

```
In [2]: import pandas as pd
import numpy as np
```

```
In [3]: df = pd.read_csv('Practice_car_Data.csv')
```

```
In [4]: df.head()
```

Out[4]:

	Car ID	Estimated Cost	Mileage(kmpl)	Engine(CC)	Purchased
0	15624510	19000	26.60	998.0	0
1	15810944	20000	19.67	1582.0	0
2	15668575	43000	18.20	1199.0	0
3	15603246	57000	20.77	1248.0	0
4	15804002	76000	15.20	1968.0	0

```
In [5]: df.shape
```

Out[5]: (400, 5)

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Car ID          400 non-null    int64  
 1   Estimated Cost  400 non-null    int64  
 2   Mileage(kmpl)  400 non-null    float64 
 3   Engine(CC)      400 non-null    float64 
 4   Purchased       400 non-null    int64  
dtypes: float64(2), int64(3)
memory usage: 15.8 KB
```

```
In [8]: df.duplicated().sum()
```

Out[8]: np.int64(0)

```
In [9]: df['Purchased'].value_counts()
```

Out[9]:

count

Purchased	
0	257
1	143

dtype: int64

```
In [11]: # since car Id is not worthy feature to effect the purchasing so it is not useful f
df.drop(columns=['Car ID'], inplace = True)
```

```
In [12]: df.head()
```

Out[12]:

	Estimated Cost	Mileage(kmpl)	Engine(CC)	Purchased
0	19000	26.60	998.0	0
1	20000	19.67	1582.0	0
2	43000	18.20	1199.0	0
3	57000	20.77	1248.0	0
4	76000	15.20	1968.0	0

```
In [152... from sklearn.model_selection import train_test_split
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
```

```
In [153... from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
In [154... X_train_scaled[1:5]
```

```
Out[154... array([[-1.22993871,  0.50378857, -0.23443574],
       [ 1.853544 , -0.78876954, -0.36237838],
       [-0.90995465,  0.11040132, -0.70656209],
       [-1.28811763,  1.06577036, -0.61826365]])
```

```
In [168... import tensorflow
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
```

```
In [198... model = Sequential()
model.add(Dense(64, activation='relu', input_dim = 3))
```

```
model.add(Dense(32,activation='relu'))  
  
#adding output Layer with one node  
model.add(Dense(1,activation='sigmoid'))
```

/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in the model instead.

```
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [199... model.summary()

Model: "sequential_18"

Layer (type)	Output Shape
dense_49 (Dense)	(None, 64)
dense_50 (Dense)	(None, 32)
dense_51 (Dense)	(None, 1)

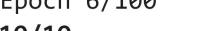
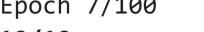
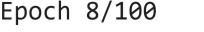
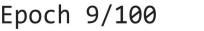
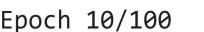
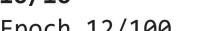
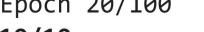
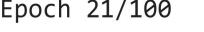
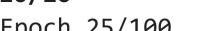
Total params: 2,369 (9.25 KB)

Trainable params: 2,369 (9.25 KB)

Non-trainable params: 0 (0.00 B)

In [200... model.compile(optimizer= 'Adam', loss = 'binary_crossentropy', metrics= ['accuracy']
as it is binary class problem so binary crossentropy

In [201... model.fit(X_train_scaled,y_train, epochs=100)

Epoch 1/100
10/10  1s 4ms/step - accuracy: 0.3589 - loss: 0.7104
Epoch 2/100
10/10  0s 4ms/step - accuracy: 0.6878 - loss: 0.6665
Epoch 3/100
10/10  0s 5ms/step - accuracy: 0.7573 - loss: 0.6285
Epoch 4/100
10/10  0s 4ms/step - accuracy: 0.7733 - loss: 0.6120
Epoch 5/100
10/10  0s 5ms/step - accuracy: 0.8248 - loss: 0.5642
Epoch 6/100
10/10  0s 4ms/step - accuracy: 0.7680 - loss: 0.5900
Epoch 7/100
10/10  0s 5ms/step - accuracy: 0.8033 - loss: 0.5318
Epoch 8/100
10/10  0s 4ms/step - accuracy: 0.7816 - loss: 0.5350
Epoch 9/100
10/10  0s 4ms/step - accuracy: 0.7847 - loss: 0.5321
Epoch 10/100
10/10  0s 4ms/step - accuracy: 0.7310 - loss: 0.5789
Epoch 11/100
10/10  0s 4ms/step - accuracy: 0.7745 - loss: 0.5444
Epoch 12/100
10/10  0s 4ms/step - accuracy: 0.7592 - loss: 0.5536
Epoch 13/100
10/10  0s 4ms/step - accuracy: 0.7712 - loss: 0.5339
Epoch 14/100
10/10  0s 4ms/step - accuracy: 0.7805 - loss: 0.5387
Epoch 15/100
10/10  0s 4ms/step - accuracy: 0.8114 - loss: 0.4951
Epoch 16/100
10/10  0s 4ms/step - accuracy: 0.8243 - loss: 0.4928
Epoch 17/100
10/10  0s 4ms/step - accuracy: 0.7801 - loss: 0.5220
Epoch 18/100
10/10  0s 5ms/step - accuracy: 0.7865 - loss: 0.5113
Epoch 19/100
10/10  0s 5ms/step - accuracy: 0.8084 - loss: 0.4997
Epoch 20/100
10/10  0s 4ms/step - accuracy: 0.7984 - loss: 0.5094
Epoch 21/100
10/10  0s 4ms/step - accuracy: 0.7916 - loss: 0.4996
Epoch 22/100
10/10  0s 4ms/step - accuracy: 0.8243 - loss: 0.4888
Epoch 23/100
10/10  0s 4ms/step - accuracy: 0.8268 - loss: 0.4761
Epoch 24/100
10/10  0s 4ms/step - accuracy: 0.7847 - loss: 0.5134
Epoch 25/100
10/10  0s 4ms/step - accuracy: 0.7688 - loss: 0.5385
Epoch 26/100
10/10  0s 5ms/step - accuracy: 0.8194 - loss: 0.4652
Epoch 27/100
10/10  0s 4ms/step - accuracy: 0.8013 - loss: 0.4894
Epoch 28/100
10/10  0s 5ms/step - accuracy: 0.8206 - loss: 0.4566

Epoch 29/100
10/10 0s 5ms/step - accuracy: 0.8064 - loss: 0.4909
Epoch 30/100
10/10 0s 4ms/step - accuracy: 0.7919 - loss: 0.4996
Epoch 31/100
10/10 0s 4ms/step - accuracy: 0.8076 - loss: 0.4740
Epoch 32/100
10/10 0s 4ms/step - accuracy: 0.7631 - loss: 0.5153
Epoch 33/100
10/10 0s 4ms/step - accuracy: 0.8046 - loss: 0.4745
Epoch 34/100
10/10 0s 4ms/step - accuracy: 0.7618 - loss: 0.5052
Epoch 35/100
10/10 0s 5ms/step - accuracy: 0.7932 - loss: 0.4770
Epoch 36/100
10/10 0s 4ms/step - accuracy: 0.7843 - loss: 0.4835
Epoch 37/100
10/10 0s 5ms/step - accuracy: 0.8114 - loss: 0.4600
Epoch 38/100
10/10 0s 4ms/step - accuracy: 0.7922 - loss: 0.4672
Epoch 39/100
10/10 0s 5ms/step - accuracy: 0.8042 - loss: 0.4545
Epoch 40/100
10/10 0s 4ms/step - accuracy: 0.7919 - loss: 0.4795
Epoch 41/100
10/10 0s 4ms/step - accuracy: 0.8169 - loss: 0.4513
Epoch 42/100
10/10 0s 4ms/step - accuracy: 0.8038 - loss: 0.4803
Epoch 43/100
10/10 0s 4ms/step - accuracy: 0.7770 - loss: 0.5119
Epoch 44/100
10/10 0s 4ms/step - accuracy: 0.7834 - loss: 0.5018
Epoch 45/100
10/10 0s 4ms/step - accuracy: 0.7581 - loss: 0.5306
Epoch 46/100
10/10 0s 4ms/step - accuracy: 0.7936 - loss: 0.4629
Epoch 47/100
10/10 0s 6ms/step - accuracy: 0.7723 - loss: 0.4936
Epoch 48/100
10/10 0s 5ms/step - accuracy: 0.7871 - loss: 0.5004
Epoch 49/100
10/10 0s 6ms/step - accuracy: 0.8308 - loss: 0.4477
Epoch 50/100
10/10 0s 4ms/step - accuracy: 0.8079 - loss: 0.4554
Epoch 51/100
10/10 0s 4ms/step - accuracy: 0.8084 - loss: 0.4500
Epoch 52/100
10/10 0s 4ms/step - accuracy: 0.7792 - loss: 0.5055
Epoch 53/100
10/10 0s 4ms/step - accuracy: 0.7943 - loss: 0.4653
Epoch 54/100
10/10 0s 4ms/step - accuracy: 0.7546 - loss: 0.5246
Epoch 55/100
10/10 0s 6ms/step - accuracy: 0.8023 - loss: 0.4474
Epoch 56/100
10/10 0s 4ms/step - accuracy: 0.7859 - loss: 0.4869

Epoch 57/100
10/10 0s 4ms/step - accuracy: 0.8129 - loss: 0.4633
Epoch 58/100
10/10 0s 4ms/step - accuracy: 0.7768 - loss: 0.5012
Epoch 59/100
10/10 0s 6ms/step - accuracy: 0.7812 - loss: 0.4886
Epoch 60/100
10/10 0s 6ms/step - accuracy: 0.8278 - loss: 0.4264
Epoch 61/100
10/10 0s 7ms/step - accuracy: 0.8194 - loss: 0.4223
Epoch 62/100
10/10 0s 7ms/step - accuracy: 0.8045 - loss: 0.4765
Epoch 63/100
10/10 0s 7ms/step - accuracy: 0.7947 - loss: 0.4696
Epoch 64/100
10/10 0s 7ms/step - accuracy: 0.8193 - loss: 0.4360
Epoch 65/100
10/10 0s 9ms/step - accuracy: 0.8181 - loss: 0.4353
Epoch 66/100
10/10 0s 7ms/step - accuracy: 0.8256 - loss: 0.4371
Epoch 67/100
10/10 0s 6ms/step - accuracy: 0.7849 - loss: 0.4839
Epoch 68/100
10/10 0s 6ms/step - accuracy: 0.7948 - loss: 0.4775
Epoch 69/100
10/10 0s 7ms/step - accuracy: 0.8170 - loss: 0.4404
Epoch 70/100
10/10 0s 7ms/step - accuracy: 0.7822 - loss: 0.4707
Epoch 71/100
10/10 0s 6ms/step - accuracy: 0.7882 - loss: 0.4700
Epoch 72/100
10/10 0s 10ms/step - accuracy: 0.7968 - loss: 0.4509
Epoch 73/100
10/10 0s 7ms/step - accuracy: 0.8171 - loss: 0.4684
Epoch 74/100
10/10 0s 7ms/step - accuracy: 0.8132 - loss: 0.4439
Epoch 75/100
10/10 0s 7ms/step - accuracy: 0.7882 - loss: 0.4612
Epoch 76/100
10/10 0s 5ms/step - accuracy: 0.8110 - loss: 0.4427
Epoch 77/100
10/10 0s 5ms/step - accuracy: 0.8026 - loss: 0.4725
Epoch 78/100
10/10 0s 4ms/step - accuracy: 0.7903 - loss: 0.4718
Epoch 79/100
10/10 0s 5ms/step - accuracy: 0.8192 - loss: 0.4657
Epoch 80/100
10/10 0s 7ms/step - accuracy: 0.8382 - loss: 0.4200
Epoch 81/100
10/10 0s 4ms/step - accuracy: 0.8036 - loss: 0.4522
Epoch 82/100
10/10 0s 5ms/step - accuracy: 0.8358 - loss: 0.4184
Epoch 83/100
10/10 0s 5ms/step - accuracy: 0.7970 - loss: 0.4858
Epoch 84/100
10/10 0s 4ms/step - accuracy: 0.8125 - loss: 0.4281

```
Epoch 85/100
10/10 0s 4ms/step - accuracy: 0.7913 - loss: 0.4728
Epoch 86/100
10/10 0s 4ms/step - accuracy: 0.7800 - loss: 0.4755
Epoch 87/100
10/10 0s 4ms/step - accuracy: 0.8228 - loss: 0.4144
Epoch 88/100
10/10 0s 4ms/step - accuracy: 0.8109 - loss: 0.4377
Epoch 89/100
10/10 0s 4ms/step - accuracy: 0.8170 - loss: 0.4270
Epoch 90/100
10/10 0s 4ms/step - accuracy: 0.7935 - loss: 0.4719
Epoch 91/100
10/10 0s 5ms/step - accuracy: 0.8156 - loss: 0.4428
Epoch 92/100
10/10 0s 5ms/step - accuracy: 0.7921 - loss: 0.4805
Epoch 93/100
10/10 0s 5ms/step - accuracy: 0.8065 - loss: 0.4668
Epoch 94/100
10/10 0s 4ms/step - accuracy: 0.7911 - loss: 0.4533
Epoch 95/100
10/10 0s 4ms/step - accuracy: 0.8300 - loss: 0.4067
Epoch 96/100
10/10 0s 5ms/step - accuracy: 0.8053 - loss: 0.4499
Epoch 97/100
10/10 0s 5ms/step - accuracy: 0.8042 - loss: 0.4370
Epoch 98/100
10/10 0s 4ms/step - accuracy: 0.7931 - loss: 0.4479
Epoch 99/100
10/10 0s 4ms/step - accuracy: 0.7704 - loss: 0.4860
Epoch 100/100
10/10 0s 4ms/step - accuracy: 0.8148 - loss: 0.4319
```

Out[201... <keras.src.callbacks.history.History at 0x7825b19775d0>

In [202... model.layers[0].get_weights()

```
Out[202]: [array([[ -8.79028291e-02,   1.44558370e-01,   2.19454840e-02,
       3.20629179e-01,  -2.01697648e-01,  -4.11203414e-01,
       2.58194864e-01,   3.53146285e-01,   3.67040694e-01,
       5.19062765e-02,   3.75066042e-01,  -3.21459144e-01,
      -1.07411623e-01,  -1.18773140e-01,   2.62567461e-01,
       1.85047671e-01,  -2.39267722e-01,  -4.05791141e-02,
      -1.18516885e-01,  -4.47206991e-03,   1.18256956e-01,
       1.26344830e-01,   3.70895505e-01,   2.29868099e-01,
       6.27746386e-03,  -3.23327065e-01,   2.72392929e-01,
       3.72209519e-01,  -3.29435945e-01,   3.74308646e-01,
       1.99009050e-02,  -1.01516128e-01,   2.37835765e-01,
       4.18299407e-01,  -1.78501546e-01,   2.92327195e-01,
      -3.91612612e-02,   3.24010789e-01,   5.69574796e-02,
       3.96583118e-02,   4.20180380e-01,  -2.47196004e-01,
      -3.93064976e-01,  -2.74976611e-01,   1.85940996e-01,
      -1.19817309e-01,  -2.65355468e-01,  -2.60752261e-01,
      -6.17357977e-02,  -2.42893890e-01,  -2.05764383e-01,
      -1.72435910e-01,   4.19462025e-01,   3.16435248e-01,
       1.91999063e-01,  -2.10728899e-01,   6.11107014e-02,
       2.63395876e-01,   1.97466269e-01,   1.93416938e-01,
      -7.56468475e-02,  -3.00329626e-01,   2.64350206e-01,
       4.06997442e-01],
      [-4.59472835e-03,   3.29130262e-01,   3.15319657e-01,
       -6.87275231e-02,   1.32650524e-01,  -2.98650891e-01,
       8.31134990e-02,  -1.39075533e-01,   6.30837828e-02,
      -1.73137933e-01,   5.07363072e-03,   2.42437974e-01,
       -6.75598383e-02,  -3.28739017e-01,  -4.76796590e-02,
      -3.61555040e-01,   1.36293158e-01,  -1.72817297e-02,
       4.71967757e-02,  -2.48383418e-01,  -3.97490747e-02,
       5.09087741e-03,  -3.42556655e-01,   2.60029227e-01,
       9.18991044e-02,   2.79845804e-01,   1.38415294e-02,
      -3.77173498e-02,  -8.09437316e-03,  -4.58214693e-02,
      -4.55373079e-02,  -6.73747957e-02,   6.93259463e-02,
      -1.01036213e-01,   1.81560025e-01,  -3.40371765e-02,
      -6.40938878e-02,  -3.36323977e-02,  -2.40026876e-01,
       3.46910618e-02,  -4.11662430e-01,   5.21624684e-02,
       3.48345309e-01,  -1.09899655e-01,   2.59646714e-01,
       3.06486189e-01,  -4.45007324e-01,   2.41597354e-01,
      -1.26387000e-01,   2.48536021e-01,   2.85082787e-01,
      -2.16825068e-01,  -1.20746948e-01,   1.05043083e-01,
      -3.89033973e-01,  -4.09829207e-02,   1.99490488e-01,
       3.22055757e-01,   1.19088762e-01,   1.15146481e-01,
      -2.45902121e-01,   7.73639828e-02,  -3.74828689e-02,
       3.17939632e-02],
      [ 5.53629361e-02,  -8.77003931e-03,   2.40789518e-01,
       -6.08171336e-02,  -5.00771739e-02,   2.45982632e-01,
       3.48551869e-01,  -2.19801664e-01,  -6.04033023e-02,
       3.57455432e-01,   7.37625957e-02,   2.68621687e-02,
       2.83632904e-01,  -3.20210695e-01,  -4.96599562e-02,
      -1.38185948e-01,  -2.94359028e-01,  -2.91173607e-01,
      -2.78608680e-01,   2.41496675e-02,  -6.98534846e-02,
      -3.21123630e-01,  -1.87724456e-01,   1.04583606e-01,
      -4.01538074e-01,  -3.09026897e-01,  -9.90110785e-02,
      -2.85433084e-02,   1.08801402e-01,  -6.77563176e-02,
      -1.74374714e-01,  -2.20134974e-01,  -1.97911456e-01,
       1.08277805e-01,   1.73635140e-01,   1.25607885e-02,
```

```

5.86689152e-02, 1.02288373e-01, -1.00457288e-01,
8.24018568e-02, -1.20046079e-01, 1.22428201e-01,
8.02332461e-02, -1.73114359e-01, 5.58320992e-02,
-3.33476186e-01, 1.59566533e-02, 4.70374748e-02,
-1.60437822e-01, 4.04613972e-01, -3.23191285e-01,
1.90984085e-01, -1.21192813e-01, -3.38599950e-01,
-1.50897965e-01, 3.23271245e-01, 1.86913088e-01,
-2.19256207e-01, 3.40222836e-01, 5.35994843e-02,
4.39872332e-02, -1.39540002e-01, 7.45069189e-03,
-3.51432333e-04]], dtype=float32),
array([ 0.12817426, 0.20521982, 0.14918701, -0.03597536, 0.09175844,
-0.05147361, 0.02190335, 0.00337734, -0.05305553, 0.03560524,
0.03404788, -0.04063964, -0.03319799, 0.03489536, -0.04311744,
0.07886904, -0.12584193, 0.17093647, 0.14751975, 0.17304407,
0.00727589, 0.09052448, 0.2191328 , -0.04587039, -0.03992491,
-0.12920305, 0.18638983, -0.08026256, -0.01061408, -0.05254402,
0.09612007, 0.01969136, 0.19294752, 0.04768126, 0.1424439 ,
0.03280235, 0.16171676, 0.04545964, 0.09123614, 0.13799311,
-0.02078512, -0.00854565, 0.01840978, -0.12238275, -0.09636377,
0.05823092, -0.0877924 , 0.04291899, 0.15706769, 0.03223462,
-0.02685755, -0.09769119, -0.01390599, 0.20203356, 0.07679092,
0.01970408, -0.04315974, 0.17346647, -0.0797968 , 0.21191488,
0.11159988, 0.14424142, -0.02150822, 0.06227436], dtype=float32)]

```

In [203...]: model.layers[1].get_weights()

```

Out[203...]: [array([[ -0.13211362, 0.14589728, 0.24322137, ... , -0.17423296,
-0.2240893 , 0.19596486],
[-0.08839979, 0.18350603, 0.2781759 , ... , 0.20231532,
-0.02336931, 0.18110175],
[-0.19904056, 0.3235288 , 0.17578326, ... , -0.05266555,
0.09134971, 0.10241529],
... ,
[-0.0086502 , 0.00857597, -0.1312894 , ... , -0.13003843,
0.0755511 , 0.22972158],
[ 0.14287758, -0.17466906, -0.3581583 , ... , 0.31288135,
-0.10591564, 0.17317042],
[ 0.29821435, -0.2934804 , -0.17796537, ... , 0.30437136,
0.12881903, 0.2711651 ]], dtype=float32),
array([ 0.04132227, 0.1495147 , 0.13114211, -0.02809754, 0.06112961,
0.07124241, -0.01728804, 0.01598972, 0.13247931, -0.01604365,
0.19784452, 0.11660361, -0.06163276, 0.04833479, -0.0184996 ,
-0.03277011, -0.03414286, -0.00168901, 0.10113084, 0.11275273,
0.11026782, -0.01490448, 0.12939027, 0.17196396, -0.06425285,
0.11785105, 0.10710909, -0.03852122, -0.05736998, -0.02098047,
-0.05876967, -0.05238512], dtype=float32)]

```

In [204...]: y_log = model.predict(X_test_scaled)
y_log[0:5]

3/3 ━━━━━━ 0s 28ms/step

```

Out[204...]: array([[0.4770495 ],
[0.4067244 ],
[0.47666952],
[0.66251916],
[0.45782974]], dtype=float32)

```

```
In [205...]: y_pred = np.where(y_log > 0.5, 1, 0)
y_pred[0:5]
```

```
Out[205...]: array([[0],
 [0],
 [0],
 [1],
 [0]])
```

```
In [206...]: from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred) # actual , predicted
```

```
Out[206...]: 0.75
```

```
In [207...]: history = model.fit(X_train_scaled, y_train, epochs = 100 , validation_split = 0.2)
```

Epoch 1/100
8/8 1s 73ms/step - accuracy: 0.8341 - loss: 0.3920 - val_accuracy: 0.7969 - val_loss: 0.4958
Epoch 2/100
8/8 0s 26ms/step - accuracy: 0.8105 - loss: 0.4127 - val_accuracy: 0.7969 - val_loss: 0.5105
Epoch 3/100
8/8 0s 26ms/step - accuracy: 0.8406 - loss: 0.4183 - val_accuracy: 0.7969 - val_loss: 0.5067
Epoch 4/100
8/8 0s 33ms/step - accuracy: 0.8210 - loss: 0.4170 - val_accuracy: 0.7969 - val_loss: 0.5071
Epoch 5/100
8/8 0s 32ms/step - accuracy: 0.8063 - loss: 0.4336 - val_accuracy: 0.7969 - val_loss: 0.5069
Epoch 6/100
8/8 0s 22ms/step - accuracy: 0.8057 - loss: 0.4309 - val_accuracy: 0.7969 - val_loss: 0.5153
Epoch 7/100
8/8 0s 40ms/step - accuracy: 0.8332 - loss: 0.4061 - val_accuracy: 0.7969 - val_loss: 0.5225
Epoch 8/100
8/8 0s 20ms/step - accuracy: 0.8038 - loss: 0.4495 - val_accuracy: 0.7969 - val_loss: 0.5230
Epoch 9/100
8/8 1s 52ms/step - accuracy: 0.8171 - loss: 0.4206 - val_accuracy: 0.7812 - val_loss: 0.5163
Epoch 10/100
8/8 0s 11ms/step - accuracy: 0.8185 - loss: 0.3951 - val_accuracy: 0.7969 - val_loss: 0.5277
Epoch 11/100
8/8 0s 11ms/step - accuracy: 0.7952 - loss: 0.4484 - val_accuracy: 0.7969 - val_loss: 0.5282
Epoch 12/100
8/8 0s 13ms/step - accuracy: 0.8053 - loss: 0.4348 - val_accuracy: 0.7969 - val_loss: 0.5256
Epoch 13/100
8/8 0s 11ms/step - accuracy: 0.8359 - loss: 0.3929 - val_accuracy: 0.7969 - val_loss: 0.5274
Epoch 14/100
8/8 0s 11ms/step - accuracy: 0.8267 - loss: 0.3844 - val_accuracy: 0.7812 - val_loss: 0.5268
Epoch 15/100
8/8 0s 11ms/step - accuracy: 0.8100 - loss: 0.4182 - val_accuracy: 0.7969 - val_loss: 0.5337
Epoch 16/100
8/8 0s 13ms/step - accuracy: 0.8436 - loss: 0.3867 - val_accuracy: 0.7969 - val_loss: 0.5336
Epoch 17/100
8/8 0s 12ms/step - accuracy: 0.8360 - loss: 0.4094 - val_accuracy: 0.7812 - val_loss: 0.5358
Epoch 18/100
8/8 0s 11ms/step - accuracy: 0.8350 - loss: 0.3839 - val_accuracy: 0.7812 - val_loss: 0.5336
Epoch 19/100
8/8 0s 11ms/step - accuracy: 0.8273 - loss: 0.4099 - val_accuracy:

```
cy: 0.7969 - val_loss: 0.5421
Epoch 20/100
8/8 0s 13ms/step - accuracy: 0.8184 - loss: 0.3992 - val_accuracy: 0.5431
cy: 0.7969 - val_loss: 0.5431
Epoch 21/100
8/8 0s 11ms/step - accuracy: 0.8435 - loss: 0.3792 - val_accuracy: 0.5401
cy: 0.7969 - val_loss: 0.5401
Epoch 22/100
8/8 0s 11ms/step - accuracy: 0.8083 - loss: 0.4494 - val_accuracy: 0.5344
cy: 0.7969 - val_loss: 0.5344
Epoch 23/100
8/8 0s 12ms/step - accuracy: 0.8460 - loss: 0.3854 - val_accuracy: 0.5361
cy: 0.7812 - val_loss: 0.5361
Epoch 24/100
8/8 0s 14ms/step - accuracy: 0.8128 - loss: 0.4237 - val_accuracy: 0.5426
cy: 0.7812 - val_loss: 0.5426
Epoch 25/100
8/8 0s 11ms/step - accuracy: 0.8319 - loss: 0.4018 - val_accuracy: 0.5473
cy: 0.7969 - val_loss: 0.5473
Epoch 26/100
8/8 0s 11ms/step - accuracy: 0.8301 - loss: 0.4009 - val_accuracy: 0.5477
cy: 0.7969 - val_loss: 0.5477
Epoch 27/100
8/8 0s 11ms/step - accuracy: 0.8417 - loss: 0.3719 - val_accuracy: 0.5497
cy: 0.7969 - val_loss: 0.5497
Epoch 28/100
8/8 0s 13ms/step - accuracy: 0.8296 - loss: 0.4053 - val_accuracy: 0.5510
cy: 0.7969 - val_loss: 0.5510
Epoch 29/100
8/8 0s 11ms/step - accuracy: 0.7999 - loss: 0.4302 - val_accuracy: 0.5484
cy: 0.7812 - val_loss: 0.5484
Epoch 30/100
8/8 0s 11ms/step - accuracy: 0.8069 - loss: 0.4168 - val_accuracy: 0.5433
cy: 0.7812 - val_loss: 0.5433
Epoch 31/100
8/8 0s 12ms/step - accuracy: 0.8180 - loss: 0.4004 - val_accuracy: 0.5543
cy: 0.7969 - val_loss: 0.5543
Epoch 32/100
8/8 0s 18ms/step - accuracy: 0.8138 - loss: 0.3928 - val_accuracy: 0.5505
cy: 0.7812 - val_loss: 0.5505
Epoch 33/100
8/8 0s 20ms/step - accuracy: 0.8064 - loss: 0.4097 - val_accuracy: 0.5572
cy: 0.7969 - val_loss: 0.5572
Epoch 34/100
8/8 0s 20ms/step - accuracy: 0.8248 - loss: 0.3868 - val_accuracy: 0.5518
cy: 0.7969 - val_loss: 0.5518
Epoch 35/100
8/8 0s 20ms/step - accuracy: 0.8208 - loss: 0.4088 - val_accuracy: 0.5539
cy: 0.7969 - val_loss: 0.5539
Epoch 36/100
8/8 0s 21ms/step - accuracy: 0.8348 - loss: 0.3860 - val_accuracy: 0.5562
cy: 0.7812 - val_loss: 0.5562
Epoch 37/100
8/8 0s 18ms/step - accuracy: 0.8033 - loss: 0.3879 - val_accuracy: 0.5558
cy: 0.7812 - val_loss: 0.5558
Epoch 38/100
```

```
8/8 _____ 0s 23ms/step - accuracy: 0.8227 - loss: 0.3935 - val_accuracy: 0.7969 - val_loss: 0.5607
Epoch 39/100
8/8 _____ 0s 17ms/step - accuracy: 0.8231 - loss: 0.3724 - val_accuracy: 0.7812 - val_loss: 0.5592
Epoch 40/100
8/8 _____ 0s 11ms/step - accuracy: 0.8170 - loss: 0.4227 - val_accuracy: 0.7969 - val_loss: 0.5596
Epoch 41/100
8/8 _____ 0s 11ms/step - accuracy: 0.8072 - loss: 0.3945 - val_accuracy: 0.7812 - val_loss: 0.5578
Epoch 42/100
8/8 _____ 0s 12ms/step - accuracy: 0.8248 - loss: 0.4035 - val_accuracy: 0.7969 - val_loss: 0.5698
Epoch 43/100
8/8 _____ 0s 11ms/step - accuracy: 0.8174 - loss: 0.3971 - val_accuracy: 0.7969 - val_loss: 0.5687
Epoch 44/100
8/8 _____ 0s 12ms/step - accuracy: 0.8139 - loss: 0.4210 - val_accuracy: 0.7969 - val_loss: 0.5581
Epoch 45/100
8/8 _____ 0s 11ms/step - accuracy: 0.8042 - loss: 0.4317 - val_accuracy: 0.7812 - val_loss: 0.5582
Epoch 46/100
8/8 _____ 0s 12ms/step - accuracy: 0.8291 - loss: 0.3787 - val_accuracy: 0.7812 - val_loss: 0.5727
Epoch 47/100
8/8 _____ 0s 12ms/step - accuracy: 0.8364 - loss: 0.3845 - val_accuracy: 0.7812 - val_loss: 0.5717
Epoch 48/100
8/8 _____ 0s 11ms/step - accuracy: 0.8017 - loss: 0.4242 - val_accuracy: 0.7969 - val_loss: 0.5701
Epoch 49/100
8/8 _____ 0s 11ms/step - accuracy: 0.8320 - loss: 0.4009 - val_accuracy: 0.7812 - val_loss: 0.5617
Epoch 50/100
8/8 _____ 0s 12ms/step - accuracy: 0.8219 - loss: 0.3960 - val_accuracy: 0.7969 - val_loss: 0.5768
Epoch 51/100
8/8 _____ 0s 12ms/step - accuracy: 0.8144 - loss: 0.4240 - val_accuracy: 0.7812 - val_loss: 0.5669
Epoch 52/100
8/8 _____ 0s 11ms/step - accuracy: 0.8116 - loss: 0.4043 - val_accuracy: 0.7812 - val_loss: 0.5743
Epoch 53/100
8/8 _____ 0s 17ms/step - accuracy: 0.8284 - loss: 0.3843 - val_accuracy: 0.7969 - val_loss: 0.5731
Epoch 54/100
8/8 _____ 0s 14ms/step - accuracy: 0.7787 - loss: 0.4302 - val_accuracy: 0.7812 - val_loss: 0.5657
Epoch 55/100
8/8 _____ 0s 11ms/step - accuracy: 0.8076 - loss: 0.4027 - val_accuracy: 0.7812 - val_loss: 0.5761
Epoch 56/100
8/8 _____ 0s 11ms/step - accuracy: 0.8127 - loss: 0.4004 - val_accuracy: 0.7812 - val_loss: 0.5784
```

Epoch 57/100
8/8 0s 11ms/step - accuracy: 0.8421 - loss: 0.3812 - val_accuracy: 0.7812 - val_loss: 0.5722
Epoch 58/100
8/8 0s 11ms/step - accuracy: 0.8420 - loss: 0.3743 - val_accuracy: 0.7812 - val_loss: 0.5748
Epoch 59/100
8/8 0s 17ms/step - accuracy: 0.8171 - loss: 0.3923 - val_accuracy: 0.7812 - val_loss: 0.5757
Epoch 60/100
8/8 0s 11ms/step - accuracy: 0.8364 - loss: 0.3882 - val_accuracy: 0.7812 - val_loss: 0.5794
Epoch 61/100
8/8 0s 17ms/step - accuracy: 0.8407 - loss: 0.3745 - val_accuracy: 0.7812 - val_loss: 0.5784
Epoch 62/100
8/8 0s 11ms/step - accuracy: 0.8401 - loss: 0.3665 - val_accuracy: 0.7812 - val_loss: 0.5829
Epoch 63/100
8/8 0s 11ms/step - accuracy: 0.8540 - loss: 0.3667 - val_accuracy: 0.7812 - val_loss: 0.5785
Epoch 64/100
8/8 0s 11ms/step - accuracy: 0.8224 - loss: 0.3783 - val_accuracy: 0.7812 - val_loss: 0.5879
Epoch 65/100
8/8 0s 11ms/step - accuracy: 0.8241 - loss: 0.3943 - val_accuracy: 0.7969 - val_loss: 0.5875
Epoch 66/100
8/8 0s 12ms/step - accuracy: 0.8428 - loss: 0.3645 - val_accuracy: 0.7812 - val_loss: 0.5769
Epoch 67/100
8/8 0s 17ms/step - accuracy: 0.8257 - loss: 0.4136 - val_accuracy: 0.7812 - val_loss: 0.5834
Epoch 68/100
8/8 0s 12ms/step - accuracy: 0.8356 - loss: 0.3688 - val_accuracy: 0.7969 - val_loss: 0.5899
Epoch 69/100
8/8 0s 18ms/step - accuracy: 0.8439 - loss: 0.3649 - val_accuracy: 0.7812 - val_loss: 0.5851
Epoch 70/100
8/8 0s 12ms/step - accuracy: 0.8360 - loss: 0.3595 - val_accuracy: 0.7812 - val_loss: 0.5857
Epoch 71/100
8/8 0s 21ms/step - accuracy: 0.8437 - loss: 0.3635 - val_accuracy: 0.7812 - val_loss: 0.5863
Epoch 72/100
8/8 0s 11ms/step - accuracy: 0.8223 - loss: 0.3789 - val_accuracy: 0.7812 - val_loss: 0.5935
Epoch 73/100
8/8 0s 12ms/step - accuracy: 0.8280 - loss: 0.3827 - val_accuracy: 0.7812 - val_loss: 0.5912
Epoch 74/100
8/8 0s 17ms/step - accuracy: 0.8022 - loss: 0.3976 - val_accuracy: 0.7812 - val_loss: 0.5859
Epoch 75/100
8/8 0s 12ms/step - accuracy: 0.8338 - loss: 0.3778 - val_accuracy:

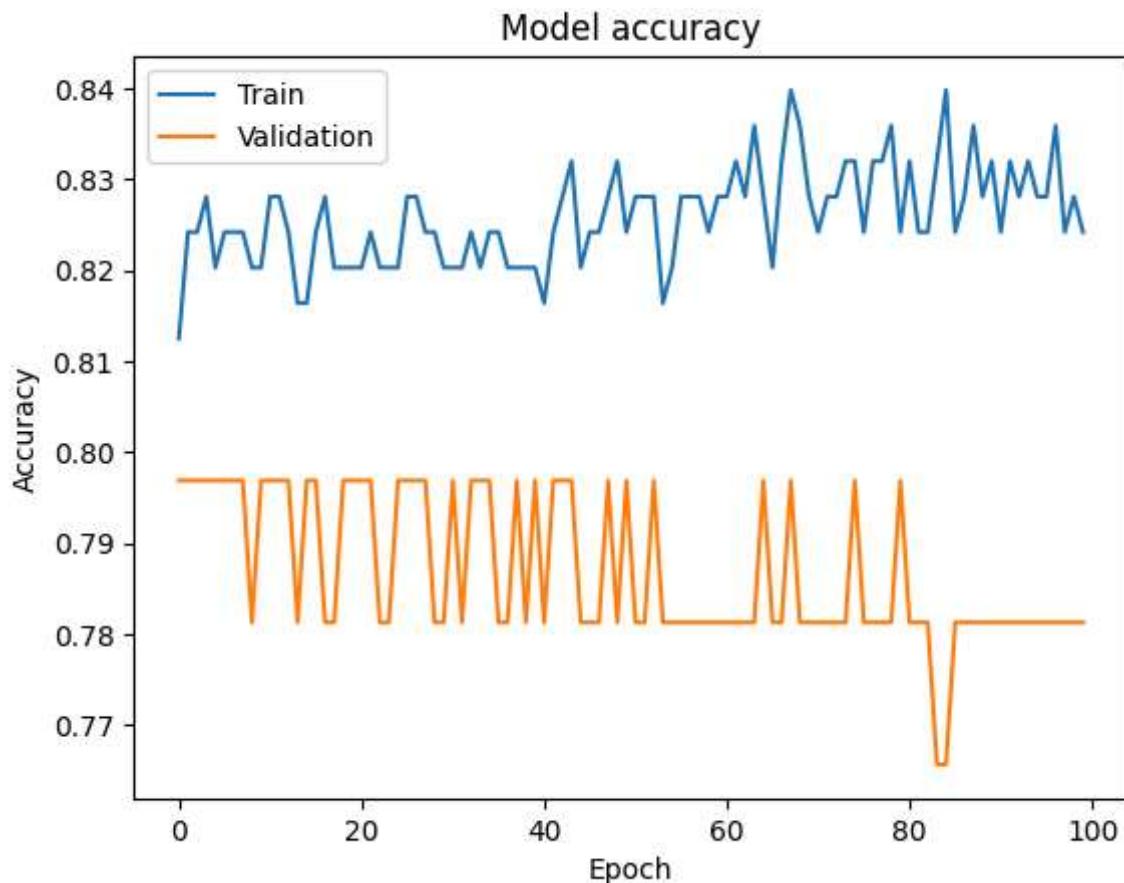
```
cy: 0.7969 - val_loss: 0.5903
Epoch 76/100
8/8 0s 14ms/step - accuracy: 0.8456 - loss: 0.3607 - val_accuracy: 0.7812 - val_loss: 0.5917
Epoch 77/100
8/8 0s 12ms/step - accuracy: 0.8324 - loss: 0.3964 - val_accuracy: 0.7812 - val_loss: 0.5940
Epoch 78/100
8/8 0s 12ms/step - accuracy: 0.8199 - loss: 0.3910 - val_accuracy: 0.7812 - val_loss: 0.6026
Epoch 79/100
8/8 0s 11ms/step - accuracy: 0.8500 - loss: 0.3640 - val_accuracy: 0.7812 - val_loss: 0.5916
Epoch 80/100
8/8 0s 11ms/step - accuracy: 0.8198 - loss: 0.3788 - val_accuracy: 0.7969 - val_loss: 0.6031
Epoch 81/100
8/8 0s 11ms/step - accuracy: 0.8173 - loss: 0.4179 - val_accuracy: 0.7812 - val_loss: 0.5954
Epoch 82/100
8/8 0s 12ms/step - accuracy: 0.8283 - loss: 0.3841 - val_accuracy: 0.7812 - val_loss: 0.5949
Epoch 83/100
8/8 0s 11ms/step - accuracy: 0.8444 - loss: 0.3464 - val_accuracy: 0.7812 - val_loss: 0.5958
Epoch 84/100
8/8 0s 13ms/step - accuracy: 0.8386 - loss: 0.3682 - val_accuracy: 0.7656 - val_loss: 0.6068
Epoch 85/100
8/8 0s 11ms/step - accuracy: 0.8075 - loss: 0.4107 - val_accuracy: 0.7656 - val_loss: 0.6040
Epoch 86/100
8/8 0s 12ms/step - accuracy: 0.8320 - loss: 0.3695 - val_accuracy: 0.7812 - val_loss: 0.5950
Epoch 87/100
8/8 0s 11ms/step - accuracy: 0.8234 - loss: 0.3710 - val_accuracy: 0.7812 - val_loss: 0.6051
Epoch 88/100
8/8 0s 12ms/step - accuracy: 0.8461 - loss: 0.3587 - val_accuracy: 0.7812 - val_loss: 0.6051
Epoch 89/100
8/8 0s 11ms/step - accuracy: 0.8150 - loss: 0.3828 - val_accuracy: 0.7812 - val_loss: 0.6066
Epoch 90/100
8/8 0s 19ms/step - accuracy: 0.8250 - loss: 0.3991 - val_accuracy: 0.7812 - val_loss: 0.6005
Epoch 91/100
8/8 0s 11ms/step - accuracy: 0.8262 - loss: 0.3630 - val_accuracy: 0.7812 - val_loss: 0.6108
Epoch 92/100
8/8 0s 12ms/step - accuracy: 0.8086 - loss: 0.3914 - val_accuracy: 0.7812 - val_loss: 0.6116
Epoch 93/100
8/8 0s 12ms/step - accuracy: 0.8256 - loss: 0.3708 - val_accuracy: 0.7812 - val_loss: 0.6082
Epoch 94/100
```

```
8/8 _____ 0s 11ms/step - accuracy: 0.7907 - loss: 0.4113 - val_accuracy: 0.7812 - val_loss: 0.6078
Epoch 95/100
8/8 _____ 0s 11ms/step - accuracy: 0.8407 - loss: 0.3626 - val_accuracy: 0.7812 - val_loss: 0.6128
Epoch 96/100
8/8 _____ 0s 17ms/step - accuracy: 0.8250 - loss: 0.3833 - val_accuracy: 0.7812 - val_loss: 0.6067
Epoch 97/100
8/8 _____ 0s 12ms/step - accuracy: 0.8214 - loss: 0.3947 - val_accuracy: 0.7812 - val_loss: 0.6091
Epoch 98/100
8/8 _____ 0s 13ms/step - accuracy: 0.8307 - loss: 0.3640 - val_accuracy: 0.7812 - val_loss: 0.6100
Epoch 99/100
8/8 _____ 0s 11ms/step - accuracy: 0.8296 - loss: 0.3661 - val_accuracy: 0.7812 - val_loss: 0.6185
Epoch 100/100
8/8 _____ 0s 11ms/step - accuracy: 0.8118 - loss: 0.3740 - val_accuracy: 0.7812 - val_loss: 0.6129
```

In [210...]

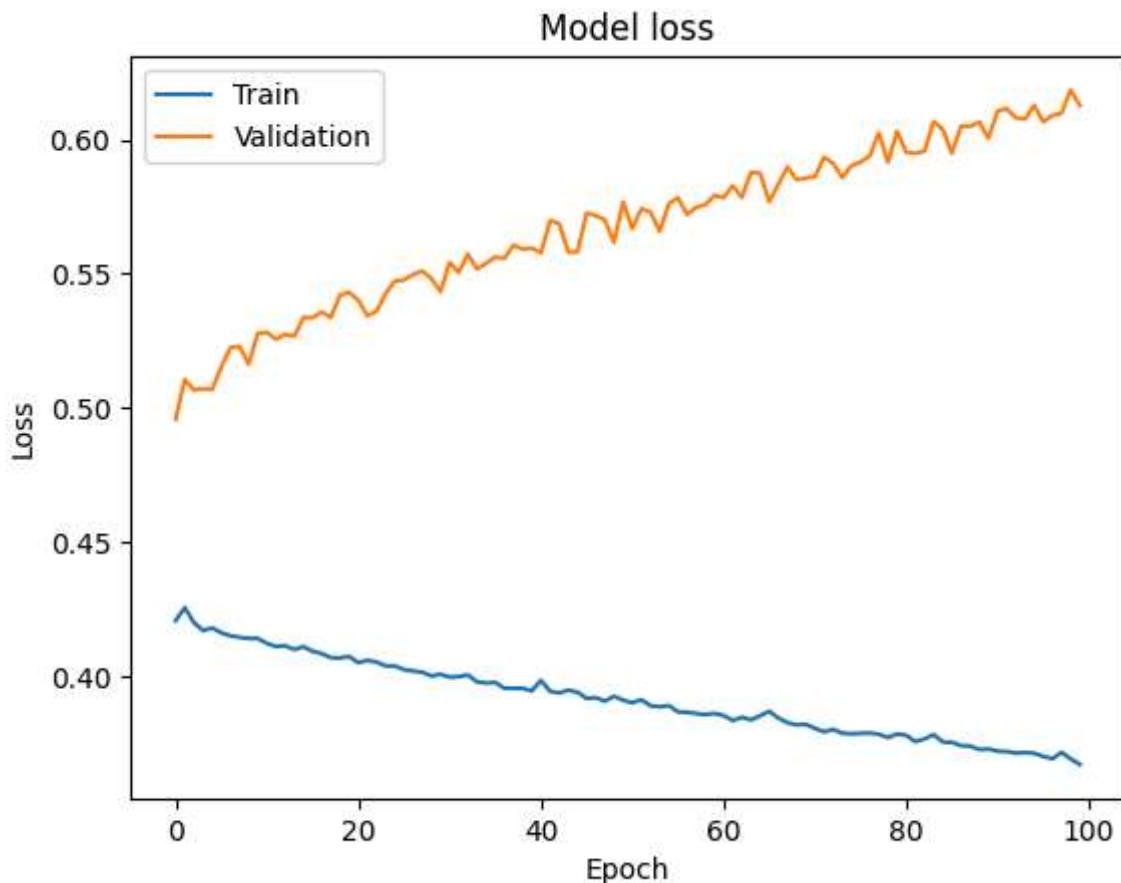
```
import matplotlib.pyplot as plt

# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```



```
In [209...]: plt.plot(history.history['loss'], label='Train')
plt.plot(history.history['val_loss'], label='Validation')
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend()
```

```
Out[209...]: <matplotlib.legend.Legend at 0x7825af6ec950>
```



```
In [211]: from sklearn.metrics import classification_report, confusion_matrix  
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.74	0.94	0.83	52
1	0.79	0.39	0.52	28
accuracy			0.75	80
macro avg	0.76	0.67	0.68	80
weighted avg	0.76	0.75	0.72	80

```
In [212]: cm = confusion_matrix(y_test, y_pred)  
cm
```

```
Out[212]: array([[49,  3],  
                  [17, 11]])
```