house-price

May 8, 2023

[13]:

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import Sequential
from sklearn.metrics import r2_score
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
```

[14]:

```
df = pd.read_csv(r'C:\Users\nithy\Downloads\Compressed\House Price India.csv')
df.drop(['id','Date'], axis=1, inplace=True)
df.head()
```

[14]:

```
   number of bedrooms  number of bathrooms  living area  lot area  \
0                   5                 2.50         3650      9050
1                   4                 2.50         2920      4000
2                   5                 2.75         2910      9480
3                   4                 2.50         3310     42998
4                   3                 2.00         2710      4500

   number of floors  waterfront present  number of views  \
0               2.0                   0                4
1               1.5                   0                0
2               1.5                   0                0
3               2.0                   0                0
4               1.5                   0                0

   condition of the house  grade of the house  \
0                       5                  10
1                       5                   8
2                       3                   8
3                       3                   9
4                       4                   8

   Area of the house(excluding basement)  …  Built Year  Renovation Year  \
0                                   3370  …        1921                0
1                                   1910  …        1909                0
```

```
2                                   2910  …        1939                0
3                                   3310  …        2001                0
4                                   1880  …        1929                0

   Postal Code  Lattitude  Longitude  living_area_renov  lot_area_renov  \
0       122003    52.8645   -114.557               2880            5400
1       122004    52.8878   -114.470               2470            4000
2       122004    52.8852   -114.468               2940            6600
3       122005    52.9532   -114.321               3350           42847
4       122006    52.9047   -114.485               2060            4500

   Number of schools nearby  Distance from the airport    Price
0                         2                         58  2380000
1                         2                         51  1400000
2                         1                         53  1200000
3                         3                         76   838000
4                         1                         51   805000

[5 rows x 21 columns]
```

[15]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 21 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   number of bedrooms                     14620 non-null  int64
 1   number of bathrooms                    14620 non-null  float64
 2   living area                            14620 non-null  int64
 3   lot area                               14620 non-null  int64
 4   number of floors                       14620 non-null  float64
 5   waterfront present                     14620 non-null  int64
 6   number of views                        14620 non-null  int64
 7   condition of the house                 14620 non-null  int64
 8   grade of the house                     14620 non-null  int64
 9   Area of the house(excluding basement)  14620 non-null  int64
 10  Area of the basement                   14620 non-null  int64
 11  Built Year                             14620 non-null  int64
 12  Renovation Year                        14620 non-null  int64
 13  Postal Code                            14620 non-null  int64
 14  Lattitude                              14620 non-null  float64
 15  Longitude                              14620 non-null  float64
 16  living_area_renov                      14620 non-null  int64
 17  lot_area_renov                         14620 non-null  int64
 18  Number of schools nearby               14620 non-null  int64
 19  Distance from the airport              14620 non-null  int64
 20  Price                                      14620 non-null  int64
dtypes: float64(4), int64(17)
memory usage: 2.3 MB
```

[16]:

```python
df.isna().sum()
```

[16]:

```
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
Lattitude                                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
```

[17]:

```python
X = df.drop('Price', axis=1)
y = df['Price']
```

[18]:

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
  ↪random_state=12)
```

[19]:

```python
print(f'\n shape of X_train - {X_train.shape}\n')
print(f' shape of X_test - {X_test.shape}\n')
print(f' shape of y_train - {y_train.shape}\n')
print(f' shape of y_test - {y_test.shape}\n')
```

```

 shape of X_train - (11696, 20)

 shape of X_test - (2924, 20)

 shape of y_train - (11696,)

 shape of y_test - (2924,)
```

[30]:

```
number_of_features = len(X.columns)

model = Sequential()

model.add(layer=Input(shape=number_of_features))

model.add(layer=Dense(units=32, activation='relu'))

model.add(layer=Dense(units=64, activation='relu'))

model.add(layer=Dense(units=128, activation='relu'))

model.add(layer=Dense(units=256, activation='relu'))

model.add(layer=Dense(units=512, activation='relu'))

model.add(layer=Dense(units=1024, activation='relu'))

model.add(layer=Dense(units=512, activation='relu'))

model.add(layer=Dense(units=256, activation='relu'))

model.add(layer=Dense(units=128, activation='relu'))

model.add(layer=Dense(units=64, activation='relu'))

model.add(layer=Dense(units=32, activation='relu'))

model.add(layer=Dense(units=16, activation='relu'))

model.add(layer=Dense(units=1, activation='linear'))
```

[31]:

```
adam = Adam(learning_rate=7e-5)

model.compile(optimizer=adam, loss='mse', metrics=['mae', 'mape'])
```

[33]:

```
history = model.fit(X_train, y_train, epochs=10)
```

```
Epoch 1/10
366/366 [==============================] - 7s 20ms/step - loss: 21744080896.0000
- mae: 106028.5234 - mape: 23.9161
Epoch 2/10
366/366 [==============================] - 8s 22ms/step - loss: 21374562304.0000
- mae: 105025.5625 - mape: 23.7174
Epoch 3/10
366/366 [==============================] - 7s 20ms/step - loss: 19224690688.0000
```

```
- mae: 100039.5000 - mape: 22.9970
Epoch 4/10
366/366 [==============================] - 8s 23ms/step - loss: 18683500544.0000
- mae: 99278.1172 - mape: 22.8519
Epoch 5/10
366/366 [==============================] - 8s 22ms/step - loss: 18234394624.0000
- mae: 98242.8594 - mape: 22.7213
Epoch 6/10
366/366 [==============================] - 8s 22ms/step - loss: 20518877184.0000
- mae: 102074.7656 - mape: 23.2780
Epoch 7/10
366/366 [==============================] - 7s 19ms/step - loss: 22670522368.0000
- mae: 106679.6953 - mape: 23.9765
Epoch 8/10
366/366 [==============================] - 7s 18ms/step - loss: 18700814336.0000
- mae: 99613.9766 - mape: 22.8452
Epoch 9/10
366/366 [==============================] - 6s 17ms/step - loss: 18543321088.0000
- mae: 98677.9766 - mape: 22.7990
Epoch 10/10
366/366 [==============================] - 7s 18ms/step - loss: 19590918144.0000
- mae: 101507.4219 - mape: 23.2146
```

[34]:

```
y_pred = model.predict(X_test)
```

```
92/92 [==============================] - 1s 6ms/step
```

[35]:

```
y_pred
```

[35]:

```
array([[387830.6 ],
       [445728.56],
       [370873.7 ],
       …,
       [281568.03],
       [358895.47],
       [703242.4 ]], dtype=float32)
```

[36]:

```
y_test
```

[36]:

```
12149    640000
13581    650000
11595    325000
2769     373000
7393     355000
          …
7362     497000
11132    400000
142      366750
```

```
1405     276000
6184     569000
Name: Price, Length: 2924, dtype: int64
```

[37]:

```
pd.DataFrame({'Actual Value':y_test.values.flatten(), 'Predicted Value':y_pred.
 ↪flatten()})
```

[37]:

```
      Actual Value  Predicted Value
0           640000     387830.59375
1           650000     445728.56250
2           325000     370873.68750
3           373000     476821.93750
4           355000     465932.37500
…              …                …
2919        497000     604781.50000
2920        400000     339529.09375
2921        366750     281568.03125
2922        276000     358895.46875
2923        569000     703242.37500

[2924 rows x 2 columns]
```

[38]:

```
r2_score(y_pred=y_pred, y_true=y_test)
```

[38]:

```
0.5681449923201896
```

[ ]:

```

```