In [4]:

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

In [5]:

```
dataset = pd.read_csv(r"C:\Users\Agath\Desktop\CCPP\Folds5x2_pp.csv")
```

In [6]:

```
dataset.head()
```

Out[6]:

	AT	V	AP	RH	PE
0	8.34	40.77	1010.84	90.01	480.48
1	23.64	58.49	1011.40	74.20	445.75
2	29.74	56.90	1007.15	41.91	438.76
3	19.07	49.69	1007.22	76.79	453.09
4	11.80	40.66	1017.13	97.20	464.43

In [7]:

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

Out[7]:

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

In [8]:

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

In [9]:

```
x.shape
```

Out[9]:

```
(9568, 4)
```

In [10]:

```
y.shape
```

Out[10]:

```
(9568, 1)
```

In [11]:

```
x.shape
```

Out[11]:

```
(9568, 4)
```

In [12]:

```
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 [13]:

```
x_train.shape
```

Out[13]:

```
(7654, 4)
```

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=Fal
se)
```

In [ ]:

```

```

In [15]:

```
y_pred = mlr.predict(x_test)
```

In [16]:

```
x_test
```

Out[16]:

```
array([[  27.4 ,   69.13, 1001.4 ,   82.47],
       [  21.94,   57.5 , 1014.64,   78.52],
       [  25.87,   57.32, 1012.06,   44.13],
       ...,
       [  20.64,   49.21, 1014.86,   58.47],
       [  23.28,   60.84, 1017.91,   67.5 ],
       [  28.7 ,   71.64, 1007.11,   69.85]])
```

In [17]:

```
y_test
```

Out[17]:

```
array([[426.18],
       [451.1 ],
       [442.87],
       ...,
       [454.2 ],
       [444.13],
       [436.58]])
```

In [18]:

```
y_pred
```

Out[18]:

```
array([[433.36241603],
       [448.38864548],
       [445.88661799],
       ...,
       [456.06061126],
       [446.91190634],
       [432.57326179]])
```

In [19]:

```
from sklearn.metrics import r2_score
accuracy = r2_score(y_test,y_pred)
```

In [20]:

```
accuracy
```

Out[20]:

```
0.9298994694436788
```

In [21]:

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

```
Using TensorFlow backend.
C:\Users\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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\Agath\anaconda3\lib\site-packages\tensorflow\python\framework\dty
pes.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of ty
pe 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)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:541: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
od as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:542: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
od as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
od as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
od as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:545: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
```

```
od as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_s
tub\dtypes.py:550: FutureWarning: Passing (type, 1) or '1type' as a synony
m of type is deprecated; in a future version of numpy, it will be understo
od as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [22]:

```
regressor = Sequential()
```

```
WARNING:tensorflow:From C:\Users\Agath\anaconda3\lib\site-packages\keras\bac
kend\tensorflow_backend.py:74: The name tf.get_default_graph is deprecated.
Please use tf.compat.v1.get_default_graph instead.
```

In [23]:

```
x_train.shape
```

Out[23]:

```
(7654, 4)
```

In [24]:

```
regressor.add(Dense(units = 4,init = 'random_uniform',activation = 'relu'))
```

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

In [25]:

```
regressor.add(Dense(units = 8,init = 'random_uniform',activation = 'relu'))
```

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

In [26]:

```
regressor.add(Dense(units = 1,init = 'random_uniform'))
```

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

In [27]:

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

```
WARNING:tensorflow:From C:\Users\Agath\anaconda3\lib\site-packages\keras\opt
imizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.com
pat.v1.train.Optimizer instead.
```

In [37]:

```
regressor.fit(x_train,y_train , batch_size =20,epochs =100)
```

```
Epoch 1/100
7654/7654 [==============================] - 0s 35us/step - loss: 26.8930
- mean_squared_error: 26.8930
Epoch 2/100
7654/7654 [==============================] - 0s 35us/step - loss: 27.1147
- mean_squared_error: 27.1147
Epoch 3/100
7654/7654 [==============================] - 0s 36us/step - loss: 27.1130
- mean_squared_error: 27.1130
Epoch 4/100
7654/7654 [==============================] - 0s 35us/step - loss: 27.4780
- mean_squared_error: 27.4780
Epoch 5/100
7654/7654 [==============================] - 0s 34us/step - loss: 27.0696
- mean_squared_error: 27.0696
Epoch 6/100
7654/7654 [==============================] - 0s 36us/step - loss: 27.0003
- mean_squared_error: 27.0003
Epoch 7/100
```

In [38]:

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

In [39]:

```
y_test
```

Out[39]:

```
array([[426.18],
       [451.1 ],
       [442.87],
       ...,
       [454.2 ],
       [444.13],
       [436.58]])
```

In [40]:

```
y_pred1
```

Out[40]:

```
array([[430.16476],
       [449.47214],
       [445.06168],
       ...,
       [456.0029 ],
       [449.05554],
       [431.4136 ]], dtype=float32)
```

In [41]:

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

In [42]:

```
accuracy1
```

Out[42]:

```
0.9118091189681656
```

In [43]:

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

In [44]:

```
regressor.predict(np.array([[ 25.87,   57.32, 1012.06,   44.13]]))
```

Out[44]:

```
array([[445.06168]], dtype=float32)
```

In [45]:

```
import seaborn as sns
```

In [59]:

```
a=dataset.head(20)
```

In [60]:

```
a
```

Out[60]:

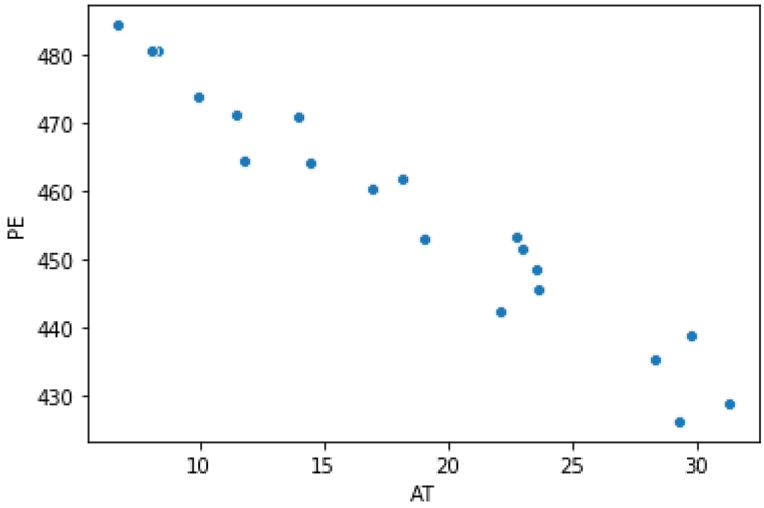
	AT	V	AP	RH	PE
0	8.34	40.77	1010.84	90.01	480.48
1	23.64	58.49	1011.40	74.20	445.75
2	29.74	56.90	1007.15	41.91	438.76
3	19.07	49.69	1007.22	76.79	453.09
4	11.80	40.66	1017.13	97.20	464.43
5	13.97	39.16	1016.05	84.60	470.96
6	22.10	71.29	1008.20	75.38	442.35
7	14.47	41.76	1021.98	78.41	464.00
8	31.25	69.51	1010.25	36.83	428.77
9	6.77	38.18	1017.80	81.13	484.31
10	28.28	68.67	1006.36	69.90	435.29
11	22.99	46.93	1014.15	49.42	451.41
12	29.30	70.04	1010.95	61.23	426.25
13	8.14	37.49	1009.04	80.33	480.66
14	16.92	44.60	1017.34	58.75	460.17
15	22.72	64.15	1021.14	60.34	453.13
16	18.14	43.56	1012.83	47.10	461.71
17	11.49	44.63	1020.44	86.04	471.08
18	9.94	40.46	1018.90	68.51	473.74
19	23.54	41.10	1002.05	38.05	448.56

In [50]:

```
sns.scatterplot(x = "AT",y = "PE" ,data = a )
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d7eb9eb088>
```

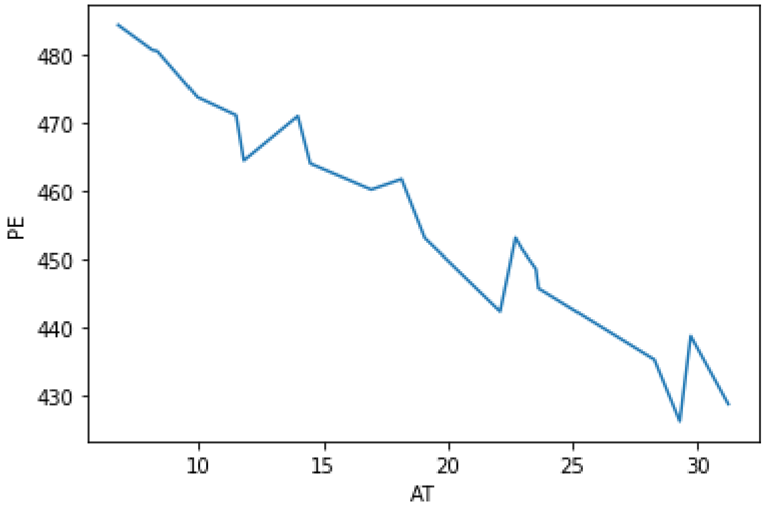


In [51]:

```
sns.lineplot(x = "AT",y = "PE" ,data = a )
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d7eba81408>
```

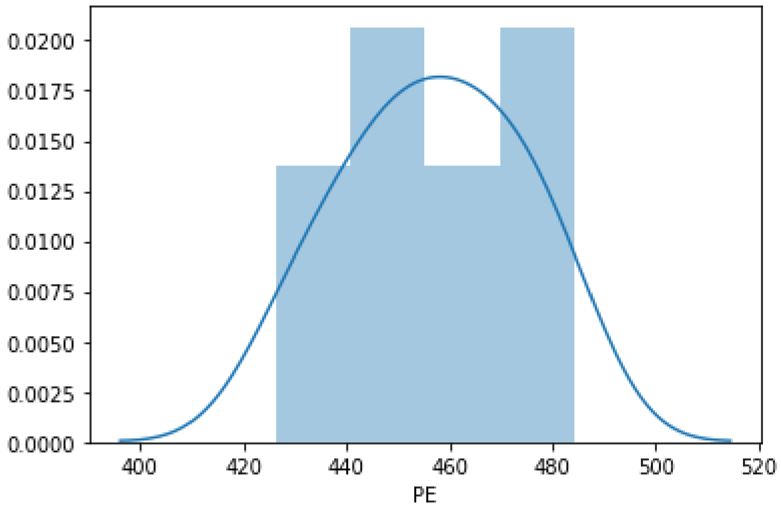


In [52]:

```
sns.distplot(a['PE'])
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d7ebb46d48>
```

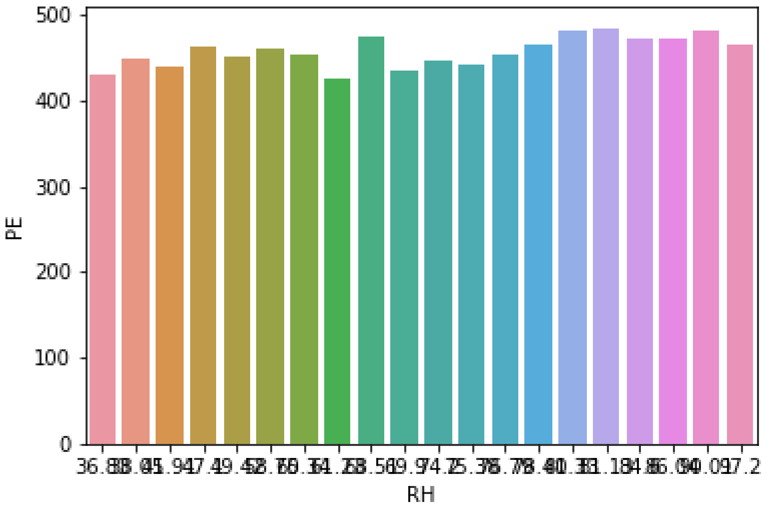


In [56]:

```
sns.barplot(x = "RH",y="PE",data =a)
```

Out[56]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d7ebeb5a48>
```

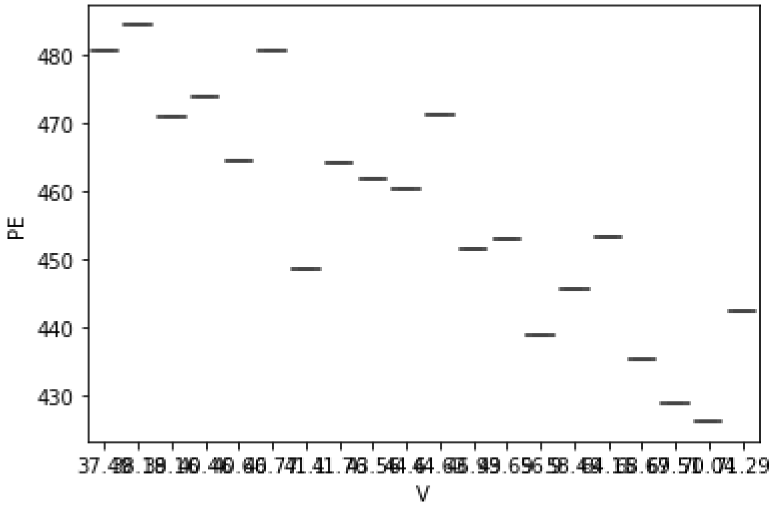


In [58]:

```
sns.boxplot(x = "V",y="PE", data =a)
```

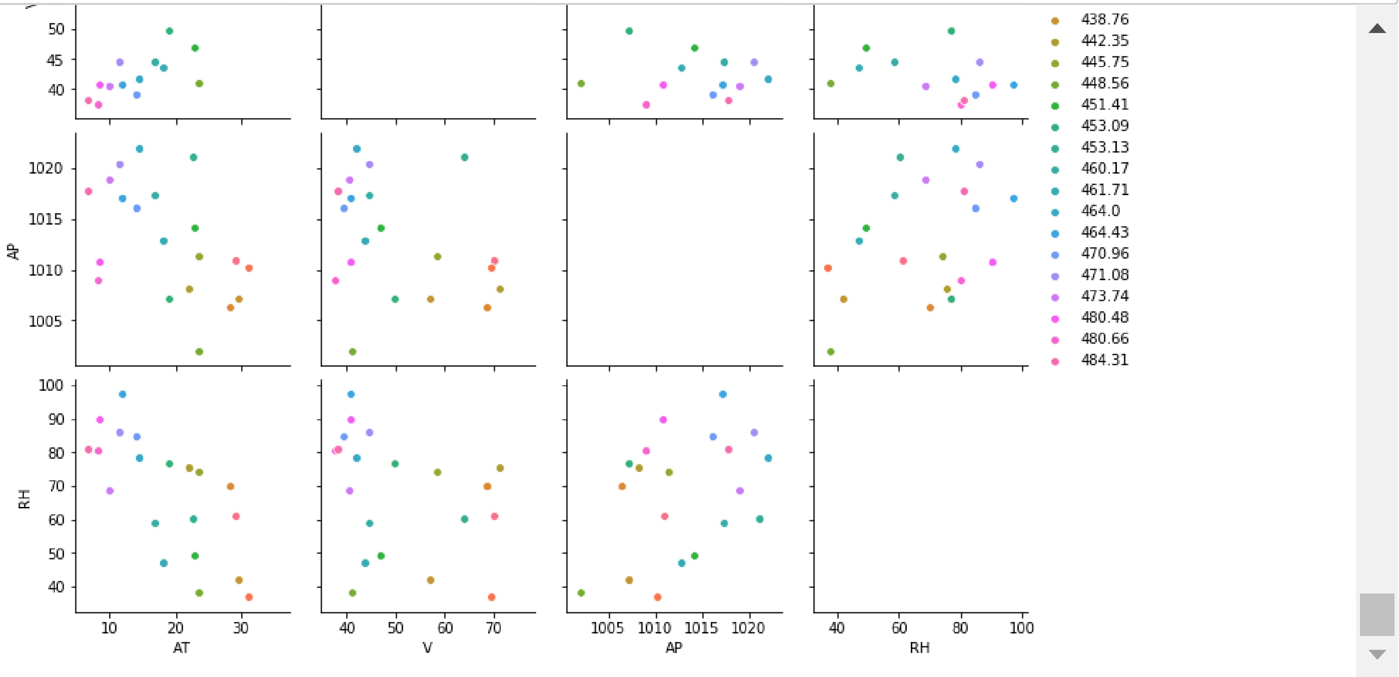
Out[58]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d7ec17fbc8>
```



In [116]:

```
sns.pairplot(a,hue = 'PE')
```



In [117]:

```
a.corr()
```

Out[117]:

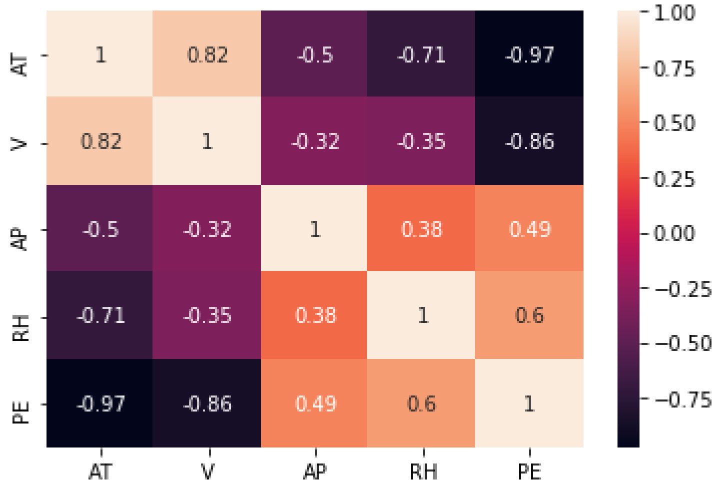
	AT	V	AP	RH	PE
AT	1.000000	0.818103	-0.496996	-0.707170	-0.973740
V	0.818103	1.000000	-0.320041	-0.345580	-0.862308
AP	-0.496996	-0.320041	1.000000	0.378671	0.487050
RH	-0.707170	-0.345580	0.378671	1.000000	0.599262
PE	-0.973740	-0.862308	0.487050	0.599262	1.000000

In [118]:

```
sns.heatmap(a.corr(),annot = True)
```

Out[118]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27780674fc8>
```



In [ ]:

```

```