

# ICP5 REPORT

```
+ Code + Text ✓ R
1s 11s
from google.colab import drive
drive.mount('/content/gdrive')

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True)

[6] path_to_csv = '/content/gdrive/My Drive/diabetes.csv'

[7] import keras
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split

# Load dataset
dataset = pd.read_csv(path_to_csv, header=None).values

# Split the dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(dataset[:, 0:8], dataset[:, 8], test_size=0.25, random_state=87)
```

```
+ Code + Text ✓ R
1s 11s
[7] # Set random seed for reproducibility
np.random.seed(155)

# Create a Sequential model
model = Sequential()

# Add Dense layers with 'relu' activation for hidden layers
model.add(Dense(20, input_dim=8, activation='relu')) # First hidden layer
model.add(Dense(15, activation='relu')) # Second hidden layer
model.add(Dense(10, activation='relu')) # Third hidden layer

# Add output layer with 'sigmoid' activation
model.add(Dense(1, activation='sigmoid'))

# Compile the model using binary_crossentropy and adam optimizer
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])

# Train the model
model_fitted = model.fit(X_train, Y_train, epochs=100, initial_epoch=0)

# Print model summary and evaluate accuracy on the test set
print(model.summary())
print(model.evaluate(X_test, Y_test))
```

+ Code + Text

```
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
[7] Epoch 1/100
18/18 ----- 2s 2ms/step - acc: 0.3190 - loss: 5.7087
Epoch 2/100
18/18 ----- 0s 2ms/step - acc: 0.4067 - loss: 2.3054
Epoch 3/100
18/18 ----- 0s 2ms/step - acc: 0.4371 - loss: 1.3191
Epoch 4/100
18/18 ----- 0s 2ms/step - acc: 0.5272 - loss: 1.0544
Epoch 5/100
18/18 ----- 0s 2ms/step - acc: 0.5203 - loss: 0.8864
Epoch 6/100
18/18 ----- 0s 2ms/step - acc: 0.5173 - loss: 0.8031
Epoch 7/100
18/18 ----- 0s 2ms/step - acc: 0.5214 - loss: 0.7698
Epoch 8/100
18/18 ----- 0s 2ms/step - acc: 0.5571 - loss: 0.6963
Epoch 9/100
18/18 ----- 0s 2ms/step - acc: 0.5678 - loss: 0.6779
Epoch 10/100
18/18 ----- 0s 2ms/step - acc: 0.5532 - loss: 0.6569
Epoch 11/100
18/18 ----- 0s 2ms/step - acc: 0.5772 - loss: 0.6605
Epoch 12/100
18/18 ----- 0s 2ms/step - acc: 0.5038 - loss: 0.6706
Epoch 13/100
18/18 ----- 0s 2ms/step - acc: 0.5560 - loss: 0.6441
```

+ Code + Text

```
Epoch 14/100
18/18 ----- 0s 2ms/step - acc: 0.5297 - loss: 0.6524
Epoch 15/100
18/18 ----- 0s 2ms/step - acc: 0.5507 - loss: 0.6466
Epoch 16/100
18/18 ----- 0s 2ms/step - acc: 0.5692 - loss: 0.6386
Epoch 17/100
18/18 ----- 0s 2ms/step - acc: 0.6033 - loss: 0.6139
Epoch 18/100
18/18 ----- 0s 2ms/step - acc: 0.6284 - loss: 0.6236
Epoch 19/100
18/18 ----- 0s 2ms/step - acc: 0.6729 - loss: 0.6436
Epoch 20/100
18/18 ----- 0s 2ms/step - acc: 0.6860 - loss: 0.6442
Epoch 21/100
18/18 ----- 0s 2ms/step - acc: 0.7245 - loss: 0.5918
Epoch 22/100
18/18 ----- 0s 2ms/step - acc: 0.7095 - loss: 0.6080
Epoch 23/100
18/18 ----- 0s 2ms/step - acc: 0.7321 - loss: 0.5977
Epoch 24/100
18/18 ----- 0s 2ms/step - acc: 0.7234 - loss: 0.5963
Epoch 25/100
18/18 ----- 0s 2ms/step - acc: 0.7408 - loss: 0.5698
Epoch 26/100
18/18 ----- 0s 2ms/step - acc: 0.6980 - loss: 0.6211
```

+ Code + Text

```
✓ [7] Epoch 27/100  
11s 18/18 ————— 0s 2ms/step - acc: 0.6840 - loss: 0.6194  
↔ Epoch 28/100  
18/18 ————— 0s 2ms/step - acc: 0.7188 - loss: 0.6051  
Epoch 29/100  
18/18 ————— 0s 2ms/step - acc: 0.7462 - loss: 0.5701  
Epoch 30/100  
18/18 ————— 0s 2ms/step - acc: 0.7327 - loss: 0.5754  
Epoch 31/100  
18/18 ————— 0s 2ms/step - acc: 0.7013 - loss: 0.6131  
Epoch 32/100  
18/18 ————— 0s 2ms/step - acc: 0.7019 - loss: 0.6206  
Epoch 33/100  
18/18 ————— 0s 2ms/step - acc: 0.7502 - loss: 0.5401  
Epoch 34/100  
18/18 ————— 0s 2ms/step - acc: 0.7264 - loss: 0.5833  
Epoch 35/100  
18/18 ————— 0s 2ms/step - acc: 0.7340 - loss: 0.5827  
Epoch 36/100  
18/18 ————— 0s 2ms/step - acc: 0.7660 - loss: 0.5447  
Epoch 37/100  
18/18 ————— 0s 2ms/step - acc: 0.7393 - loss: 0.5799  
Epoch 38/100  
18/18 ————— 0s 2ms/step - acc: 0.7176 - loss: 0.5618  
Epoch 39/100  
18/18 ————— 0s 2ms/step - acc: 0.7424 - loss: 0.5797
```

+ Code + Text

```
✓ Epoch 40/100  
11s 18/18 ————— 0s 2ms/step - acc: 0.7393 - loss: 0.5519  
↔ Epoch 41/100  
18/18 ————— 0s 2ms/step - acc: 0.7300 - loss: 0.5793  
Epoch 42/100  
18/18 ————— 0s 2ms/step - acc: 0.7427 - loss: 0.5491  
Epoch 43/100  
18/18 ————— 0s 2ms/step - acc: 0.7539 - loss: 0.5540  
Epoch 44/100  
18/18 ————— 0s 2ms/step - acc: 0.7690 - loss: 0.5501  
Epoch 45/100  
18/18 ————— 0s 2ms/step - acc: 0.7384 - loss: 0.5427  
Epoch 46/100  
18/18 ————— 0s 2ms/step - acc: 0.7378 - loss: 0.5531  
Epoch 47/100  
18/18 ————— 0s 2ms/step - acc: 0.7457 - loss: 0.5522  
Epoch 48/100  
18/18 ————— 0s 2ms/step - acc: 0.7155 - loss: 0.5701  
Epoch 49/100  
18/18 ————— 0s 2ms/step - acc: 0.6924 - loss: 0.5807  
Epoch 50/100  
18/18 ————— 0s 2ms/step - acc: 0.7113 - loss: 0.5718  
Epoch 51/100  
18/18 ————— 0s 2ms/step - acc: 0.7524 - loss: 0.5103  
Epoch 52/100  
18/18 ————— 0s 2ms/step - acc: 0.7496 - loss: 0.5274
```

+ Code + Text

✓ 11s

▶

↺

Epoch 53/100  
18/18 ————— 0s 2ms/step - acc: 0.7143 - loss: 0.5703  
Epoch 54/100  
18/18 ————— 0s 2ms/step - acc: 0.7197 - loss: 0.5559  
Epoch 55/100  
18/18 ————— 0s 2ms/step - acc: 0.7173 - loss: 0.5576  
Epoch 56/100  
18/18 ————— 0s 2ms/step - acc: 0.7615 - loss: 0.5316  
Epoch 57/100  
18/18 ————— 0s 2ms/step - acc: 0.7287 - loss: 0.5506  
Epoch 58/100  
18/18 ————— 0s 2ms/step - acc: 0.7461 - loss: 0.5225  
Epoch 59/100  
18/18 ————— 0s 2ms/step - acc: 0.7556 - loss: 0.5372  
Epoch 60/100  
18/18 ————— 0s 2ms/step - acc: 0.7567 - loss: 0.5275  
Epoch 61/100  
18/18 ————— 0s 2ms/step - acc: 0.7489 - loss: 0.5255  
Epoch 62/100  
18/18 ————— 0s 2ms/step - acc: 0.7605 - loss: 0.5334  
Epoch 63/100  
18/18 ————— 0s 2ms/step - acc: 0.7875 - loss: 0.5061  
Epoch 64/100  
18/18 ————— 0s 2ms/step - acc: 0.7417 - loss: 0.5325  
Epoch 65/100  
18/18 ————— 0s 2ms/step - acc: 0.7579 - loss: 0.5309

+ Code + Text

✓ 11s

▶

↺

Epoch 66/100  
18/18 ————— 0s 2ms/step - acc: 0.7366 - loss: 0.5271  
Epoch 67/100  
18/18 ————— 0s 2ms/step - acc: 0.7525 - loss: 0.5385  
Epoch 68/100  
18/18 ————— 0s 2ms/step - acc: 0.7637 - loss: 0.5188  
Epoch 69/100  
18/18 ————— 0s 2ms/step - acc: 0.7603 - loss: 0.5020  
Epoch 70/100  
18/18 ————— 0s 2ms/step - acc: 0.7391 - loss: 0.5333  
Epoch 71/100  
18/18 ————— 0s 2ms/step - acc: 0.7824 - loss: 0.4815  
Epoch 72/100  
18/18 ————— 0s 2ms/step - acc: 0.7692 - loss: 0.5031  
Epoch 73/100  
18/18 ————— 0s 2ms/step - acc: 0.7294 - loss: 0.5682  
Epoch 74/100  
18/18 ————— 0s 2ms/step - acc: 0.7502 - loss: 0.5285  
Epoch 75/100  
18/18 ————— 0s 2ms/step - acc: 0.7369 - loss: 0.5336  
Epoch 76/100  
18/18 ————— 0s 2ms/step - acc: 0.7692 - loss: 0.5219  
Epoch 77/100  
18/18 ————— 0s 2ms/step - acc: 0.7408 - loss: 0.5452  
Epoch 78/100  
18/18 ————— 0s 2ms/step - acc: 0.7461 - loss: 0.5277

+ Code+ Text

11s

▶

↺

Epoch 79/100  
18/18 ————— 0s 2ms/step - acc: 0.7776 - loss: 0.4981  
Epoch 80/100  
18/18 ————— 0s 2ms/step - acc: 0.7296 - loss: 0.5524  
Epoch 81/100  
18/18 ————— 0s 2ms/step - acc: 0.7573 - loss: 0.5163  
Epoch 82/100  
18/18 ————— 0s 2ms/step - acc: 0.7412 - loss: 0.5288  
Epoch 83/100  
18/18 ————— 0s 2ms/step - acc: 0.7412 - loss: 0.5460  
Epoch 84/100  
18/18 ————— 0s 2ms/step - acc: 0.7534 - loss: 0.5281  
Epoch 85/100  
18/18 ————— 0s 2ms/step - acc: 0.7295 - loss: 0.5537  
Epoch 86/100  
18/18 ————— 0s 2ms/step - acc: 0.7619 - loss: 0.5079  
Epoch 87/100  
18/18 ————— 0s 2ms/step - acc: 0.7474 - loss: 0.4902  
Epoch 88/100  
18/18 ————— 0s 2ms/step - acc: 0.7400 - loss: 0.5420  
Epoch 89/100  
18/18 ————— 0s 2ms/step - acc: 0.7941 - loss: 0.4994  
Epoch 90/100  
18/18 ————— 0s 2ms/step - acc: 0.7542 - loss: 0.5333  
Epoch 91/100  
18/18 ————— 0s 2ms/step - acc: 0.7308 - loss: 0.5384  
Epoch 92/100

+ Code+ Text

11s

▶

↺

Epoch 92/100  
18/18 ————— 0s 2ms/step - acc: 0.7535 - loss: 0.5134  
Epoch 93/100  
18/18 ————— 0s 2ms/step - acc: 0.7440 - loss: 0.5140  
Epoch 94/100  
18/18 ————— 0s 2ms/step - acc: 0.7574 - loss: 0.5116  
Epoch 95/100  
18/18 ————— 0s 2ms/step - acc: 0.7535 - loss: 0.5300  
Epoch 96/100  
18/18 ————— 0s 2ms/step - acc: 0.7358 - loss: 0.5309  
Epoch 97/100  
18/18 ————— 0s 2ms/step - acc: 0.7353 - loss: 0.5178  
Epoch 98/100  
18/18 ————— 0s 2ms/step - acc: 0.7446 - loss: 0.5244  
Epoch 99/100  
18/18 ————— 0s 2ms/step - acc: 0.7546 - loss: 0.5112  
Epoch 100/100  
18/18 ————— 0s 2ms/step - acc: 0.7265 - loss: 0.5301  
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 20)	180
dense_1 (Dense)	(None, 10)	315
dense_2 (Dense)	(None, 10)	160

+ Code + Text

1s

dense\_3 (Dense)

(None, 1)

11

RAM

Disk

↺

Total params: 2,000 (7.82 KB)  
Trainable params: 666 (2.60 KB)  
Non-trainable params: 0 (0.00 B)  
Optimizer params: 1,334 (5.21 KB)  
None  
6/6 ————— 0s 2ms/step - acc: 0.7188 - loss: 0.5526  
[0.5714862942695618, 0.7083333134651184]

10s

[8] import keras  
import pandas as pd  
import numpy as np  
from keras.models import Sequential  
from keras.layers import Dense  
from sklearn.model\_selection import train\_test\_split  
  
# Load dataset  
dataset = pd.read\_csv(path\_to\_csv, header=None).values  
  
# Split the dataset into training and testing sets  
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(dataset[:, 0:8], dataset[:, 8], test\_size=0.25, random\_state=87)

+ Code + Text

10s

[8] # Set random seed for reproducibility  
np.random.seed(155)  
  
# Create a Sequential model  
model = Sequential()  
  
# Add Dense layers with 'relu' activation for hidden layers  
model.add(Dense(20, input\_dim=8, activation='relu')) # First hidden layer  
model.add(Dense(15, activation='relu')) # Second hidden layer  
model.add(Dense(10, activation='relu')) # Third hidden layer  
  
# Add output layer with 'sigmoid' activation  
model.add(Dense(1, activation='sigmoid'))  
  
# Compile the model using binary crossentropy and adam optimizer  
model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['acc'])  
  
# Train the model  
model\_fitted = model.fit(X\_train, Y\_train, epochs=100, initial\_epoch=0)  
  
# Print model summary and evaluate accuracy on the test set  
print(model.summary())  
print(model.evaluate(X\_test, Y\_test))

+ Code + Text

```
✓ [8] Epoch 1/100
10s /usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `in
    super().__init__(activity_regularizer=activity_regularizer, **kwargs)
18/18 ————— 2s 4ms/step - acc: 0.6069 - loss: 2.6103
Epoch 2/100
18/18 ————— 0s 3ms/step - acc: 0.4098 - loss: 1.3969
Epoch 3/100
18/18 ————— 0s 2ms/step - acc: 0.4338 - loss: 1.1178
Epoch 4/100
18/18 ————— 0s 2ms/step - acc: 0.6133 - loss: 0.8093
Epoch 5/100
18/18 ————— 0s 2ms/step - acc: 0.5246 - loss: 0.8328
Epoch 6/100
18/18 ————— 0s 2ms/step - acc: 0.5479 - loss: 0.7467
Epoch 7/100
18/18 ————— 0s 2ms/step - acc: 0.6121 - loss: 0.7069
Epoch 8/100
18/18 ————— 0s 2ms/step - acc: 0.5838 - loss: 0.7007
Epoch 9/100
18/18 ————— 0s 2ms/step - acc: 0.6276 - loss: 0.7060
Epoch 10/100
18/18 ————— 0s 2ms/step - acc: 0.6822 - loss: 0.6447
Epoch 11/100
18/18 ————— 0s 2ms/step - acc: 0.6323 - loss: 0.6508
Epoch 12/100
18/18 ————— 0s 2ms/step - acc: 0.6680 - loss: 0.6438
```

+ Code + Text

```
✓ [8] Epoch 13/100
10s 18/18 ————— 0s 2ms/step - acc: 0.6713 - loss: 0.6273
Epoch 14/100
18/18 ————— 0s 2ms/step - acc: 0.6869 - loss: 0.6233
Epoch 15/100
18/18 ————— 0s 2ms/step - acc: 0.6623 - loss: 0.6170
Epoch 16/100
18/18 ————— 0s 2ms/step - acc: 0.6733 - loss: 0.6254
Epoch 17/100
18/18 ————— 0s 2ms/step - acc: 0.6852 - loss: 0.6094
Epoch 18/100
18/18 ————— 0s 2ms/step - acc: 0.6919 - loss: 0.6023
Epoch 19/100
18/18 ————— 0s 2ms/step - acc: 0.7038 - loss: 0.6025
Epoch 20/100
18/18 ————— 0s 2ms/step - acc: 0.7065 - loss: 0.5702
Epoch 21/100
18/18 ————— 0s 2ms/step - acc: 0.7127 - loss: 0.5799
Epoch 22/100
18/18 ————— 0s 2ms/step - acc: 0.7280 - loss: 0.5637
Epoch 23/100
18/18 ————— 0s 2ms/step - acc: 0.6904 - loss: 0.5891
Epoch 24/100
18/18 ————— 0s 2ms/step - acc: 0.7165 - loss: 0.5772
Epoch 25/100
18/18 ————— 0s 2ms/step - acc: 0.7032 - loss: 0.5721
```

+ Code + Text

```
✓ 10s 8 Epoch 26/100
18/18 ————— 0s 2ms/step - acc: 0.6943 - loss: 0.6148
↔ Epoch 27/100
18/18 ————— 0s 2ms/step - acc: 0.6907 - loss: 0.5895
Epoch 28/100
18/18 ————— 0s 2ms/step - acc: 0.7139 - loss: 0.5606
Epoch 29/100
18/18 ————— 0s 2ms/step - acc: 0.7101 - loss: 0.5705
Epoch 30/100
18/18 ————— 0s 2ms/step - acc: 0.7075 - loss: 0.5848
Epoch 31/100
18/18 ————— 0s 2ms/step - acc: 0.7040 - loss: 0.5891
Epoch 32/100
18/18 ————— 0s 2ms/step - acc: 0.7141 - loss: 0.5516
Epoch 33/100
18/18 ————— 0s 2ms/step - acc: 0.7080 - loss: 0.5693
Epoch 34/100
18/18 ————— 0s 2ms/step - acc: 0.7145 - loss: 0.5758
Epoch 35/100
18/18 ————— 0s 2ms/step - acc: 0.7288 - loss: 0.5572
Epoch 36/100
18/18 ————— 0s 2ms/step - acc: 0.7356 - loss: 0.5421
Epoch 37/100
18/18 ————— 0s 2ms/step - acc: 0.7327 - loss: 0.5439
Epoch 38/100
18/18 ————— 0s 2ms/step - acc: 0.6894 - loss: 0.5916
```

+ Code + Text

```
✓ 10s [8] Epoch 39/100
18/18 ————— 0s 2ms/step - acc: 0.6899 - loss: 0.5667
↔ Epoch 40/100
18/18 ————— 0s 2ms/step - acc: 0.7233 - loss: 0.5868
Epoch 41/100
18/18 ————— 0s 2ms/step - acc: 0.6997 - loss: 0.5847
Epoch 42/100
18/18 ————— 0s 2ms/step - acc: 0.7397 - loss: 0.5511
Epoch 43/100
18/18 ————— 0s 2ms/step - acc: 0.6904 - loss: 0.5578
Epoch 44/100
18/18 ————— 0s 2ms/step - acc: 0.7027 - loss: 0.5815
Epoch 45/100
18/18 ————— 0s 2ms/step - acc: 0.7123 - loss: 0.5803
Epoch 46/100
18/18 ————— 0s 2ms/step - acc: 0.7184 - loss: 0.5542
Epoch 47/100
18/18 ————— 0s 2ms/step - acc: 0.7063 - loss: 0.5961
Epoch 48/100
18/18 ————— 0s 2ms/step - acc: 0.6950 - loss: 0.5709
Epoch 49/100
18/18 ————— 0s 2ms/step - acc: 0.7237 - loss: 0.5583
Epoch 50/100
18/18 ————— 0s 2ms/step - acc: 0.7089 - loss: 0.5568
Epoch 51/100
18/18 ————— 0s 2ms/step - acc: 0.7244 - loss: 0.5613
```



+ Code + Text

✓ 10s

8

Epoch 52/100  
18/18 0s 2ms/step - acc: 0.7393 - loss: 0.5547

↺

Epoch 53/100  
18/18 0s 2ms/step - acc: 0.7321 - loss: 0.5306

Epoch 54/100  
18/18 0s 2ms/step - acc: 0.6965 - loss: 0.5663

Epoch 55/100  
18/18 0s 2ms/step - acc: 0.7224 - loss: 0.5497

Epoch 56/100  
18/18 0s 2ms/step - acc: 0.7690 - loss: 0.5140

Epoch 57/100  
18/18 0s 2ms/step - acc: 0.7089 - loss: 0.5548

Epoch 58/100  
18/18 0s 2ms/step - acc: 0.7242 - loss: 0.5640

Epoch 59/100  
18/18 0s 2ms/step - acc: 0.7055 - loss: 0.5518

Epoch 60/100  
18/18 0s 2ms/step - acc: 0.7469 - loss: 0.5266

Epoch 61/100  
18/18 0s 2ms/step - acc: 0.7345 - loss: 0.5327

Epoch 62/100  
18/18 0s 2ms/step - acc: 0.7365 - loss: 0.5531

Epoch 63/100  
18/18 0s 2ms/step - acc: 0.7314 - loss: 0.5246

Epoch 64/100  
18/18 0s 2ms/step - acc: 0.7077 - loss: 0.5969

+ Code + Text

✓ 10s

8

Epoch 65/100  
18/18 0s 2ms/step - acc: 0.7405 - loss: 0.5322

↺

Epoch 66/100  
18/18 0s 2ms/step - acc: 0.7223 - loss: 0.5392

Epoch 67/100  
18/18 0s 3ms/step - acc: 0.7577 - loss: 0.5323

Epoch 68/100  
18/18 0s 2ms/step - acc: 0.7115 - loss: 0.5403

Epoch 69/100  
18/18 0s 2ms/step - acc: 0.7405 - loss: 0.5426

Epoch 70/100  
18/18 0s 2ms/step - acc: 0.7508 - loss: 0.5392

Epoch 71/100  
18/18 0s 2ms/step - acc: 0.7321 - loss: 0.5465

Epoch 72/100  
18/18 0s 2ms/step - acc: 0.7093 - loss: 0.5626

Epoch 73/100  
18/18 0s 2ms/step - acc: 0.7263 - loss: 0.5426

Epoch 74/100  
18/18 0s 2ms/step - acc: 0.7489 - loss: 0.5164

Epoch 75/100  
18/18 0s 2ms/step - acc: 0.7499 - loss: 0.5568

Epoch 76/100  
18/18 0s 2ms/step - acc: 0.7441 - loss: 0.5123

Epoch 77/100  
18/18 0s 2ms/step - acc: 0.7385 - loss: 0.5295

+ Code + Text

```
Epoch 78/100
18/18 ————— 0s 2ms/step - acc: 0.7137 - loss: 0.5510
Epoch 79/100
18/18 ————— 0s 2ms/step - acc: 0.7633 - loss: 0.5052
```

```
[9] path_to_csv1 = '/content/gdrive/My Drive/breastcancer.csv'
```

```
[10] import keras
import pandas
from keras.models import Sequential
from keras.layers import Dense, Activation

# load dataset
from sklearn.model_selection import train_test_split
#from sklearn.preprocessing import StandardScaler
import pandas as pd
import numpy as np

dataset = pd.read_csv(path_to_csv1, header=None).values

X = dataset[1:, 2:-1] # Features
Y = dataset[1:, -1]  # Labels (M or B)
```

+ Code + Text

```
[10] # Convert labels to binary format
Y = np.where(Y == 'M', 1, 0) # M -> 1, B -> 0

#Convert to numeric
X = X.astype(np.float64) # Convert X to numeric

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
                                                    test_size=0.25, random_state=87)

np.random.seed(155)
my_first_nn = Sequential() # create model
my_first_nn.add(Dense(20, input_dim=30, activation='relu')) # hidden layer
my_first_nn.add(Dense(30, activation='relu')) # hidden layer
my_first_nn.add(Dense(40, activation='relu')) # hidden layer
my_first_nn.add(Dense(50, activation='relu')) # hidden layer

my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                     initial_epoch=0)

print(my_first_nn.summary())
print(my_first_nn.evaluate(X_test, Y_test))
```

```
+ Code + Text
[10] Epoch 1/100
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape` to `Input`
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
14/14 ----- 2s 2ms/step - acc: 0.4632 - loss: 21.6955
Epoch 2/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 6.5355e-18
Epoch 3/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.7323e-21
Epoch 4/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 9.4181e-23
Epoch 5/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 5.3417e-23
Epoch 6/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.4529e-23
Epoch 7/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 6.7339e-24
Epoch 8/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 3.4758e-23
Epoch 9/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 6.2038e-23
Epoch 10/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 7.3121e-24
Epoch 11/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 4.3365e-23
```

```
+ Code + Text
[10] Epoch 13/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.3942e-23
Epoch 14/100
14/14 ----- 0s 3ms/step - acc: 1.0000 - loss: 1.7827e-23
Epoch 15/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 8.3061e-24
Epoch 16/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.7197e-23
Epoch 17/100
14/14 ----- 0s 3ms/step - acc: 1.0000 - loss: 7.8483e-24
Epoch 18/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 4.4026e-23
Epoch 19/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 2.8664e-23
Epoch 20/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 9.2286e-24
Epoch 21/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.1670e-23
Epoch 22/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 2.3073e-23
Epoch 23/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.0445e-23
Epoch 24/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 2.3174e-23
Epoch 25/100
14/14 ----- 0s 2ms/step - acc: 1.0000 - loss: 1.9787e-23
```

+ Code + Text

```
✓ [10] Epoch 26/100
14s 14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.5664e-24
Epoch 27/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.8833e-23
Epoch 28/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.1647e-23
Epoch 29/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.2143e-23
Epoch 30/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.5648e-23
Epoch 31/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.7845e-23
Epoch 32/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.7683e-24
Epoch 33/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.6667e-23
Epoch 34/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 8.4303e-24
Epoch 35/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.4461e-23
Epoch 36/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.6889e-23
Epoch 37/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 4.3336e-23
Epoch 38/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 4.3806e-23
```

+ Code + Text

```
✓ [10] Epoch 39/100
14s 14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.8012e-24
Epoch 40/100
14/14 ————— 0s 4ms/step - acc: 1.0000 - loss: 7.5741e-24
Epoch 41/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 9.9436e-24
Epoch 42/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 3.4460e-23
Epoch 43/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.4380e-23
Epoch 44/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.1865e-23
Epoch 45/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 2.7703e-23
Epoch 46/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 5.0736e-24
Epoch 47/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.1375e-23
Epoch 48/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.8679e-24
Epoch 49/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.6588e-23
Epoch 50/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 7.7012e-24
Epoch 51/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 8.1386e-24
```

+ Code + Text

```
✓ [10] Epoch 52/100
14s 14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.4281e-23
Epoch 53/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.1816e-23
Epoch 54/100
14/14 ————— 0s 4ms/step - acc: 1.0000 - loss: 6.1814e-23
Epoch 55/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.4491e-23
Epoch 56/100
14/14 ————— 0s 4ms/step - acc: 1.0000 - loss: 1.8226e-23
Epoch 57/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 2.8015e-23
Epoch 58/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 7.6719e-24
Epoch 59/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 9.0052e-24
Epoch 60/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.5264e-24
Epoch 61/100
14/14 ————— 0s 6ms/step - acc: 1.0000 - loss: 6.2031e-24
Epoch 62/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 8.4618e-24
Epoch 63/100
14/14 ————— 0s 4ms/step - acc: 1.0000 - loss: 1.1060e-23
Epoch 64/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.7328e-24
```

+ Code + Text

```
✓ [10] Epoch 65/100
14s 14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.0989e-23
Epoch 66/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.9461e-24
Epoch 67/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.1563e-23
Epoch 68/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.4600e-23
Epoch 69/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.7841e-23
Epoch 70/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.6593e-23
Epoch 71/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.9723e-23
Epoch 72/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.3922e-23
Epoch 73/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.3356e-24
Epoch 74/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.2959e-23
Epoch 75/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.0094e-23
Epoch 76/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.0033e-23
Epoch 77/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.1748e-23
```

+ Code+ Text

✓14s

🔄

Epoch 78/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.4641e-23  
Epoch 79/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.4753e-23  
Epoch 80/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.9713e-23  
Epoch 81/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.3618e-23  
Epoch 82/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.4926e-23  
Epoch 83/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.4703e-23  
Epoch 84/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.9802e-24  
Epoch 85/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.4335e-23  
Epoch 86/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.4886e-23  
Epoch 87/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.4854e-23  
Epoch 88/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.7824e-23  
Epoch 89/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.7138e-24  
Epoch 90/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.1187e-23

+ Code+ Text

✓14s

🔄

Epoch 91/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.4328e-24  
Epoch 92/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.3039e-23  
Epoch 93/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.6697e-23  
Epoch 94/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.3393e-23  
Epoch 95/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.2194e-24  
Epoch 96/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.2559e-23  
Epoch 97/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.1150e-23  
Epoch 98/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.9519e-23  
Epoch 99/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.4001e-23  
Epoch 100/100  
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.4392e-23  
Model: "sequential\_2"

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 20)	620
dense_9 (Dense)	(None, 30)	630

+ Code + Text

✓ [10] 14s	dense_10 (Dense)	(None, 40)	1,240
↔	dense_11 (Dense)	(None, 50)	2,050
	dense_12 (Dense)	(None, 1)	51

Total params: 13,775 (53.81 KB)  
Trainable params: 4,591 (17.93 KB)  
Non-trainable params: 0 (0.00 B)  
Optimizer params: 9,184 (35.88 KB)  
None  
5/5 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.2542e-19  
[2.4141023193055175e-19, 1.0]

```
✓ [15] 12s import keras
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load dataset (ensure to replace 'path_to_csv1' with the actual path to your CSV file)
dataset = pd.read_csv(path_to_csv1, header=None).values
```

+ Code + Text

```
✓ [15] 2s # Features and Labels extraction
X = dataset[1:, 2:-1] # Assuming your features are from the 3rd column to the second last
Y = dataset[1:, -1] # Labels (M or B)

# Convert labels to binary format
Y = np.where(Y == 'M', 1, 0) # M -> 1, B -> 0

# Convert features to numeric
X = X.astype(np.float64)

# Split data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=87)

# Normalize the data
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Create and compile the model
np.random.seed(155)
my_first_nn = Sequential()
my_first_nn.add(Dense(20, input_dim=X_train.shape[1], activation='relu')) # Input and hidden layer
my_first_nn.add(Dense(30, activation='relu')) # Hidden layer
my_first_nn.add(Dense(40, activation='relu')) # Hidden layer
```

+ Code + Text

```
✓ [15] my_first_nn.add(Dense(50, activation='relu')) # Hidden layer
2s my_first_nn.add(Dense(1, activation='sigmoid')) # Output layer

my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])

# Fit the model
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100, initial_epoch=0)

# Model summary
print(my_first_nn.summary())

# Model evaluation
print(my_first_nn.evaluate(X_test, Y_test))
```

⚡ /usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input\_shape` to

```
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/100
14/14 ━━━━━━━━━━━ 4s 4ms/step - acc: 0.2897 - loss: 0.7786
Epoch 2/100
14/14 ━━━━━━━━━━━ 0s 3ms/step - acc: 1.0000 - loss: 0.4263
Epoch 3/100
14/14 ━━━━━━━━━━━ 0s 4ms/step - acc: 1.0000 - loss: 0.1664
Epoch 4/100
14/14 ━━━━━━━━━━━ 0s 3ms/step - acc: 1.0000 - loss: 0.0396
```

+ Code + Text

```
✓ [15] Epoch 5/100
2s 14/14 ━━━━━━━━━━━ 0s 5ms/step - acc: 1.0000 - loss: 0.0133
⚡ Epoch 6/100
14/14 ━━━━━━━━━━━ 0s 4ms/step - acc: 1.0000 - loss: 0.0063
Epoch 7/100
14/14 ━━━━━━━━━━━ 0s 3ms/step - acc: 1.0000 - loss: 0.0037
Epoch 8/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 0.0023
Epoch 9/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 0.0018
Epoch 10/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 0.0016
Epoch 11/100
14/14 ━━━━━━━━━━━ 0s 3ms/step - acc: 1.0000 - loss: 9.3270e-04
Epoch 12/100
14/14 ━━━━━━━━━━━ 0s 3ms/step - acc: 1.0000 - loss: 0.0011
Epoch 13/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 0.0010
Epoch 14/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 7.6782e-04
Epoch 15/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 7.2956e-04
Epoch 16/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 4.0933e-04
Epoch 17/100
14/14 ━━━━━━━━━━━ 0s 2ms/step - acc: 1.0000 - loss: 3.1689e-04
```



+ Code + Text

```
✓ [15] Epoch 18/100
12s 14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.1724e-04
Epoch 19/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.5264e-04
Epoch 20/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.0318e-04
Epoch 21/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.5145e-04
Epoch 22/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.2682e-04
Epoch 23/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.0170e-04
Epoch 24/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.3490e-04
Epoch 25/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.2762e-04
Epoch 26/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.4767e-04
Epoch 27/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.1058e-04
Epoch 28/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 8.7684e-05
Epoch 29/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.5570e-05
Epoch 30/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.0252e-04
```

+ Code + Text

```
✓ [15] Epoch 31/100
12s 14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.4525e-05
Epoch 32/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.0625e-04
Epoch 33/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.5004e-05
Epoch 34/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.0934e-05
Epoch 35/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.3536e-05
Epoch 36/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.4080e-05
Epoch 37/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.5930e-05
Epoch 38/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.4563e-05
Epoch 39/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.4467e-05
Epoch 40/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 3.2966e-05
Epoch 41/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.0412e-05
Epoch 42/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.8139e-05
Epoch 43/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 3.7831e-05
```

+ Code + Text

```
✓ [15] Epoch 44/100
2s 14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.2630e-05
↔ Epoch 45/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.1653e-05
Epoch 46/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.7575e-05
Epoch 47/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.0681e-05
Epoch 48/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.7836e-05
Epoch 49/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.4274e-05
Epoch 50/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.9887e-05
Epoch 51/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.4160e-05
Epoch 52/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.3795e-05
Epoch 53/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.9672e-05
Epoch 54/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 2.0558e-05
Epoch 55/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.2282e-05
Epoch 56/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.0298e-05
```

+ Code + Text

```
✓ [15] Epoch 57/100
2s 14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 2.1414e-05
↔ Epoch 58/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.1045e-05
Epoch 59/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.5833e-06
Epoch 60/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.2424e-05
Epoch 61/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.4807e-06
Epoch 62/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.9724e-05
Epoch 63/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.7996e-05
Epoch 64/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.2692e-05
Epoch 65/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.3232e-05
Epoch 66/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.0194e-05
Epoch 67/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.8407e-06
Epoch 68/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.3721e-05
Epoch 69/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.7360e-06
Epoch 70/100
```

+ Code + Text

```
[15] Epoch 70/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.1026e-05
Epoch 71/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 8.8767e-06
Epoch 72/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.6110e-06
Epoch 73/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 1.0805e-05
Epoch 74/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.2086e-06
Epoch 75/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.8259e-06
Epoch 76/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 1.1085e-05
Epoch 77/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.4063e-06
Epoch 78/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.6654e-06
Epoch 79/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.1753e-06
Epoch 80/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.7801e-06
Epoch 81/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 7.5057e-06
Epoch 82/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 5.8740e-06
```

+ Code + Text

```
[15] Epoch 83/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 5.1262e-06
Epoch 84/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 8.9123e-06
Epoch 85/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 9.8815e-06
Epoch 86/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.3333e-06
Epoch 87/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 9.3847e-06
Epoch 88/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.2030e-06
Epoch 89/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.3796e-06
Epoch 90/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.5705e-06
Epoch 91/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.0552e-06
Epoch 92/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.5655e-06
Epoch 93/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 5.0981e-06
Epoch 94/100
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 4.8528e-06
Epoch 95/100
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 6.0609e-06
```

+ Code + Text

```
✓ [15] Epoch 96/100  
12s 14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 4.0042e-06  
Epoch 97/100  
14/14 ————— 0s 3ms/step - acc: 1.0000 - loss: 4.8173e-06  
Epoch 98/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.0151e-06  
Epoch 99/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 6.1184e-06  
Epoch 100/100  
14/14 ————— 0s 2ms/step - acc: 1.0000 - loss: 2.8973e-06  
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
dense_13 (Dense)	(None, 20)	620
dense_14 (Dense)	(None, 30)	630
dense_15 (Dense)	(None, 40)	1,240
dense_16 (Dense)	(None, 50)	2,050
dense_17 (Dense)	(None, 1)	51

Total params: 13,775 (53.81 KB)  
Trainable params: 4,591 (17.93 KB)  
Non-trainable params: 0 (0.00 B)  
Optimizer params: 9,184 (35.88 KB)

+ Code + Text

```
✓ [15] None  
12s 5/5 ————— 0s 3ms/step - acc: 1.0000 - loss: 7.9473e-06  
[7.064252713462338e-06, 1.0]
```

```
✓ [16] path_to_csv1 = '/content/gdrive/My Drive/breastcancer.csv'
```

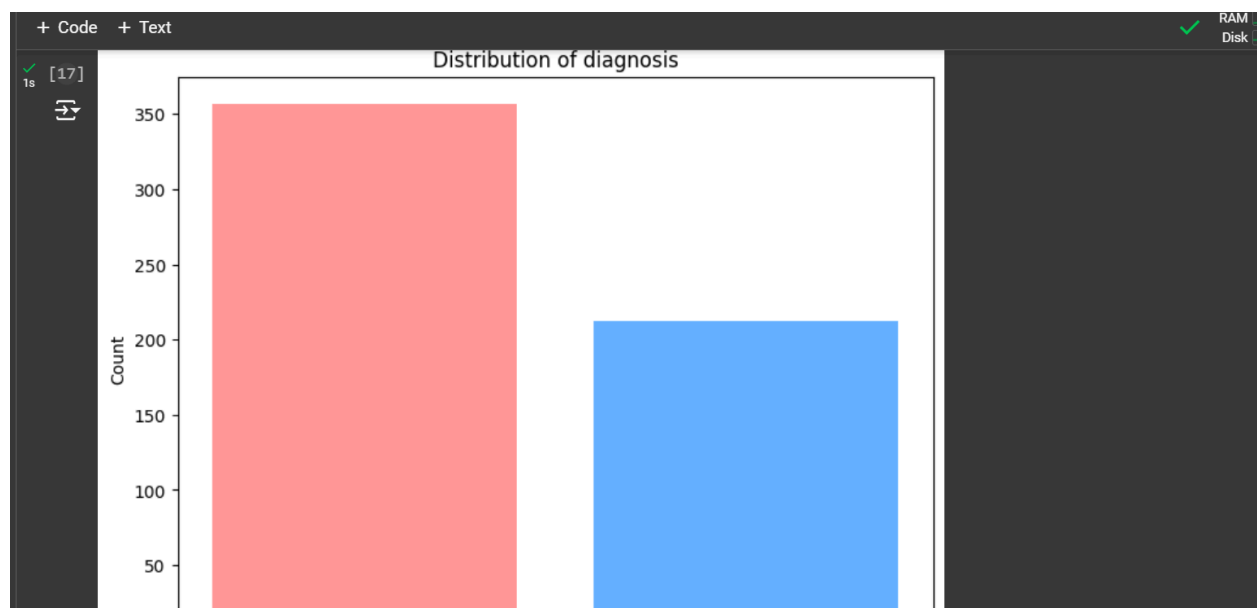
```
✓ [17] import pandas as pd  
1s import matplotlib.pyplot as plt  
  
# Load the dataset  
data = pd.read_csv('/content/gdrive/My Drive/breastcancer.csv')  
  
# Print the column names to help you choose the correct column  
print(data.columns)  
  
# Replace 'diagnosis' with the actual column name for labels, if needed  
label_column = 'diagnosis' # Example: 'diagnosis' for benign/malignant  
  
# Count the occurrences of each class  
label_counts = data[label_column].value_counts()
```

```
+ Code + Text
# Create a bar graph
[17] plt.figure(figsize=(8, 6))
plt.bar(label_counts.index, label_counts.values, color=['#ff9999', '#66b3ff'])

# Add title and labels
plt.title(f'Distribution of {label_column}')
plt.xlabel('Class')
plt.ylabel('Count')

# Show the plot
plt.show()

Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```



GITHUB REPO:- <https://github.com/akshaykumarpthem/bda.git>