MEASURE ENERGY CONSUMPTION

NAME:S.SUGANTHAN-82262104305

EMAIL:ssuganthan370@gmail.com

Phase 3 submission document

Project Title: MEASURE ENERGY CONSUMPTION

MEASURE ENERGY CONSUMPTION

INTRODUCTION:

Measuring energy consumption is a crucial aspect of understanding and managing our energy usage, which has significant implications for sustainability, cost management, and environmental impact. Energy consumption refers to the amount of energy used by various devices, systems, or processes in a given period. This energy can come from various sources, such as electricity, natural gas, gasoline, or other fuels. The practice of measuring energy consumption is essential for several reasons:

1. **Resource Management:** Efficient energy management is crucial for optimizing the use of finite resources, such as fossil fuels, and minimizing waste. Measuring energy consumption helps identify areas where energy is being used inefficiently, leading to potential savings.

2. **Cost Reduction:** Energy costs represent a significant portion of the operating expenses for individuals, businesses, and governments. By measuring energy consumption, you can pinpoint areas of high usage and take steps to reduce costs, whether through energy-efficient technology adoption or behavioral changes.

3. **Environmental Impact:** High energy consumption is often associated with increased greenhouse gas emissions and environmental degradation. Accurate measurement is a prerequisite for making informed decisions to reduce the environmental impact of energy use, such as transitioning to cleaner energy sources and adopting sustainable practices.

4. **Energy Efficiency**: Measuring energy consumption is fundamental for tracking the effectiveness of energy efficiency initiatives and retrofits. It provides the data necessary to assess the success of energy-saving projects and improvements.

5. **Compliance and Regulation**: Many regions have regulations and standards in place to monitor and limit energy consumption, especially in sectors like industry and construction. Measuring energy usage is crucial to ensure compliance with these regulations.

6. **Billing and Allocation**: For utility providers, measuring energy consumption is essential for billing customers accurately. In multi-unit buildings, accurate measurement helps allocate costs fairly among residents or tenants.

To measure energy consumption effectively, various tools and methods can be employed. These include smart meters, sub-metering systems, energy monitoring software, and energy audits. Additionally, individuals and organizations can take advantage of technology and data analytics to gain insights into energy usage patterns and make informed decisions to reduce consumption and improve energy efficiency.

Import relevant python packages

Let's use the electrical meter data to create clusters of typical load profiles for analysis. First we can load our conventional packages

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
```

Next let's load all the packages we will need for analysis

```
import sklearn
from sklearn import metrics
from sklearn.neighbors import KNeighborsRegressor
from scipy.cluster.vq import kmeans, vq, whiten
from scipy.spatial.distance import cdist
import numpy as np
from datetime import datetime
```

Electricity Prediction for Measurement and Verification

Prediction is a common machine learning (ML) technique used on building energy consumption data. This process is valuable for anomaly detection, load profile-based building control and measurement and verification procedures.

The graphic below comes from the IPMVP to show how prediction can be used for M&V to calculate how much energy **would have** been consumed if an energy savings intervention had not been implemented.

Load electricity data and weather data

First we can load the data from the BDG in the same as our previous weather analysis influence notebook from the Construction Phase videos

```
elec_all_data = pd.read_csv("../input/buildingdatagenomeproject2/electricity_cleaned.
csv", index_col='timestamp', parse_dates=True)
```

```
In [4]:
elec_all_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 17544 entries, 2016-01-01 00:00:00 to 2017-12-31 23:00:00
Columns: 1578 entries, Panther_parking_Lorriane to Mouse_science_Micheal
dtypes: float64(1578)
memory usage: 211.3 MB
```

```
buildingname = 'Panther_office_Hannah'
```

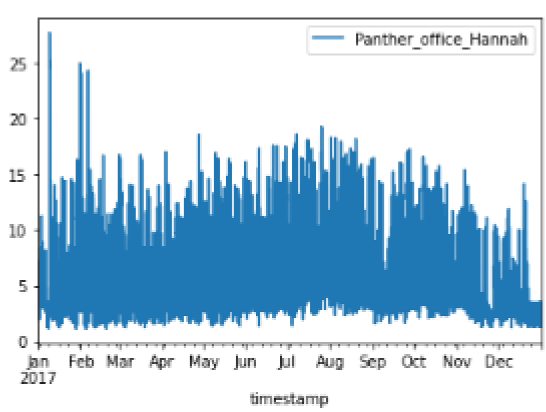
In [6]:
```
office_example_prediction_data = pd.DataFrame(elec_all_data[buildingname].truncate(be
fore='2017-01-01')).fillna(method='ffill')
```

In [7]:
```
office_example_prediction_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8760 entries, 2017-01-01 00:00:00 to 2017-12-31 23:00:00
Data columns (total 1 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Panther_office_Hannah  8760 non-null   float64
dtypes: float64(1)
memory usage: 136.9 KB
```

```
office_example_prediction_data.plot()
```

```
<AxesSubplot:xlabel='timestamp'>
```

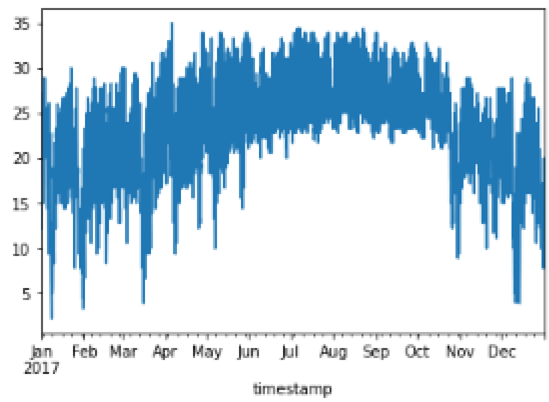


```python
weather_data = pd.read_csv("../input/buildingdatagenomeproject2/weather.csv",
index_col='timestamp', parse_dates=True)
weather_data_site = weather_data[weather_data.site_id == 'Panther'].truncate(
before='2017-01-01')
weather_data_site.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8760 entries, 2017-01-01 00:00:00 to 2017-12-31 23:00:00
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   site_id         8760 non-null   object
 1   airTemperature  8760 non-null   float64
 2   cloudCoverage   5047 non-null   float64
 3   dewTemperature  8760 non-null   float64
 4   precipDepth1HR  8752 non-null   float64
 5   precipDepth6HR  329 non-null    float64
 6   seaLvlPressure  8522 non-null   float64
 7   windDirection   8511 non-null   float64
 8   windSpeed       8760 non-null   float64
dtypes: float64(8), object(1)
memory usage: 684.4+ KB
```

```python
weather_hourly = weather_data_site.resample("H").mean()
weather_hourly_nooutlier = weather_hourly[weather_hourly > -40]
weather_hourly_nooutlier_nogaps = weather_hourly_nooutlier.fillna(method='ffi
ll')
temperature = weather_hourly_nooutlier_nogaps["airTemperature"]
temperature.plot()
```

```
<AxesSubplot:xlabel='timestamp'>
```



Create Train and Test Datasets

The model is given a set of data that will be used to **train** the model to predict a specific objectice. In this case, we will use a few simple time series features as well as outdoor air temperature to predict how much energy a building uses.

For this demonstration, we will use three months of data from April, May, and June to prediction July.

```
training_months = [4,5,6]
test_months = [7]
```

```
trainingdata = office_example_prediction_data[office_example_prediction_data.
index.month.isin(training_months)]
testdata = office_example_prediction_data[office_example_prediction_data.inde
x.month.isin(test_months)]
trainingdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2184 entries, 2017-04-01 00:00:00 to 2017-06-30 23:00:00
Data columns (total 1 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Panther_office_Hannah  2184 non-null   float64
dtypes: float64(1)
memory usage: 34.1 KB
```

Encoding Categorical Variables

We use the pandas .get_dummies() function to change the temporal variables of *time of day* and *day of week* into categories that the model can use more effectively. This process is known as encoding.

```
train_features = pd.concat([pd.get_dummies(trainingdata.index.hour),
                                     pd.get_dummies(trainingdata.index.dayofw
eek),
                                     pd.DataFrame(temperature[temperature.ind
ex.month.isin(training_months)].values)], axis=1).dropna()
train_features.head()
```

	0	1	2	3	4	5	6	7	8	9	...	22	23	0	1	2	3	4	5	6	0
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	21.7
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	21.0
2	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	18.9
3	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	20.6
4	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	21.0

5 rows × 32 columns

Train a K-Neighbor Model

This model was chosen after following the process in the cheat sheet until a model that worked and provided good results was found.

```
model = KNeighborsRegressor().fit(np.array(train_features), np.array(training
data.values));
```

linkcode

```
test_features = np.array(pd.concat([pd.get_dummies(testdata.index.hour),
                                    pd.get_dummies(testdata.index.dayofweek),
                                    pd.DataFrame(temperature[temperature.inde
x.month.isin(test_months)].values)], axis=1).dropna())
```

Use the Model to predict for the *Test* period

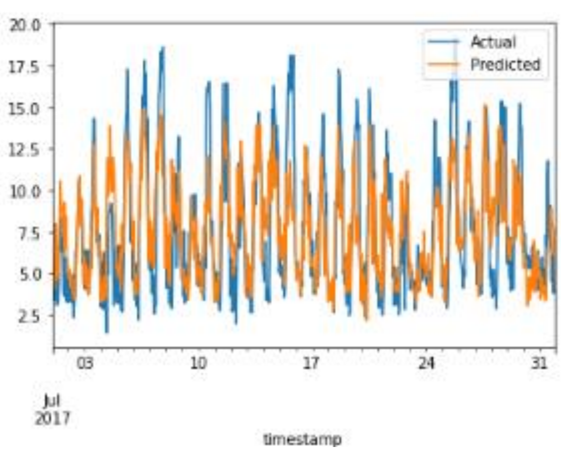
Then the model is given the test_features from the period which we want to predict. We can then merge those results and see how the model did

```
predictions = model.predict(test_features)
predicted_vs_actual = pd.concat([testdata, pd.DataFrame(predictions, index=testdata.i
ndex)], axis=1)
predicted_vs_actual.columns = ["Actual", "Predicted"]
predicted_vs_actual.head()
```

	Actual	Predicted
timestamp		
2017-07-01 00:00:00	5.3370	5.75910
2017-07-01 01:00:00	3.8547	6.02898
2017-07-01 02:00:00	5.5751	4.39686
2017-07-01 03:00:00	4.1248	4.23180
2017-07-01 04:00:00	3.3497	4.03858

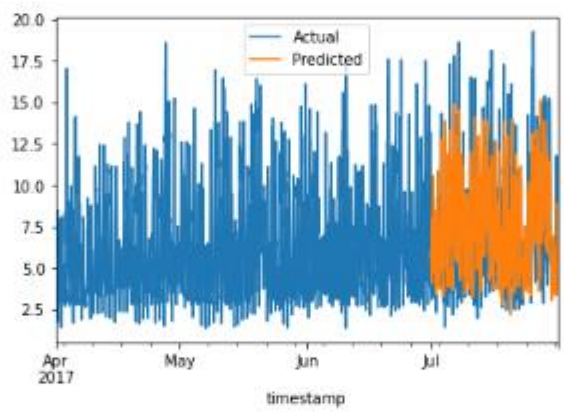
```
predicted_vs_actual.plot()
```

```
<AxesSubplot:xlabel='timestamp'>
```



```
trainingdata.columns = ["Actual"]
predicted_vs_actual_plus_training = pd.concat([trainingdata, predicted_vs_act
ual], sort=True)
predicted_vs_actual_plus_training.plot()
```

```
<AxesSubplot:xlabel='timestamp'>
```



Evaluation metrics

In order to understand quantitatively how the model performed, we can use various evaluation metrics to understand how well the model compared to reality.

In this situation, let's use the error metric Mean Absolute Percentage Error (MAPE)

```
# Calculate the absolute errors
errors = abs(predicted_vs_actual['Predicted'] - predicted_vs_actual['Actual'])
# Calculate mean absolute percentage error (MAPE) and add to list
MAPE = 100 * np.mean((errors / predicted_vs_actual['Actual']))
MAPE
```

```
34.22379683897996
```

```
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("ignor
e") from pandas.plotting
import lag_plotfrom pylab
import rcParams
from statsmodels.tsa.seasonal import
seasonal_decomposefrom pandas import DataFrame
from pandas import
linkcode
df=pd.read_csv("../input/hourly-energy-
consumption/AEP_hourly.csv",index_col='Datetime',parse_dates=True)
df.head()
```

output:

	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

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

```
                     AEP_MW
Datetime
2004-10-01 01:00:00  12379.0
2004-10-01 02:00:00  11935.0
2004-10-01 03:00:00  11692.0
2004-10-01 04:00:00  11597.0
2004-10-01 05:00:00  11681.0
...                      ...
2018-08-02 20:00:00  17673.0
2018-08-02 21:00:00  17303.0
2018-08-02 22:00:00  17001.0
2018-08-02 23:00:00  15964.0
2018-08-03 00:00:00  14809.0

[121273 rows x 1 columns]
```

```python
df.shape()
```

```
(121273, 1)
```

```python
df.info()
```

```
<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
```

```
df.describe()
```

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

Conclusion:

In conclusion, measuring energy consumption is a vital practice that empowers us to make informed decisions about energy usage, reduce costs, mitigate environmental impact, and contribute to a more sustainable future. Whether at the individual, industrial, or governmental level, understanding energy consumption is a critical step toward responsible and efficient energy management.