

Regression

Double-click (or enter) to edit

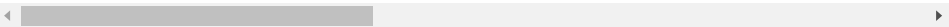
```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/House Price India.csv')
```

```
df.head()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	numb vie
0	6762810145	42491	5	2.50	3650	9050	2.0	0	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	

5 rows × 23 columns



```
df.isnull().sum()
```

id	0
Date	0
number of bedrooms	0
number of bathrooms	0
living area	0
lot area	0
number of floors	0
waterfront present	0
number of views	0
condition of the house	0
grade of the house	0
Area of the house(excluding basement)	0
Area of the basement	0
Built Year	0
Renovation Year	0
Postal Code	0
Latitude	0
Longitude	0
living_area_renov	0
lot_area_renov	0
Number of schools nearby	0
Distance from the airport	0
Price	0
dtype: int64	

```
df.info()
```

<class 'pandas.core.frame.DataFrame'>			
RangeIndex: 14620 entries, 0 to 14619			
Data columns (total 23 columns):			
#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	id	14620 non-null	int64
1	Date	14620 non-null	int64

```
2  number of bedrooms      14620 non-null int64
3  number of bathrooms     14620 non-null float64
4  living area             14620 non-null int64
5  lot area                14620 non-null int64
6  number of floors        14620 non-null float64
7  waterfront present      14620 non-null int64
8  number of views         14620 non-null int64
9  condition of the house  14620 non-null int64
10 grade of the house      14620 non-null int64
11 Area of the house(excluding basement) 14620 non-null int64
12 Area of the basement    14620 non-null int64
13 Built Year              14620 non-null int64
14 Renovation Year         14620 non-null int64
15 Postal Code             14620 non-null int64
16 Lattitude               14620 non-null float64
17 Longitude               14620 non-null float64
18 living_area_renov       14620 non-null int64
19 lot_area_renov          14620 non-null int64
20 Number of schools nearby 14620 non-null int64
21 Distance from the airport 14620 non-null int64
22 Price                   14620 non-null int64
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

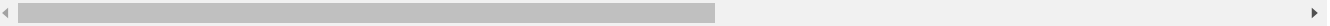
```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
df.head()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renova
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	

5 rows × 23 columns



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

from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3,random_state=0)

xtrain.shape, xtest.shape

((10234, 4), (4386, 4))

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

```
regressor = Sequential()
regressor.add(Dense(4,activation='relu'))
regressor.add(Dense(12,activation='relu'))
regressor.add(Dense(8,activation='relu'))
regressor.add(Dense(9,activation='relu'))
regressor.add(Dense(1,activation='linear'))

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

regressor.fit(xtrain,ytrain,batch_size=10,epochs=300)

1024/1024 [=====] - 2s 2ms/step - loss: 4208089.0000 - mse: 4208089.0000
Epoch 273/300
1024/1024 [=====] - 3s 3ms/step - loss: 4204387.5000 - mse: 4204387.5000
Epoch 274/300
1024/1024 [=====] - 2s 2ms/step - loss: 4200702.5000 - mse: 4200702.5000
Epoch 275/300
1024/1024 [=====] - 2s 2ms/step - loss: 4197013.5000 - mse: 4197013.5000
Epoch 276/300
1024/1024 [=====] - 2s 2ms/step - loss: 4193336.7500 - mse: 4193336.7500
Epoch 277/300
1024/1024 [=====] - 2s 2ms/step - loss: 4189655.0000 - mse: 4189655.0000
Epoch 278/300
1024/1024 [=====] - 3s 3ms/step - loss: 4185969.0000 - mse: 4185969.0000
Epoch 279/300
1024/1024 [=====] - 2s 2ms/step - loss: 4182281.0000 - mse: 4182281.0000
Epoch 280/300
1024/1024 [=====] - 2s 2ms/step - loss: 4178601.2500 - mse: 4178601.2500
Epoch 281/300
1024/1024 [=====] - 2s 2ms/step - loss: 4174928.7500 - mse: 4174928.7500
Epoch 282/300
1024/1024 [=====] - 2s 2ms/step - loss: 4171257.0000 - mse: 4171257.0000
Epoch 283/300
1024/1024 [=====] - 2s 2ms/step - loss: 4167588.0000 - mse: 4167588.0000
Epoch 284/300
1024/1024 [=====] - 3s 3ms/step - loss: 4163914.5000 - mse: 4163914.5000
Epoch 285/300
1024/1024 [=====] - 2s 2ms/step - loss: 4160240.5000 - mse: 4160240.5000
Epoch 286/300
1024/1024 [=====] - 2s 2ms/step - loss: 4156575.0000 - mse: 4156574.7500
Epoch 287/300
1024/1024 [=====] - 2s 2ms/step - loss: 4152912.2500 - mse: 4152912.2500
Epoch 288/300
1024/1024 [=====] - 2s 2ms/step - loss: 4149248.5000 - mse: 4149248.5000
Epoch 289/300
1024/1024 [=====] - 2s 2ms/step - loss: 4145592.7500 - mse: 4145592.7500
Epoch 290/300
1024/1024 [=====] - 3s 3ms/step - loss: 4141931.7500 - mse: 4141931.7500
Epoch 291/300
1024/1024 [=====] - 2s 2ms/step - loss: 4138282.0000 - mse: 4138282.0000
Epoch 292/300
1024/1024 [=====] - 2s 2ms/step - loss: 4134633.7500 - mse: 4134633.5000
Epoch 293/300
1024/1024 [=====] - 2s 2ms/step - loss: 4130982.0000 - mse: 4130982.5000
Epoch 294/300
1024/1024 [=====] - 2s 2ms/step - loss: 4127334.0000 - mse: 4127334.0000
Epoch 295/300
1024/1024 [=====] - 3s 3ms/step - loss: 4123681.7500 - mse: 4123681.7500
Epoch 296/300
1024/1024 [=====] - 3s 2ms/step - loss: 4120037.7500 - mse: 4120037.7500
Epoch 297/300
1024/1024 [=====] - 2s 2ms/step - loss: 4116398.0000 - mse: 4116398.0000
Epoch 298/300
1024/1024 [=====] - 2s 2ms/step - loss: 4112757.7500 - mse: 4112757.7500
Epoch 299/300
1024/1024 [=====] - 2s 2ms/step - loss: 4109117.2500 - mse: 4109117.2500
Epoch 300/300
1024/1024 [=====] - 2s 2ms/step - loss: 4105478.0000 - mse: 4105478.0000
<keras.callbacks.History at 0x7f8da9418eb0>
```

```
ypred = regressor.predict(xtest)
```

```
from sklearn.metrics import r2_score
```

```
r2_score(ytest,ypred)*100

-374.53426194770475
```

```
ypred.flatten()

array([303.58432, 303.58432, 303.58432, ..., 303.58432, 303.58432,
       303.58432], dtype=float32)
```

```
pd.DataFrame({'Actual value':ytest.flatten(),
              'Predicted value':ypred.flatten()}).head(10)
```

	Actual value	Predicted value	
0	1440	303.58432	
1	4270	303.58432	
2	1010	303.58432	
3	1970	303.58432	
4	2320	303.58432	
5	1390	303.58432	
6	4070	303.58432	
7	2050	303.58432	
8	6880	303.58432	
9	2690	303.58432	

```
regressor.predict([[1000456,23235554,1232334,2]])

1/1 [=====] - 0s 79ms/step
array([[ -6377741.]], dtype=float32)1/1 [=====] - 0s 75ms/step
array([[ -6377741.]], dtype=float32)
```

```
regressor.predict([[1000,2354,1234,2]])

1/1 [=====] - 0s 44ms/step
array([[ -339.50894]], dtype=float32)
```