**1. Data importing and preprocessing**

* Import cleaned data
* Required libraries are imported
* Check for GPU availability to be used by the model
* Data split into features (X) and target (y)
* Data split into teain\_validation and test
* Feature scaling by StandardScaler

Collecting keras-tuner

Downloading keras\_tuner-1.4.7-py3-none-any.whl.metadata (5.4 kB)

Requirement already satisfied: keras in /usr/local/lib/python3.12/dist-packages (from keras-tuner) (3.10.0)

Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from keras-tuner) (25.0)

Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from keras-tuner) (2.32.4)

Collecting kt-legacy (from keras-tuner)

Downloading kt\_legacy-1.0.5-py3-none-any.whl.metadata (221 bytes)

Requirement already satisfied: absl-py in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (1.4.0)

Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (2.0.2)

Requirement already satisfied: rich in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (13.9.4)

Requirement already satisfied: namex in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (0.1.0)

Requirement already satisfied: h5py in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (3.14.0)

Requirement already satisfied: optree in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (0.17.0)

Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.12/dist-packages (from keras->keras-tuner) (0.5.3)

Requirement already satisfied: charset\_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->keras-tuner) (3.4.3)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->keras-tuner) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->keras-tuner) (2.5.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->keras-tuner) (2025.8.3)

Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from optree->keras->keras-tuner) (4.15.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich->keras->keras-tuner) (4.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich->keras->keras-tuner) (2.19.2)

Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0->rich->keras->keras-tuner) (0.1.2)

Downloading keras\_tuner-1.4.7-py3-none-any.whl (129 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.1/129.1 kB 3.6 MB/s eta 0:00:00

Downloading kt\_legacy-1.0.5-py3-none-any.whl (9.6 kB)

Installing collected packages: kt-legacy, keras-tuner

Successfully installed keras-tuner-1.4.7 kt-legacy-1.0.5

GPU Available to use: []

Mounted at /content/drive

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

First five rows of X and y:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

First five rows of X:

Inventory Level Units Sold Units Ordered Price Discount \

0 231 127 55 33.50 20

1 204 150 66 63.01 20

2 102 65 51 27.99 10

3 469 61 164 32.72 10

4 166 14 135 73.64 0

Holiday/Promotion Competitor Pricing Store ID\_S001 Store ID\_S002 \

0 0 29.69 1.0 0.0

1 0 66.16 1.0 0.0

2 1 31.32 1.0 0.0

3 1 34.74 1.0 0.0

4 0 68.95 1.0 0.0

Store ID\_S003 ... Weather Condition\_Cloudy Weather Condition\_Rainy \

0 0.0 ... 0.0 1.0

1 0.0 ... 0.0 0.0

2 0.0 ... 0.0 0.0

3 0.0 ... 1.0 0.0

4 0.0 ... 0.0 0.0

Weather Condition\_Snowy Weather Condition\_Sunny Seasonality\_Autumn \

0 0.0 0.0 1.0

1 0.0 1.0 1.0

2 0.0 1.0 0.0

3 0.0 0.0 1.0

4 0.0 1.0 0.0

Seasonality\_Spring Seasonality\_Summer Seasonality\_Winter dayofweek\_sin \

0 0.0 0.0 0.0 -0.974928

1 0.0 0.0 0.0 -0.974928

2 0.0 1.0 0.0 -0.974928

3 0.0 0.0 0.0 -0.974928

4 0.0 1.0 0.0 -0.974928

dayofweek\_cos

0 -0.222521

1 -0.222521

2 -0.222521

3 -0.222521

4 -0.222521

[5 rows x 51 columns]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

First five rows of y:

0 135.47

1 144.04

2 74.02

3 62.18

4 9.26

Name: Demand Forecast, dtype: float64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Results of above cell show that the data imported properly**

# 2. Model building - TensorFlow

Trial 10 Complete [00h 00m 35s]

val\_mae: 11.167926788330078

Best val\_mae So Far: 7.728631019592285

Total elapsed time: 00h 15m 31s

Epoch 1/100

/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/input\_layer.py:27: UserWarning: Argument `input\_shape` is deprecated. Use `shape` instead.

warnings.warn(

/usr/local/lib/python3.12/dist-packages/keras/src/saving/saving\_lib.py:802: UserWarning: Skipping variable loading for optimizer 'adam', because it has 2 variables whereas the saved optimizer has 10 variables.

saveable.load\_own\_variables(weights\_store.get(inner\_path))

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 96.5812 - mae: 8.1424 - val\_loss: 85.1180 - val\_mae: 7.8500

Epoch 2/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 97.2747 - mae: 8.1811 - val\_loss: 83.0307 - val\_mae: 7.7768

Epoch 3/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 97.2103 - mae: 8.1775 - val\_loss: 82.5364 - val\_mae: 7.7719

Epoch 4/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **7s** 5ms/step - loss: 95.9238 - mae: 8.1125 - val\_loss: 81.8570 - val\_mae: 7.7419

Epoch 5/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 95.6307 - mae: 8.1060 - val\_loss: 81.3262 - val\_mae: 7.7114

Epoch 6/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 96.2873 - mae: 8.1371 - val\_loss: 83.4812 - val\_mae: 7.8094

Epoch 7/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 95.4460 - mae: 8.1055 - val\_loss: 81.7139 - val\_mae: 7.7261

Epoch 8/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 95.1676 - mae: 8.0993 - val\_loss: 81.0486 - val\_mae: 7.7242

Epoch 9/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 2ms/step - loss: 94.9000 - mae: 8.0657 - val\_loss: 81.7404 - val\_mae: 7.7197

Epoch 10/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 95.0163 - mae: 8.0783 - val\_loss: 82.4981 - val\_mae: 7.7570

Epoch 11/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 95.6776 - mae: 8.1236 - val\_loss: 83.4327 - val\_mae: 7.7973

Epoch 12/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 95.4400 - mae: 8.0892 - val\_loss: 82.1225 - val\_mae: 7.7476

Epoch 13/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 95.1999 - mae: 8.0966 - val\_loss: 85.9465 - val\_mae: 7.8570

Epoch 14/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 4ms/step - loss: 94.8196 - mae: 8.1017 - val\_loss: 82.0767 - val\_mae: 7.7554

Epoch 15/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.2566 - mae: 8.0633 - val\_loss: 81.4109 - val\_mae: 7.7262

Epoch 16/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 2ms/step - loss: 96.1629 - mae: 8.1509 - val\_loss: 81.0670 - val\_mae: 7.7098

Epoch 17/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.1518 - mae: 8.0391 - val\_loss: 82.3684 - val\_mae: 7.7556

Epoch 18/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 94.3440 - mae: 8.0495 - val\_loss: 80.8492 - val\_mae: 7.6984

Epoch 19/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.3964 - mae: 8.0564 - val\_loss: 82.6917 - val\_mae: 7.7614

Epoch 20/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 2ms/step - loss: 93.8802 - mae: 8.0308 - val\_loss: 81.9259 - val\_mae: 7.7433

Epoch 21/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 4ms/step - loss: 93.6724 - mae: 8.0185 - val\_loss: 82.1783 - val\_mae: 7.7491

Epoch 22/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 93.9921 - mae: 8.0269 - val\_loss: 82.0545 - val\_mae: 7.7486

Epoch 23/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 2ms/step - loss: 92.6336 - mae: 8.0097 - val\_loss: 81.7853 - val\_mae: 7.7338

Epoch 24/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 93.7303 - mae: 8.0441 - val\_loss: 82.2123 - val\_mae: 7.7468

Epoch 25/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 94.7466 - mae: 8.0835 - val\_loss: 81.9548 - val\_mae: 7.7430

Epoch 26/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.0423 - mae: 8.0795 - val\_loss: 80.3896 - val\_mae: 7.6798

Epoch 27/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 2ms/step - loss: 93.3464 - mae: 8.0298 - val\_loss: 82.7828 - val\_mae: 7.7809

Epoch 28/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 4ms/step - loss: 93.3772 - mae: 8.0108 - val\_loss: 80.9295 - val\_mae: 7.7141

Epoch 29/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 93.5043 - mae: 8.0370 - val\_loss: 81.4548 - val\_mae: 7.7257

Epoch 30/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 93.6004 - mae: 8.0571 - val\_loss: 80.8249 - val\_mae: 7.7096

Epoch 31/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.2031 - mae: 8.0676 - val\_loss: 82.4735 - val\_mae: 7.7491

Epoch 32/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **5s** 3ms/step - loss: 94.0046 - mae: 8.0651 - val\_loss: 80.6065 - val\_mae: 7.6987

Epoch 33/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 93.1797 - mae: 8.0437 - val\_loss: 81.4225 - val\_mae: 7.7261

Epoch 34/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 92.6938 - mae: 7.9994 - val\_loss: 81.6045 - val\_mae: 7.7381

Epoch 35/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **6s** 4ms/step - loss: 92.5540 - mae: 8.0253 - val\_loss: 80.5051 - val\_mae: 7.6922

Epoch 36/100

**1462/1462** ━━━━━━━━━━━━━━━━━━━━ **4s** 3ms/step - loss: 94.1496 - mae: 8.0492 - val\_loss: 82.5869 - val\_mae: 7.7700

**457/457** ━━━━━━━━━━━━━━━━━━━━ **1s** 1ms/step

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Final Evaluation:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Mean Squared Error (MSE): 81.13

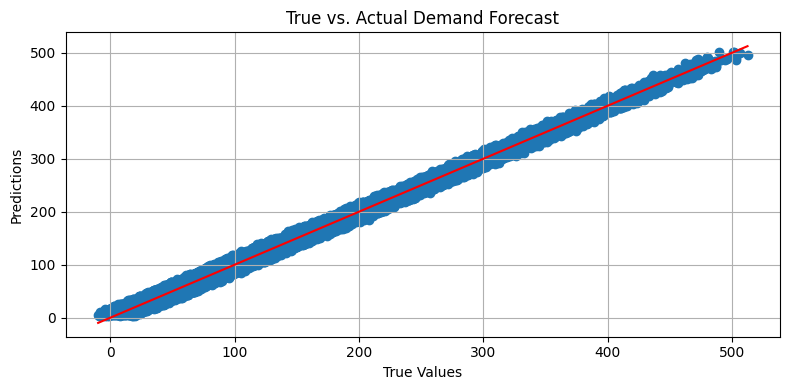
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*R-squared (R²): 0.99

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

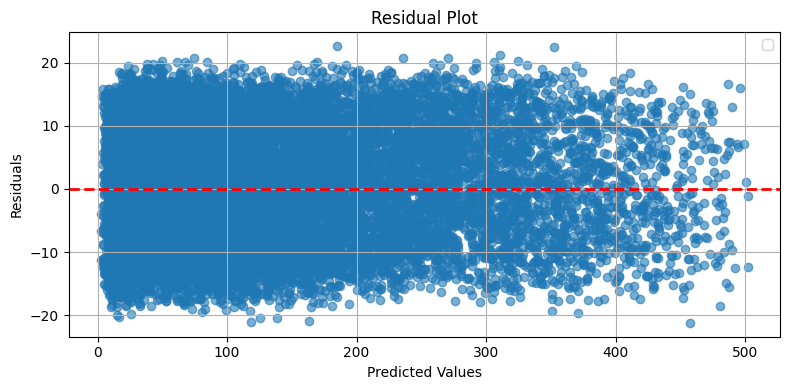
# 3. Performance analysis of the model by below plots

* Actual vs predicted
* Residual error
* Convergence curve

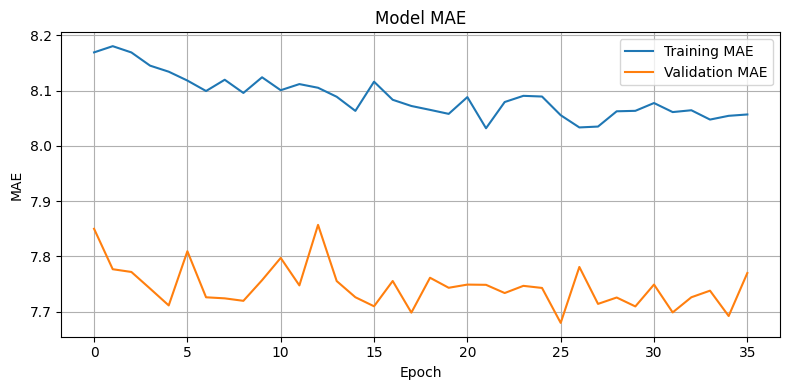


/tmp/ipython-input-3420489471.py:22: UserWarning: No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

plt.legend()







**According to plots and results of the model, it has been trained well.**

# 6. Save the best model