Creating and Saving the Model

In [1]:

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

In [2]:

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

In [3]:

```
dataset.head()
```

Out[3]:

	AT	V	AP	RH	PE
0	14.96	41.76	1024.07	73.17	463.26
1	25.18	62.96	1020.04	59.08	444.37
2	5.11	39.40	1012.16	92.14	488.56
3	20.86	57.32	1010.24	76.64	446.48
4	10.82	37.50	1009.23	96.62	473.90

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]:

```
(47840, 4)
```

In [7]:

```
y.shape
```

Out[7]:

```
(47840, 1)
```

In [8]:

```
x.shape
```

Out[8]:

```
(47840, 4)
```

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]:

```
(38272, 4)
```

In [11]:

```
from sklearn.linear_model import LinearRegression
mlr  = LinearRegression()
mlr.fit(x_train,y_train)
```

Out[11]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fal
se)
```

In [ ]:

```

```

In [12]:

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

In [13]:

```
x_test
```

Out[13]:

```
array([[  29.7 ,   57.19, 1008.41,   55.07],
       [  23.27,   64.69, 1006.84,   70.53],
       [  11.44,   44.2 , 1018.14,   84.9 ],
       ...,
       [  16.  ,   43.34, 1013.58,   78.87],
       [   5.15,   40.78, 1025.34,   95.84],
       [  16.77,   42.28, 1007.53,   73.19]])
```

In [14]:

```
y_test
```

Out[14]:

```
array([[436.29],
       [438.08],
       [476.12],
       ...,
       [461.7 ],
       [483.03],
       [465.52]])
```

In [15]:

```
y_pred
```

Out[15]:

```
array([[436.39329725],
       [444.80652228],
       [471.4625261 ],
       ...,
       [463.31280148],
       [483.43344779],
       [462.55077955]])
```

In [16]:

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

In [17]:

```
accuracy
```

Out[17]:

```
0.9294606737992249
```

In [18]:

```
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\dtype
s.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\dtype
s.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\dtype
s.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\dtype
s.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\dtype
s.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\dtype
s.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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_stu
b\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\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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\Agath\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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 [19]:

```
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 [20]:

```
x_train.shape
```

Out[20]:

```
(38272, 4)
```

In [21]:

```
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 [22]:

```
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 [23]:

```
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 [24]:

```
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 [25]:

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

```
2 - mean_squared_error: 26.8832
Epoch 97/100
38272/38272 [==============================] - 2s 41us/step - loss: 26.762
9 - mean_squared_error: 26.7629
Epoch 98/100
38272/38272 [==============================] - 2s 39us/step - loss: 26.712
6 - mean_squared_error: 26.7126
Epoch 99/100
38272/38272 [==============================] - 2s 39us/step - loss: 27.062
3 - mean_squared_error: 27.0623
Epoch 100/100
38272/38272 [==============================] - 2s 40us/step - loss: 27.027
5 - mean_squared_error: 27.0275
```

Out[25]:

```
<keras.callbacks.History at 0x1bac6854fc8>
```

In [26]:

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

In [27]:

```
y_test
```

Out[27]:

```
array([[436.29],
       [438.08],
       [476.12],
       ...,
       [461.7 ],
       [483.03],
       [465.52]])
```

In [28]:

```
y_pred1
```

Out[28]:

```
array([[434.4963 ],
       [440.8289 ],
       [470.35477],
       ...,
       [461.31876],
       [484.25854],
       [457.90683]], dtype=float32)
```

In [29]:

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

In [30]:

```
accuracy1
```

Out[30]:

```
0.902936827925405
```

In [31]:

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

In [32]:

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

Out[32]:

```
array([[443.81635]], dtype=float32)
```

Model has been created and saved