50_startups-regression

In [1]:

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

In [2]:

```
dataset = pd.read_csv(r"C:\Users\kotha\Downloads\50_startups.csv")
```

In [3]:

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

Out[3]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.2	136897.80	471784.10	New York	192261.83
1	162597.7	151377.59	443898.53	California	191792.06

In [4]:

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

Out[4]:

```
R&D Spend          False
Administration     False
Marketing Spend    False
State              False
Profit             False
dtype: bool
```

In [5]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['State'] = le.fit_transform(dataset['State'])
```

In [6]:

```
dataset.head()
```

Out[6]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	2	192261.83
1	162597.70	151377.59	443898.53	0	191792.06
2	153441.51	101145.55	407934.54	1	191050.39
3	144372.41	118671.85	383199.62	2	182901.99
4	142107.34	91391.77	366168.42	1	166187.94

In [7]:

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

In [8]:

```
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
z=one.fit_transform(x[:,3:4]).toarray()
x=np.delete(x,3,axis=1)
x=np.concatenate((z,x),axis=1)
```

```
from sklearn.preprocessing import OneHotEncoder
one=OneHotEncoder()
z=one.fit_transform(x[:,3:4]).toarray()

x=np.delete(x,3,axis=1)
x=np.concatenate((z,x),axis=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]:

```
(40, 6)
```

In [11]:

```
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 [12]:

```
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 [13]:

```
x_train.shape
```

Out[13]:

```
(40, 6)
```

In [14]:

```
regressor.add(Dense(units = 6,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=6, activation
="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [15]:

```
regressor.add(Dense(units = 12,init = 'random_uniform',activation = 'relu')) # 1st hidden l
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarn
ing: Update your `Dense` call to the Keras 2 API: `Dense(units=12, activatio
n="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [16]:

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

```
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 [17]:

```
regressor.add(Dense(units = 9,init = 'random_uniform',activation = 'relu')) # third Hidden
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarn
ing: Update your `Dense` call to the Keras 2 API: `Dense(units=9, activation
="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [18]:

```
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 [19]:

```
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 [32]:

```
regressor.fit(x_train,y_train,batch_size=32,epochs=500)
```

```
Epoch 1/500
40/40 [==============================] - 0s 115us/step - loss: 152199627.2
000 - mean_squared_error: 152199627.2000
Epoch 2/500
40/40 [==============================] - 0s 48us/step - loss: 153872550.40
00 - mean_squared_error: 153872550.4000
Epoch 3/500
40/40 [==============================] - 0s 0us/step - loss: 152240553.600
0 - mean_squared_error: 152240553.6000
Epoch 4/500
40/40 [==============================] - 0s 145us/step - loss: 149062284.8
000 - mean_squared_error: 149062284.8000
Epoch 5/500
40/40 [==============================] - 0s 181us/step - loss: 149601785.6
000 - mean_squared_error: 149601785.6000
Epoch 6/500
40/40 [==============================] - 0s 187us/step - loss: 152042803.2
000 - mean_squared_error: 152042803.2000
Epoch 7/500
```

In [23]:

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

In [24]:

```
y_pred
```

Out[24]:

```
array([[119817.766],
       [122347.05 ],
       [128282.266],
       [ 61851.574],
       [174912.98 ],
       [122507.8  ],
       [ 51392.31 ],
       [103579.77 ],
       [118320.97 ],
       [160319.61 ]], dtype=float32)
```

In [25]:

```
y_test
```

Out[25]:

```
array([[103282.38],
       [144259.4 ],
       [146121.95],
       [ 77798.83],
       [191050.39],
       [105008.31],
       [ 81229.06],
       [ 97483.56],
       [110352.25],
       [166187.94]])
```

In [26]:

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

In [27]:

```
accuracy
```

Out[27]:

```
0.7718242300169484
```

model.save('regressor.h5')

In [28]:

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

In [29]:

```
x_train.shape
```

Out[29]:

```
(40, 6)
```

In [31]:

```
regressor.predict(np.array([[1,0,0,63408.86,129219.6,46085.25]]))
```

Out[31]:

```
array([[93789.21]], dtype=float32)
```

In [ ]:

```
```

In [ ]:

```
```