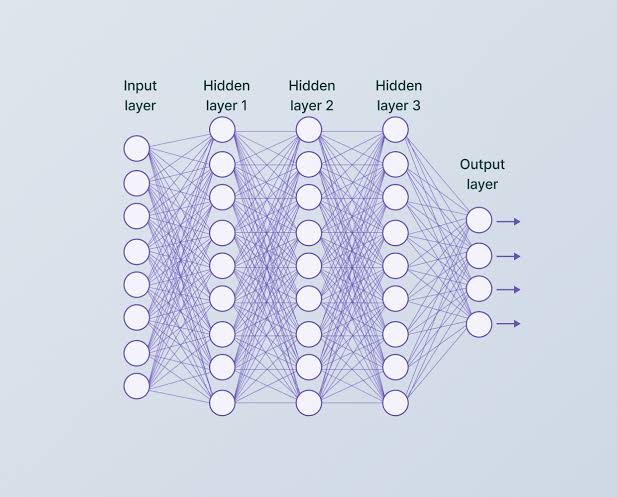
**Deep Learning Architectures:**

Deep learning architectures, particularly neural networks, can capture complex patterns in energy consumption data. Consider these approaches:

**1. Recurrent Neural Networks (RNNs):**

* RNNs, including Long Short-Term Memory (LSTM) networks, are well-suited for time series data like energy consumption.
* They can capture long-term dependencies and sequential patterns in the data.

**2. Convolutional Neural Networks (CNNs):**

* CNNs, primarily used for image analysis, can also be applied to time series data when patterns exist in the temporal dimension.
* They can extract relevant features from energy consumption time series.

**3. Hybrid Models:**

* Combine RNNs and CNNs in hybrid architectures to capture both temporal and spatial patterns in energy consumption data.
* These models are particularly useful when dealing with multivariate time series data from multiple sensors or sources.

**4. Attention Mechanisms:**

* Implement attention mechanisms in deep learning models to give more weight to certain time steps or features, allowing the model to focus on the most relevant Information for prediction.

**5. Autoencoders:**

* Autoencoders can be used for feature extraction and dimensionality reduction before feeding data into other deep learning models.
* They help in reducing noise and encoding essential information.

**6. Transfer Learning:**

* Consider using pre-trained neural networks, such as those for natural language processing or image recognition, and fine-tune them for energy consumption prediction.
* Transfer learning can leverage existing knowledge to improve model performance.

**7. Reinforcement Learning (RL):**

* Explore RL techniques for dynamic energy consumption prediction and optimization in real-time.
* RL models can learn to control building systems (e.g., HVAC, lighting) to minimize energy usage while maintaining comfort.

**8. Uncertainty Quantification:**

* Enhance deep learning models by incorporating uncertainty quantification techniques, such as Bayesian neural networks or dropout layers, to provide probabilistic forecasts and assess prediction confidence.

To make the most of ensemble methods and deep learning architectures, it’s essential to experiment with different configurations, hyperparameters, and training strategies. Additionally, ensure you have sufficient data, both for training and validation, to optimize and evaluate model performance effectively.

**Some of the other techniques are:**

**1. Time Series Forecasting Models:**

* **ARIMA (AutoRegressive Integrated Moving Average):** ARIMA models are widely used for time series forecasting. They capture trends, seasonality, and autocorrelation in energy consumption data.
* **Prophet:** Developed by Facebook, Prophet is designed for forecasting with daily observations that display patterns on different time scales. It can handle missing data and holidays.
* **Exponential Smoothing (ETS):** ETS methods, like Holt-Winters, capture different components of a time series, including level, trend, and seasonality.

**2. Feature Engineering:**

* Create informative features by extracting time-based attributes like day of the week, month, or year, as well as lag features (previous energy consumption values) and rolling statistics (e.g., moving averages).

**3. Seasonal Decomposition:**

* Use techniques like seasonal decomposition of time series (STL) to separate the time series data into its components, such as trend, seasonality, and residuals. Analyze and model these components individually.

**4. Exogenous Variables:**

* Integrate external factors that influence energy consumption, such as weather data (temperature, humidity), occupancy, building characteristics, or holidays, as exogenous variables in your models.

**5. Model Evaluation and Hyperparameter Tuning:**

* Assess the performance of your models using appropriate evaluation metrics (e.g., Mean Absolute Error, Root Mean Squared Error). Employ cross-validation to prevent overfitting.
* Tune hyperparameters to optimize model accuracy and generalization.

**6. Anomaly Detection:**

* Develop anomaly detection models alongside forecasting models to identify abnormal energy consumption patterns. Techniques like Isolation Forest or One-Class SVM can be used for this purpose.

**7. Bayesian Methods:**

* Explore Bayesian forecasting models that can incorporate uncertainty and probabilistic forecasts, providing a more realistic view of potential outcomes.

**8. Reinforcement Learning:**

* Consider using reinforcement learning to optimize energy consumption in real-time by learning policies that control building systems (e.g., HVAC, lighting) to minimize energy usage while maintaining comfort.

**9. Online Learning:**

* Implement online learning techniques that continuously update models as new data arrives, ensuring adaptability to changing consumption patterns.

**10. Explainable AI (XAI):**

* Employ XAI techniques to make AI-driven energy consumption predictions interpretable and actionable, helping users understand the basis of forecasts.

**11. Integration with IoT:**

* Integrate data from IoT sensors for real-time monitoring and enhance the accuracy of predictions by incorporating live data feeds.

**12. Energy Optimization:**

* Extend prediction models to include optimization algorithms that suggest energy-saving actions based on forecasted consumption patterns.

**13. Evaluation of Uncertainty:**

* Assess and quantify uncertainty in predictions to provide confidence intervals or probabilistic forecasts, which can be valuable for decision-making.

Experimenting with these innovative techniques and combining them as needed can lead to more accurate, robust, and actionable energy consumption predictions. Tailor your approach to the specific context and goals of your energy forecasting project.

**Data Cleaning and Missing Value Handling:**

* Check the dataset for missing values and anomalies. It’s crucial to address these issues ensure the accuracy of the analysis.
* Handle missing data using appropriate techniques such as:
* Imputation: Filling missing values with a suitable estimate (e.g., mean, median, forward-fill, backward-fill).
* Interpolation: If the dataset has a time component, consider interpolating missing values based on surrounding time points.
* Removal: If data is severely corrupted or missing, you might need to exclude affected records.
* Outlier Handling: Identify and address any outliers that might skew the analysis.

**Program:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

RED = "\033[91m"

GREEN = "\033[92m"

YELLOW = "\033[93m"

BLUE = "\033[94m"

RESET= "\033[em"

df = pd.read\_csv("AEP\_hourly.csv")

df["Datetime"] = pd.to\_datetime (df["Datetime"])

# DATA CLEANING

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)

# DATA ANALYSIS

print(BLUE + "\nDATA ANALYSIS" + RESET)

#-- Summary Statistics

summary\_stats = df.describe()

print(GREEN + "Summary Statistics: " + RESET)

print(summary\_stats)

# Data Visualization

#Line plot for energy consumption over time

plt.figure(figsize=(12, 6))

plt.plot(df.index, df["AEP\_MW"], label="Energy Consumption (AEP\_MW)")

plt.xlabel("Datetime")

plt.ylabel("Energy Consumption (MW)")

plt.title("Energy Consumption Over Time")

plt.legend()

plt.grid()

plt.show()

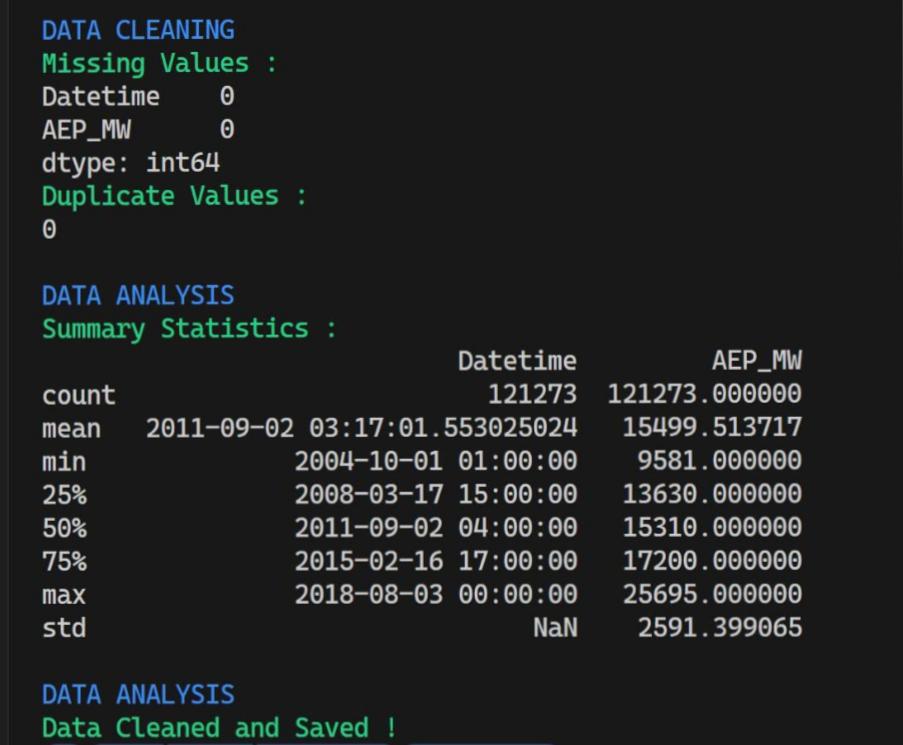
# SAVING THE FILE

df.to\_csv("cleaned\_AEP\_hourly.csv", index=False)

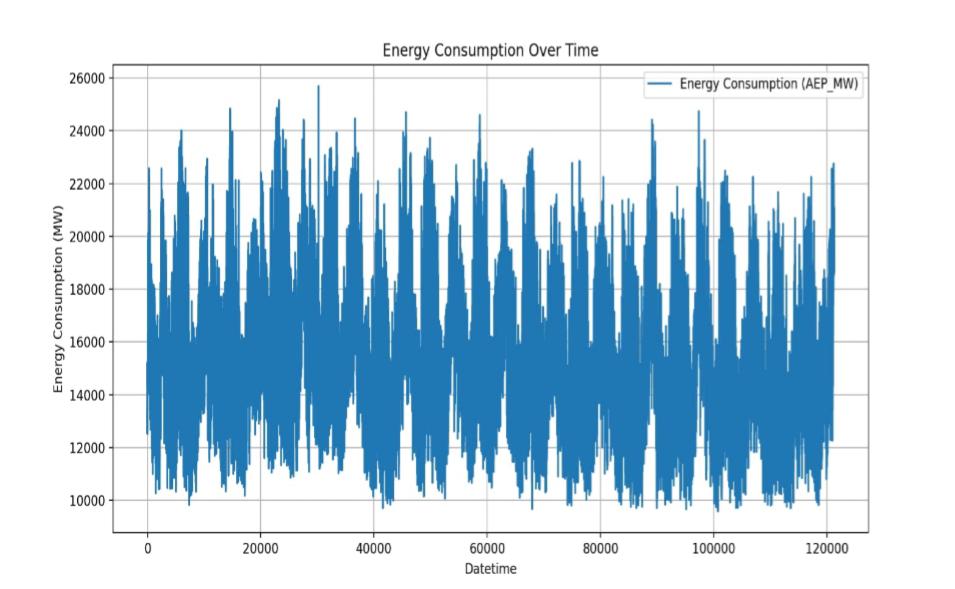
print(BLUE + "\nDATA ANALYSIS" + RESET)

print (GREEN+"Data Cleaned and Saved 1" + RESET)

**Output:**



**Graphical Representation:**

****