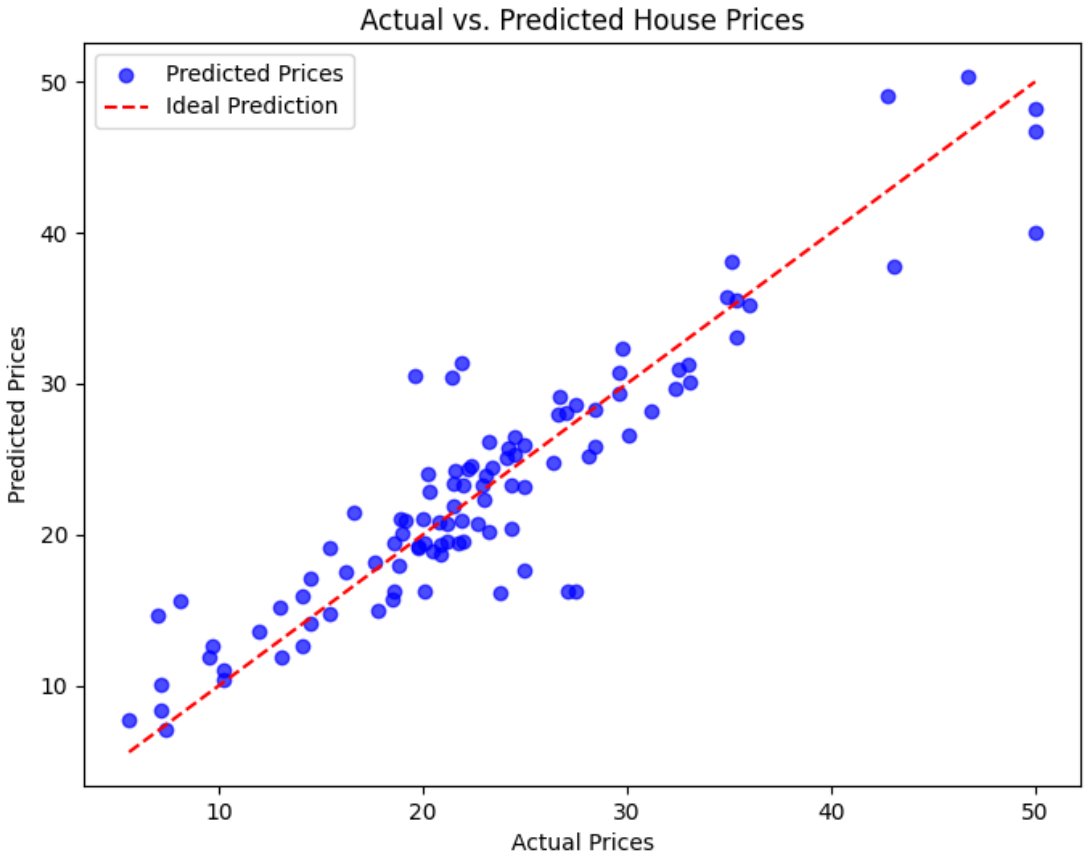
Practical 01

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import boston_housing

(X_train, y_train), (X_test, y_test) = boston_housing.load_data()
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = Sequential([
Dense(64, input_shape=(X_train.shape[1],), activation='relu'),
Dense(32, activation='relu'),
Dense(16, activation='relu'),
Dense(1, activation='linear')
])
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='mse',
metrics=['mae'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=500,
batch_size=10, verbose=1)
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"\n Mean Absolute Error (MAE): {mae:.2f}")
print(f" Mean Squared Error (MSE): {mse:.2f}")
print(f" Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f" R² Score: {r2:.2f} (Higher is better, max = 1)")
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, alpha=0.7, color='blue', label="Predicted Prices")
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linestyle="dashed", label="Ideal Prediction")
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs. Predicted House Prices")
plt.legend()
plt.show()
```

```
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 775us/step - loss: 0.6310 - mae: 0.5742 - val_loss: 13.4696 - val_mae: 2.6002
Epoch 497/500
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 780us/step - loss: 0.9240 - mae: 0.6783 - val_loss: 13.2371 - val_mae: 2.5843
Epoch 498/500
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 801us/step - loss: 0.6083 - mae: 0.5728 - val_loss: 12.6806 - val_mae: 2.5135
Epoch 499/500
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 793us/step - loss: 0.6503 - mae: 0.6094 - val_loss: 12.8712 - val_mae: 2.5379
Epoch 500/500
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 791us/step - loss: 0.5242 - mae: 0.5089 - val_loss: 12.8330 - val_mae: 2.5717
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step

 Mean Absolute Error (MAE): 2.57
 Mean Squared Error (MSE): 12.83
 Root Mean Squared Error (RMSE): 3.58
 R² Score: 0.85 (Higher is better, max = 1)
2025-04-22 10:36:34.007 Python[10392:8492683] +[IMKClient subclass]: chose IMKClient_Modern
2025-04-22 10:36:34.007 Python[10392:8492683] +[IMKInputSession subclass]: chose IMKInputSession_Modern
```


Actual vs. Predicted House Prices
Predicted Prices
Ideal Prediction
Predicted Prices
Actual Prices
10
20
30
40
50