In [ ]:

In [7]:
```
import pandas as pd
import numpy as np
data = pd.read_csv(r"C:\Users\jitendra\Desktop\suhani\DATASET\1_boston_housing.csv")
```

In [3]:
```
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
```

In [4]:
```
from sklearn.preprocessing import MinMaxScaler
mms = MinMaxScaler()
mms.fit(X_train)
X_train = mms.transform(X_train)
X_test = mms.transform(X_test)
```

In [5]:
```
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(1, activation='linear', name='dense_output'))

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

```
WARNING:tensorflow:From C:\Users\jitendra\anaconda3\Anaconda\Lib\site-packages\keras\src
\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please u
se tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

WARNING:tensorflow:From C:\Users\jitendra\anaconda3\Anaconda\Lib\site-packages\keras\src
\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.ge
t_default_graph instead.

WARNING:tensorflow:From C:\Users\jitendra\anaconda3\Anaconda\Lib\site-packages\keras\src
\optimizers\__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.co
mpat.v1.train.Optimizer instead.

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

 dense_2 (Dense)             (None, 64)                8256

 dense_output (Dense)        (None, 1)                 65

=================================================================
Total params: 10113 (39.50 KB)
Trainable params: 10113 (39.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [8]:
```
data.head(n=10)
```
Out[8]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9

In [9]:
```
print(data.shape)
```
```
(506, 14)
```

In [10]:
```
data.isnull().sum()
```
Out[10]:
```
crim       0
zn         0
indus      0
chas       0
nox        0
rm         0
age        0
dis        0
rad        0
tax        0
ptratio    0
b          0
lstat      0
MEDV       0
dtype: int64
```

In [11]:
```
data.describe()
```
Out[11]:

	crim	zn	indus	chas	nox	rm	age	dis	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.0(
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.5
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.7(
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.0(
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.0(
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.0(
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.0(
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.0(

In [12]:
```
data.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
```

In [15]:
```
correlation = data.corr()
```

In [16]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
fig,axes = plt.subplots(figsize=(15,12))
sns.heatmap(correlation,square = True,annot = True)
```

Out[16]:
```
<Axes: >
```

In [17]:
```python
mean = X_train.mean(axis=0)
std = X_train.std(axis=0)
X_train = (X_train - mean) / std
X_test = (X_test - mean) / std
```

In [18]:
```python
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
#Fitting the model
regressor.fit(X_train,y_train)
```

Out[18]:
```
▾ LinearRegression
LinearRegression()
```

In [19]:
```python
y_pred = regressor.predict(X_test)
# Predicting RMSE the Test set results
from sklearn.metrics import mean_squared_error
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
print(rmse)
```

```
5.329714327288869
```

In [20]:
```
from sklearn.metrics import r2_score
r2 = r2_score(y_test, y_pred)
print(r2)
```

```
0.6485645742370704
```

In [21]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [22]:
```
import keras
from keras.layers import Dense, Activation,Dropout
from keras.models import Sequential
model = Sequential()
model.add(Dense(128,activation = 'relu',input_dim =13))
model.add(Dense(64,activation = 'relu'))
model.add(Dense(32,activation = 'relu'))
model.add(Dense(16,activation = 'relu'))
model.add(Dense(1))
```

In [27]:
```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
```

In [29]:
```
history = model.fit(X_train, y_train, epochs=100, validation_split=0.05, verbose = 1)
```

```
Epoch 1/100
WARNING:tensorflow:From C:\Users\jitendra\anaconda3\Anaconda\Lib\site-packages\keras\src
\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use t
f.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\jitendra\anaconda3\Anaconda\Lib\site-packages\keras\src
\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is depr
ecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

11/11 [==============================] - 11s 145ms/step - loss: 585.2556 - mae: 22.3499
- val_loss: 596.2977 - val_mae: 22.5246
Epoch 2/100
11/11 [==============================] - 0s 27ms/step - loss: 528.4331 - mae: 21.0493 -
val_loss: 514.0519 - val_mae: 20.7619
Epoch 3/100
11/11 [==============================] - 0s 42ms/step - loss: 414.8126 - mae: 18.2842 -
val_loss: 356.5651 - val_mae: 16.8917
Epoch 4/100
11/11 [==============================] - 0s 26ms/step - loss: 228.5727 - mae: 12.7512 -
val_loss: 139.6987 - val_mae: 9.6914
Epoch 5/100
11/11 [==============================] - 0s 26ms/step - loss: 87.0985 - mae: 7.4056 - va
l_loss: 61.2518 - val_mae: 6.0208
Epoch 6/100
11/11 [==============================] - 0s 26ms/step - loss: 69.0950 - mae: 6.4817 - va
l_loss: 38.7943 - val_mae: 4.9534
Epoch 7/100
11/11 [==============================] - 0s 39ms/step - loss: 41.5758 - mae: 4.8532 - va
l_loss: 28.8961 - val_mae: 4.3492
Epoch 8/100
11/11 [==============================] - 0s 33ms/step - loss: 30.9614 - mae: 4.0821 - va
l_loss: 17.2308 - val_mae: 3.4260
Epoch 9/100
11/11 [==============================] - 0s 31ms/step - loss: 25.0295 - mae: 3.7171 - va
l_loss: 15.3707 - val_mae: 3.1138
Epoch 10/100
11/11 [==============================] - 0s 47ms/step - loss: 22.3749 - mae: 3.4992 - va
l_loss: 14.1067 - val_mae: 3.0348
Epoch 11/100
11/11 [==============================] - 0s 36ms/step - loss: 20.5045 - mae: 3.2837 - va
l_loss: 13.5427 - val_mae: 2.8788
Epoch 12/100
11/11 [==============================] - 0s 35ms/step - loss: 18.9938 - mae: 3.1463 - va
l_loss: 11.9552 - val_mae: 2.8914
Epoch 13/100
11/11 [==============================] - 1s 53ms/step - loss: 17.9182 - mae: 3.1265 - va
l_loss: 11.3451 - val_mae: 2.7676
Epoch 14/100
11/11 [==============================] - 0s 36ms/step - loss: 16.9439 - mae: 2.9523 - va
l_loss: 10.6373 - val_mae: 2.6149
Epoch 15/100
11/11 [==============================] - 0s 47ms/step - loss: 15.9681 - mae: 2.8969 - va
l_loss: 10.0033 - val_mae: 2.6655
Epoch 16/100
11/11 [==============================] - 1s 67ms/step - loss: 15.2325 - mae: 2.8260 - va
l_loss: 9.2314 - val_mae: 2.4704
Epoch 17/100
11/11 [==============================] - 0s 28ms/step - loss: 14.4948 - mae: 2.7440 - va
l_loss: 8.7567 - val_mae: 2.4011
Epoch 18/100
11/11 [==============================] - 0s 27ms/step - loss: 13.8937 - mae: 2.6656 - va
l_loss: 8.1658 - val_mae: 2.2796
Epoch 19/100
11/11 [==============================] - 0s 34ms/step - loss: 13.7290 - mae: 2.7141 - va
l_loss: 7.8507 - val_mae: 2.2792
Epoch 20/100
11/11 [==============================] - 1s 60ms/step - loss: 13.3003 - mae: 2.5494 - va
l_loss: 7.3493 - val_mae: 2.1279
```

```
Epoch 21/100
11/11 [==============================] - 0s 36ms/step - loss: 12.4017 - mae: 2.5662 - va
l_loss: 8.0241 - val_mae: 2.4960
Epoch 22/100
11/11 [==============================] - 0s 27ms/step - loss: 12.2212 - mae: 2.5034 - va
l_loss: 6.9930 - val_mae: 2.2019
Epoch 23/100
11/11 [==============================] - 0s 32ms/step - loss: 11.5573 - mae: 2.4511 - va
l_loss: 8.0183 - val_mae: 2.4032
Epoch 24/100
11/11 [==============================] - 0s 27ms/step - loss: 11.3125 - mae: 2.3992 - va
l_loss: 6.9163 - val_mae: 2.1917
Epoch 25/100
11/11 [==============================] - 0s 26ms/step - loss: 11.1273 - mae: 2.4198 - va
l_loss: 7.7265 - val_mae: 2.3141
Epoch 26/100
11/11 [==============================] - 0s 27ms/step - loss: 10.7422 - mae: 2.3319 - va
l_loss: 7.1624 - val_mae: 2.1988
Epoch 27/100
11/11 [==============================] - 0s 39ms/step - loss: 10.3417 - mae: 2.2993 - va
l_loss: 7.5602 - val_mae: 2.2642
Epoch 28/100
11/11 [==============================] - 0s 32ms/step - loss: 10.0903 - mae: 2.2949 - va
l_loss: 7.7257 - val_mae: 2.3023
Epoch 29/100
11/11 [==============================] - 0s 27ms/step - loss: 9.9196 - mae: 2.2352 - val
_loss: 7.1177 - val_mae: 2.1562
Epoch 30/100
11/11 [==============================] - 0s 37ms/step - loss: 10.0221 - mae: 2.2801 - va
l_loss: 7.9539 - val_mae: 2.2689
Epoch 31/100
11/11 [==============================] - 0s 28ms/step - loss: 9.7540 - mae: 2.1975 - val
_loss: 7.4254 - val_mae: 2.2018
Epoch 32/100
11/11 [==============================] - 0s 29ms/step - loss: 9.7290 - mae: 2.2644 - val
_loss: 8.1674 - val_mae: 2.2386
Epoch 33/100
11/11 [==============================] - 0s 28ms/step - loss: 9.6185 - mae: 2.1856 - val
_loss: 8.0326 - val_mae: 2.3014
Epoch 34/100
11/11 [==============================] - 0s 42ms/step - loss: 9.2515 - mae: 2.1909 - val
_loss: 8.4889 - val_mae: 2.2535
Epoch 35/100
11/11 [==============================] - 0s 28ms/step - loss: 8.9824 - mae: 2.1499 - val
_loss: 7.8751 - val_mae: 2.2566
Epoch 36/100
11/11 [==============================] - 0s 29ms/step - loss: 8.8803 - mae: 2.1430 - val
_loss: 8.8717 - val_mae: 2.3266
Epoch 37/100
11/11 [==============================] - 0s 39ms/step - loss: 9.0676 - mae: 2.1753 - val
_loss: 7.5180 - val_mae: 2.1030
Epoch 38/100
11/11 [==============================] - 0s 28ms/step - loss: 8.8774 - mae: 2.1367 - val
_loss: 8.1002 - val_mae: 2.2648
Epoch 39/100
11/11 [==============================] - 0s 30ms/step - loss: 8.7559 - mae: 2.1651 - val
_loss: 8.8025 - val_mae: 2.2785
Epoch 40/100
11/11 [==============================] - 0s 28ms/step - loss: 8.9796 - mae: 2.1139 - val
_loss: 8.9794 - val_mae: 2.3580
Epoch 41/100
11/11 [==============================] - 0s 42ms/step - loss: 8.6096 - mae: 2.1568 - val
_loss: 9.9753 - val_mae: 2.3709
Epoch 42/100
11/11 [==============================] - 0s 25ms/step - loss: 8.1599 - mae: 2.0848 - val
_loss: 7.9795 - val_mae: 2.0889
Epoch 43/100
11/11 [==============================] - 0s 30ms/step - loss: 8.2041 - mae: 2.0646 - val
_loss: 10.5150 - val_mae: 2.4153
```

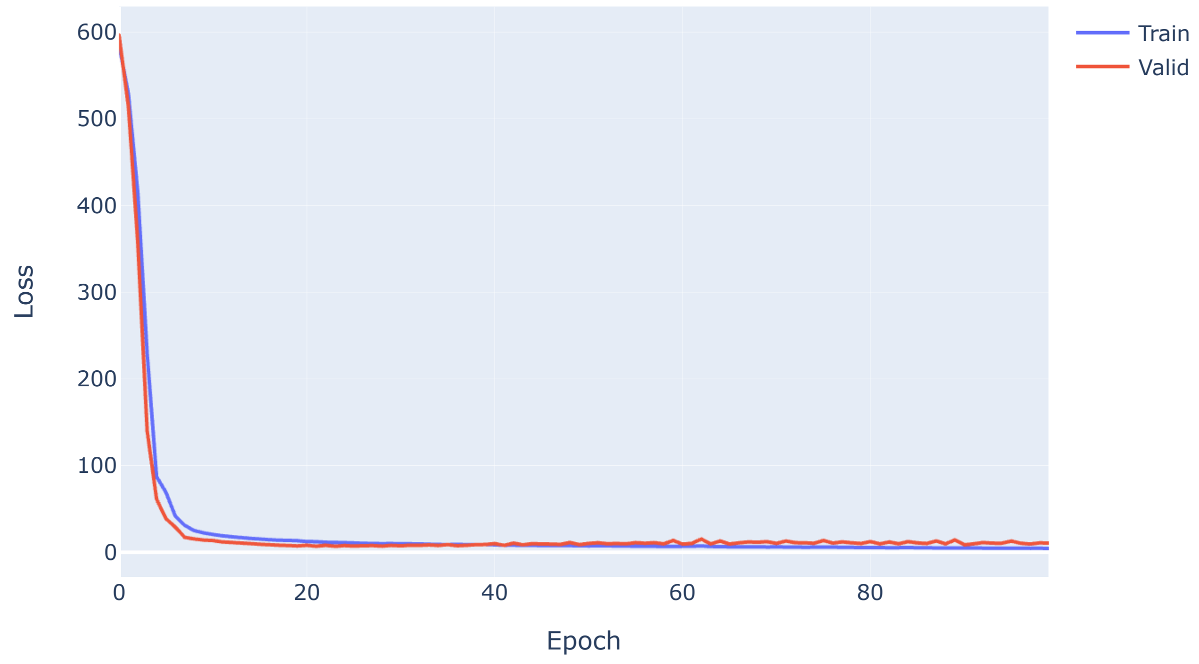
```
Epoch 44/100
11/11 [==============================] - 0s 41ms/step - loss: 7.9932 - mae: 2.0557 - val
_loss: 8.6287 - val_mae: 2.1062
Epoch 45/100
11/11 [==============================] - 0s 27ms/step - loss: 8.0056 - mae: 2.0338 - val
_loss: 9.9041 - val_mae: 2.3241
Epoch 46/100
11/11 [==============================] - 0s 33ms/step - loss: 7.8219 - mae: 2.0116 - val
_loss: 9.5402 - val_mae: 2.3069
Epoch 47/100
11/11 [==============================] - 0s 29ms/step - loss: 7.6886 - mae: 1.9982 - val
_loss: 9.4378 - val_mae: 2.1830
Epoch 48/100
11/11 [==============================] - 0s 40ms/step - loss: 7.6818 - mae: 2.0047 - val
_loss: 9.0714 - val_mae: 2.1586
Epoch 49/100
11/11 [==============================] - 0s 31ms/step - loss: 7.6828 - mae: 1.9658 - val
_loss: 11.2426 - val_mae: 2.4920
Epoch 50/100
11/11 [==============================] - 0s 38ms/step - loss: 7.5752 - mae: 1.9972 - val
_loss: 8.8415 - val_mae: 2.1104
Epoch 51/100
11/11 [==============================] - 0s 42ms/step - loss: 7.3747 - mae: 1.9598 - val
_loss: 10.1686 - val_mae: 2.2783
Epoch 52/100
11/11 [==============================] - 0s 32ms/step - loss: 7.5040 - mae: 1.9861 - val
_loss: 11.0064 - val_mae: 2.4006
Epoch 53/100
11/11 [==============================] - 0s 25ms/step - loss: 7.5464 - mae: 1.9776 - val
_loss: 9.8080 - val_mae: 2.2374
Epoch 54/100
11/11 [==============================] - 0s 43ms/step - loss: 7.2072 - mae: 1.9704 - val
_loss: 10.0752 - val_mae: 2.2426
Epoch 55/100
11/11 [==============================] - 0s 28ms/step - loss: 7.2391 - mae: 1.9481 - val
_loss: 9.7234 - val_mae: 2.1608
Epoch 56/100
11/11 [==============================] - 0s 29ms/step - loss: 6.9917 - mae: 1.9041 - val
_loss: 11.0961 - val_mae: 2.2905
Epoch 57/100
11/11 [==============================] - 0s 30ms/step - loss: 6.9732 - mae: 1.8906 - val
_loss: 10.4070 - val_mae: 2.1456
Epoch 58/100
```

```
11/11 [==============================] - 0s 39ms/step - loss: 6.9353 - mae: 1.9238 - val
_loss: 11.0708 - val_mae: 2.2439
Epoch 59/100
11/11 [==============================] - 0s 26ms/step - loss: 6.7107 - mae: 1.8869 - val
_loss: 9.6628 - val_mae: 2.1254
Epoch 60/100
11/11 [==============================] - 0s 24ms/step - loss: 6.7900 - mae: 1.8688 - val
_loss: 13.6515 - val_mae: 2.6828
Epoch 61/100
11/11 [==============================] - 0s 26ms/step - loss: 6.8937 - mae: 1.9582 - val
_loss: 9.3899 - val_mae: 2.0708
Epoch 62/100
11/11 [==============================] - 0s 37ms/step - loss: 7.0095 - mae: 1.9203 - val
_loss: 10.2449 - val_mae: 2.1583
Epoch 63/100
11/11 [==============================] - 0s 28ms/step - loss: 7.3316 - mae: 1.9027 - val
_loss: 15.3057 - val_mae: 2.8083
Epoch 64/100
11/11 [==============================] - 0s 23ms/step - loss: 6.6596 - mae: 1.9098 - val
_loss: 9.6575 - val_mae: 2.1116
Epoch 65/100
11/11 [==============================] - 0s 21ms/step - loss: 6.5523 - mae: 1.8338 - val
_loss: 13.0014 - val_mae: 2.5703
Epoch 66/100
11/11 [==============================] - 0s 35ms/step - loss: 6.3066 - mae: 1.8250 - val
_loss: 9.5504 - val_mae: 2.0946
Epoch 67/100
11/11 [==============================] - 0s 29ms/step - loss: 6.2927 - mae: 1.8301 - val
_loss: 10.8695 - val_mae: 2.3231
Epoch 68/100
11/11 [==============================] - 0s 23ms/step - loss: 6.2117 - mae: 1.8101 - val
_loss: 11.9418 - val_mae: 2.3970
Epoch 69/100
11/11 [==============================] - 0s 26ms/step - loss: 6.2400 - mae: 1.8284 - val
_loss: 11.5588 - val_mae: 2.3644
Epoch 70/100
11/11 [==============================] - 0s 39ms/step - loss: 6.0867 - mae: 1.7902 - val
_loss: 12.2379 - val_mae: 2.4350
Epoch 71/100
11/11 [==============================] - 0s 22ms/step - loss: 6.2054 - mae: 1.8148 - val
_loss: 10.2314 - val_mae: 2.2290
Epoch 72/100
11/11 [==============================] - 0s 33ms/step - loss: 6.0691 - mae: 1.7780 - val
_loss: 12.9785 - val_mae: 2.5009
Epoch 73/100
11/11 [==============================] - 0s 24ms/step - loss: 6.1077 - mae: 1.7669 - val
_loss: 11.0406 - val_mae: 2.3469
Epoch 74/100
11/11 [==============================] - 0s 38ms/step - loss: 5.7664 - mae: 1.7637 - val
_loss: 10.8766 - val_mae: 2.2563
Epoch 75/100
11/11 [==============================] - 0s 24ms/step - loss: 5.9289 - mae: 1.7859 - val
_loss: 10.3696 - val_mae: 2.2330
Epoch 76/100
11/11 [==============================] - 0s 24ms/step - loss: 5.8895 - mae: 1.7500 - val
_loss: 13.6397 - val_mae: 2.6219
Epoch 77/100
11/11 [==============================] - 0s 30ms/step - loss: 6.1197 - mae: 1.7580 - val
_loss: 10.5364 - val_mae: 2.2132
Epoch 78/100
11/11 [==============================] - 0s 48ms/step - loss: 5.6197 - mae: 1.7315 - val
_loss: 12.0012 - val_mae: 2.3778
Epoch 79/100
11/11 [==============================] - 0s 33ms/step - loss: 5.6176 - mae: 1.7144 - val
_loss: 10.8913 - val_mae: 2.2944
Epoch 80/100
11/11 [==============================] - 0s 33ms/step - loss: 5.5539 - mae: 1.6979 - val
_loss: 10.1399 - val_mae: 2.2420
```

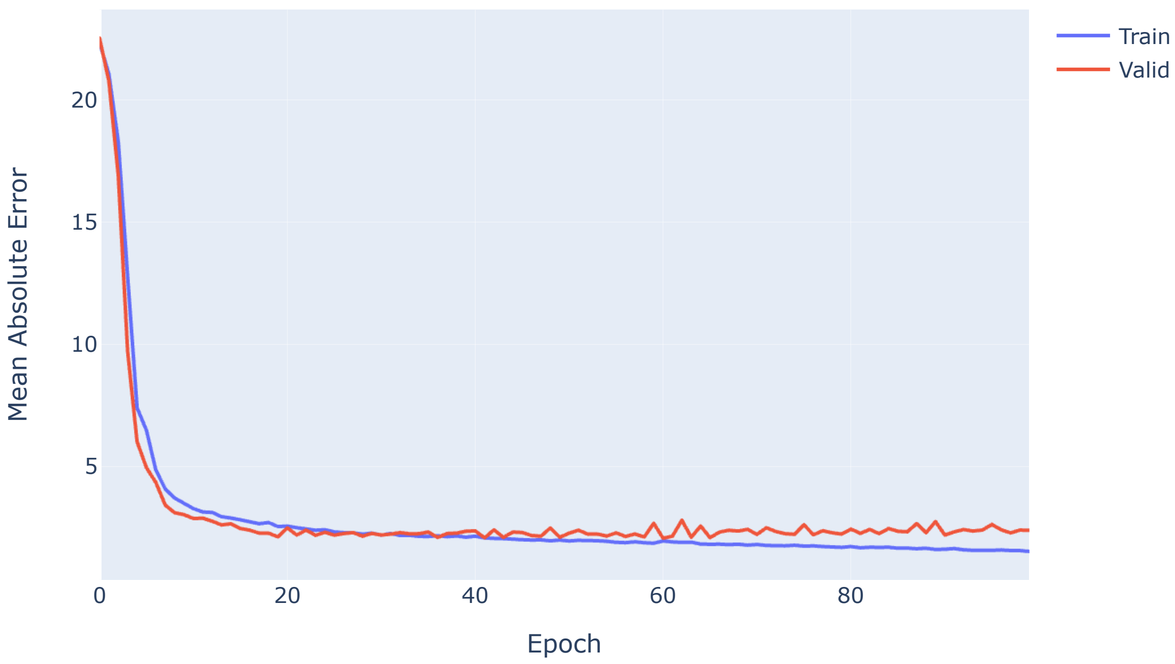
```
Epoch 81/100
11/11 [==============================] - 0s 45ms/step - loss: 5.5309 - mae: 1.7306 - val
_loss: 12.3128 - val_mae: 2.4446
Epoch 82/100
11/11 [==============================] - 0s 27ms/step - loss: 5.3871 - mae: 1.6817 - val
_loss: 9.7143 - val_mae: 2.2702
Epoch 83/100
11/11 [==============================] - 0s 27ms/step - loss: 5.3153 - mae: 1.7025 - val
_loss: 11.8781 - val_mae: 2.4303
Epoch 84/100
11/11 [==============================] - 1s 55ms/step - loss: 5.3404 - mae: 1.6983 - val
_loss: 9.7900 - val_mae: 2.2606
Epoch 85/100
11/11 [==============================] - 0s 27ms/step - loss: 5.5037 - mae: 1.7025 - val
_loss: 12.3043 - val_mae: 2.4673
Epoch 86/100
11/11 [==============================] - 1s 88ms/step - loss: 5.2242 - mae: 1.6652 - val
_loss: 10.8030 - val_mae: 2.3560
Epoch 87/100
11/11 [==============================] - 0s 33ms/step - loss: 5.1994 - mae: 1.6684 - val
_loss: 10.1529 - val_mae: 2.3359
Epoch 88/100
11/11 [==============================] - 0s 32ms/step - loss: 4.9820 - mae: 1.6428 - val
_loss: 13.0737 - val_mae: 2.6694
Epoch 89/100
11/11 [==============================] - 0s 34ms/step - loss: 5.0021 - mae: 1.6627 - val
_loss: 9.7038 - val_mae: 2.2990
Epoch 90/100
11/11 [==============================] - 0s 40ms/step - loss: 5.0632 - mae: 1.6122 - val
_loss: 14.3217 - val_mae: 2.7515
Epoch 91/100
11/11 [==============================] - 0s 32ms/step - loss: 4.8921 - mae: 1.6223 - val
_loss: 8.5998 - val_mae: 2.2014
Epoch 92/100
11/11 [==============================] - 2s 156ms/step - loss: 4.9494 - mae: 1.6502 - va
l_loss: 9.7320 - val_mae: 2.3338
Epoch 93/100
11/11 [==============================] - 1s 74ms/step - loss: 4.8163 - mae: 1.5973 - val
_loss: 11.2011 - val_mae: 2.4265
Epoch 94/100
11/11 [==============================] - 1s 80ms/step - loss: 4.7272 - mae: 1.5767 - val
_loss: 10.4882 - val_mae: 2.3658
Epoch 95/100
11/11 [==============================] - 0s 28ms/step - loss: 4.6943 - mae: 1.5760 - val
_loss: 10.4736 - val_mae: 2.4078
Epoch 96/100
11/11 [==============================] - 0s 30ms/step - loss: 4.7537 - mae: 1.5754 - val
_loss: 12.9197 - val_mae: 2.6341
Epoch 97/100
11/11 [==============================] - 0s 32ms/step - loss: 4.7413 - mae: 1.5917 - val
_loss: 10.3873 - val_mae: 2.4167
Epoch 98/100
11/11 [==============================] - 0s 47ms/step - loss: 4.5733 - mae: 1.5683 - val
_loss: 9.4391 - val_mae: 2.2909
Epoch 99/100
11/11 [==============================] - 0s 29ms/step - loss: 4.6363 - mae: 1.5659 - val
_loss: 10.8212 - val_mae: 2.4048
Epoch 100/100
11/11 [==============================] - 0s 27ms/step - loss: 4.4476 - mae: 1.5319 - val
_loss: 10.5794 - val_mae: 2.3931
```

In [ ]:

In [31]:
```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Scattergl(y=history.history['loss'],
name='Train'))
fig.add_trace(go.Scattergl(y=history.history['val_loss'],
name='Valid'))
fig.update_layout(height=500, width=700,
xaxis_title='Epoch',
yaxis_title='Loss')
fig.show()
```



In [32]:
```
fig = go.Figure()
fig.add_trace(go.Scattergl(y=history.history['mae'],
name='Train'))
fig.add_trace(go.Scattergl(y=history.history['val_mae'],
name='Valid'))
fig.update_layout(height=500, width=700,
xaxis_title='Epoch',
yaxis_title='Mean Absolute Error')
fig.show()
```



In [33]:
```
y_pred = model.predict(X_test)
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 [==============================] - 1s 8ms/step
5/5 [==============================] - 2s 8ms/step - loss: 16.6110 - mae: 2.5777
Mean squared error on test data:  16.611026763916016
Mean absolute error on test data:  2.5777082443237305
```

In [34]:
```
from sklearn.metrics import mean_absolute_error
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
y_pred_lr = lr_model.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
mae_lr = mean_absolute_error(y_test, y_pred_lr)
```

In [35]:
```
print('Mean squared error on test data: ', mse_lr)
print('Mean absolute error on test data: ', mae_lr)
from sklearn.metrics import r2_score
r2 = r2_score(y_test, y_pred)
print(r2)
```

```
Mean squared error on test data:  28.405854810508238
Mean absolute error on test data:  3.691362677116256
0.7944894528123007
```

In [36]:
```
# Predicting RMSE the Test set results
from sklearn.metrics import mean_squared_error
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
print(rmse)
```

```
4.075662565490449
```

In [37]:
```
import sklearn
new_data = sklearn.preprocessing.StandardScaler().fit_transform(([[0.1, 10.0,
5.0, 0, 0.4, 6.0, 50, 6.0, 1, 400, 20, 300, 10]]))
prediction = model.predict(new_data)
print("Predicted house price:", prediction)
```

```
1/1 [==============================] - 0s 193ms/step
Predicted house price: [[10.683548]]
```

In [38]:
```
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 17ms/step - loss: 16.6110 - mae: 2.5777
Mean squared error on test data:  16.611026763916016
Mean absolute error on test data:  2.5777082443237305
```

In [ ]: