

```
#Libraries import
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras import models,datasets,layers
import matplotlib.pyplot as plt

#Reading Data
df = pd.read_csv("diabetes1.csv")
x = df.iloc[:, :8]
y = df["Outcome"]

x_std = StandardScaler().fit_transform(x)
x_train,x_test,y_train,y_test = train_test_split(x_std,y,test_size = 0.1)
```

ADAM Optimizer to create model

```
adam_model = models.Sequential()
# First hidden layer:
adam_model.add(layers.Dense(100,activation="relu"))
# Second hidden layer:
adam_model.add(layers.Dense(50,activation="relu"))
# Third hidden layer:
adam_model.add(layers.Dense(25,activation="relu"))
# Fourth hidden layer:
adam_model.add(layers.Dense(12,activation="relu"))
# Fifth hidden layer:
adam_model.add(layers.Dense(6,activation="relu"))
# Output layer:
adam_model.add(layers.Dense(1,activation="sigmoid"))

#Model Compilation
# optimizer = adam
# binary_crossentropy function used to compute the cross-entropy loss between true labels
adam_model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
```

```
history = adam_model.fit(x_train,y_train,epochs=50,validation_data=(x_test,y_test))
```

```
Epoch 1/50
22/22 [=====] - 1s 9ms/step - loss: 0.6093 - accuracy: 0.
Epoch 2/50
22/22 [=====] - 0s 2ms/step - loss: 0.5137 - accuracy: 0.
Epoch 3/50
22/22 [=====] - 0s 2ms/step - loss: 0.4891 - accuracy: 0.
Epoch 4/50
22/22 [=====] - 0s 2ms/step - loss: 0.4704 - accuracy: 0.
Epoch 5/50
22/22 [=====] - 0s 2ms/step - loss: 0.4583 - accuracy: 0.
```

```

Epoch 6/50
22/22 [=====] - 0s 2ms/step - loss: 0.4356 - accuracy: 0.8
Epoch 7/50
22/22 [=====] - 0s 2ms/step - loss: 0.4298 - accuracy: 0.7
Epoch 8/50
22/22 [=====] - 0s 2ms/step - loss: 0.4182 - accuracy: 0.7
Epoch 9/50
22/22 [=====] - 0s 3ms/step - loss: 0.4095 - accuracy: 0.8
Epoch 10/50
22/22 [=====] - 0s 2ms/step - loss: 0.3925 - accuracy: 0.8
Epoch 11/50
22/22 [=====] - 0s 2ms/step - loss: 0.3841 - accuracy: 0.8
Epoch 12/50
22/22 [=====] - 0s 3ms/step - loss: 0.3753 - accuracy: 0.8
Epoch 13/50
22/22 [=====] - 0s 3ms/step - loss: 0.3723 - accuracy: 0.8
Epoch 14/50
22/22 [=====] - 0s 2ms/step - loss: 0.3686 - accuracy: 0.8
Epoch 15/50
22/22 [=====] - 0s 3ms/step - loss: 0.3555 - accuracy: 0.8
Epoch 16/50
22/22 [=====] - 0s 2ms/step - loss: 0.3468 - accuracy: 0.8
Epoch 17/50
22/22 [=====] - 0s 2ms/step - loss: 0.3425 - accuracy: 0.8
Epoch 18/50
22/22 [=====] - 0s 2ms/step - loss: 0.3431 - accuracy: 0.8
Epoch 19/50
22/22 [=====] - 0s 2ms/step - loss: 0.3370 - accuracy: 0.8
Epoch 20/50
22/22 [=====] - 0s 2ms/step - loss: 0.3222 - accuracy: 0.8
Epoch 21/50
22/22 [=====] - 0s 2ms/step - loss: 0.3222 - accuracy: 0.8
Epoch 22/50
22/22 [=====] - 0s 2ms/step - loss: 0.3276 - accuracy: 0.8
Epoch 23/50
22/22 [=====] - 0s 2ms/step - loss: 0.3221 - accuracy: 0.8
Epoch 24/50
22/22 [=====] - 0s 2ms/step - loss: 0.3026 - accuracy: 0.8
Epoch 25/50
22/22 [=====] - 0s 2ms/step - loss: 0.2947 - accuracy: 0.8
Epoch 26/50
22/22 [=====] - 0s 2ms/step - loss: 0.2900 - accuracy: 0.8
Epoch 27/50
22/22 [=====] - 0s 2ms/step - loss: 0.2828 - accuracy: 0.8
Epoch 28/50
22/22 [=====] - 0s 3ms/step - loss: 0.2746 - accuracy: 0.8
Epoch 29/50

```

```
result = adam_model.evaluate(x_test,y_test)
```

```
3/3 [=====] - 0s 4ms/step - loss: 0.9853 - accuracy: 0.6753
```

```
# Loss of the model:
```

```
print(f"Loss: {result[0]}")
```

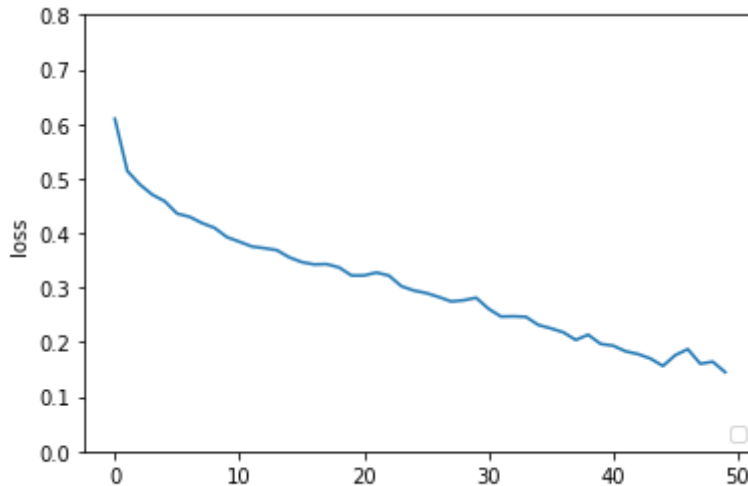
```
# Accuracy of the model:
```

```
print(f"Accuracy: {result[1]*100}")
```

Loss: 0.9853241443634033
Accuracy: 67.53246784210205

```
plt.xlabel('Epoch')
plt.ylabel('loss')
plt.ylim([0, 0.8])
plt.legend(loc='lower right')
test_loss, test_acc = adam_model.evaluate(x_test, y_test, verbose=2)
plt.plot(history.history['loss'], label='loss')
```

WARNING:matplotlib.legend:No handles with labels found to put in legend.
3/3 - 0s - loss: 0.9853 - accuracy: 0.6753 - 23ms/epoch - 8ms/step
[<matplotlib.lines.Line2D at 0x7fe32b212b10>]



RMSProp to create a model

```
RMSprop_model = models.Sequential()
# First hidden layer:
RMSprop_model.add(layers.Dense(100,activation="relu"))
# First hidden layer:
RMSprop_model.add(layers.Dense(50,activation="relu"))
# Second hidden layer:
RMSprop_model.add(layers.Dense(25,activation="relu"))
# Third hidden layer:
RMSprop_model.add(layers.Dense(12,activation="relu"))
# Fourth hidden layer:
RMSprop_model.add(layers.Dense(6,activation="relu"))
# Output layer:
RMSprop_model.add(layers.Dense(1,activation="sigmoid"))

# optimizer = RMSprop
RMSprop_model.compile(optimizer="RMSprop",loss="binary_crossentropy",metrics=["accuracy"])
history = RMSprop_model.fit(x_train,y_train,epochs=50,validation_data=(x_test,y_test))
```

```
Epoch 1/50
22/22 [=====] - 1s 8ms/step - loss: 0.6544 - accuracy: 0.
Epoch 2/50
22/22 [=====] - 0s 2ms/step - loss: 0.5579 - accuracy: 0.
```

```
Epoch 3/50
22/22 [=====] - 0s 2ms/step - loss: 0.4954 - accuracy: 0.8
Epoch 4/50
22/22 [=====] - 0s 2ms/step - loss: 0.4724 - accuracy: 0.8
Epoch 5/50
22/22 [=====] - 0s 2ms/step - loss: 0.4589 - accuracy: 0.8
Epoch 6/50
22/22 [=====] - 0s 2ms/step - loss: 0.4507 - accuracy: 0.8
Epoch 7/50
22/22 [=====] - 0s 2ms/step - loss: 0.4479 - accuracy: 0.8
Epoch 8/50
22/22 [=====] - 0s 2ms/step - loss: 0.4409 - accuracy: 0.8
Epoch 9/50
22/22 [=====] - 0s 3ms/step - loss: 0.4402 - accuracy: 0.8
Epoch 10/50
22/22 [=====] - 0s 2ms/step - loss: 0.4330 - accuracy: 0.8
Epoch 11/50
22/22 [=====] - 0s 2ms/step - loss: 0.4337 - accuracy: 0.8
Epoch 12/50
22/22 [=====] - 0s 2ms/step - loss: 0.4286 - accuracy: 0.8
Epoch 13/50
22/22 [=====] - 0s 2ms/step - loss: 0.4270 - accuracy: 0.8
Epoch 14/50
22/22 [=====] - 0s 2ms/step - loss: 0.4195 - accuracy: 0.8
Epoch 15/50
22/22 [=====] - 0s 2ms/step - loss: 0.4164 - accuracy: 0.8
Epoch 16/50
22/22 [=====] - 0s 2ms/step - loss: 0.4155 - accuracy: 0.8
Epoch 17/50
22/22 [=====] - 0s 2ms/step - loss: 0.4120 - accuracy: 0.8
Epoch 18/50
22/22 [=====] - 0s 2ms/step - loss: 0.4085 - accuracy: 0.8
Epoch 19/50
22/22 [=====] - 0s 2ms/step - loss: 0.4047 - accuracy: 0.8
Epoch 20/50
22/22 [=====] - 0s 2ms/step - loss: 0.4032 - accuracy: 0.8
Epoch 21/50
22/22 [=====] - 0s 2ms/step - loss: 0.4023 - accuracy: 0.8
Epoch 22/50
22/22 [=====] - 0s 2ms/step - loss: 0.3900 - accuracy: 0.8
Epoch 23/50
22/22 [=====] - 0s 2ms/step - loss: 0.3921 - accuracy: 0.8
Epoch 24/50
22/22 [=====] - 0s 3ms/step - loss: 0.3879 - accuracy: 0.8
Epoch 25/50
22/22 [=====] - 0s 2ms/step - loss: 0.3856 - accuracy: 0.8
Epoch 26/50
22/22 [=====] - 0s 3ms/step - loss: 0.3773 - accuracy: 0.8
Epoch 27/50
22/22 [=====] - 0s 2ms/step - loss: 0.3761 - accuracy: 0.8
Epoch 28/50
22/22 [=====] - 0s 2ms/step - loss: 0.3758 - accuracy: 0.8
```

```
result = RMSprop_model.evaluate(x_test,y_test)
```

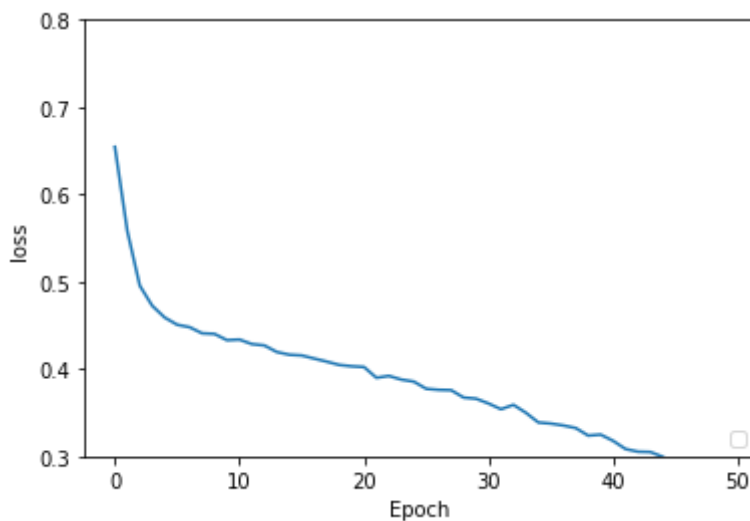
```
3/3 [=====] - 0s 5ms/step - loss: 0.5833 - accuracy: 0.7273
```

```
# Loss of the model:
print(f"Loss: {result[0]*100}")
# Accuracy of the model:
print(f"Accuracy: {result[1]*100}")
```

```
Loss: 58.331137895584106
Accuracy: 72.72727489471436
```

```
plt.xlabel('Epoch')
plt.ylabel('loss')
plt.ylim([0.3, 0.8])
plt.legend(loc='lower right')
test_loss, test_acc = RMSprop_model.evaluate(x_test, y_test, verbose=2)
plt.plot(history.history['loss'], label='loss')
```

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
3/3 - 0s - loss: 0.5833 - accuracy: 0.7273 - 15ms/epoch - 5ms/step
[<matplotlib.lines.Line2D at 0x7fe32ba0bf50>]
```



Nadam optimizer to create a model

```
Nadam_model = models.Sequential()
# First hidden layer:
Nadam_model.add(layers.Dense(100,activation="relu"))
# First hidden layer:
Nadam_model.add(layers.Dense(50,activation="relu"))
# Second hidden layer:
Nadam_model.add(layers.Dense(12,activation="relu"))
# Third hidden layer:
Nadam_model.add(layers.Dense(6,activation="relu"))
# Output layer:
Nadam_model.add(layers.Dense(1,activation="sigmoid"))

# optimizer = Nadam
Nadam_model.compile(optimizer="Nadam",loss="binary_crossentropy",metrics=["accuracy"])
history = Nadam_model.fit(x_train,y_train,epochs=50,validation_data=(x_test,y_test))
```

```
Epoch 1/50
22/22 [=====] - 1s 7ms/step - loss: 0.6909 - accuracy: 0.
Epoch 2/50
22/22 [=====] - 0s 2ms/step - loss: 0.6460 - accuracy: 0.
Epoch 3/50
22/22 [=====] - 0s 2ms/step - loss: 0.5344 - accuracy: 0.
Epoch 4/50
22/22 [=====] - 0s 2ms/step - loss: 0.4696 - accuracy: 0.
Epoch 5/50
22/22 [=====] - 0s 2ms/step - loss: 0.4514 - accuracy: 0.
Epoch 6/50
22/22 [=====] - 0s 2ms/step - loss: 0.4466 - accuracy: 0.
Epoch 7/50
22/22 [=====] - 0s 2ms/step - loss: 0.4388 - accuracy: 0.
Epoch 8/50
22/22 [=====] - 0s 2ms/step - loss: 0.4322 - accuracy: 0.
Epoch 9/50
22/22 [=====] - 0s 2ms/step - loss: 0.4303 - accuracy: 0.
Epoch 10/50
22/22 [=====] - 0s 2ms/step - loss: 0.4217 - accuracy: 0.
Epoch 11/50
22/22 [=====] - 0s 2ms/step - loss: 0.4220 - accuracy: 0.
Epoch 12/50
22/22 [=====] - 0s 2ms/step - loss: 0.4127 - accuracy: 0.
Epoch 13/50
22/22 [=====] - 0s 2ms/step - loss: 0.4118 - accuracy: 0.
Epoch 14/50
22/22 [=====] - 0s 3ms/step - loss: 0.4052 - accuracy: 0.
Epoch 15/50
22/22 [=====] - 0s 2ms/step - loss: 0.4019 - accuracy: 0.
Epoch 16/50
22/22 [=====] - 0s 2ms/step - loss: 0.3970 - accuracy: 0.
Epoch 17/50
22/22 [=====] - 0s 2ms/step - loss: 0.3921 - accuracy: 0.
Epoch 18/50
22/22 [=====] - 0s 2ms/step - loss: 0.3884 - accuracy: 0.
Epoch 19/50
22/22 [=====] - 0s 2ms/step - loss: 0.3818 - accuracy: 0.
Epoch 20/50
22/22 [=====] - 0s 2ms/step - loss: 0.3822 - accuracy: 0.
Epoch 21/50
22/22 [=====] - 0s 3ms/step - loss: 0.3739 - accuracy: 0.
Epoch 22/50
22/22 [=====] - 0s 2ms/step - loss: 0.3733 - accuracy: 0.
Epoch 23/50
22/22 [=====] - 0s 2ms/step - loss: 0.3620 - accuracy: 0.
Epoch 24/50
22/22 [=====] - 0s 2ms/step - loss: 0.3606 - accuracy: 0.
Epoch 25/50
22/22 [=====] - 0s 2ms/step - loss: 0.3539 - accuracy: 0.
Epoch 26/50
22/22 [=====] - 0s 2ms/step - loss: 0.3491 - accuracy: 0.
Epoch 27/50
22/22 [=====] - 0s 3ms/step - loss: 0.3427 - accuracy: 0.
Epoch 28/50
22/22 [=====] - 0s 2ms/step - loss: 0.3366 - accuracy: 0.
Epoch 29/50
```

```
result = Nadam_model.evaluate(x_test,y_test)
```

```
3/3 [=====] - 0s 2ms/step - loss: 0.6313 - accuracy: 0.7273
```

```
# Loss of the model:
```

```
print(f"Loss: {result[0]*100}")
```

```
# Accuracy of the model:
```

```
print(f"Accuracy: {result[1]*100}")
```

```
Loss: 63.129788637161255
```

```
Accuracy: 72.72727489471436
```

```
plt.xlabel('Epoch')
```

```
plt.ylabel('loss')
```

```
plt.ylim([0.3, 0.8])
```

```
plt.legend(loc='lower right')
```

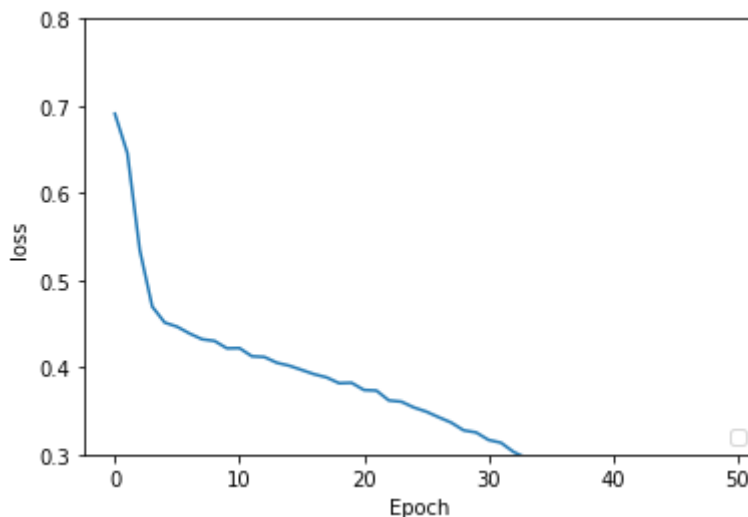
```
test_loss, test_acc = Nadam_model.evaluate(x_test, y_test, verbose=2)
```

```
plt.plot(history.history['loss'], label='loss')
```

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```

```
3/3 - 0s - loss: 0.6313 - accuracy: 0.7273 - 14ms/epoch - 5ms/step
```

```
[<matplotlib.lines.Line2D at 0x7fe32cc530d0>]
```



SGD Optimizer to create a model

```
SGD_model = models.Sequential()
```

```
# First hidden layer:
```

```
SGD_model.add(layers.Dense(100,activation="relu"))
```

```
# First hidden layer:
```

```
SGD_model.add(layers.Dense(50,activation="relu"))
```

```
# Second hidden layer:
```

```
SGD_model.add(layers.Dense(12,activation="relu"))
```

```
# Third hidden layer:
```

```
SGD_model.add(layers.Dense(6,activation="relu"))
```

```
# Output layer:
```

```
SGD_model.add(layers.Dense(1,activation="sigmoid"))
```

```
# optimizer = SGD
SGD_model.compile(optimizer="SGD",loss="binary_crossentropy",metrics=["accuracy"])
history = SGD_model.fit(x_train,y_train,epochs=50,validation_data=(x_test,y_test))
```

```
Epoch 1/50
22/22 [=====] - 1s 19ms/step - loss: 0.6952 - accuracy: 0.
Epoch 2/50
22/22 [=====] - 0s 6ms/step - loss: 0.6767 - accuracy: 0.
Epoch 3/50
22/22 [=====] - 0s 8ms/step - loss: 0.6627 - accuracy: 0.
Epoch 4/50
22/22 [=====] - 0s 4ms/step - loss: 0.6495 - accuracy: 0.
Epoch 5/50
22/22 [=====] - 0s 2ms/step - loss: 0.6346 - accuracy: 0.
Epoch 6/50
22/22 [=====] - 0s 2ms/step - loss: 0.6199 - accuracy: 0.
Epoch 7/50
22/22 [=====] - 0s 2ms/step - loss: 0.6065 - accuracy: 0.
Epoch 8/50
22/22 [=====] - 0s 2ms/step - loss: 0.5943 - accuracy: 0.
Epoch 9/50
22/22 [=====] - 0s 2ms/step - loss: 0.5822 - accuracy: 0.
Epoch 10/50
22/22 [=====] - 0s 2ms/step - loss: 0.5711 - accuracy: 0.
Epoch 11/50
22/22 [=====] - 0s 2ms/step - loss: 0.5599 - accuracy: 0.
Epoch 12/50
22/22 [=====] - 0s 2ms/step - loss: 0.5487 - accuracy: 0.
Epoch 13/50
22/22 [=====] - 0s 2ms/step - loss: 0.5374 - accuracy: 0.
Epoch 14/50
22/22 [=====] - 0s 2ms/step - loss: 0.5272 - accuracy: 0.
Epoch 15/50
22/22 [=====] - 0s 3ms/step - loss: 0.5182 - accuracy: 0.
Epoch 16/50
22/22 [=====] - 0s 2ms/step - loss: 0.5102 - accuracy: 0.
Epoch 17/50
22/22 [=====] - 0s 2ms/step - loss: 0.5031 - accuracy: 0.
Epoch 18/50
22/22 [=====] - 0s 2ms/step - loss: 0.4972 - accuracy: 0.
Epoch 19/50
22/22 [=====] - 0s 2ms/step - loss: 0.4923 - accuracy: 0.
Epoch 20/50
22/22 [=====] - 0s 2ms/step - loss: 0.4880 - accuracy: 0.
Epoch 21/50
22/22 [=====] - 0s 2ms/step - loss: 0.4844 - accuracy: 0.
Epoch 22/50
22/22 [=====] - 0s 2ms/step - loss: 0.4804 - accuracy: 0.
Epoch 23/50
22/22 [=====] - 0s 3ms/step - loss: 0.4771 - accuracy: 0.
Epoch 24/50
22/22 [=====] - 0s 2ms/step - loss: 0.4740 - accuracy: 0.
Epoch 25/50
22/22 [=====] - 0s 2ms/step - loss: 0.4716 - accuracy: 0.
Epoch 26/50
22/22 [=====] - 0s 2ms/step - loss: 0.4690 - accuracy: 0.
Epoch 27/50
22/22 [=====] - 0s 2ms/step - loss: 0.4666 - accuracy: 0.
```



```
Epoch 28/50
22/22 [=====] - 0s 2ms/step - loss: 0.4641 - accuracy: 0.7273
Epoch 29/50
```

```
result = SGD_model.evaluate(x_test,y_test)
```

```
3/3 [=====] - 0s 3ms/step - loss: 0.4931 - accuracy: 0.7273
```

```
# Loss of the model:
```

```
print(f"Loss: {result[0]*100}")
```

```
# Accuracy of the model:
```

```
print(f"Accuracy: {result[1]*100}")
```

```
Loss: 49.313756823539734
```

```
Accuracy: 72.72727489471436
```

```
plt.xlabel('Epoch')
```

```
plt.ylabel('loss')
```

```
plt.ylim([0.3, 0.8])
```

```
plt.legend(loc='lower right')
```

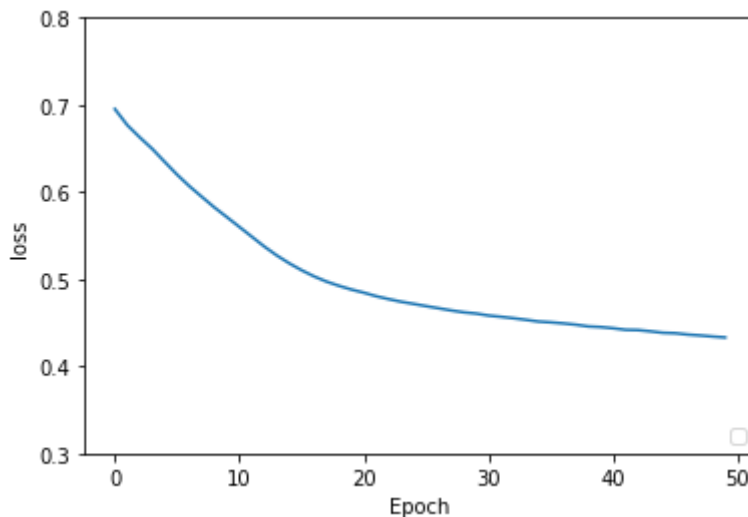
```
test_loss, test_acc = SGD_model.evaluate(x_test, y_test, verbose=2)
```

```
plt.plot(history.history['loss'], label='loss')
```

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```

```
3/3 - 0s - loss: 0.4931 - accuracy: 0.7273 - 19ms/epoch - 6ms/step
```

```
[<matplotlib.lines.Line2D at 0x7fe32ca61390>]
```



[Colab paid products](#) - [Cancel contracts here](#)

