concrete strength-regression

In [52]:

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

In [53]:

```
dataset = pd.read_csv(r"C:\Users\kotha\Downloads\Concrete strength.csv")
```

In [3]:

```
dataset
```

Out[3]:

	cement	slag	flyash	water	superplasticizer	coarseaggregate	fineaggregate	age	csMI
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.
...	...	...	...	...	...	...	...	...	
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.

1030 rows × 9 columns

In [4]:

```
dataset.head(2)
```

Out[4]:

	cement	slag	flyash	water	superplasticizer	coarseaggregate	fineaggregate	age	csMPa
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89

In [5]:

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

Out[5]:

```
cement              False
slag                False
flyash              False
water               False
superplasticizer    False
coarseaggregate     False
fineaggregate       False
age                 False
csMPa               False
dtype: bool
```

In [6]:

```
x  = dataset.iloc[:,0:8].values
y = dataset.iloc[:,8:9].values
```

In [7]:

```
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 [8]:

```
x_train.shape
```

Out[8]:

```
(824, 8)
```

In [9]:

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

```
Using TensorFlow backend.
C:\Users\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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\kotha\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 [10]:

```
regressor = Sequential()
```

```
WARNING:tensorflow:From C:\Users\kotha\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 [11]:

```
x_train.shape
```

Out[11]:

```
(824, 8)
```

In [12]:

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

```
C:\Users\kotha\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 [13]:

```
regressor.add(Dense(units = 16,init = 'random_uniform',activation = 'relu'))
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarn
ing: Update your `Dense` call to the Keras 2 API: `Dense(units=16, activatio
n="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [14]:

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

```
C:\Users\kotha\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 [15]:

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

```
WARNING:tensorflow:From C:\Users\kotha\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 [43]:

```
regressor.fit(x_train,y_train,batch_size=32,epochs=10000)
```

```
Epoch 1/10000
824/824 [==============================] - 0s 58us/step - loss: 35.4142 -
mean_squared_error: 35.4142
Epoch 2/10000
824/824 [==============================] - 0s 56us/step - loss: 36.2044 -
mean_squared_error: 36.2044
Epoch 3/10000
824/824 [==============================] - 0s 54us/step - loss: 36.0323 -
mean_squared_error: 36.0323
Epoch 4/10000
824/824 [==============================] - 0s 60us/step - loss: 35.4921 -
mean_squared_error: 35.4921
Epoch 5/10000
824/824 [==============================] - 0s 52us/step - loss: 37.4188 -
mean_squared_error: 37.4188
Epoch 6/10000
824/824 [==============================] - 0s 71us/step - loss: 37.3103 -
mean_squared_error: 37.3103
Epoch 7/10000
```

In [44]:

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

In [45]:

```
y_pred
```

Out[45]:

```
array([[30.914309 ],
       [ 9.091808 ],
       [73.82176  ],
       [55.97666  ],
       [12.5306835],
       [44.37255  ],
       [56.015198 ],
       [23.522661 ],
       [63.566628 ],
       [48.34412  ],
       [21.431349 ],
       [46.254322 ],
       [37.21066  ],
       [16.846037 ],
       [58.392376 ],
       [61.42476  ],
       [30.537127 ],
       [34.95448  ],
```

In [46]:

```
y_test
```

Out[46]:

```
array([[26.06],
       [10.35],
       [79.3 ],
       [74.99],
       [ 9.69],
       [47.1 ],
       [59.  ],
       [22.72],
       [61.89],
       [52.12],
       [17.54],
       [48.15],
       [38.33],
       [17.2 ],
       [56.83],
       [55.25],
       [33.36],
       [34.68],
```

In [47]:

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

In [48]:

```
accuracy
```

Out[48]:

```
0.8728790348565478
```

In [49]:

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

In [50]:

```
x_train.shape
```

Out[50]:

```
(824, 8)
```

In [51]:

```
regressor.predict(np.array([[237.5,237.5,0,228,0,932,594,365]]))
```

Out[51]:

```
array([[43.079132]], dtype=float32)
```

In [ ]:

```

```