

```
!pip install scikit-learn==1.1.1
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from sklearn.datasets import load_boston
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
Collecting scikit-learn==1.1.1
  Downloading scikit_learn-1.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (30.4 MB)
    30.4/30.4 MB 30.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1.1.1) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1.1.1) (1.11.4)
Requirement already satisfied: joblib>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1.1.1) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1.1.1) (3.4.0)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.2.2
    Uninstalling scikit-learn-1.2.2:
      Successfully uninstalled scikit-learn-1.2.2
  ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
  bigframes 1.2.0 requires scikit-learn>=1.2.2, but you have scikit-learn 1.1.1 which is incompatible.
  Successfully installed scikit-learn-1.1.1
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
# Load the Boston Housing Price Prediction dataset
boston = load_boston()
X = boston.data
y = boston.target
```

```
# Normalize the input data
scaler = StandardScaler().fit(X)
X = scaler.transform(X)
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Define the model
model = Sequential()
model.add(Dense(64, input_dim=13, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))
```

```
# Compile the model
model.compile(loss='mean_squared_error', optimizer='adam')
```

```
# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=16, verbose=0)
```

```
<keras.src.callbacks.History at 0x7c234e1df580>
```

```
# Evaluate the model on the testing set
mse = model.evaluate(X_test, y_test, verbose=0)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 10.95145034790039
```

