bostonhouse

April 28, 2024

[ ]:
```python
import numpy as np
import pandas as pd
df = pd.read_csv("/content/drive/MyDrive/DL/1_boston_housing.csv")
df.head()
```

[ ]:
```
      crim    zn  indus  chas    nox     rm   age     dis  rad  tax  ptratio  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

        b  lstat  MEDV
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90   5.33  36.2
```

[ ]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   crim     506 non-null    float64
 1   zn       506 non-null    float64
 2   indus    506 non-null    float64
 3   chas     506 non-null    int64
 4   nox      506 non-null    float64
 5   rm       506 non-null    float64
 6   age      506 non-null    float64
 7   dis      506 non-null    float64
 8   rad      506 non-null    int64
 9   tax      506 non-null    int64
 10  ptratio  506 non-null    float64
 11  b        506 non-null    float64
```

```
 12  lstat     506 non-null    float64
 13  MEDV      506 non-null    float64
dtypes: float64(11), int64(3)
memory usage: 55.5 KB
```

[ ]:
```
from sklearn.model_selection import train_test_split

X = df.loc[:, df.columns != 'MEDV']
y = df.loc[:, df.columns == 'MEDV']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=123)
```

[ ]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error

# Assuming you have already split your data into training and testing sets␣
 ↪(X_train, X_test, y_train, y_test)

# Linear Regression model
regressor = LinearRegression()

# Fitting the model
regressor.fit(X_train, y_train)
```

[ ]:
```
LinearRegression()
```

[ ]:
```
# Predictions on the test set
y_pred = regressor.predict(X_test)

# Calculating mean squared error and mean absolute error
mse_lr = mean_squared_error(y_test, y_pred)
mae_lr = mean_absolute_error(y_test, y_pred)

print('Mean squared error on test data: ', mse_lr)
print('Mean absolute error on test data: ', mae_lr)
```

```
Mean squared error on test data:  28.405854810508146
Mean absolute error on test data:  3.6913626771162664
```

[ ]:
```
from sklearn.preprocessing import StandardScaler
mms = StandardScaler()
mms.fit(X_train)
X_train = mms.transform(X_train)
X_test = mms.transform(X_test)
```

[ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
model = Sequential()

model.add(Dense(128, input_shape=(13, ), activation='relu', name='dense_1'))
model.add(Dense(64, activation='relu', name='dense_2'))
model.add(Dense(32, activation='relu', name='dense_3'))
model.add(Dense(16, activation='relu', name='dense_4'))
model.add(Dense(1, activation='relu', name='dense_output'))

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

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 128)               1792

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 32)                2080

 dense_4 (Dense)             (None, 16)                528

 dense_output (Dense)        (None, 1)                 17

=================================================================
Total params: 12673 (49.50 KB)
Trainable params: 12673 (49.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[ ]:
```
history = model.fit(X_train, y_train, epochs=100, validation_split=0.05,␣
 ↪verbose = 1)
```

```
Epoch 1/100
11/11 [==============================] - 2s 23ms/step - loss: 575.6537 - mae:
22.1221 - val_loss: 576.5711 - val_mae: 22.1259
Epoch 2/100
11/11 [==============================] - 0s 5ms/step - loss: 495.1672 - mae:
20.1684 - val_loss: 450.8975 - val_mae: 19.2964
Epoch 3/100
11/11 [==============================] - 0s 5ms/step - loss: 327.2580 - mae:
15.8599 - val_loss: 223.3898 - val_mae: 12.8459
Epoch 4/100
11/11 [==============================] - 0s 5ms/step - loss: 124.4332 - mae:
9.0221 - val_loss: 47.5648 - val_mae: 4.9631
Epoch 5/100
```

```
11/11 [==============================] - 0s 5ms/step - loss: 67.8966 - mae:
6.2678 - val_loss: 29.0603 - val_mae: 4.2359
Epoch 6/100
11/11 [==============================] - 0s 5ms/step - loss: 38.4034 - mae:
4.5157 - val_loss: 24.8668 - val_mae: 3.8740
Epoch 7/100
11/11 [==============================] - 0s 5ms/step - loss: 27.6280 - mae:
3.7044 - val_loss: 13.4622 - val_mae: 2.6501
Epoch 8/100
11/11 [==============================] - 0s 5ms/step - loss: 23.3305 - mae:
3.5865 - val_loss: 11.8256 - val_mae: 2.4234
Epoch 9/100
11/11 [==============================] - 0s 5ms/step - loss: 20.5428 - mae:
3.2447 - val_loss: 11.9806 - val_mae: 2.5500
Epoch 10/100
11/11 [==============================] - 0s 5ms/step - loss: 18.5107 - mae:
3.0898 - val_loss: 11.3341 - val_mae: 2.6661
Epoch 11/100
11/11 [==============================] - 0s 5ms/step - loss: 17.3425 - mae:
3.0028 - val_loss: 10.3772 - val_mae: 2.5477
Epoch 12/100
11/11 [==============================] - 0s 6ms/step - loss: 16.0525 - mae:
2.8680 - val_loss: 9.4586 - val_mae: 2.4696
Epoch 13/100
11/11 [==============================] - 0s 7ms/step - loss: 15.2510 - mae:
2.8649 - val_loss: 8.7481 - val_mae: 2.3798
Epoch 14/100
11/11 [==============================] - 0s 5ms/step - loss: 14.3053 - mae:
2.7309 - val_loss: 7.9584 - val_mae: 2.1728
Epoch 15/100
11/11 [==============================] - 0s 5ms/step - loss: 13.6324 - mae:
2.6289 - val_loss: 7.6652 - val_mae: 2.0998
Epoch 16/100
11/11 [==============================] - 0s 5ms/step - loss: 13.2467 - mae:
2.6358 - val_loss: 7.3439 - val_mae: 2.0469
Epoch 17/100
11/11 [==============================] - 0s 5ms/step - loss: 12.4381 - mae:
2.5150 - val_loss: 7.4387 - val_mae: 2.0482
Epoch 18/100
11/11 [==============================] - 0s 5ms/step - loss: 12.0310 - mae:
2.4696 - val_loss: 7.0743 - val_mae: 1.9703
Epoch 19/100
11/11 [==============================] - 0s 4ms/step - loss: 11.4895 - mae:
2.4199 - val_loss: 7.2182 - val_mae: 2.0069
Epoch 20/100
11/11 [==============================] - 0s 6ms/step - loss: 11.1225 - mae:
2.3653 - val_loss: 7.4251 - val_mae: 2.0441
Epoch 21/100
```

```
11/11 [==============================] - 0s 5ms/step - loss: 10.9209 - mae:
2.3783 - val_loss: 7.6012 - val_mae: 2.0459
Epoch 22/100
11/11 [==============================] - 0s 5ms/step - loss: 10.7109 - mae:
2.3320 - val_loss: 7.9825 - val_mae: 2.0880
Epoch 23/100
11/11 [==============================] - 0s 5ms/step - loss: 10.1821 - mae:
2.2560 - val_loss: 7.3414 - val_mae: 2.0100
Epoch 24/100
11/11 [==============================] - 0s 4ms/step - loss: 10.1566 - mae:
2.2720 - val_loss: 8.7596 - val_mae: 2.1805
Epoch 25/100
11/11 [==============================] - 0s 5ms/step - loss: 9.9464 - mae:
2.2422 - val_loss: 8.3408 - val_mae: 2.0807
Epoch 26/100
11/11 [==============================] - 0s 5ms/step - loss: 9.7240 - mae:
2.2135 - val_loss: 9.0558 - val_mae: 2.2543
Epoch 27/100
11/11 [==============================] - 0s 5ms/step - loss: 9.5356 - mae:
2.2084 - val_loss: 8.4129 - val_mae: 2.0596
Epoch 28/100
11/11 [==============================] - 0s 5ms/step - loss: 9.3356 - mae:
2.1948 - val_loss: 9.2000 - val_mae: 2.1455
Epoch 29/100
11/11 [==============================] - 0s 5ms/step - loss: 8.9372 - mae:
2.1112 - val_loss: 9.1660 - val_mae: 2.0979
Epoch 30/100
11/11 [==============================] - 0s 5ms/step - loss: 8.8070 - mae:
2.1203 - val_loss: 9.2944 - val_mae: 2.1207
Epoch 31/100
11/11 [==============================] - 0s 5ms/step - loss: 8.8012 - mae:
2.1513 - val_loss: 9.3776 - val_mae: 2.1199
Epoch 32/100
11/11 [==============================] - 0s 5ms/step - loss: 8.4997 - mae:
2.0662 - val_loss: 10.6675 - val_mae: 2.3048
Epoch 33/100
11/11 [==============================] - 0s 5ms/step - loss: 8.3794 - mae:
2.0680 - val_loss: 10.7615 - val_mae: 2.2569
Epoch 34/100
11/11 [==============================] - 0s 5ms/step - loss: 8.3587 - mae:
2.0804 - val_loss: 8.9162 - val_mae: 1.9760
Epoch 35/100
11/11 [==============================] - 0s 5ms/step - loss: 8.1260 - mae:
2.0279 - val_loss: 10.5830 - val_mae: 2.2423
Epoch 36/100
11/11 [==============================] - 0s 5ms/step - loss: 7.9090 - mae:
2.0223 - val_loss: 9.6021 - val_mae: 2.1441
Epoch 37/100
```

```
11/11 [==============================] - 0s 7ms/step - loss: 7.8128 - mae:
1.9900 - val_loss: 9.7770 - val_mae: 2.1223
Epoch 38/100
11/11 [==============================] - 0s 5ms/step - loss: 7.7199 - mae:
1.9792 - val_loss: 10.1151 - val_mae: 2.1829
Epoch 39/100
11/11 [==============================] - 0s 5ms/step - loss: 7.7019 - mae:
1.9872 - val_loss: 9.7837 - val_mae: 2.1473
Epoch 40/100
11/11 [==============================] - 0s 5ms/step - loss: 7.6286 - mae:
1.9536 - val_loss: 10.7361 - val_mae: 2.2084
Epoch 41/100
11/11 [==============================] - 0s 5ms/step - loss: 7.2406 - mae:
1.9143 - val_loss: 10.2387 - val_mae: 2.1447
Epoch 42/100
11/11 [==============================] - 0s 5ms/step - loss: 7.2091 - mae:
1.9076 - val_loss: 10.7103 - val_mae: 2.1416
Epoch 43/100
11/11 [==============================] - 0s 5ms/step - loss: 7.2630 - mae:
1.9026 - val_loss: 10.0700 - val_mae: 2.1940
Epoch 44/100
11/11 [==============================] - 0s 5ms/step - loss: 7.2288 - mae:
1.9100 - val_loss: 10.1638 - val_mae: 2.1425
Epoch 45/100
11/11 [==============================] - 0s 6ms/step - loss: 6.9424 - mae:
1.8672 - val_loss: 10.9349 - val_mae: 2.2297
Epoch 46/100
11/11 [==============================] - 0s 5ms/step - loss: 6.8648 - mae:
1.8691 - val_loss: 9.6392 - val_mae: 2.0353
Epoch 47/100
11/11 [==============================] - 0s 5ms/step - loss: 6.8744 - mae:
1.8489 - val_loss: 10.3861 - val_mae: 2.1083
Epoch 48/100
11/11 [==============================] - 0s 5ms/step - loss: 6.7826 - mae:
1.8468 - val_loss: 11.3101 - val_mae: 2.2088
Epoch 49/100
11/11 [==============================] - 0s 5ms/step - loss: 6.5257 - mae:
1.8130 - val_loss: 10.1045 - val_mae: 2.1258
Epoch 50/100
11/11 [==============================] - 0s 5ms/step - loss: 6.7292 - mae:
1.8268 - val_loss: 10.6866 - val_mae: 2.1276
Epoch 51/100
11/11 [==============================] - 0s 5ms/step - loss: 6.6449 - mae:
1.8339 - val_loss: 10.3599 - val_mae: 2.0862
Epoch 52/100
11/11 [==============================] - 0s 5ms/step - loss: 6.7290 - mae:
1.8440 - val_loss: 10.7999 - val_mae: 2.1365
Epoch 53/100
```

```
11/11 [==============================] - 0s 5ms/step - loss: 6.3929 - mae:
1.7827 - val_loss: 11.2738 - val_mae: 2.0973
Epoch 54/100
11/11 [==============================] - 0s 5ms/step - loss: 6.4309 - mae:
1.8115 - val_loss: 10.1321 - val_mae: 2.0261
Epoch 55/100
11/11 [==============================] - 0s 4ms/step - loss: 6.3405 - mae:
1.7849 - val_loss: 9.7666 - val_mae: 1.9875
Epoch 56/100
11/11 [==============================] - 0s 5ms/step - loss: 6.1300 - mae:
1.7605 - val_loss: 9.6525 - val_mae: 1.9901
Epoch 57/100
11/11 [==============================] - 0s 5ms/step - loss: 5.9482 - mae:
1.7179 - val_loss: 10.3535 - val_mae: 2.0041
Epoch 58/100
11/11 [==============================] - 0s 5ms/step - loss: 5.9568 - mae:
1.7317 - val_loss: 9.8177 - val_mae: 1.9761
Epoch 59/100
11/11 [==============================] - 0s 5ms/step - loss: 5.8106 - mae:
1.7148 - val_loss: 10.8562 - val_mae: 2.0966
Epoch 60/100
11/11 [==============================] - 0s 5ms/step - loss: 5.8679 - mae:
1.7391 - val_loss: 9.1167 - val_mae: 1.9043
Epoch 61/100
11/11 [==============================] - 0s 6ms/step - loss: 5.8781 - mae:
1.7187 - val_loss: 10.9807 - val_mae: 2.1239
Epoch 62/100
11/11 [==============================] - 0s 6ms/step - loss: 5.6600 - mae:
1.6960 - val_loss: 9.5215 - val_mae: 1.9513
Epoch 63/100
11/11 [==============================] - 0s 5ms/step - loss: 5.5321 - mae:
1.6771 - val_loss: 11.7740 - val_mae: 2.2756
Epoch 64/100
11/11 [==============================] - 0s 5ms/step - loss: 5.9410 - mae:
1.7411 - val_loss: 9.6136 - val_mae: 1.9935
Epoch 65/100
11/11 [==============================] - 0s 6ms/step - loss: 5.6162 - mae:
1.7117 - val_loss: 9.1349 - val_mae: 1.8898
Epoch 66/100
11/11 [==============================] - 0s 5ms/step - loss: 5.4396 - mae:
1.6527 - val_loss: 10.4305 - val_mae: 2.0992
Epoch 67/100
11/11 [==============================] - 0s 5ms/step - loss: 5.2318 - mae:
1.6257 - val_loss: 8.2997 - val_mae: 1.9269
Epoch 68/100
11/11 [==============================] - 0s 4ms/step - loss: 5.4168 - mae:
1.6331 - val_loss: 8.8860 - val_mae: 1.9678
Epoch 69/100
```

```
11/11 [==============================] - 0s 5ms/step - loss: 5.2594 - mae:
1.6124 - val_loss: 9.1836 - val_mae: 1.9825
Epoch 70/100
11/11 [==============================] - 0s 5ms/step - loss: 5.1120 - mae:
1.5919 - val_loss: 11.0205 - val_mae: 2.1775
Epoch 71/100
11/11 [==============================] - 0s 5ms/step - loss: 5.4511 - mae:
1.6281 - val_loss: 8.5903 - val_mae: 1.9352
Epoch 72/100
11/11 [==============================] - 0s 5ms/step - loss: 4.9280 - mae:
1.5735 - val_loss: 9.7532 - val_mae: 2.0531
Epoch 73/100
11/11 [==============================] - 0s 5ms/step - loss: 4.9212 - mae:
1.5683 - val_loss: 7.8694 - val_mae: 1.9178
Epoch 74/100
11/11 [==============================] - 0s 5ms/step - loss: 5.4069 - mae:
1.6631 - val_loss: 11.5396 - val_mae: 2.2655
Epoch 75/100
11/11 [==============================] - 0s 5ms/step - loss: 5.1972 - mae:
1.6270 - val_loss: 9.2957 - val_mae: 1.9973
Epoch 76/100
11/11 [==============================] - 0s 5ms/step - loss: 4.8972 - mae:
1.5688 - val_loss: 12.2088 - val_mae: 2.4134
Epoch 77/100
11/11 [==============================] - 0s 5ms/step - loss: 5.7669 - mae:
1.7409 - val_loss: 8.8129 - val_mae: 1.9120
Epoch 78/100
11/11 [==============================] - 0s 7ms/step - loss: 4.8708 - mae:
1.5567 - val_loss: 8.9943 - val_mae: 2.0436
Epoch 79/100
11/11 [==============================] - 0s 5ms/step - loss: 4.7634 - mae:
1.5516 - val_loss: 9.9061 - val_mae: 2.0723
Epoch 80/100
11/11 [==============================] - 0s 5ms/step - loss: 4.8491 - mae:
1.5463 - val_loss: 9.4317 - val_mae: 1.9966
Epoch 81/100
11/11 [==============================] - 0s 5ms/step - loss: 4.6415 - mae:
1.5455 - val_loss: 10.0888 - val_mae: 2.0971
Epoch 82/100
11/11 [==============================] - 0s 5ms/step - loss: 4.5750 - mae:
1.5229 - val_loss: 10.3211 - val_mae: 2.0763
Epoch 83/100
11/11 [==============================] - 0s 5ms/step - loss: 4.3582 - mae:
1.5045 - val_loss: 8.2224 - val_mae: 2.0125
Epoch 84/100
11/11 [==============================] - 0s 5ms/step - loss: 4.4517 - mae:
1.4808 - val_loss: 10.8078 - val_mae: 2.2010
Epoch 85/100
```

```
11/11 [==============================] - 0s 5ms/step - loss: 4.4134 - mae:
1.4978 - val_loss: 9.5990 - val_mae: 2.0491
Epoch 86/100
11/11 [==============================] - 0s 5ms/step - loss: 4.4947 - mae:
1.5214 - val_loss: 10.4532 - val_mae: 2.0968
Epoch 87/100
11/11 [==============================] - 0s 5ms/step - loss: 4.4224 - mae:
1.4686 - val_loss: 9.9915 - val_mae: 2.0324
Epoch 88/100
11/11 [==============================] - 0s 5ms/step - loss: 4.7466 - mae:
1.5333 - val_loss: 8.3806 - val_mae: 2.1140
Epoch 89/100
11/11 [==============================] - 0s 5ms/step - loss: 4.4178 - mae:
1.5181 - val_loss: 9.6633 - val_mae: 2.2019
Epoch 90/100
11/11 [==============================] - 0s 5ms/step - loss: 4.1679 - mae:
1.4646 - val_loss: 8.6010 - val_mae: 1.9789
Epoch 91/100
11/11 [==============================] - 0s 5ms/step - loss: 4.0542 - mae:
1.4193 - val_loss: 11.4210 - val_mae: 2.2112
Epoch 92/100
11/11 [==============================] - 0s 5ms/step - loss: 4.0425 - mae:
1.4288 - val_loss: 9.6377 - val_mae: 2.0078
Epoch 93/100
11/11 [==============================] - 0s 5ms/step - loss: 3.8605 - mae:
1.3804 - val_loss: 9.0980 - val_mae: 1.9980
Epoch 94/100
11/11 [==============================] - 0s 5ms/step - loss: 3.9189 - mae:
1.4435 - val_loss: 7.5765 - val_mae: 1.9845
Epoch 95/100
11/11 [==============================] - 0s 6ms/step - loss: 3.8924 - mae:
1.3981 - val_loss: 9.5147 - val_mae: 2.1248
Epoch 96/100
11/11 [==============================] - 0s 6ms/step - loss: 3.7366 - mae:
1.3856 - val_loss: 8.1894 - val_mae: 2.0050
Epoch 97/100
11/11 [==============================] - 0s 5ms/step - loss: 3.7707 - mae:
1.3896 - val_loss: 9.2924 - val_mae: 2.1421
Epoch 98/100
11/11 [==============================] - 0s 5ms/step - loss: 3.8668 - mae:
1.3844 - val_loss: 8.4724 - val_mae: 1.9684
Epoch 99/100
11/11 [==============================] - 0s 5ms/step - loss: 3.5257 - mae:
1.3200 - val_loss: 9.7832 - val_mae: 1.9984
Epoch 100/100
11/11 [==============================] - 0s 4ms/step - loss: 3.6604 - mae:
1.3539 - val_loss: 10.9911 - val_mae: 2.1160
```

[ ]:
```
mse_nn, mae_nn = model.evaluate(X_test, y_test)

print('Mean squared error on test data: ', mse_nn)
print('Mean absolute error on test data: ', mae_nn)
```

```
5/5 [==============================] - 0s 3ms/step - loss: 18.1131 - mae: 2.6427
Mean squared error on test data:  18.113122940063477
Mean absolute error on test data:  2.642714738845825
```

[ ]:
```
import sklearn
new_data = [[11.5779, 0,18.1, 0, 0.7, 5.036, 97, 1.77, 3, 666, 20.2, 396.9, 25.
↪68]]
new_data = sklearn.preprocessing.StandardScaler().fit_transform((new_data))
prediction = model.predict(new_data)
print("Predicted house price:", prediction)#9.7  ==394
```

```
1/1 [==============================] - 0s 198ms/step
Predicted house price: [[11.825968]]
```