Reg no :721921243049

Name: Hariprasath.R

Project: ***Measure Energy Consumption***

Introduction:

- Energy consumption in the body is a product of the basal metabolic rate and the physical activity level.
- The physical activity level are defined for a non-pregnant, non-lactating adult as that person's total energy expenditure (TEE) in a 24-hour period, divided by his or her basal metabolic rate (BMR)

Objective:

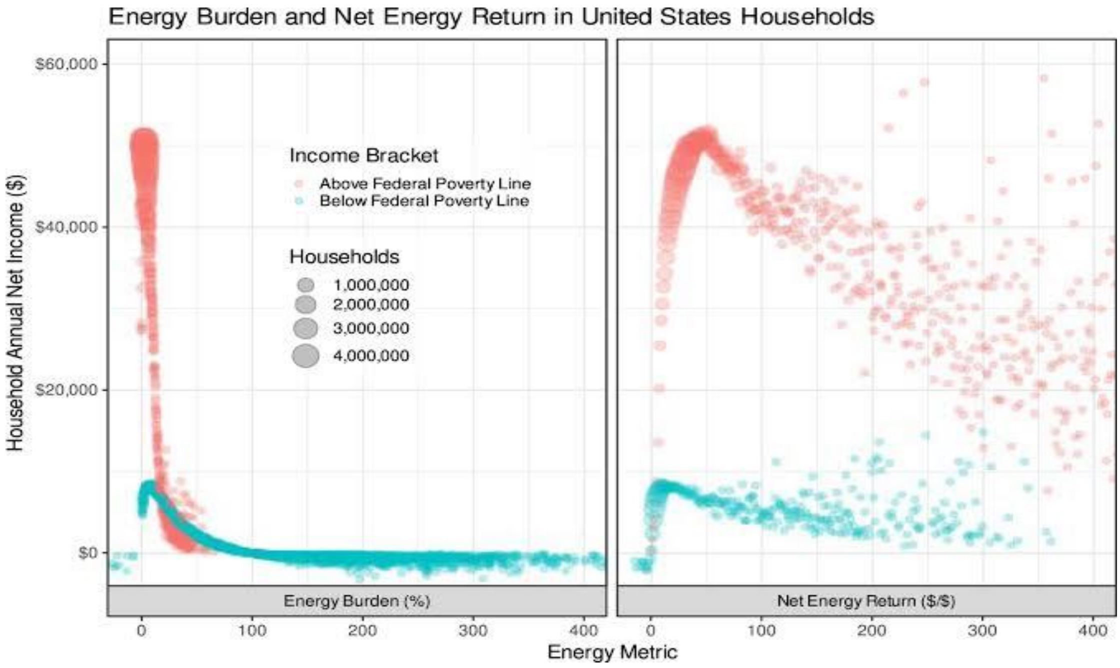
- Initiate the development process by selecting a suitable dataset and preparing it for analysis.
- In PHASE 1 we discussed about the problem definition and their application used in artificial intelligence.
- In phase 2 we discussed about the explore innovative technologies used in artificial intelligence.

PHASE 3:

- To initiate the development process for measuring energy consumption, you'll need to follow these steps:

1.***Define the Problem***:

- Clearly define the goal of your energy consumption analysis.
- What are you trying to achieve, and what insights are you seeking.



2. ***Select a Dataset***:

- Choose a dataset that contains relevant information for your analysis.
- You may find energy consumption data from sources like utility companies, government agencies, or research organizations.
- Ensure the dataset is up-to-date and covers the necessary variables, such as time, location, and energy usage.

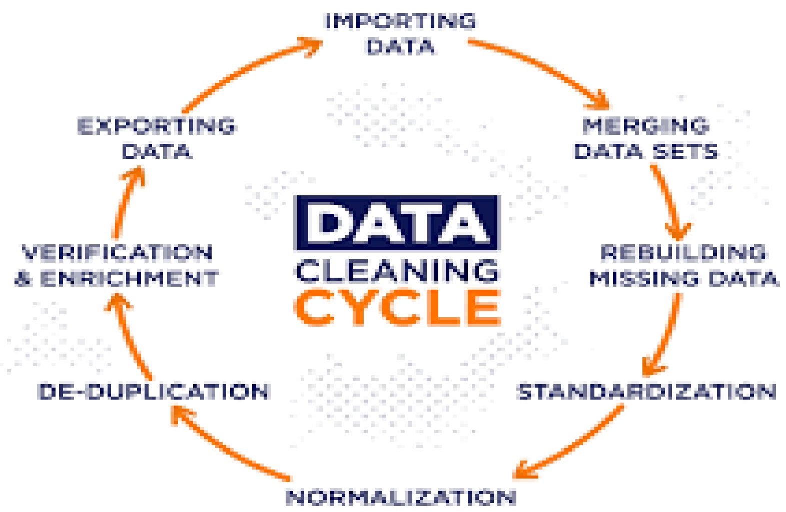
3.***Data Collection***:

- Acquire the selected dataset.
- This may involve downloading it from a website, requesting it from the source, or even collecting data directly if you have the means to do so.
- Here the measure energy dataset link is given below.

  https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

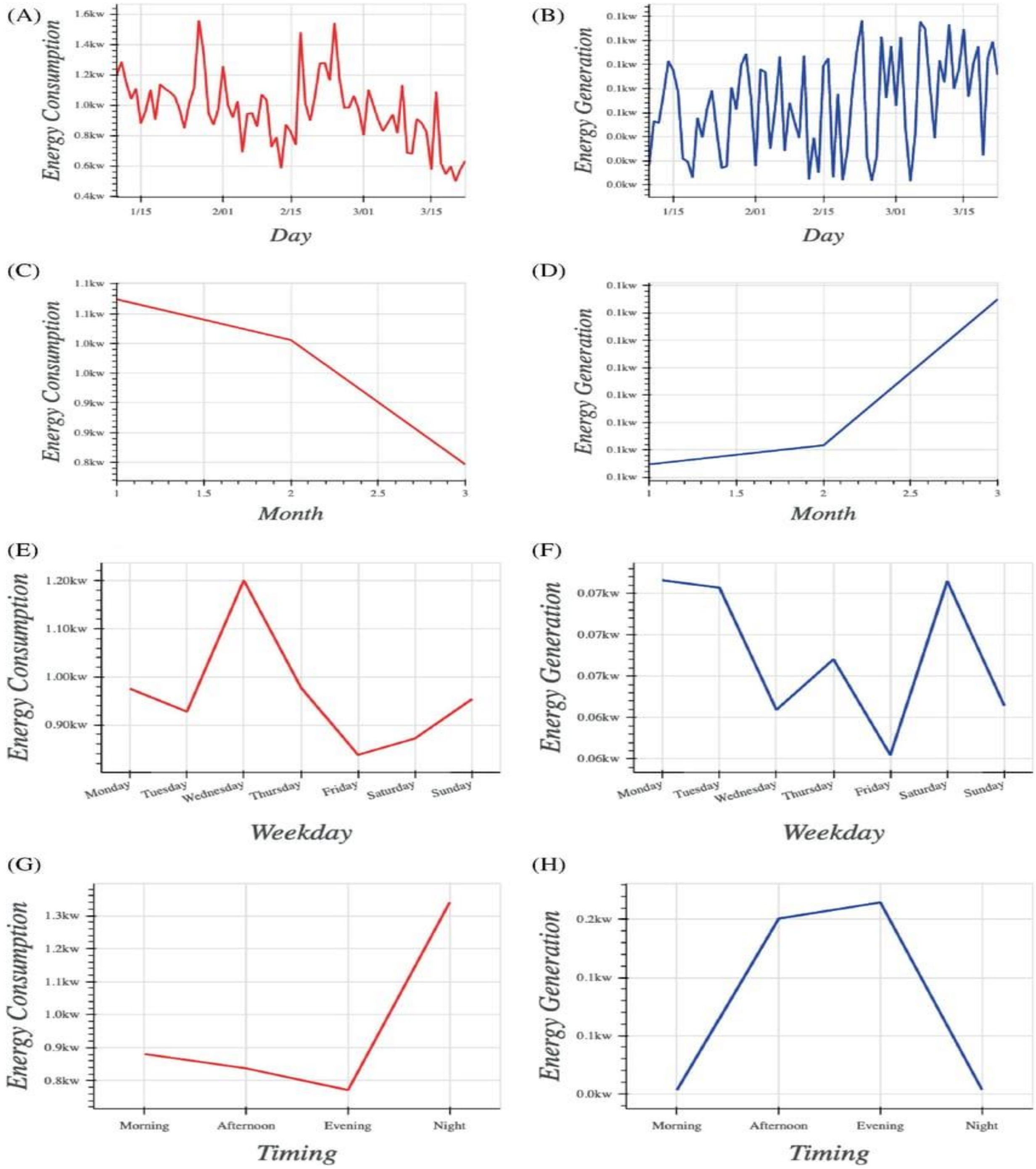
4.***Data Cleaning***:

- Prepare the dataset by cleaning it. This includes handling missing values, correcting errors, and ensuring data consistency.
- Ensure that the data is in a format that you can work with, such as CSV, Excel, or a database.



5.***Exploratory Data Analysis (EDA):***

- Perform EDA to understand the dataset better.
- This involves creating visualizations, summarizing statistics, and identifying patterns or anomalies in the data.
- Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.
- Visualize trends, seasonality, and correlations between energy consumption and other variables.

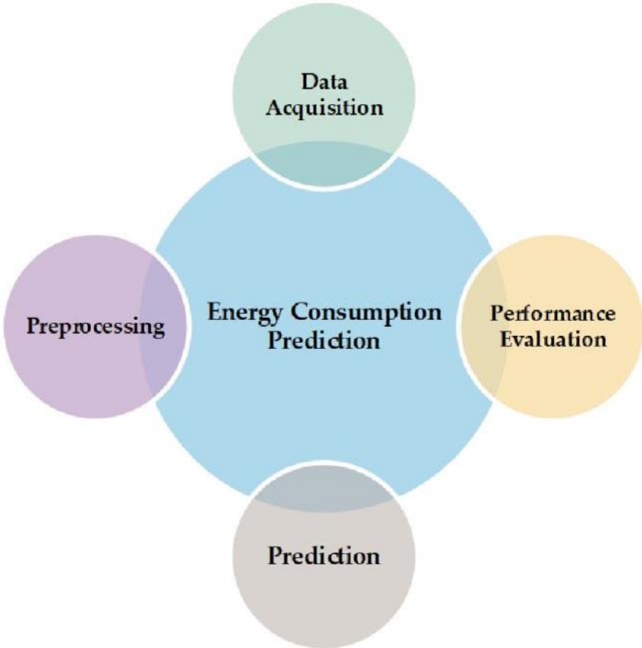


6.***Data Preprocessing***:

- This step involves transforming and preparing the data for analysis.
- You may need to normalize, scale, or engineer features to make them suitable for modeling.

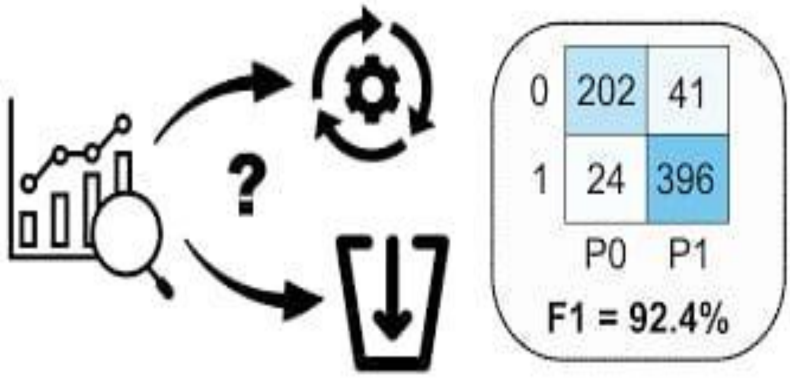
- There are four types of data processing

1. Data cleaning
2. Data integration
3. Data transformation
4. Data reduction



7.***Model development***:

- Energy consumption modeling seeks to determine energy requirements as a function of input parameters.
- Models may be used for determining the requirements of energy supply and the consumer consumption variations while an upgrade or addition of technology exist.
- Use appropriate metrics such as
  I. Mean Absolute Error (MAE)
  II. Mean Squared Error (MSE)
  III. Root Mean Squared Error (RMSE)



8.Evaluation:

- Assess the performance of your models using appropriate metrics.
- This could include measures like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).

9. Select Analytical Methods:

- Choose the analytical methods and algorithms that are appropriate for measuring energy consumption.
- This could involve regression analysis, time series analysis, or machine learning techniques.

10. Interpret Results:

- Interpret the results of your analysis.
- What do your models and metrics reveal about energy consumption patterns?

11.Visualization:

- Create visualizations to communicate your findings effectively.
- Visual aids can be instrumental in conveying your results to stakeholders.

12.Documentation and Reporting:

- Document your process, results, and insights in a clear and concise manner.
- This documentation will be crucial for presenting your findings and for future reference.

13.Continual Improvement:

- Energy consumption patterns can change over time.
- Consider setting up a system for continual monitoring and analysis to stay up-to-date with the latest trends.

Source code:

Int[1]:

```
Import numpy as np
Import pandas as pd
Import matplotlib.pyplot as plt
Import matplotlib.dates as mdates
%matplotlib inline
Import seaborn as sns
Import warnings
Warnings.filterwarnings("ignore")
From pandas.plotting import lag_plot
From pylab import rcParams
```

From statsmodels.tsa.seasonal import seasonal_decompose

From pandas import DataFrame

From pandas import concat

Int[2]:

Df=pd.read_csv(“../input/hourly-energy-consumption/AEP_hourly.csv”,index_col=’Datetime’,parse_dates=True)

Df.head()

Out[2]:

	AEP_MW
Datetime	
2004-12-31 01:00:00	**13478.0**
2004-12-31 02:00:00	**12865.0**
2004-12-31 03:00:00	**12577.0**
2004-12-31 04:00:00	**12517.0**
2004-12-31 05:00:00	**12670.0**

Int[3]:

```
df.sort_values(by='Datetime', inplace=True)
print(df)
```

Int[4]:

```
df.shape
```

Out[4]:

```
(121273, 1)
```

Int[5]:

```
df.info()
```

Out[5]:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 121273 entries, 2004-10-01 01:00:00 to 2018-08-03 00:00:00
Data columns (total 1 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   AEP_MW  121273 non-null  float64
dtypes: float64(1)
memory usage: 1.9 MB
```

Int[6]:

df.describe()

Out[6]:

	AEP_MW
count	121273.000000
mean	15499.513717
std	2591.399065
min	9581.000000
25%	13630.000000
50%	15310.000000
75%	17200.000000
100%	25695.000000

Int[7]:

```
df.index = pd.to_datetime(df.index)
```

Int[8]:

```
# Extract all Data Like Year MOnth Day Time etc
df["Month"] = df.index.month
df["Year"] = df.index.year
```

```
df["Date"] = df.index.date
df["Hour"] = df.index.hour
df["Week"] = df.index.week
df["Day"] = df.index.day_name()
df.head()
```

Out[8]:

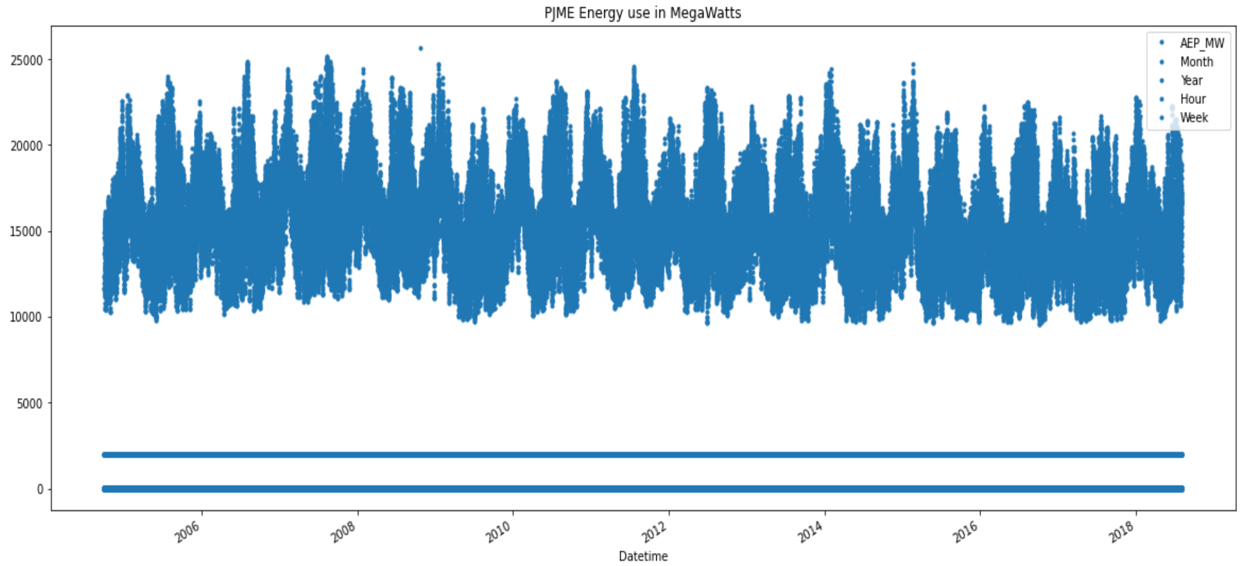
	AEP_MW	Month	Year	Date	Hour	Week	Day
Datetime							
2004-10-01 01:00:00	12379.0	10	2004	2004-10-01 1	1	4	Friday
2004-10-01 02:00:00	11935.0	10	2004	2004-10-01 1	2	4	Friday

2004-10-01 03:00:00	**11692.0**	**10**	**2004**	**2004-10-01 1**	**3**	**4**	**Friday**
2004-10-01 04:00:00	**11597.0**	**10**	**2004**	**2004-10-01 1**	**4**	**4**	**Friday**
2004-10-01 05:00:00	**05:00:00**	**10**	**2004**	**2004-10-01 1**	**5**	**4**	**Friday**

Int[9]:

```
df.plot(title="PJME Energy use in MegaWatts",
        figsize=(20, 8),
        style=".",
        color=sns.color_palette()[0])

plt.show()
```

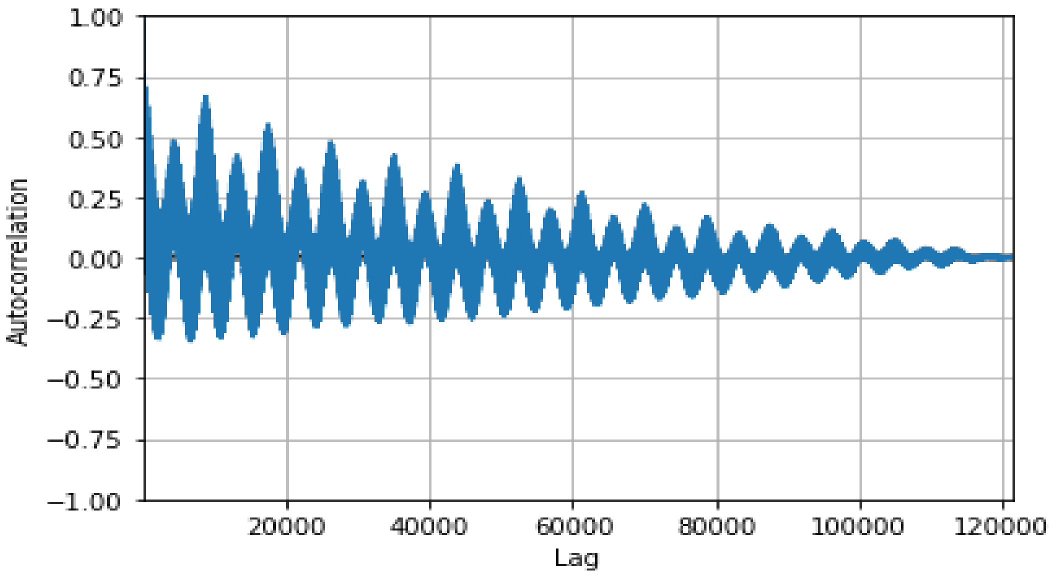
Out[9]:



Int[10]:

```
from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df['AEP_MW'])
plt.show()
```

Out[10]:



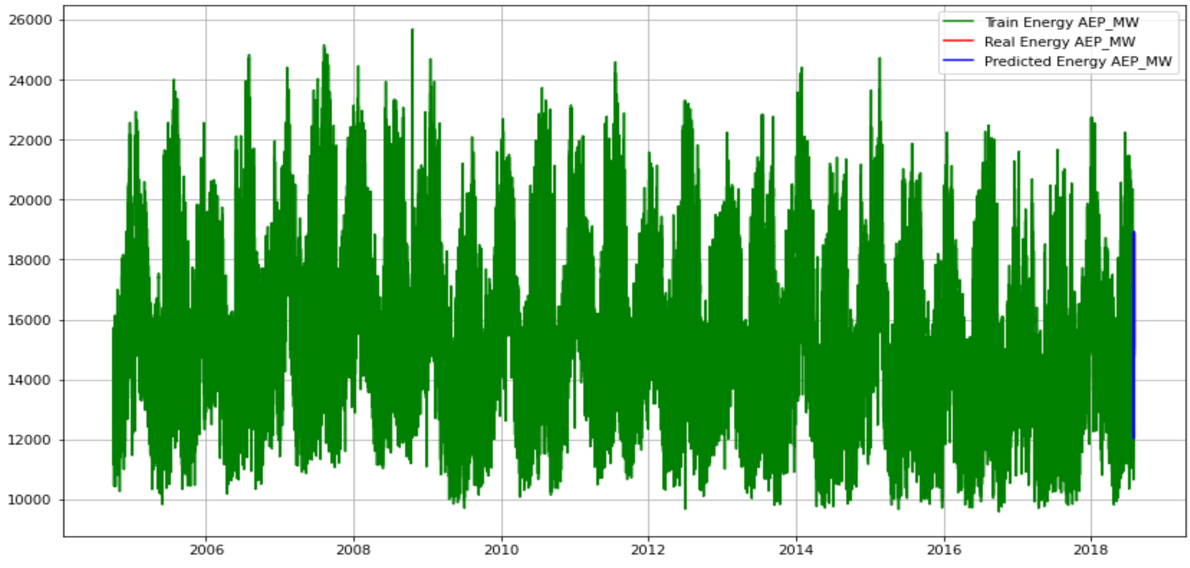
Int[11]:

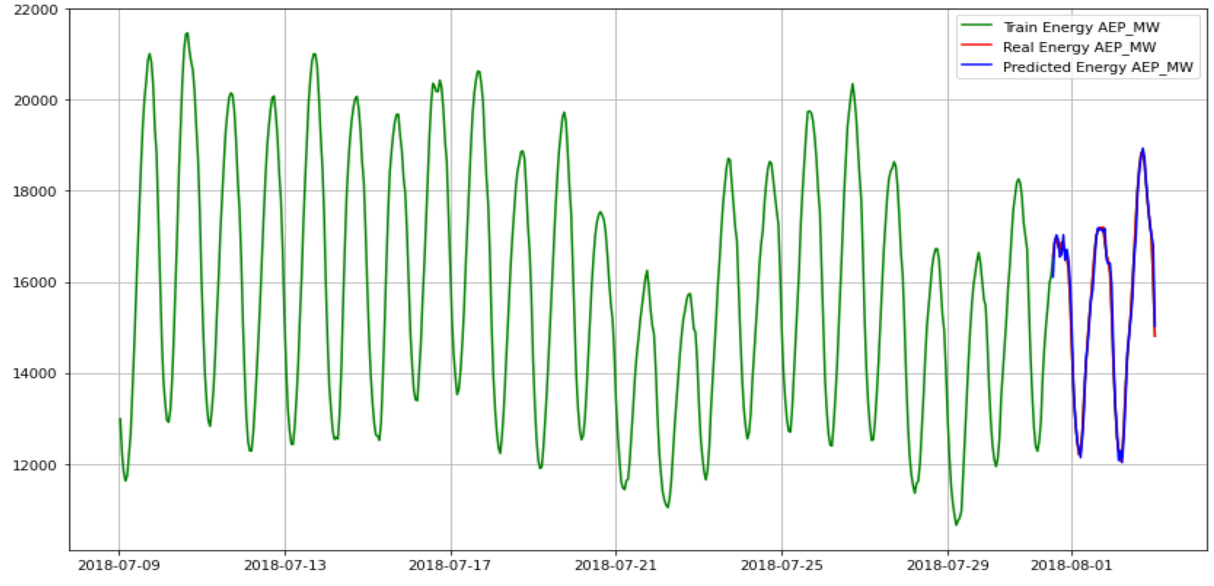
```
#Train Arima Model
train_arima = train_data['AEP_MW']
test_arima = test_data['AEP_MW']

history = [x for x in train_arima]
y = test_arima
# make first prediction
predictions = list()
model = sm.tsa.arima.ARIMA(history,
order=(5,1,0))
model_fit = model.fit()
yhat = model_fit.forecast()[0]
predictions.append(yhat)
history.append(y[0])
# rolling forecasts
for i in range(1, len(y)):
    # predict
    model = sm.tsa.arima.ARIMA(history,
order=(5,1,0))
    model_fit = model.fit()
    yhat = model_fit.forecast()[0]
    # invert transformed prediction
```

```
    predictions.append(yhat)
    # observation
    obs = y[i]
    history.append(obs)

plt.figure(figsize=(14,8))
plt.plot(df.index, df['AEP_MW'], color='green', label = 'Train Energy AEP_MW')
plt.plot(test_data.index, y, color = 'red', label = 'Real Energy AEP_MW')
plt.plot(test_data.index, predictions, color = 'blue', label = 'Predicted Energy AEP_MW')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(14,8))
plt.plot(df.index[-600:], df['AEP_MW'].tail(600), color='green', label = 'Train Energy AEP_MW')
plt.plot(test_data.index, y, color = 'red', label = 'Real Energy AEP_MW')
plt.plot(test_data.index, predictions, color = 'blue', label = 'Predicted Energy AEP_MW')
```

```
plt.legend()
plt.grid(True)
plt.show()

print('MSE: '+str(mean_squared_error(y, predictions)))
print('MAE: '+str(mean_absolute_error(y, predictions)))
print('RMSE: '+str(sqrt(mean_squared_error(y, predictions))))
```

Out[11]:





Train Energy AEP_MW
Real Energy AEP_MW
Predicted Energy AEP_MW
22000
20000
18000
16000
14000
12000
2018-07-09
2018-07-13
2018-07-17
2018-07-21
2018-07-25
2018-07-29
2018-08-01

Conclusion:

- **Remember that the accuracy of your predictions will depend on the quality and quantity of data, as well as the choice of the most appropriate modeling techniques.**
- **Regularly updating the model with new data is crucial to ensure it remains accurate as consumption patterns evolve.**