

```

from google.colab import auth
import gspread
from google.auth import default
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import StandardScaler

```

```


auth.authenticate_user()
creds, _ = default()
gc = gspread.authorize(creds)

```

```

worksheet = gc.open('student_data').sheet1
rows = worksheet.get_all_values()
df = pd.DataFrame(rows[1:], columns=rows[0])
print(df.shape)

```

 (20, 2)

```

worksheet=gc.open('student_data').sheet1
rows = worksheet.get_all_values()


```

```

dataset1 = pd.DataFrame(rows[1:], columns=rows[0])
dataset1 = dataset1.astype({'Input': 'int'})
dataset1 = dataset1.astype({'output': 'int'})

```

```
dataset1.head()
```



	Input	output
0	1	11
1	2	21
2	3	31
3	4	41
4	5	51

```

X = dataset1[['Input']].values
y_train = dataset1[['output']].values

```


```
#x
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y_train, test_size=0.33, random_state=33)
```

```
scaler = MinMaxScaler()
```

```
scaler.fit(X_train)
```



MinMaxScaler

MinMaxScaler()

```
X_train1= scaler.transform(X_train)
```

```

ai_brain = Sequential([
Dense(8,activation='relu'),
Dense(4,activation='relu'),
Dense(1)
])

```

```
ai_brain.compile(optimizer= 'rmsprop', loss="mse")
```

```
ai_brain.fit(X_train1,y_train,epochs=2000)
```



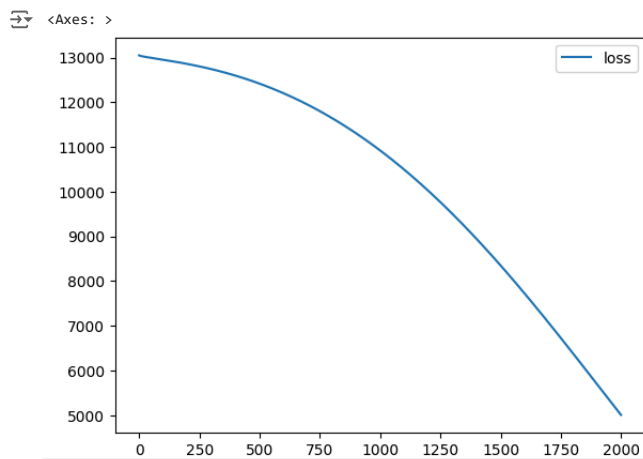
```

Epoch 785/2000
1/1 ————— 0s 56ms/step - loss: 11697.1768
Epoch 786/2000
1/1 ————— 0s 60ms/step - loss: 11694.0557
Epoch 787/2000
1/1 ————— 0s 59ms/step - loss: 11690.9307
Epoch 788/2000
1/1 ————— 0s 56ms/step - loss: 11687.8008
Epoch 789/2000
1/1 ————— 0s 35ms/step - loss: 11684.6660
Epoch 790/2000
1/1 ————— 0s 58ms/step - loss: 11681.5273
Epoch 791/2000
1/1 ————— 0s 59ms/step - loss: 11678.3848
Epoch 792/2000
1/1 ————— 0s 58ms/step - loss: 11675.2393
Epoch 793/2000
1/1 ————— 0s 60ms/step - loss: 11672.0879
Epoch 794/2000
1/1 ————— 0s 43ms/step - loss: 11668.9316
Epoch 795/2000
1/1 ————— 0s 36ms/step - loss: 11665.7715
Epoch 796/2000
1/1 ————— 0s 54ms/step - loss: 11662.6084
Epoch 797/2000
1/1 ————— 0s 123ms/step - loss: 11659.4395
Epoch 798/2000
1/1 ————— 0s 47ms/step - loss: 11656.2676
Epoch 799/2000
1/1 ————— 0s 53ms/step - loss: 11653.0918
Epoch 800/2000
1/1 ————— 0s 59ms/step - loss: 11649.9102
Epoch 801/2000
1/1 ————— 0s 34ms/step - loss: 11646.7236
Epoch 802/2000
1/1 ————— 0s 60ms/step - loss: 11643.5352
Epoch 803/2000
1/1 ————— 0s 55ms/step - loss: 11640.3398
Epoch 804/2000
1/1 ————— 0s 61ms/step - loss: 11637.1416
Epoch 805/2000
1/1 ————— 0s 57ms/step - loss: 11633.9424
Epoch 806/2000

```

```
loss_df = pd.DataFrame(ai_brain.history.history)
```

```
loss_df.plot()
```



```
scaler = StandardScaler()
scaler.fit(X_train)
```

```
StandardScaler
```

```
StandardScaler()
```

```
X_test1 = scaler.transform(X_test)
```

```
ai_brain.evaluate(X_test1,y_test)
```

```
1/1 ————— 0s 94ms/step - loss: 6373.4326
6373.4326171875
```

```
X_n1 = [[50]]
```

```
X_n1_1 = scaler.transform(X_n1)
```

```
ai_brain.predict(X_n1_1)
```

```
1/1 ————— 0s 46ms/step
array([[235.92375]], dtype=float32)
```

