

```
##### Regression using Sequential Model
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

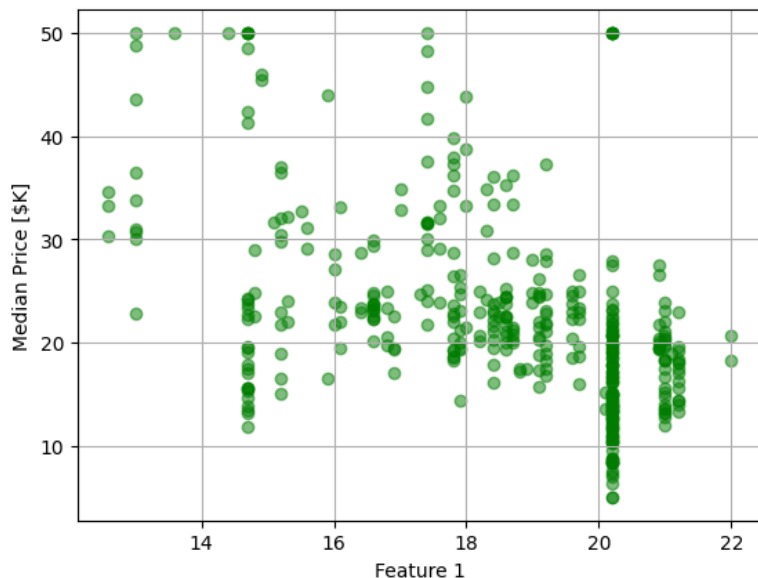
# Load the Boston housing dataset.
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.boston_housing.load_data()
print(X_train.shape)
print("Input features: ", X_train[0])
print("Output target: ", y_train[0])

x_train=X_train[:,10]
x_test=X_test[:,10]
print(x_train.shape)
print(x_test.shape)

(404,)
(102,)

plt.xlabel("Feature 1")
plt.ylabel("Median Price [$K]")
plt.grid("on")
plt.scatter(x_train, y_train, color="green", alpha=0.5)
```

<matplotlib.collections.PathCollection at 0x7f4d0fb0bdc0>



```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(1, input_shape=(1,)))
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1)	2

```
=====  
Total params: 2 (8.00 Byte)  
Trainable params: 2 (8.00 Byte)  
Non-trainable params: 0 (0.00 Byte)  
=====
```

```
model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.005), loss="mse")
```

```
hist = model.fit(x_train,y_train, batch_size=32,epochs=100,=validation_split0.3)
```

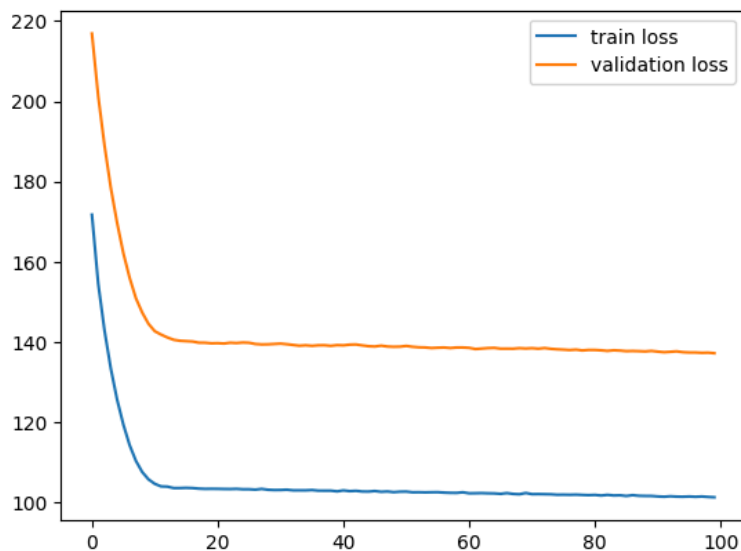
```
Epoch 1/100  
9/9 [=====] - 1s 26ms/step - loss: 171.7238 - val_loss: 216.8642  
Epoch 2/100  
9/9 [=====] - 0s 8ms/step - loss: 154.5656 - val_loss: 201.2910  
Epoch 3/100  
9/9 [=====] - 0s 6ms/step - loss: 142.8803 - val_loss: 188.9941  
Epoch 4/100
```

```
9/9 [=====] - 0s 6ms/step - loss: 133.4072 - val_loss: 178.4404
Epoch 5/100
9/9 [=====] - 0s 8ms/step - loss: 125.6729 - val_loss: 169.7411
Epoch 6/100
9/9 [=====] - 0s 7ms/step - loss: 119.4216 - val_loss: 162.1401
Epoch 7/100
9/9 [=====] - 0s 7ms/step - loss: 114.2785 - val_loss: 155.9785
Epoch 8/100
9/9 [=====] - 0s 7ms/step - loss: 110.3651 - val_loss: 150.8847
Epoch 9/100
9/9 [=====] - 0s 8ms/step - loss: 107.5921 - val_loss: 147.3375
Epoch 10/100
9/9 [=====] - 0s 8ms/step - loss: 105.7634 - val_loss: 144.5169
Epoch 11/100
9/9 [=====] - 0s 6ms/step - loss: 104.6491 - val_loss: 142.6853
Epoch 12/100
9/9 [=====] - 0s 6ms/step - loss: 104.0146 - val_loss: 141.8422
Epoch 13/100
9/9 [=====] - 0s 8ms/step - loss: 103.9583 - val_loss: 141.1319
Epoch 14/100
9/9 [=====] - 0s 8ms/step - loss: 103.5950 - val_loss: 140.5478
Epoch 15/100
9/9 [=====] - 0s 7ms/step - loss: 103.5996 - val_loss: 140.3027
Epoch 16/100
9/9 [=====] - 0s 8ms/step - loss: 103.6557 - val_loss: 140.2130
Epoch 17/100
9/9 [=====] - 0s 9ms/step - loss: 103.5921 - val_loss: 140.1179
Epoch 18/100
9/9 [=====] - 0s 8ms/step - loss: 103.4610 - val_loss: 139.8283
Epoch 19/100
9/9 [=====] - 0s 6ms/step - loss: 103.3962 - val_loss: 139.8108
Epoch 20/100
9/9 [=====] - 0s 12ms/step - loss: 103.4131 - val_loss: 139.6698
Epoch 21/100
9/9 [=====] - 0s 10ms/step - loss: 103.4010 - val_loss: 139.7046
Epoch 22/100
9/9 [=====] - 0s 18ms/step - loss: 103.3672 - val_loss: 139.6165
Epoch 23/100
9/9 [=====] - 0s 19ms/step - loss: 103.3529 - val_loss: 139.8127
Epoch 24/100
9/9 [=====] - 0s 11ms/step - loss: 103.3995 - val_loss: 139.7407
Epoch 25/100
9/9 [=====] - 0s 16ms/step - loss: 103.2996 - val_loss: 139.8632
Epoch 26/100
9/9 [=====] - 0s 12ms/step - loss: 103.2972 - val_loss: 139.8015
Epoch 27/100
9/9 [=====] - 0s 15ms/step - loss: 103.2056 - val_loss: 139.4976
Epoch 28/100
9/9 [=====] - 0s 21ms/step - loss: 103.3851 - val_loss: 139.3951
Epoch 29/100
```

```
dict=hist.history
print(dict.keys())
```

```
dict_keys(['loss', 'val_loss'])
```

```
plt.plot(dict['loss'],label='train loss')
plt.plot(dict['val_loss'],label='validation loss')
plt.legend()
plt.show()
```



```
x = [3]
y_pred = model.predict(x)
print(y_pred)

1/1 [=====] - 0s 106ms/step
[[5.0800824]]

y_pred=model.predict(x_test)

4/4 [=====] - 0s 3ms/step

plt.scatter(x_test, y_test, label='Ground Truth', color='green', alpha=0.5)
plt.plot(x_test, y_pred, color='k', label='Model Predictions')

model.evaluate(x_test)

##### Muliple Linear Regression
import tensorflow as tf
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.boston_housing.load_data()
print(X_train.shape)
print(X_test.shape)

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/boston\_housing.npz
57026/57026 [=====] - 0s 0us/step
(404, 13)
(102, 13)

new_model=tf.keras.models.Sequential()
new_model.add(tf.keras.layers.Dense(120,activation='relu',input_shape=(13,)))
new_model.add(tf.keras.layers.Dense(1))
new_model.summary()
new_model.compile(optimizer='rmsprop',loss='mse')
new_hist=new_model.fit(X_train,y_train,batch_size=32,validation_split=0.3,epochs=100)

dict=new_hist.history
dict.keys()

dict_keys(['loss', 'val_loss'])

plt.plot(dict['loss'],label='train loss')
plt.plot(dict['val_loss'],label='validation loss')
plt.legend()
plt.show()

model.evaluate(X_test)
```

Start coding or [generate](#) with AI.