bostan_housing

May 2, 2025

[16]:
```
# Step 1: Import libraries
import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

[18]:
```
# Step 2: Load dataset correctly (for space-separated values stored as strings)
raw_df = pd.read_csv("housing (1).csv", header=None)
```

[20]:
```
# Check if it's a single-column issue
if raw_df.shape[1] == 1:
    # Convert space-separated string rows into numerical columns
    df = raw_df[0].str.strip().str.split(r"\s+", expand=True)
    df = df.apply(pd.to_numeric)
else:
    df = raw_df.copy()  # If already structured correctly

print("Parsed Data Shape:", df.shape)
print("Preview:")
print(df.head())
```

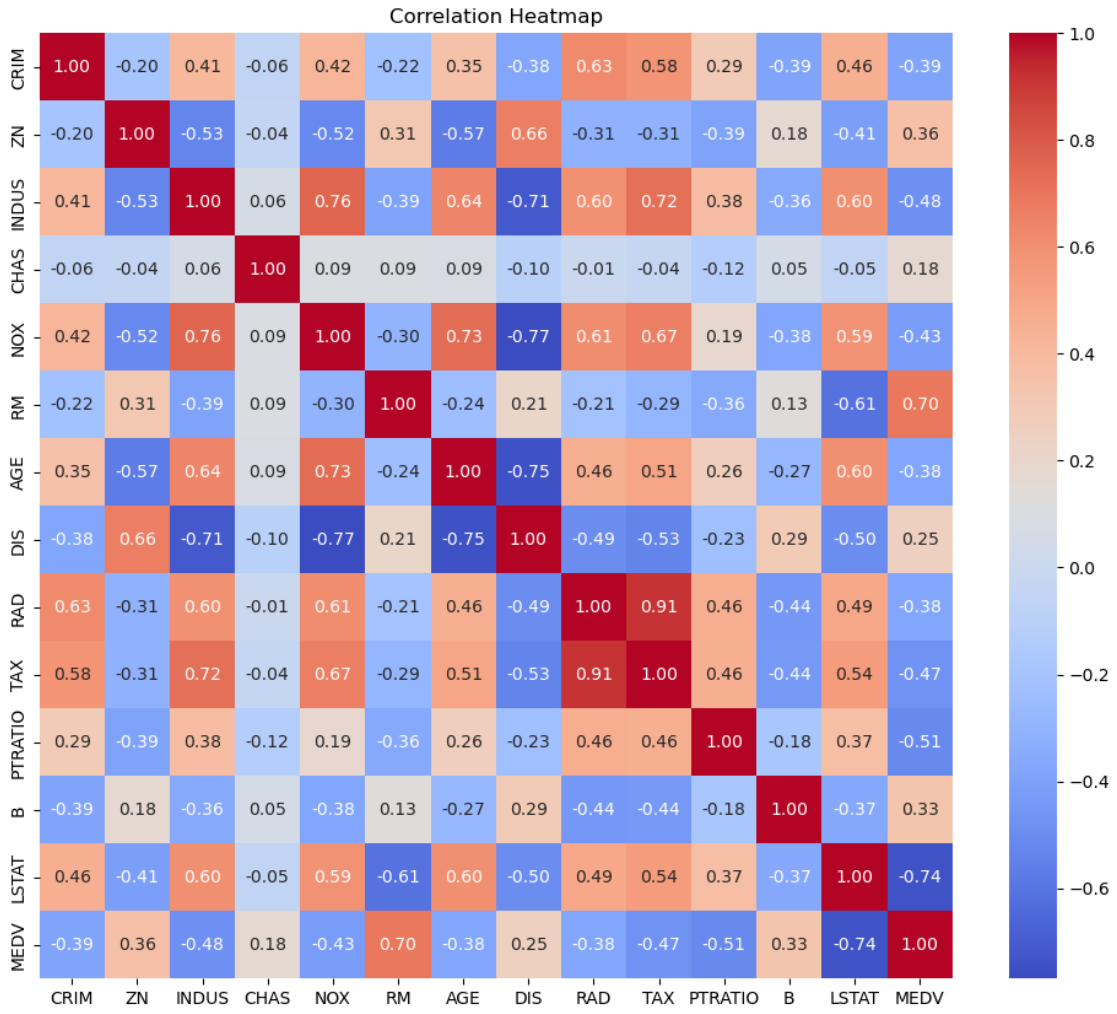
```
Parsed Data Shape: (506, 14)
Preview:
        0     1     2   3      4      5     6       7   8      9     10  \
0  0.00632  18.0  2.31   0  0.538  6.575  65.2  4.0900   1  296.0  15.3
1  0.02731   0.0  7.07   0  0.469  6.421  78.9  4.9671   2  242.0  17.8
2  0.02729   0.0  7.07   0  0.469  7.185  61.1  4.9671   2  242.0  17.8
3  0.03237   0.0  2.18   0  0.458  6.998  45.8  6.0622   3  222.0  18.7
4  0.06905   0.0  2.18   0  0.458  7.147  54.2  6.0622   3  222.0  18.7

       11    12    13
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
```

[22]:
```
# Step 3: Add column names (Boston Housing has 13 features + 1 target = 14 columns)
column_names = [
    "CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM", "AGE",
    "DIS", "RAD", "TAX", "PTRATIO", "B", "LSTAT", "MEDV"
]
df.columns = column_names
```

[24]:
```
# Step 4: Visualize correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



[26]:
```
# Step 5: Prepare features and target
X = df.drop("MEDV", axis=1).values
y = df["MEDV"].values
```

[28]:
```
# Step 6: Train-test split
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=42)
```

[30]:
```
# Step 7: Normalize data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[32]:
```
# Step 8: Build DNN model for regression
def build_model():
    model = keras.Sequential([
        layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
        layers.Dense(64, activation='relu'),
        layers.Dense(1)  # Output layer
    ])
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.01),
                  loss='mse', metrics=['mae'])
    return model
```

[34]:
```
# Step 9: Train the model
model = build_model()
history = model.fit(X_train, y_train, epochs=100, batch_size=8,
                    validation_split=0.2, verbose=1)
```

```
Epoch 1/100
/home/mca/anaconda3/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-05-02 11:35:07.051243: E
external/local_xla/xla/stream_executor/cuda/cuda_platform.cc:51] failed call to
cuInit: INTERNAL: CUDA error: Failed call to cuInit: CUDA_ERROR_NO_DEVICE: no
CUDA-capable device is detected
41/41             1s 6ms/step - loss:
319.9324 - mae: 13.9673 - val_loss: 27.3439 - val_mae: 3.6190
Epoch 2/100
41/41             0s 2ms/step - loss:
17.3947 - mae: 3.1808 - val_loss: 19.6372 - val_mae: 3.1946
Epoch 3/100
```

```
41/41            0s 2ms/step - loss:
17.7205 - mae: 3.0650 - val_loss: 19.9870 - val_mae: 2.9667
Epoch 4/100
41/41            0s 2ms/step - loss:
13.3470 - mae: 2.8039 - val_loss: 19.0064 - val_mae: 2.9996
Epoch 5/100
41/41            0s 2ms/step - loss:
15.5051 - mae: 2.9246 - val_loss: 13.4843 - val_mae: 2.6040
Epoch 6/100
41/41            0s 2ms/step - loss:
13.1173 - mae: 2.6480 - val_loss: 12.9658 - val_mae: 2.5890
Epoch 7/100
41/41            0s 1ms/step - loss:
11.7404 - mae: 2.4988 - val_loss: 14.6038 - val_mae: 2.9261
Epoch 8/100
41/41            0s 1ms/step - loss:
9.3069 - mae: 2.3025 - val_loss: 18.8185 - val_mae: 3.2698
Epoch 9/100
41/41            0s 2ms/step - loss:
11.2690 - mae: 2.6148 - val_loss: 25.4795 - val_mae: 3.7983
Epoch 10/100
41/41            0s 1ms/step - loss:
11.3581 - mae: 2.5488 - val_loss: 14.2088 - val_mae: 2.7491
Epoch 11/100
41/41            0s 1ms/step - loss:
7.3257 - mae: 2.0173 - val_loss: 13.4177 - val_mae: 2.6490
Epoch 12/100
41/41            0s 1ms/step - loss:
11.1729 - mae: 2.6390 - val_loss: 17.0438 - val_mae: 2.9481
Epoch 13/100
41/41            0s 2ms/step - loss:
9.7622 - mae: 2.3756 - val_loss: 17.8415 - val_mae: 3.2836
Epoch 14/100
41/41            0s 2ms/step - loss:
12.5278 - mae: 2.7520 - val_loss: 11.9773 - val_mae: 2.6152
Epoch 15/100
41/41            0s 2ms/step - loss:
11.8423 - mae: 2.4859 - val_loss: 13.3303 - val_mae: 2.6809
Epoch 16/100
41/41            0s 1ms/step - loss:
6.3379 - mae: 1.9480 - val_loss: 13.7991 - val_mae: 2.7016
Epoch 17/100
41/41            0s 2ms/step - loss:
7.2494 - mae: 2.0445 - val_loss: 11.3243 - val_mae: 2.4289
Epoch 18/100
41/41            0s 2ms/step - loss:
8.0657 - mae: 2.1052 - val_loss: 10.9418 - val_mae: 2.4897
Epoch 19/100
```

```
41/41            0s 1ms/step - loss:
11.5384 - mae: 2.4711 - val_loss: 12.1729 - val_mae: 2.6407
Epoch 20/100
41/41            0s 1ms/step - loss:
8.4679 - mae: 2.1825 - val_loss: 12.2624 - val_mae: 2.5919
Epoch 21/100
41/41            0s 1ms/step - loss:
10.3431 - mae: 2.4620 - val_loss: 11.1585 - val_mae: 2.4505
Epoch 22/100
41/41            0s 1ms/step - loss:
7.4493 - mae: 1.9656 - val_loss: 10.9414 - val_mae: 2.4288
Epoch 23/100
41/41            0s 1ms/step - loss:
7.7861 - mae: 2.0773 - val_loss: 11.0525 - val_mae: 2.4743
Epoch 24/100
41/41            0s 2ms/step - loss:
8.4833 - mae: 2.1895 - val_loss: 11.3359 - val_mae: 2.5232
Epoch 25/100
41/41            0s 2ms/step - loss:
11.0244 - mae: 2.4949 - val_loss: 12.6245 - val_mae: 2.5872
Epoch 26/100
41/41            0s 1ms/step - loss:
7.5265 - mae: 2.1650 - val_loss: 9.2525 - val_mae: 2.2646
Epoch 27/100
41/41            0s 1ms/step - loss:
8.5461 - mae: 2.2514 - val_loss: 9.3610 - val_mae: 2.2933
Epoch 28/100
41/41            0s 1ms/step - loss:
7.2339 - mae: 2.1117 - val_loss: 9.2666 - val_mae: 2.1730
Epoch 29/100
41/41            0s 2ms/step - loss:
4.4710 - mae: 1.5652 - val_loss: 11.9863 - val_mae: 2.5702
Epoch 30/100
41/41            0s 2ms/step - loss:
5.5113 - mae: 1.8272 - val_loss: 9.1226 - val_mae: 2.2194
Epoch 31/100
41/41            0s 2ms/step - loss:
6.0594 - mae: 1.8865 - val_loss: 12.1785 - val_mae: 2.4660
Epoch 32/100
41/41            0s 2ms/step - loss:
6.3699 - mae: 1.9076 - val_loss: 24.7461 - val_mae: 3.9883
Epoch 33/100
41/41            0s 1ms/step - loss:
11.3223 - mae: 2.5465 - val_loss: 11.0210 - val_mae: 2.5581
Epoch 34/100
41/41            0s 1ms/step - loss:
9.2280 - mae: 2.3285 - val_loss: 10.2462 - val_mae: 2.3896
Epoch 35/100
```

```
41/41            0s 1ms/step - loss:
6.6346 - mae: 1.8926 - val_loss: 12.2377 - val_mae: 2.5306
Epoch 36/100
41/41            0s 1ms/step - loss:
5.8532 - mae: 1.7929 - val_loss: 11.9790 - val_mae: 2.4933
Epoch 37/100
41/41            0s 2ms/step - loss:
5.5414 - mae: 1.8741 - val_loss: 7.5856 - val_mae: 2.0161
Epoch 38/100
41/41            0s 1ms/step - loss:
4.5961 - mae: 1.6458 - val_loss: 8.3469 - val_mae: 2.1261
Epoch 39/100
41/41            0s 1ms/step - loss:
5.6217 - mae: 1.7688 - val_loss: 8.5199 - val_mae: 2.0974
Epoch 40/100
41/41            0s 2ms/step - loss:
4.9564 - mae: 1.5696 - val_loss: 10.8117 - val_mae: 2.4358
Epoch 41/100
41/41            0s 2ms/step - loss:
5.4053 - mae: 1.7230 - val_loss: 9.2621 - val_mae: 2.2221
Epoch 42/100
41/41            0s 2ms/step - loss:
6.1117 - mae: 1.8495 - val_loss: 15.7769 - val_mae: 2.8744
Epoch 43/100
41/41            0s 2ms/step - loss:
7.3086 - mae: 2.1190 - val_loss: 11.2469 - val_mae: 2.4245
Epoch 44/100
41/41            0s 1ms/step - loss:
4.3315 - mae: 1.5851 - val_loss: 8.4981 - val_mae: 2.0677
Epoch 45/100
41/41            0s 2ms/step - loss:
5.3525 - mae: 1.6803 - val_loss: 9.4468 - val_mae: 2.3390
Epoch 46/100
41/41            0s 2ms/step - loss:
4.5016 - mae: 1.6426 - val_loss: 9.0253 - val_mae: 2.1753
Epoch 47/100
41/41            0s 2ms/step - loss:
5.4079 - mae: 1.7840 - val_loss: 11.2490 - val_mae: 2.4977
Epoch 48/100
41/41            0s 1ms/step - loss:
5.7589 - mae: 1.8365 - val_loss: 10.3886 - val_mae: 2.3354
Epoch 49/100
41/41            0s 2ms/step - loss:
4.8627 - mae: 1.6322 - val_loss: 9.0542 - val_mae: 2.1163
Epoch 50/100
41/41            0s 2ms/step - loss:
5.4266 - mae: 1.6828 - val_loss: 9.2126 - val_mae: 2.1780
Epoch 51/100
```

```
41/41            0s 2ms/step - loss:
4.7737 - mae: 1.6347 - val_loss: 10.3785 - val_mae: 2.1943
Epoch 52/100
41/41            0s 2ms/step - loss:
4.8856 - mae: 1.6347 - val_loss: 11.2464 - val_mae: 2.4273
Epoch 53/100
41/41            0s 1ms/step - loss:
5.0940 - mae: 1.6339 - val_loss: 11.6111 - val_mae: 2.4547
Epoch 54/100
41/41            0s 1ms/step - loss:
5.5240 - mae: 1.6828 - val_loss: 9.2453 - val_mae: 2.2909
Epoch 55/100
41/41            0s 2ms/step - loss:
4.3654 - mae: 1.5269 - val_loss: 10.3268 - val_mae: 2.3402
Epoch 56/100
41/41            0s 1ms/step - loss:
4.5995 - mae: 1.6500 - val_loss: 12.6699 - val_mae: 2.6568
Epoch 57/100
41/41            0s 1ms/step - loss:
5.4062 - mae: 1.7729 - val_loss: 9.4964 - val_mae: 2.1968
Epoch 58/100
41/41            0s 1ms/step - loss:
5.4774 - mae: 1.7777 - val_loss: 10.1531 - val_mae: 2.4331
Epoch 59/100
41/41            0s 1ms/step - loss:
4.5825 - mae: 1.5933 - val_loss: 10.2249 - val_mae: 2.3166
Epoch 60/100
41/41            0s 1ms/step - loss:
5.7329 - mae: 1.7802 - val_loss: 7.3830 - val_mae: 1.9311
Epoch 61/100
41/41            0s 1ms/step - loss:
5.5850 - mae: 1.7988 - val_loss: 11.3854 - val_mae: 2.4431
Epoch 62/100
41/41            0s 1ms/step - loss:
7.5431 - mae: 2.0686 - val_loss: 10.6977 - val_mae: 2.5021
Epoch 63/100
41/41            0s 1ms/step - loss:
5.2118 - mae: 1.6848 - val_loss: 8.1884 - val_mae: 2.2010
Epoch 64/100
41/41            0s 1ms/step - loss:
5.2930 - mae: 1.8081 - val_loss: 8.8637 - val_mae: 2.1484
Epoch 65/100
41/41            0s 1ms/step - loss:
5.7019 - mae: 1.8090 - val_loss: 8.2546 - val_mae: 2.0457
Epoch 66/100
41/41            0s 2ms/step - loss:
4.1495 - mae: 1.5282 - val_loss: 9.9925 - val_mae: 2.3030
Epoch 67/100
```

```
41/41            0s 2ms/step - loss:
3.3629 - mae: 1.3740 - val_loss: 10.2545 - val_mae: 2.3429
Epoch 68/100
41/41            0s 1ms/step - loss:
3.6080 - mae: 1.4219 - val_loss: 9.9093 - val_mae: 2.3464
Epoch 69/100
41/41            0s 1ms/step - loss:
6.4094 - mae: 1.9128 - val_loss: 10.0567 - val_mae: 2.2321
Epoch 70/100
41/41            0s 1ms/step - loss:
4.0308 - mae: 1.4446 - val_loss: 8.1290 - val_mae: 2.0891
Epoch 71/100
41/41            0s 1ms/step - loss:
5.3237 - mae: 1.8089 - val_loss: 8.7754 - val_mae: 2.1925
Epoch 72/100
41/41            0s 2ms/step - loss:
5.0475 - mae: 1.7148 - val_loss: 7.8393 - val_mae: 1.9689
Epoch 73/100
41/41            0s 1ms/step - loss:
3.2192 - mae: 1.3527 - val_loss: 9.2786 - val_mae: 2.0950
Epoch 74/100
41/41            0s 2ms/step - loss:
3.6460 - mae: 1.4818 - val_loss: 9.0191 - val_mae: 2.0632
Epoch 75/100
41/41            0s 2ms/step - loss:
3.9772 - mae: 1.5188 - val_loss: 9.3464 - val_mae: 2.2728
Epoch 76/100
41/41            0s 2ms/step - loss:
4.0937 - mae: 1.5400 - val_loss: 8.6289 - val_mae: 2.2404
Epoch 77/100
41/41            0s 1ms/step - loss:
6.0207 - mae: 1.8134 - val_loss: 11.6931 - val_mae: 2.3121
Epoch 78/100
41/41            0s 2ms/step - loss:
6.0921 - mae: 1.8601 - val_loss: 9.0944 - val_mae: 2.1911
Epoch 79/100
41/41            0s 2ms/step - loss:
4.7998 - mae: 1.6997 - val_loss: 7.9846 - val_mae: 2.0429
Epoch 80/100
41/41            0s 2ms/step - loss:
3.0239 - mae: 1.3059 - val_loss: 8.2974 - val_mae: 2.0058
Epoch 81/100
41/41            0s 1ms/step - loss:
5.7928 - mae: 1.8351 - val_loss: 10.1681 - val_mae: 2.2465
Epoch 82/100
41/41            0s 2ms/step - loss:
5.7331 - mae: 1.7341 - val_loss: 8.5992 - val_mae: 2.2129
Epoch 83/100
```

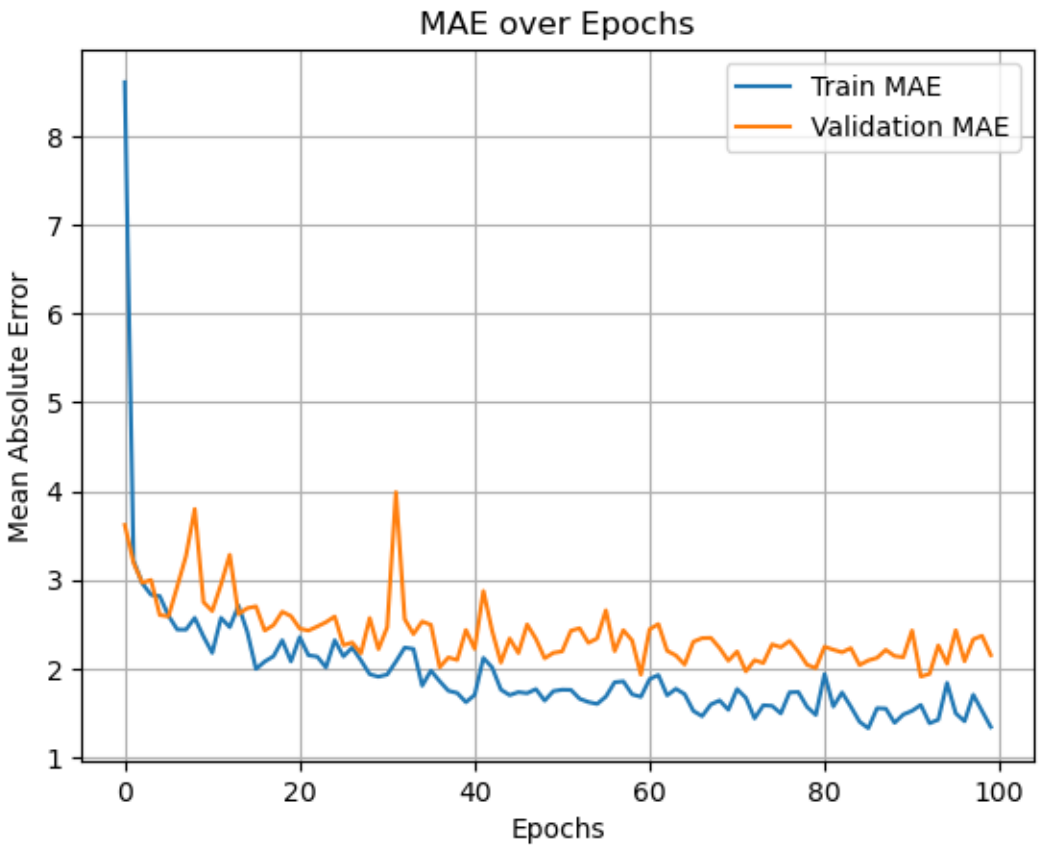
```
41/41            0s 1ms/step - loss:
5.6271 - mae: 1.7616 - val_loss: 9.0712 - val_mae: 2.1841
Epoch 84/100
41/41            0s 2ms/step - loss:
3.9910 - mae: 1.5643 - val_loss: 10.1639 - val_mae: 2.2305
Epoch 85/100
41/41            0s 1ms/step - loss:
3.6437 - mae: 1.3921 - val_loss: 8.5444 - val_mae: 2.0404
Epoch 86/100
41/41            0s 1ms/step - loss:
2.8376 - mae: 1.2851 - val_loss: 9.0183 - val_mae: 2.0924
Epoch 87/100
41/41            0s 2ms/step - loss:
4.2004 - mae: 1.5464 - val_loss: 9.4921 - val_mae: 2.1217
Epoch 88/100
41/41            0s 2ms/step - loss:
3.8771 - mae: 1.4979 - val_loss: 9.8725 - val_mae: 2.2116
Epoch 89/100
41/41            0s 2ms/step - loss:
2.5570 - mae: 1.1857 - val_loss: 8.7245 - val_mae: 2.1419
Epoch 90/100
41/41            0s 1ms/step - loss:
3.7726 - mae: 1.4928 - val_loss: 9.3226 - val_mae: 2.1285
Epoch 91/100
41/41            0s 2ms/step - loss:
3.6944 - mae: 1.4343 - val_loss: 10.6567 - val_mae: 2.4295
Epoch 92/100
41/41            0s 2ms/step - loss:
4.4655 - mae: 1.6116 - val_loss: 7.4433 - val_mae: 1.9109
Epoch 93/100
41/41            0s 2ms/step - loss:
4.5129 - mae: 1.5259 - val_loss: 7.0446 - val_mae: 1.9404
Epoch 94/100
41/41            0s 1ms/step - loss:
3.9181 - mae: 1.4238 - val_loss: 10.2883 - val_mae: 2.2613
Epoch 95/100
41/41            0s 2ms/step - loss:
6.3012 - mae: 1.9315 - val_loss: 8.4652 - val_mae: 2.0554
Epoch 96/100
41/41            0s 2ms/step - loss:
3.5171 - mae: 1.4425 - val_loss: 12.0901 - val_mae: 2.4329
Epoch 97/100
41/41            0s 2ms/step - loss:
4.1454 - mae: 1.4958 - val_loss: 9.0876 - val_mae: 2.0834
Epoch 98/100
41/41            0s 2ms/step - loss:
3.7486 - mae: 1.4168 - val_loss: 10.6720 - val_mae: 2.3304
Epoch 99/100
```

```
41/41              0s 2ms/step - loss:
4.1874 - mae: 1.5239 - val_loss: 10.7697 - val_mae: 2.3699
Epoch 100/100
41/41              0s 2ms/step - loss:
4.3205 - mae: 1.5135 - val_loss: 10.7430 - val_mae: 2.1524
```

[36]:
```python
# Step 10: Evaluate the model
test_mse, test_mae = model.evaluate(X_test, y_test, verbose=0)
print(f"\nTest MAE: {test_mae:.2f}")
```

```
Test MAE: 2.13
```

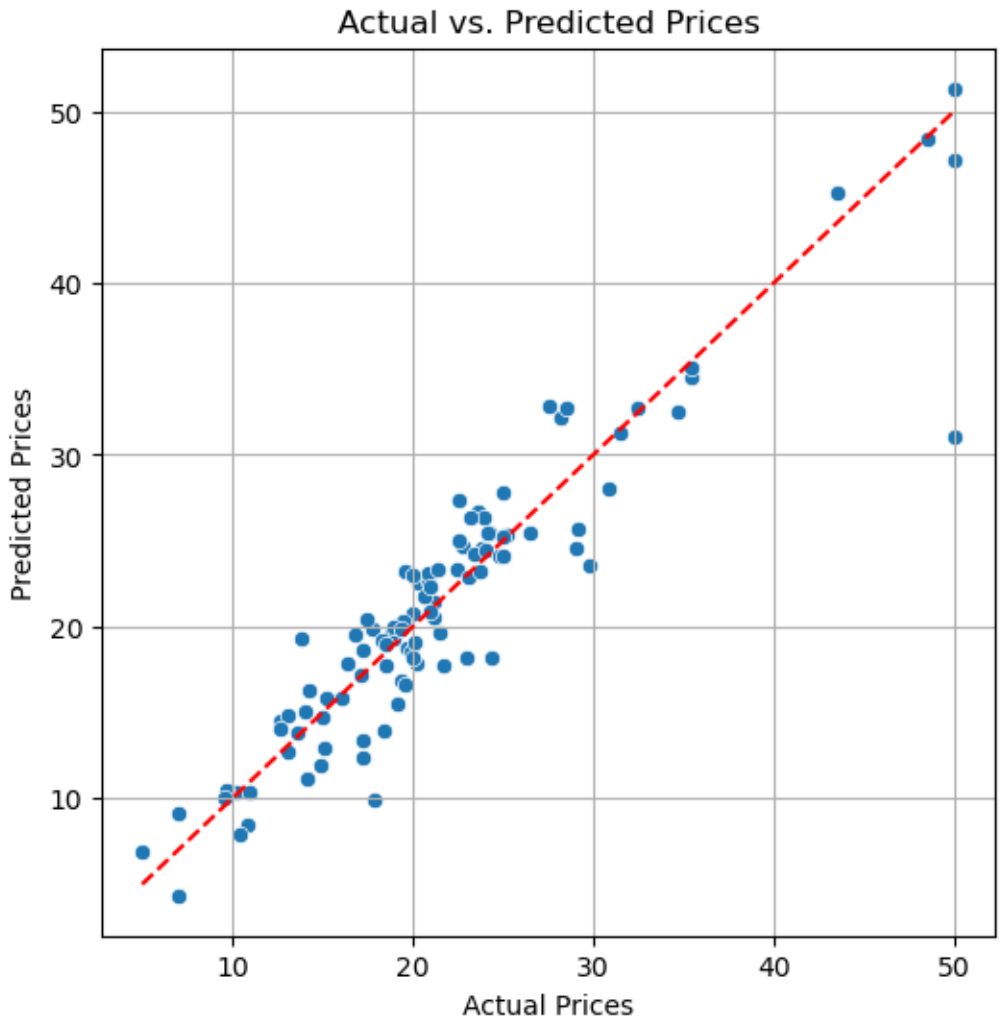
[38]:
```python
# Step 11: Plot MAE history
plt.plot(history.history['mae'], label='Train MAE')
plt.plot(history.history['val_mae'], label='Validation MAE')
plt.xlabel('Epochs')
plt.ylabel('Mean Absolute Error')
plt.title('MAE over Epochs')
plt.legend()
plt.grid(True)
plt.show()
```



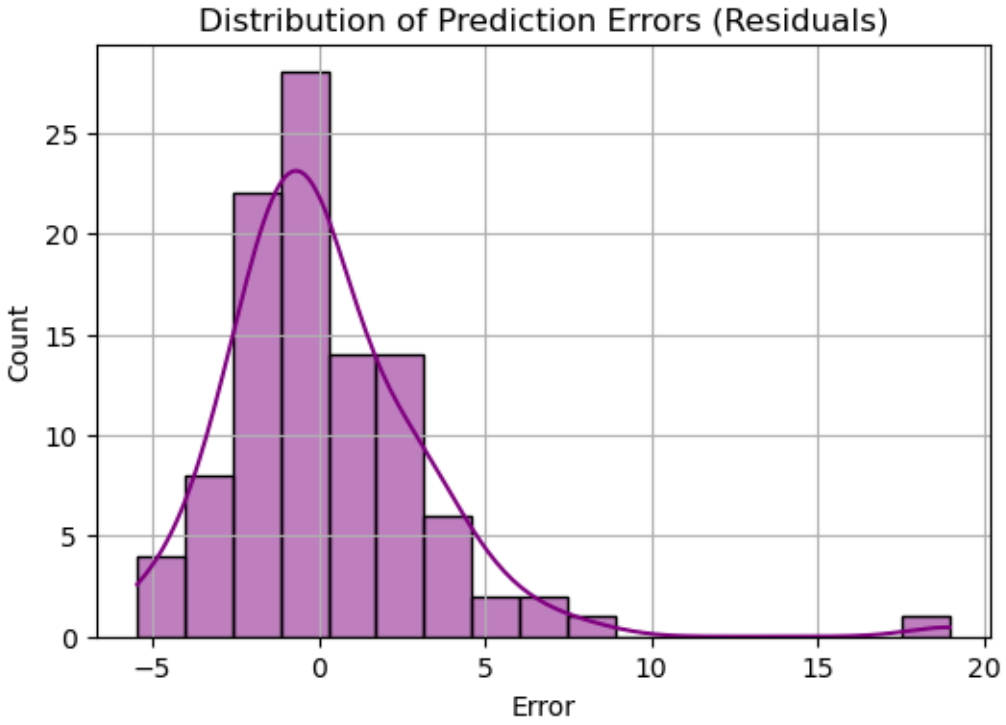
[40]:
```
# Step 12: Predictions
predictions = model.predict(X_test).flatten()
```

```
4/4             0s 9ms/step
```

[42]:
```
# Scatter plot of true vs predicted values
plt.figure(figsize=(6, 6))
sns.scatterplot(x=y_test, y=predictions)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs. Predicted Prices')
plt.grid(True)
plt.show()
```



[44]:
```
# Residual plot
residuals = y_test - predictions
plt.figure(figsize=(6, 4))
sns.histplot(residuals, kde=True, color='purple')
plt.title('Distribution of Prediction Errors (Residuals)')
plt.xlabel('Error')
plt.grid(True)
plt.show()
```



[46]:
```
# Show some sample predictions
print("\nSample predictions:")
for i in range(5):
    print(f"Predicted: {predictions[i]:.2f}, Actual: {y_test[i]}")
```

```
Sample predictions:
Predicted: 26.72, Actual: 23.6
Predicted: 32.73, Actual: 32.4
Predicted: 13.84, Actual: 13.6
Predicted: 24.71, Actual: 22.8
Predicted: 15.88, Actual: 16.1
```

[ ]: