In [1]:

```
import pandas
from datetime import date
from sklearn.metrics import r2_score
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
```

In [2]:

```
data = pandas.read_csv("../Dataset/AlteredColumns.csv")
```

In [3]:

```
data.head(5)
```

Out[3]:

	PeriodStart	AirTemp	CloudOpacity	DewpointTemp	Ghi	PrecipitableWater	Relati
0	2022-01-31T00:00:00Z	5.3	4.0	4.0	0	11.1	
1	2022-01-31T00:05:00Z	5.3	3.5	3.9	0	11.2	
2	2022-01-31T00:10:00Z	5.2	2.0	3.9	0	11.2	
3	2022-01-31T00:15:00Z	5.2	0.4	3.8	0	11.2	
4	2022-01-31T00:20:00Z	5.2	0.0	3.7	0	11.2	

In [4]:

```
data["PeriodStart"]=data["PeriodStart"].astype(str)
data["PeriodStart"]=data["PeriodStart"].str.replace('T',' ')
data["PeriodStart"]=data["PeriodStart"].str.replace('Z','')
data.head(5)
```

Out[4]:

	PeriodStart	AirTemp	CloudOpacity	DewpointTemp	Ghi	PrecipitableWater	Relativ
0	2022-01-31 00:00:00	5.3	4.0	4.0	0	11.1	
1	2022-01-31 00:05:00	5.3	3.5	3.9	0	11.2	
2	2022-01-31 00:10:00	5.2	2.0	3.9	0	11.2	
3	2022-01-31 00:15:00	5.2	0.4	3.8	0	11.2	
4	2022-01-31 00:20:00	5.2	0.0	3.7	0	11.2	

In [5]:

```
data["PeriodStart"]=pandas.to_datetime(data["PeriodStart"], format='%Y-%m-%d %H:
data.head(5)
```

Out[5]:

	PeriodStart	AirTemp	CloudOpacity	DewpointTemp	Ghi	PrecipitableWater	Relativ
0	2022-01-31 00:00:00	5.3	4.0	4.0	0	11.1	
1	2022-01-31 00:05:00	5.3	3.5	3.9	0	11.2	
2	2022-01-31 00:10:00	5.2	2.0	3.9	0	11.2	
3	2022-01-31 00:15:00	5.2	0.4	3.8	0	11.2	
4	2022-01-31 00:20:00	5.2	0.0	3.7	0	11.2	

In [6]:

```
data["hour"] = data["PeriodStart"].map(lambda x: x.hour)
data["minutes"] = data["PeriodStart"].map(lambda x: x.minute)
data["day"] = data['PeriodStart'].map(lambda x: x.day)
data["month"] = data['PeriodStart'].map(lambda x: x.month)
data["year"] = data['PeriodStart'].map(lambda x: x.year)
data = data.drop(['PeriodStart'], axis=1)
data.head(5)
```

Out[6]:

	AirTemp	CloudOpacity	DewpointTemp	Ghi	PrecipitableWater	RelativeHumidity	S
0	5.3	4.0	4.0	0	11.1	91.2	
1	5.3	3.5	3.9	0	11.2	91.1	
2	5.2	2.0	3.9	0	11.2	90.9	
3	5.2	0.4	3.8	0	11.2	90.8	
4	5.2	0.0	3.7	0	11.2	90.6	

In [7]:

```
X = data.drop(["Ghi"], axis=1)
y = data["Ghi"]
```

In [8]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random
```

In [9]:

```
model = MLPRegressor()
```

In [10]:

```
model.fit(X_train, y_train)
```

```
C:\Users\u8kar\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\neural_network\_multilayer_perceptron.py:686: ConvergenceWarning: Stochastic Opt
imizer: Maximum iterations (200) reached and the optimization hasn't converged ye
t.
  warnings.warn(
```

Out[10]:

▾ MLPRegressor

MLPRegressor()

```
In [11]: y_pred = model.predict(X_test)
```

```
In [12]: acc = r2_score(y_test, y_pred)*100
```

```
In [13]: print("Accuracy of the model is: ", acc)
```

```
Accuracy of the model is:  96.9083431526634
```