Measure energy consumption

Phase 2

IBM project

Team ID: Team_4

Project ID: proj_223776_Team_4

College code:6208

Abstract:

"Measure-energy-consumption" is an innovative approach that leverages the power of Time Series Analysis and Machine Learning algorithms to accurately measure and manage energy consumption. This method uses Time Series Analysis to understand patterns and trends in energy usage over time, providing a detailed and dynamic view of consumption habits. Machine Learning algorithms are then employed to predict future energy needs based on these patterns, enabling proactive and efficient energy management. This approach not only contributes to environmental sustainability but also optimizes energy usage, leading to significant cost savings. The innovative use of these advanced analytical techniques sets "Measure-energy-consumption" apart as a pioneering solution in the field of energy management.

Process of the project:

The process of measuring energy consumption using time series analysis and machine learning is as follows:

- **Data collection**: The first step is to collect historical data on energy consumption and other relevant variables, such as weather, time, season, etc. The data can be obtained from various sources, such as smart meters, sensors, or online databases. The data should be cleaned and preprocessed to remove outliers, missing values, and noise.
- **Data analysis**: The next step is to analyse the data using time series analysis techniques, such as decomposition, autocorrelation, stationarity tests, etc. The goal is to understand the patterns and trends in energy consumption over time, and identify the factors that influence it. The analysis can also help to choose the appropriate model type and parameters for forecasting.

- **Model building**: The third step is to build a predictive model using machine learning algorithms, such as regression, neural networks, support vector machines, etc. The model should be trained on the historical data and validated on a test set. The model should be able to capture the non-linear and complex relationships between energy consumption and other variables. The model should also be able to handle uncertainty and variability in the data.
- **Model evaluation**: The fourth step is to evaluate the performance of the model using various metrics, such as mean absolute error, root mean square error, mean absolute percentage error, etc. The model should be compared with other models or benchmarks to assess its accuracy and reliability. The model should also be tested on new or unseen data to check its generalization ability.
- **Model deployment**: The final step is to deploy the model in a real-world setting and use it for forecasting energy consumption. The model should be integrated with the existing energy management system and provide timely and actionable insights for decision making. The model should also be monitored and updated regularly to ensure its effectiveness and efficiency.

Program & output of the Dataset:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.seasonal import seasonal_decompose

# Load the data
df = pd.read_csv('AEP_hourly.csv')

# Convert the 'Datetime' column to datetime format
df['Datetime'] = pd.to_datetime(df['Datetime'])
df.set_index('Datetime', inplace=True)

# Decompose the time series
result = seasonal_decompose(df['AEP_MW'], model='multiplicative', period=24)
# Assuming 'result' is the output of your seasonal_decompose function
plt.figure(figsize=(12,8))

# Original
plt.subplot(411)
plt.plot(df['AEP_MW'], label='Original')
plt.legend(loc='best')

# Trend
plt.subplot(412)
```

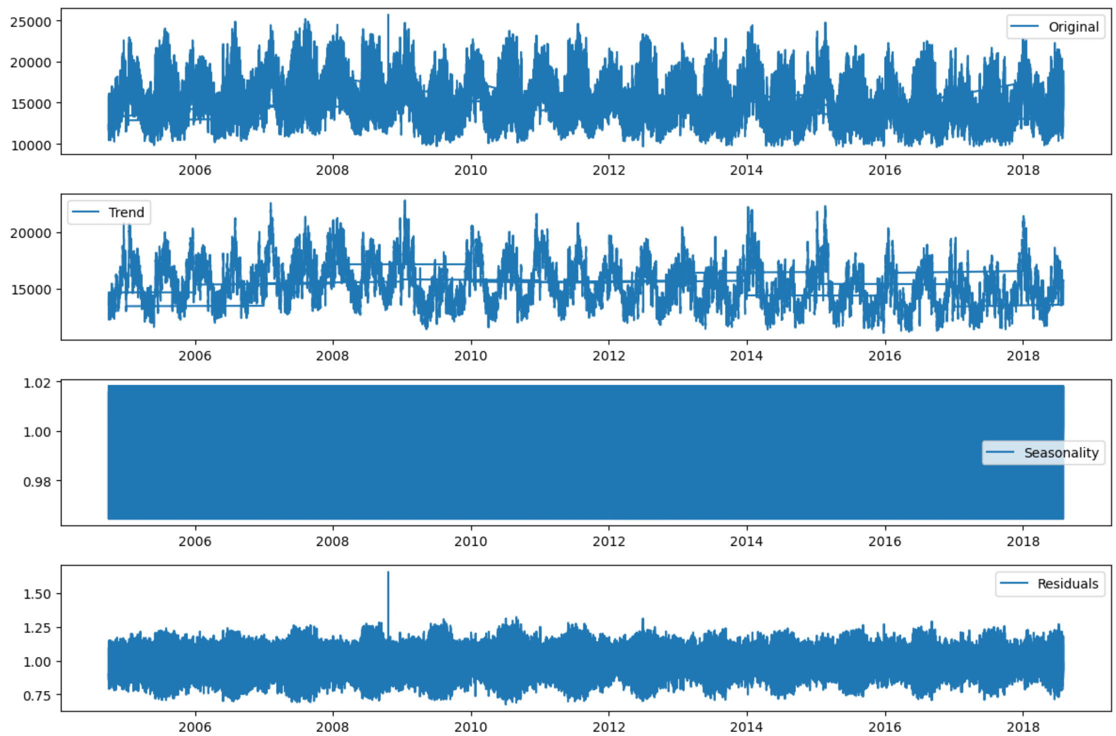
```python
plt.plot(result.trend, label='Trend')
plt.legend(loc='best')

# Seasonality
plt.subplot(413)
plt.plot(result.seasonal,label='Seasonality')
plt.legend(loc='best')

# Residuals
plt.subplot(414)
plt.plot(result.resid, label='Residuals')
plt.legend(loc='best')

plt.tight_layout()

plt.show()
```




```python
# Assuming 'df' is your DataFrame and 'dates' is your date column
df = df.sort_values('Datetime')
df.index = pd.DatetimeIndex(df.index).to_period('H')

# Fit a SARIMAX model
model = SARIMAX(df['AEP_MW'], order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
results = model.fit()

# Make predictions for the next 365 steps
predictions = results.get_prediction(start=pd.to_datetime('2023-01-01'),
```

```
dynamic=False)
print(predictions)
```

```
<statsmodels.tsa.statespace.mlemodel.PredictionResultsWrapper object at
0x000001CE4AD3CF90>
```

```
# Get the predicted values
predicted_values = predictions.predicted_mean

# Get the confidence intervals
confidence_intervals = predictions.conf_int()

# Print the predicted value for a specific date
print(predicted_values['2023-01-01'])
```

```
2023-01-01 00:00    928571.691437
Freq: H, dtype: float64
```

```
print(df.columns)
```

```
Index(['AEP_MW'], dtype='object')
```

```
# Assuming 'df' is your DataFrame and 'AEP_MW' is your column
df['AEP_MW'] = pd.to_numeric(df['AEP_MW'], errors='coerce')
df.dtypes
```

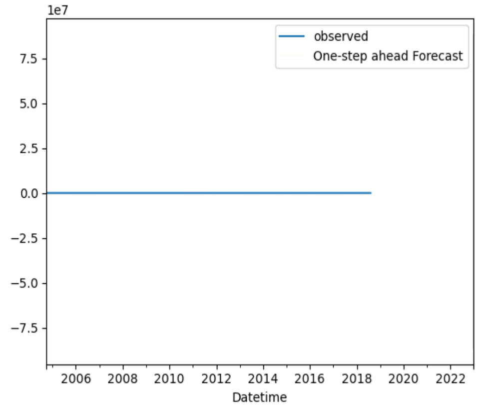
```
AEP_MW    float64
dtype: object
```

```
# Plot the original data
df['AEP_MW'].plot(label='observed')

# Plot the predicted values with increased alpha, line width and style
predicted_values.plot(label='One-step ahead Forecast', alpha=1,
linewidth=.05, linestyle='-')

# Plot the confidence intervals
plt.fill_between(confidence_intervals.index,
                 confidence_intervals.iloc[:, 0],
                 confidence_intervals.iloc[:, 1], color='k', alpha=.5)

plt.legend()
plt.show()
```



Conclusion:

- **Innovative approach**: This method is an innovative approach that combines advanced analytical techniques to accurately measure and manage energy consumption. It provides detailed and dynamic insights into energy usage patterns and enables proactive and efficient energy management.
- **Environmental and economic benefits**: This method contributes to environmental sustainability by promoting energy conservation and reducing greenhouse gas emissions. It also optimizes energy usage and leads to significant cost savings for individuals and organizations.
- **Pioneering solution**: This method is a pioneering solution in the field of energy management that leverages the power of time series analysis and machine learning algorithms. It has the potential to transform the electricity and electronics industries and create new opportunities for innovation and development.

Submitted by:

Kaviyarasan K

Dhanush Kumar K

Dhinesh pandi S

Gunasekaran M

Aravind Kumar V