NAAN MUTHALVAN

IBM COLLABARATE

ARTIFICIAL INTELLIGENCE

PROJECT TITLE

MEASURE ENERGY CONSUMPTION

NAME : MUTHULAKSHMI.R
DEPT & YEAR : CSE & III yr
REG.NO : 712221104009
COLLEGE : PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

DATA ANALYSIS AND PREPROCESSING

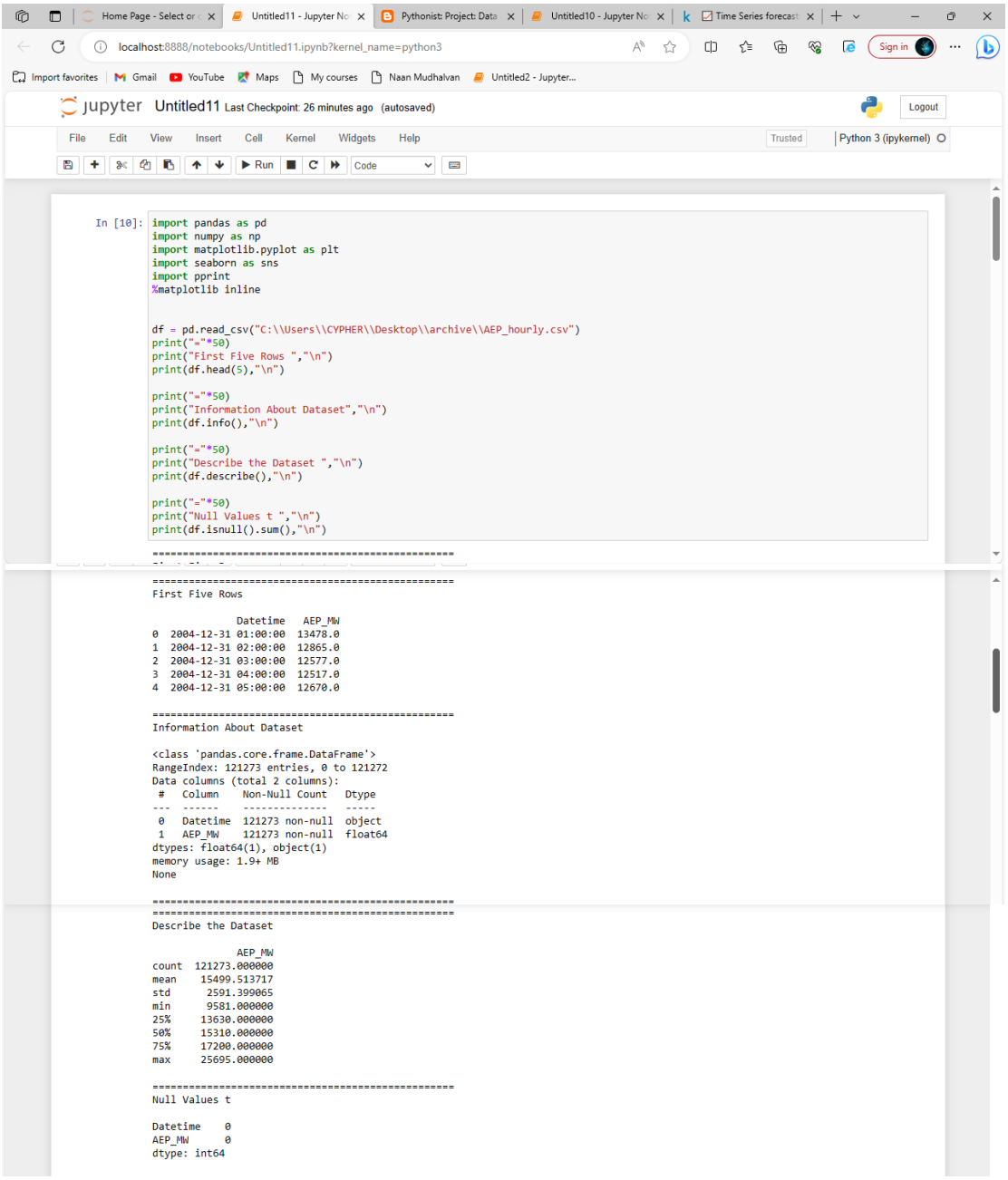
STEP 1:

Import library

Code :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pprint
%matplotlib inline

df = pd.read_csv("AEP_hourly.csv")
print("="*50)
print("First Five Rows ","\n")
print(df.head (5),"\n")
print("="*50)
print("Information About Dataset","\n")
print(df.info(),"\n")
print("="*50)
print("Describe the Dataset ","\n")
print(df.describe (),"\n")
print("="*50)
print("Null Values t ","\n")
print(df.isnull ().sum (),"\n")
```

Output :


Untitled11 - Jupyter No
localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3
jupyter Untitled11 Last Checkpoint: 26 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [10]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pprint
%matplotlib inline
df = pd.read_csv("C:\\Users\\CYPHER\\Desktop\\archive\\AEP_hourly.csv")
print("="*50)
print("First Five Rows ","\n")
print(df.head(5),"\n")
print("="*50)
print("Information About Dataset","\n")
print(df.info(),"\n")
print("="*50)
print("Describe the Dataset ","\n")
print(df.describe(),"\n")
print("="*50)
print("Null Values t ","\n")
print(df.isnull().sum(),"\n")
==
First Five Rows
Datetime AEP_MW
0 2004-12-31 01:00:00 13478.0
1 2004-12-31 02:00:00 12865.0
2 2004-12-31 03:00:00 12577.0
3 2004-12-31 04:00:00 12517.0
4 2004-12-31 05:00:00 12670.0
==
Information About Dataset
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121273 entries, 0 to 121272
Data columns (total 2 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Datetime 121273 non-null object
1 AEP_MW 121273 non-null float64
dtypes: float64(1), object(1)
memory usage: 1.9+ MB
None
==
==
Describe the Dataset
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
==
Null Values t
Datetime 0
AEP_MW 0
dtype: int64

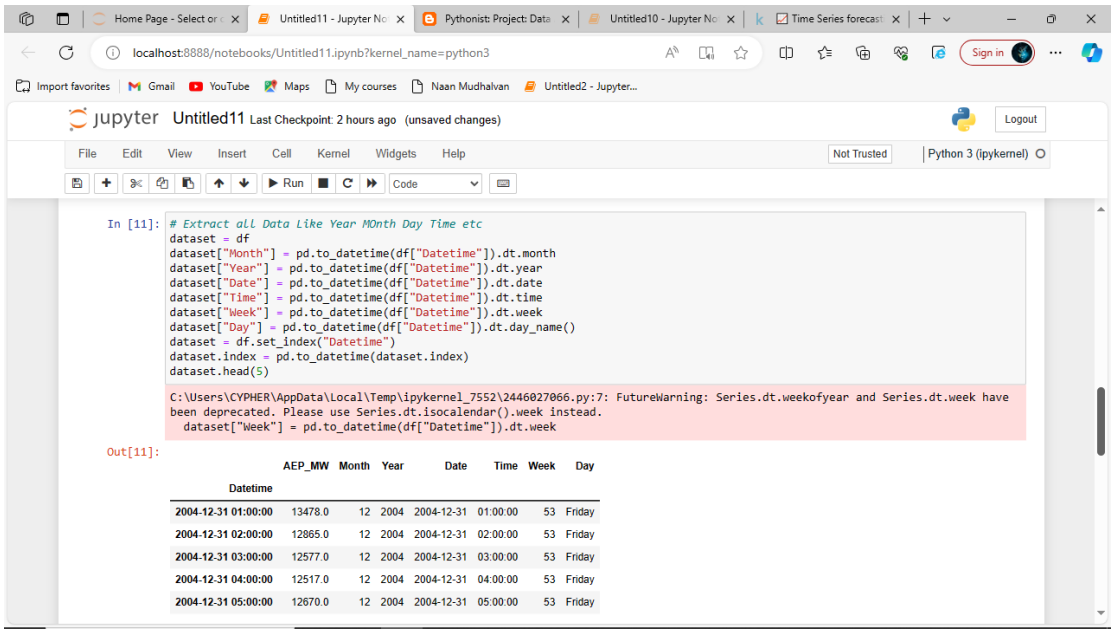
STEP 2:

Reformat the Date Time Columns

Code :

```
# Extract all Data Like Year MOnth Day Time etc
dataset = df
dataset["Month"] = pd.to_datetime(df["Datetime"]).dt.month
dataset["Year"] = pd.to_datetime(df["Datetime"]).dt.year
dataset["Date"] = pd.to_datetime(df["Datetime"]).dt.date
dataset["Time"] = pd.to_datetime(df["Datetime"]).dt.time
dataset["Week"] = pd.to_datetime(df["Datetime"]).dt.week
dataset["Day"] = pd.to_datetime(df["Datetime"]).dt.day_name()
dataset = df.set_index("Datetime")
dataset.index = pd.to_datetime(dataset.index)
dataset.head(5)
```

Output :



STEP 3:

Show the Energy Consumption Each Year

Code :

```
# How many Unique Year do we Have in Dataset
print(df.Year.unique(),"\n")
print("Total Number of Unique Year", df.Year.nunique(), "\n")
```

Output :

Code :

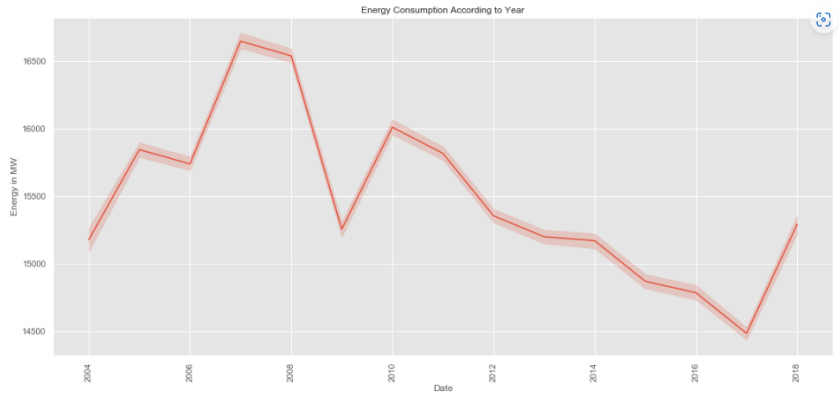
```
from matplotlib import style

fig = plt.figure()
ax1 = plt.subplot2grid((1,1), (0,0))

style.use('ggplot')

sns.lineplot(x=dataset["Year"], y=dataset["AEP_MW"], data=df)
sns.set(rc={'figure.figsize':(15,6)})

plt.title("Energy consumptionnin Year 2004")
plt.xlabel("Date")
plt.ylabel("Energy in MW")
plt.grid(True)
plt.legend()
```

```
for label in ax1.xaxis.get_ticklabels():
    label.set_rotation(90)

plt.title("Energy Consumption According to Year")
```

Output :

```
In [304]: from matplotlib import style

          fig = plt.figure()
          ax1 = plt.subplot2grid((1,1), (0,0))

          style.use('ggplot')

          sns.lineplot(x=dataset["Year"], y=dataset["AEP_MW"], data=df)
          sns.set(rc={'figure.figsize':(15,6)})

          plt.title("Energy consumptionnin Year 2004")
          plt.xlabel("Date")
          plt.ylabel("Energy in MW")
          plt.grid(True)
          plt.legend()

          for label in ax1.xaxis.get_ticklabels():
              label.set_rotation(90)


          plt.title("Energy Consumption According to Year")
```

```
Out[304]: Text(0.5, 1.0, 'Energy Consumption According to Year')
```

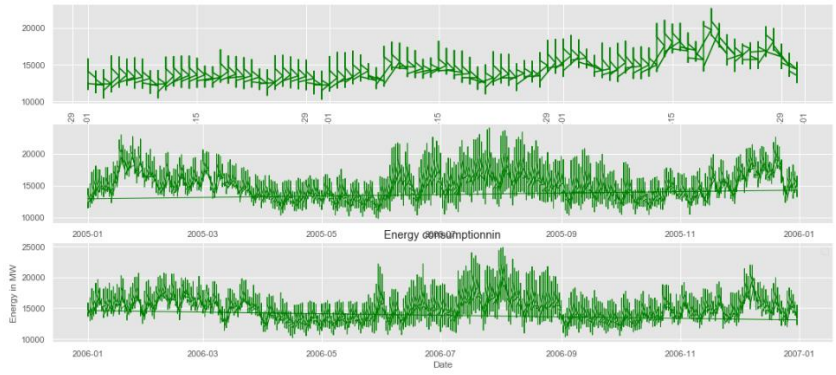


Code :

```
from matplotlib import style

fig = plt.figure()

ax1= fig.add_subplot(311)
ax2= fig.add_subplot(312)
ax3= fig.add_subplot(313)

style.use('ggplot')

y_2004 = dataset["2004"]["AEP_MW"].to_list()
x_2004 = dataset["2004"]["Date"].to_list()
ax1.plot(x_2004,y_2004, color="green", linewidth=1.7)

y_2005 = dataset["2005"]["AEP_MW"].to_list()
x_2005 = dataset["2005"]["Date"].to_list()
ax2.plot(x_2005, y_2005, color="green", linewidth=1)

y_2006 = dataset["2006"]["AEP_MW"].to_list()
x_2006 = dataset["2006"]["Date"].to_list()
ax3.plot(x_2006, y_2006, color="green", linewidth=1)

plt.rcParams["figure.figsize"] = (18,8)
plt.title("Energy consumptionnin")
plt.xlabel("Date")
plt.ylabel("Energy in MW")
plt.grid(True, alpha=1)
plt.legend()

for label in ax1.xaxis.get_ticklabels():
    label.set_rotation(90)
```

Output :

In [339]:
```
from matplotlib import style


fig = plt.figure()

ax1= fig.add_subplot(311)
ax2= fig.add_subplot(312)
ax3= fig.add_subplot(313)


style.use('ggplot')

y_2004 = dataset["2004"]["AEP_MW"].to_list()
x_2004 = dataset["2004"]["Date"].to_list()
ax1.plot(x_2004,y_2004, color="green", linewidth=1.7)


y_2005 = dataset["2005"]["AEP_MW"].to_list()
x_2005 = dataset["2005"]["Date"].to_list()
ax2.plot(x_2005, y_2005, color="green", linewidth=1)


y_2006 = dataset["2006"]["AEP_MW"].to_list()
x_2006 = dataset["2006"]["Date"].to_list()
ax3.plot(x_2006, y_2006, color="green", linewidth=1)



plt.rcParams["figure.figsize"] = (18,8)
plt.title("Energy consumptionnin")
plt.xlabel("Date")
plt.ylabel("Energy in MW")
plt.grid(True, alpha=1)
plt.legend()

for label in ax1.xaxis.get_ticklabels():
    label.set_rotation(90)
```



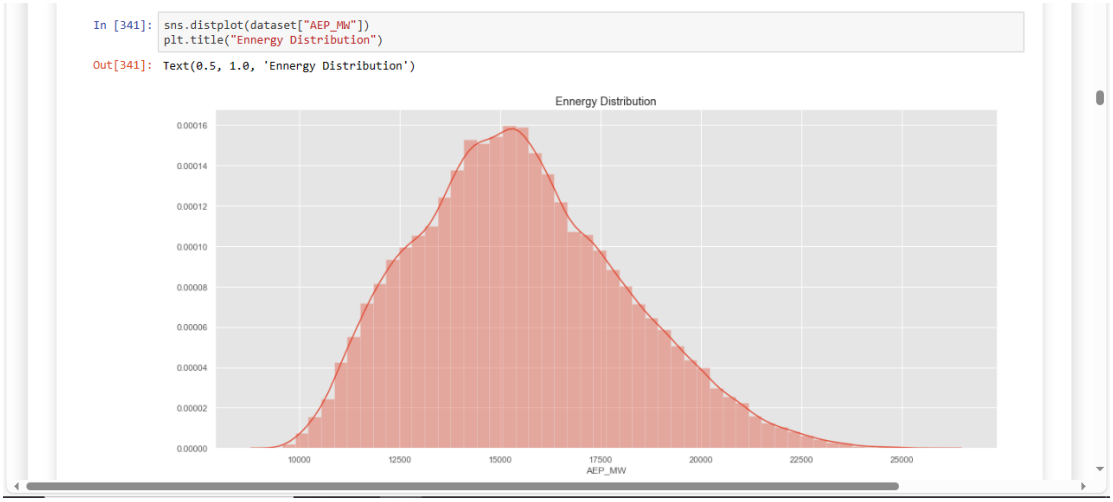
STEP 4:

Energy with Respect To Time

Code :

```
sns.distplot(dataset["AEP_MW"])
plt.title("Energy Distribution")
```

Output :



Code :

```
fig = plt.figure()
ax1= fig.add_subplot(111)

sns.lineplot(x=dataset["Time"],y=dataset["AEP_MW"], data=df)
plt.title("Energy Consumption vs Time ")
plt.xlabel("Time")
plt.grid(True, alpha=1)
plt.legend()

for label in ax1.xaxis.get_ticklabels():
    label.set_rotation(90)
```

Output :