In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
dataset=pd.read_csv(r"Downloads/Folds5.csv")
```

In [3]:

```
dataset.head(20)
```

Out[3]:

```
       AT      V       AP     RH      PE
 0  23.25  71.29  1008.05  71.36  442.21
 1  13.87  42.99  1007.45  81.52  471.12
 2  16.91  43.96  1013.32  79.87  465.86
 3  10.09  37.14  1012.99  72.59  473.66
 4  12.72  40.60  1013.45  86.16  471.23
 5  17.77  52.90  1020.11  81.51  457.98
 6  24.14  63.21  1012.28  89.08  440.74
 7  29.05  75.60  1017.63  51.16  439.14
 8  17.44  45.01  1013.65  75.13  456.08
 9  11.17  44.77  1018.20  79.60  470.85
10   8.36  38.91  1016.27  88.82  482.09
11  16.70  36.99  1006.19  89.33  464.70
12  15.99  43.50  1021.34  73.22  460.70
13  28.31  50.05  1005.75  50.17  436.75
14  28.61  52.84  1006.00  45.73  437.39
15  10.56  40.22  1005.08  89.71  478.02
16  12.10  44.77  1019.10  75.85  463.11
17  33.83  67.90  1005.94  29.86  428.12
18  21.95  59.43  1010.29  74.67  442.84
19  32.67  67.83  1007.81  42.16  425.75
```

In [4]:

```
dataset.isnull().any()
```

Out[4]:

```
AT    False
V     False
AP    False
RH    False
PE    False
dtype: bool
```

In [5]:

```
x=dataset.iloc[:,0:4].values
y=dataset.iloc[:,4:5].values
```

In [6]:
```
x.shape
```
Out[6]:
```
(9568, 4)
```
In [7]:
```
y.shape
```
Out[7]:
```
(9568, 1)
```
In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2, random_state = 0)
```
In [10]:
```
x_train.shape
```
Out[10]:
```
(7654, 4)
```
In [11]:
```
x_test.shape
```
Out[11]:
```
(1914, 4)
```
In [12]:
```
y_train.shape
```
Out[12]:
```
(7654, 1)
```
In [13]:
```
y_test.shape
```
Out[13]:
```
(1914, 1)
```
In [14]:
```
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
mlr.fit(x_train,y_train)
```
Out[14]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```
In [15]:
```
y_pred =mlr.predict(x_test)
```
In [16]:
```
x_test
```
Out[16]:
```
array([[  13.87,   41.74, 1021.42,   64.02],
       [  26.36,   54.5 , 1015.35,   66.87],
       [  25.1 ,   74.87, 1010.01,   79.41],
       ...,
       [  27.35,   60.75, 1008.98,   67.56],
       [  16.92,   50.9 , 1013.27,   82.97],
       [  26.37,   54.5 , 1014.48,   66.31]])
```

```
In [17]: y_test
```

```
Out[17]: array([[472.12],
                [451.81],
                [439.93],
                ...,
                [435.14],
                [460.86],
                [451.78]])
```

```
In [18]: y_pred
```

```
Out[18]: array([[470.70306937],
                [442.19802338],
                [437.70172362],
                ...,
                [438.30370835],
                [459.08869914],
                [442.21409443]])
```

```
In [19]: from sklearn.metrics import r2_score
         accuracy = r2_score(y_test,y_pred)
```

```
In [20]: accuracy
```

```
Out[20]: 0.924894899672575
```

In [24]:

```
import keras
from keras.models import Sequential
from keras.layers import Dense
```

```
Using TensorFlow backend.
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: Future
Warning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\BEST\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [26]:
```
regressor = Sequential()
```

```
WARNING:tensorflow:From C:\Users\BEST\anaconda3\lib\site-packages\keras\backend\tensorflo
w_backend.py:74: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_def
ault_graph instead.
```

In [27]:
```
x_train.shape
```

Out[27]:
```
(7654, 4)
```

In [34]:
```
regressor.add(Dense(units=4,init = 'random_uniform',activation = 'relu'))
```

```
C:\Users\BEST\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(units=4, activation="relu", kernel_initializer="random_unif
orm")`
  """Entry point for launching an IPython kernel.
```

In [35]:
```
regressor.add(Dense(units=8,init = 'random_uniform',activation = 'relu'))
```

```
C:\Users\BEST\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(units=8, activation="relu", kernel_initializer="random_unif
orm")`
  """Entry point for launching an IPython kernel.
```

In [41]:
```
regressor.add(Dense(units = 1,init="random_uniform"))
```

```
C:\Users\BEST\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(units=1, kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [42]:
```
regressor.compile(optimizer = 'adam',loss = 'mse',metrics =['mse'])
```

In [43]:
```
regressor.fit(x_train,y_train,batch_size = 34,epochs = 100)
```

```
WARNING:tensorflow:From C:\Users\BEST\anaconda3\lib\site-packages\keras\backend\tensor
flow_backend.py:986: The name tf.assign_add is deprecated. Please use tf.compat.v1.assign_
add instead.

WARNING:tensorflow:From C:\Users\BEST\anaconda3\lib\site-packages\keras\backend\tensor
flow_backend.py:973: The name tf.assign is deprecated. Please use tf.compat.v1.assign instea
d.

Epoch 1/100
7654/7654 [==============================] - 2s 304us/step - loss: 160328.5380 - mea
n_squared_error: 160328.5380
Epoch 2/100
7654/7654 [==============================] - 0s 53us/step - loss: 4020.7849 - mean_s
quared_error: 4020.7849
Epoch 3/100
7654/7654 [==============================] - 0s 51us/step - loss: 424.5211 - mean_sq
uared_error: 424.5211
Epoch 4/100
7654/7654 [==============================] - 0s 51us/step - loss: 410.6388 - mean_sq
```

In [44]:

```
y_pred1 = regressor.predict(x_test)
```

In [45]:

```
y_test
```

Out[45]:

```
array([[472.12],
       [451.81],
       [439.93],
       ...,
       [435.14],
       [460.86],
       [451.78]])
```

In [46]:

```
y_pred1
```

Out[46]:

```
array([[471.2934 ],
       [443.68494],
       [436.30096],
       ...,
       [437.06735],
       [457.79306],
       [443.2874 ]], dtype=float32)
```

In [47]:

```
from sklearn.metrics import r2_score
accuracy1 = r2_score(y_test,y_pred1)
```

In [48]:

```
accuracy1
```

Out[48]:

```
0.906382984800807
```

In [49]:

```
regressor.save('electricity.h5')
```

In [50]:

```
import seaborn as sns
```

In [51]:

```
a=dataset.head(10)
```

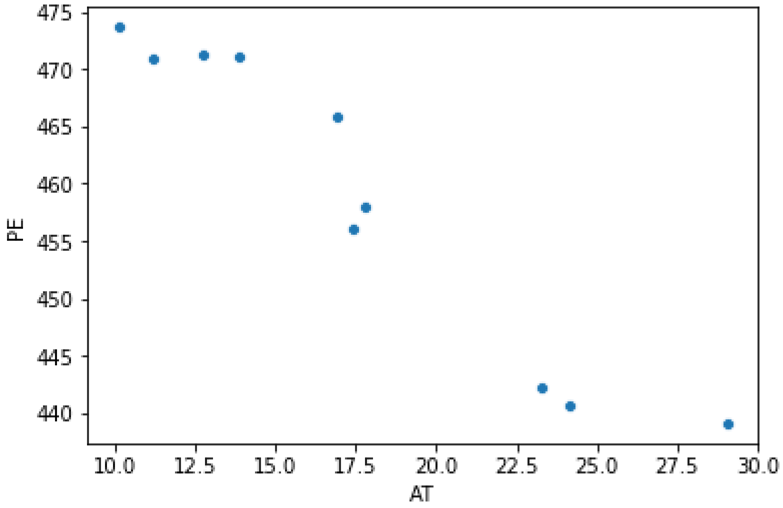
In [52]:
```
a
```

Out[52]:

	AT	V	AP	RH	PE
0	23.25	71.29	1008.05	71.36	442.21
1	13.87	42.99	1007.45	81.52	471.12
2	16.91	43.96	1013.32	79.87	465.86
3	10.09	37.14	1012.99	72.59	473.66
4	12.72	40.60	1013.45	86.16	471.23
5	17.77	52.90	1020.11	81.51	457.98
6	24.14	63.21	1012.28	89.08	440.74
7	29.05	75.60	1017.63	51.16	439.14
8	17.44	45.01	1013.65	75.13	456.08
9	11.17	44.77	1018.20	79.60	470.85

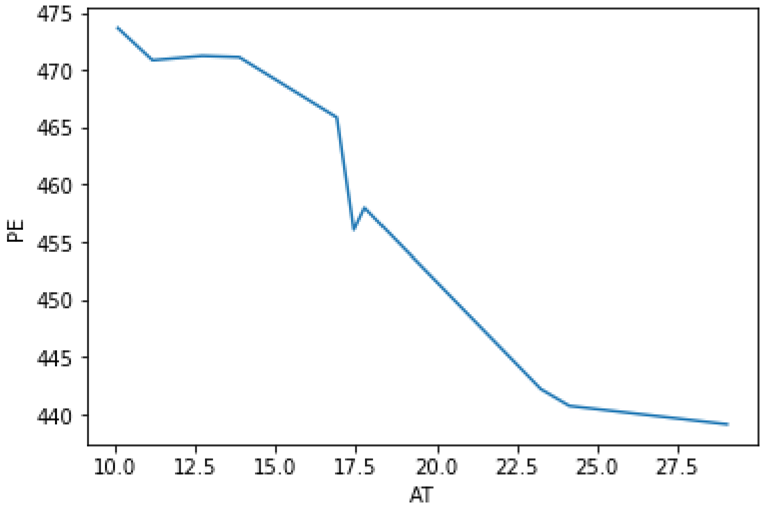
In [54]:
```
sns.scatterplot(x="AT",y = "PE",data=a)
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4262d9c08>
```



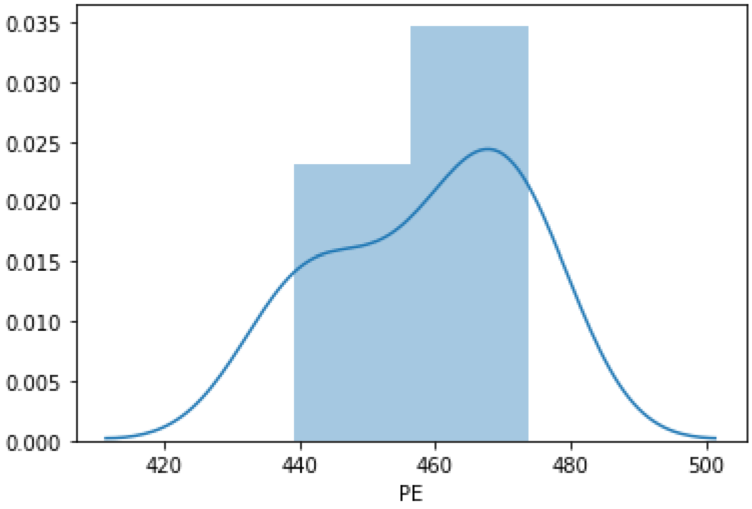
In [55]:
```
sns.lineplot(x="AT",y = "PE",data=a)
```

Out[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e42639d9c8>
```



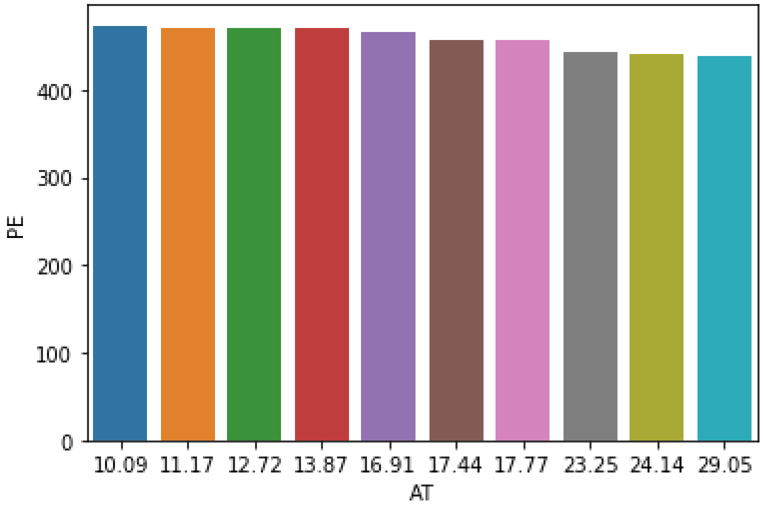
In [57]:
```
sns.distplot(a["PE"])
```

Out[57]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e426471348>
```



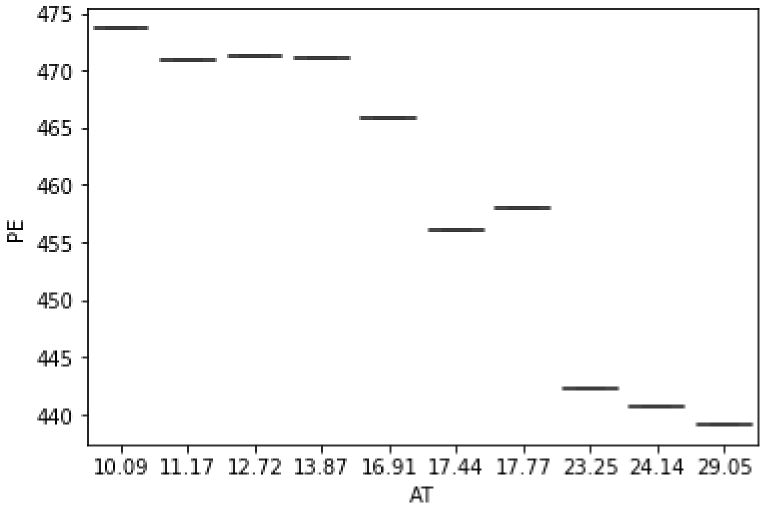
In [58]:
```
sns.barplot(x="AT",y = "PE",data=a)
```

Out[58]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4264c58c8>
```



In [59]:
```
sns.boxplot(x="AT",y="PE",data=a)
```

Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e426577588>
```



In [60]:
```
sns.pairplot(a,hue = "PE")
```

```
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\BEST\anaconda3\lib\site-packages\seaborn\distributions.py:288: UserWarning: Data
must have variance to compute a kernel density estimate.
```

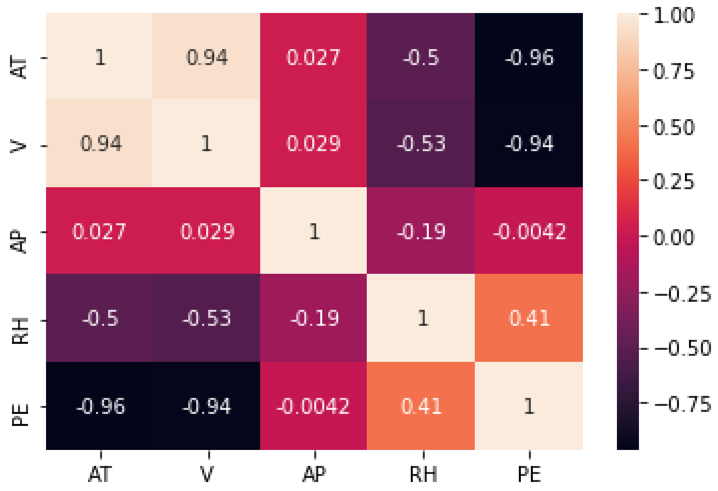
In [61]:
```
a.corr()
```

Out[61]:

	AT	V	AP	RH	PE
AT	1.000000	0.940407	0.026779	-0.502206	-0.962110
V	0.940407	1.000000	0.028525	-0.533201	-0.940854
AP	0.026779	0.028525	1.000000	-0.191591	-0.004194
RH	-0.502206	-0.533201	-0.191591	1.000000	0.410857
PE	-0.962110	-0.940854	-0.004194	0.410857	1.000000

In [62]:
```
sns.heatmap(a.corr(),annot = True)
```

Out[62]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e42759e248>
```



In [ ]:

```
```