

NAAN MUTHALVAN

IBM COLLABARATE

ARTIFICIAL INTELLIGENCE

PROJECT TITLE

MEASURE ENERGY CONSUMPTION

NAME : PRAVEEN KUMAR M

DEPT & YEAR : CSE & III yr

REG.NO : 712221104014

COLLEGE : PARK CLLEGE 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")
```

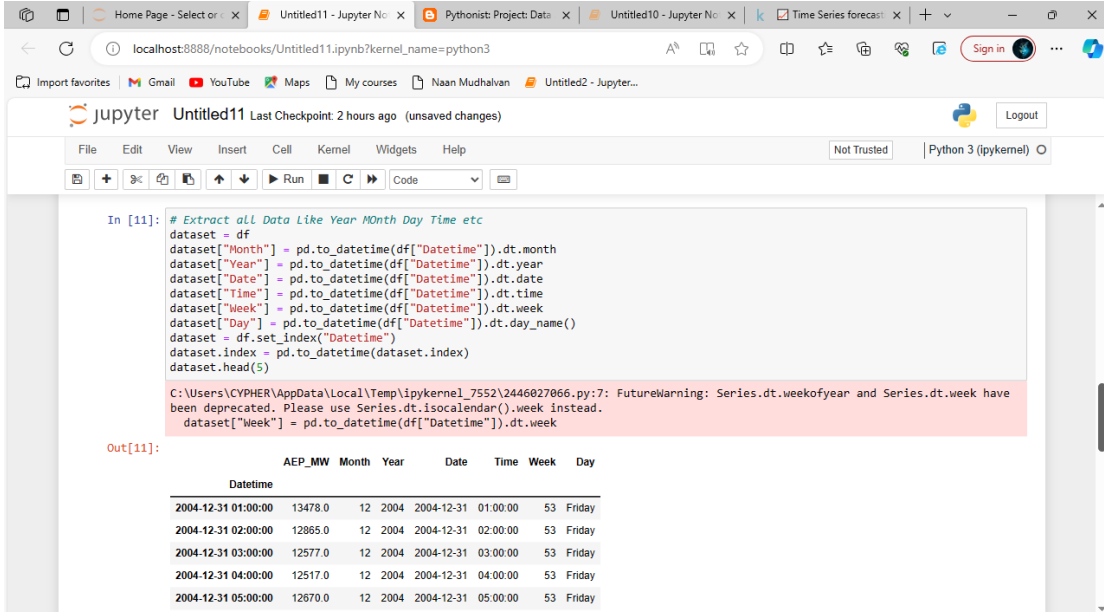

STEP 2:

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 :



The screenshot shows a Jupyter Notebook interface with the following components:

- Browser Tabs:** Home Page - Select or..., Untitled11 - Jupyter No..., Pythonist: Project Data..., Untitled10 - Jupyter No..., Time Series forecast...
- Address Bar:** localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3
- Page Header:** jupyter Untitled11 Last Checkpoint: 2 hours ago (unsaved changes) Logout
- Menu Bar:** File Edit View Insert Cell Kernel Widgets Help
- Toolbar:** Import favorites, Gmail, YouTube, Maps, My courses, Naan Mudhalvan, Untitled2 - Jupyter..., Run, Code
- Code Cell:**

```
In [11]: # 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 Cell:**

Out[11]:

| | AEP_MW | Month | Year | Date | Time | Week | Day |
|---------------------|---------|-------|------|------------|----------|------|--------|
| Datetime | | | | | | | |
| 2004-12-31 01:00:00 | 13478.0 | 12 | 2004 | 2004-12-31 | 01:00:00 | 53 | Friday |
| 2004-12-31 02:00:00 | 12865.0 | 12 | 2004 | 2004-12-31 | 02:00:00 | 53 | Friday |
| 2004-12-31 03:00:00 | 12577.0 | 12 | 2004 | 2004-12-31 | 03:00:00 | 53 | Friday |
| 2004-12-31 04:00:00 | 12517.0 | 12 | 2004 | 2004-12-31 | 04:00:00 | 53 | Friday |
| 2004-12-31 05:00:00 | 12670.0 | 12 | 2004 | 2004-12-31 | 05:00:00 | 53 | Friday |

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 :

```
In [96]: # How many Unique Year do we Have in Dataset
print(df.Year.unique(), "\n")
print("Total Number of Unique Year", df.Year.nunique(), "\n")

[2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
 2018]

Total Number of Unique Year 15
```

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)})

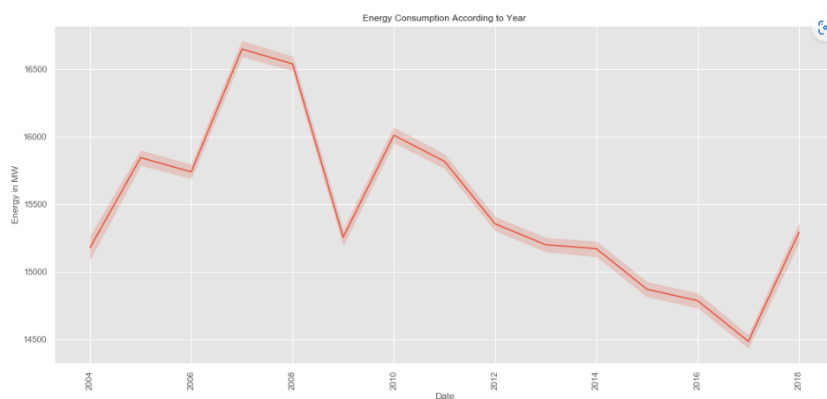
plt.title("Energy consumptionn in Year 2004")
plt.xlabel("Date")
plt.ylabel("Energy in MW")
plt.grid(True)
plt.legend()

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')

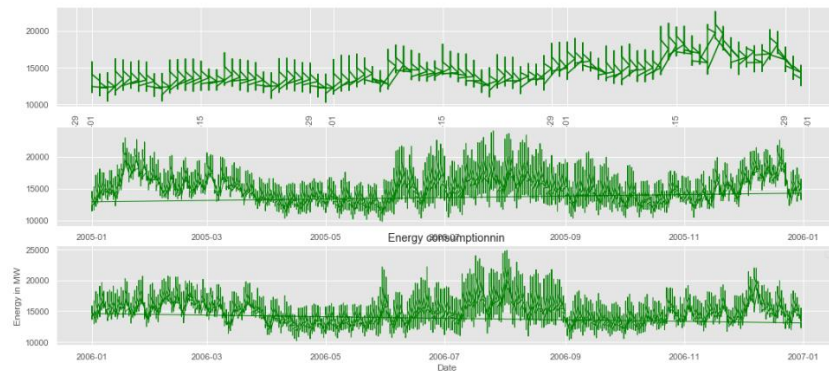
y_2004 = dataset["2004"]["AEP_MW"].to_list()
x_2004 = dataset["2004"]["Date"].to_list()
ax1.plot(x_2004,y_2004, color="green", linewidth=1.7)

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)
```



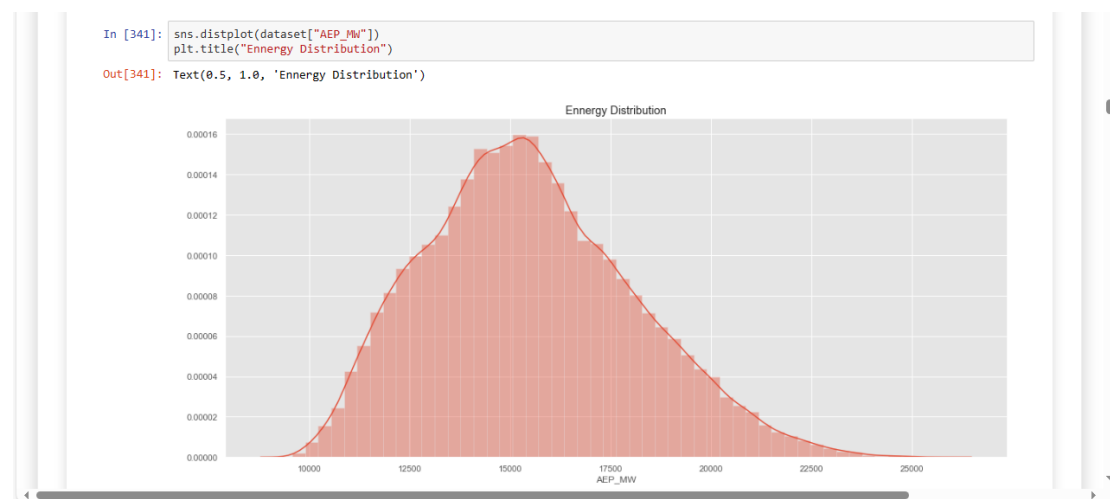
STEP 4:

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 :

```
In [356]: 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)
```

