In [1]:

```
from google.colab import drive
drive.mount('/content/gdrive',force_remount=True)
```

```
Mounted at /content/gdrive
```

Pima-Indians-diabetes

In [ ]:

```
path = '/content/gdrive/My Drive/' + 'sampledatasets/pima-indians-diabetes.csv'
```

In [3]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
```

In [ ]:

```
raw_data = pd.read_csv(path)
raw_data
```

Out[ ]:

	6	148	72	35	0	33.6	0.627	50	1
0	1	85	66	29	0	26.6	0.351	31	0
1	8	183	64	0	0	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.167	21	0
3	0	137	40	35	168	43.1	2.288	33	1
4	5	116	74	0	0	25.6	0.201	30	0
...	...	...	...	...	...	...	...	...	...
762	10	101	76	48	180	32.9	0.171	63	0
763	2	122	70	27	0	36.8	0.340	27	0
764	5	121	72	23	112	26.2	0.245	30	0
765	1	126	60	0	0	30.1	0.349	47	1
766	1	93	70	31	0	30.4	0.315	23	0

767 rows × 9 columns

In [ ]:

```
X = raw_data.iloc[:,:-1].values
Y = raw_data.iloc[:,-1].values
X.shape, Y.shape
```

Out[ ]:

```
((767, 8), (767,))
```

In [ ]:

```
X_t, X_test, Y_t, Y_test = train_test_split(X, Y, test_size = 0.1)
X_t.shape, X_test.shape, Y_t.shape, Y_test.shape
```

Out[ ]:

```
((690, 8), (77, 8), (690,), (77,))
```

In [ ]:

```
X_train, X_valid, Y_train, Y_valid = train_test_split(X_t, Y_t, test_size = 0.1)
X_train.shape, X_valid.shape, Y_train.shape, Y_valid.shape
```

Out[ ]:

```
((621, 8), (69, 8), (621,), (69,))
```

In [ ]:

```
print('before : ', X_train[0])
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_valid = scaler.transform(X_valid)
X_test = scaler.transform(X_test)
print('after : ', X_train[0])
```

```
before :  [  2.     101.      58.      35.      90.      21.8      0.155   22.   ]
after :  [0.11764706 0.50753769 0.47540984 0.35353535 0.10638298 0.32488823
 0.03039384 0.01666667]
```

In [4]:

```
import tensorflow as tf
import matplotlib.pyplot as plt
```

In [ ]:

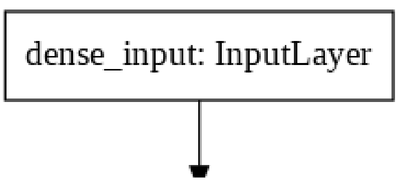
```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(12, input_dim = 8, activation = 'elu'))
model.add(tf.keras.layers.Dense(8,  activation = 'elu'))
model.add(tf.keras.layers.Dense(8, activation = 'elu'))
model.add(tf.keras.layers.Dense(8, activation = 'sigmoid'))
model.add(tf.keras.layers.Dense(1, activation = 'sigmoid'))
print(model.summary())

model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

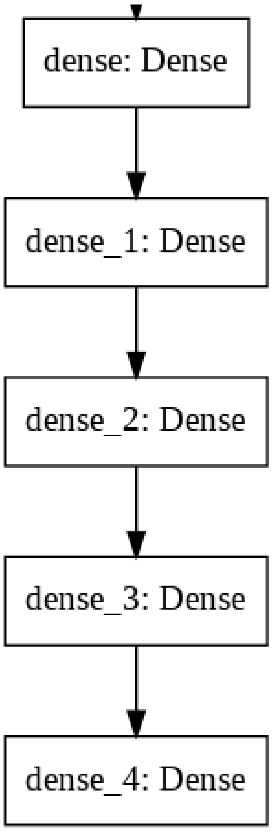
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense (Dense)                (None, 12)                108       
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 104       
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 72        
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 72        
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 9         
=================================================================
Total params: 365
Trainable params: 365
Non-trainable params: 0
_________________________________________________________________
None
```

In [ ]:

```
tf.keras.utils.plot_model(model)
```

Out[ ]:





In [ ]:

```
history = model.fit(X_train, Y_train, epochs = 100, verbose = 1, validation_data = (X_va
lid,Y_valid))
```

```
Epoch 1/100
20/20 [==============================] - 0s 11ms/step - loss: 0.7220 - accuracy: 0.3559 -
val_loss: 0.7012 - val_accuracy: 0.3913
Epoch 2/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6835 - accuracy: 0.6055 -
val_loss: 0.6836 - val_accuracy: 0.6087
Epoch 3/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6679 - accuracy: 0.6457 -
val_loss: 0.6774 - val_accuracy: 0.6087
Epoch 4/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6573 - accuracy: 0.6457 -
val_loss: 0.6757 - val_accuracy: 0.6087
Epoch 5/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6531 - accuracy: 0.6457 -
val_loss: 0.6748 - val_accuracy: 0.6087
Epoch 6/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6501 - accuracy: 0.6457 -
val_loss: 0.6735 - val_accuracy: 0.6087
Epoch 7/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6466 - accuracy: 0.6457 -
val_loss: 0.6702 - val_accuracy: 0.6087
Epoch 8/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6415 - accuracy: 0.6457 -
val_loss: 0.6644 - val_accuracy: 0.6087
Epoch 9/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6341 - accuracy: 0.6457 -
val_loss: 0.6577 - val_accuracy: 0.6087
Epoch 10/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6242 - accuracy: 0.6570 -
val_loss: 0.6460 - val_accuracy: 0.6087
Epoch 11/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6116 - accuracy: 0.6779 -
val_loss: 0.6358 - val_accuracy: 0.6232
Epoch 12/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5984 - accuracy: 0.7021 -
val_loss: 0.6268 - val_accuracy: 0.6667
Epoch 13/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5867 - accuracy: 0.6940 -
val_loss: 0.6158 - val_accuracy: 0.6667
```

```
Epoch 14/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5773 - accuracy: 0.7118 -
val_loss: 0.6097 - val_accuracy: 0.6522
Epoch 15/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5634 - accuracy: 0.7085 -
val_loss: 0.6015 - val_accuracy: 0.6667
Epoch 16/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5515 - accuracy: 0.7246 -
val_loss: 0.5937 - val_accuracy: 0.6812
Epoch 17/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5444 - accuracy: 0.7311 -
val_loss: 0.5980 - val_accuracy: 0.6812
Epoch 18/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5366 - accuracy: 0.7375 -
val_loss: 0.5859 - val_accuracy: 0.6812
Epoch 19/100
20/20 [==============================] - 0s 3ms/step - loss: 0.5274 - accuracy: 0.7343 -
val_loss: 0.5838 - val_accuracy: 0.6812
Epoch 20/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5204 - accuracy: 0.7424 -
val_loss: 0.5892 - val_accuracy: 0.7101
Epoch 21/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5162 - accuracy: 0.7440 -
val_loss: 0.5858 - val_accuracy: 0.7101
Epoch 22/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5104 - accuracy: 0.7488 -
val_loss: 0.5795 - val_accuracy: 0.7101
Epoch 23/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5033 - accuracy: 0.7601 -
val_loss: 0.5733 - val_accuracy: 0.7101
Epoch 24/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5018 - accuracy: 0.7536 -
val_loss: 0.5719 - val_accuracy: 0.7101
Epoch 25/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5021 - accuracy: 0.7552 -
val_loss: 0.5754 - val_accuracy: 0.7101
Epoch 26/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4965 - accuracy: 0.7617 -
val_loss: 0.5682 - val_accuracy: 0.7101
Epoch 27/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4913 - accuracy: 0.7649 -
val_loss: 0.5643 - val_accuracy: 0.6812
Epoch 28/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4868 - accuracy: 0.7617 -
val_loss: 0.5694 - val_accuracy: 0.7246
Epoch 29/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4840 - accuracy: 0.7665 -
val_loss: 0.5620 - val_accuracy: 0.7101
Epoch 30/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4822 - accuracy: 0.7649 -
val_loss: 0.5610 - val_accuracy: 0.7101
Epoch 31/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4797 - accuracy: 0.7697 -
val_loss: 0.5584 - val_accuracy: 0.6812
Epoch 32/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4795 - accuracy: 0.7649 -
val_loss: 0.5605 - val_accuracy: 0.7101
Epoch 33/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4758 - accuracy: 0.7681 -
val_loss: 0.5546 - val_accuracy: 0.6957
Epoch 34/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4750 - accuracy: 0.7601 -
val_loss: 0.5563 - val_accuracy: 0.7101
Epoch 35/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4782 - accuracy: 0.7633 -
val_loss: 0.5522 - val_accuracy: 0.6812
Epoch 36/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4714 - accuracy: 0.7681 -
val_loss: 0.5568 - val_accuracy: 0.7101
Epoch 37/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4732 - accuracy: 0.7729 -
val_loss: 0.5493 - val_accuracy: 0.6812
```

```
Epoch 38/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4789 - accuracy: 0.7617 -
val_loss: 0.5499 - val_accuracy: 0.6957
Epoch 39/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4741 - accuracy: 0.7665 -
val_loss: 0.5513 - val_accuracy: 0.6957
Epoch 40/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4698 - accuracy: 0.7681 -
val_loss: 0.5488 - val_accuracy: 0.6957
Epoch 41/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4713 - accuracy: 0.7665 -
val_loss: 0.5468 - val_accuracy: 0.6957
Epoch 42/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4685 - accuracy: 0.7713 -
val_loss: 0.5612 - val_accuracy: 0.7391
Epoch 43/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4697 - accuracy: 0.7649 -
val_loss: 0.5470 - val_accuracy: 0.6957
Epoch 44/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4696 - accuracy: 0.7729 -
val_loss: 0.5516 - val_accuracy: 0.7101
Epoch 45/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4691 - accuracy: 0.7762 -
val_loss: 0.5441 - val_accuracy: 0.6957
Epoch 46/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4766 - accuracy: 0.7762 -
val_loss: 0.5430 - val_accuracy: 0.6957
Epoch 47/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4705 - accuracy: 0.7713 -
val_loss: 0.5449 - val_accuracy: 0.6957
Epoch 48/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4675 - accuracy: 0.7649 -
val_loss: 0.5467 - val_accuracy: 0.6957
Epoch 49/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4724 - accuracy: 0.7729 -
val_loss: 0.5399 - val_accuracy: 0.6957
Epoch 50/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4691 - accuracy: 0.7762 -
val_loss: 0.5472 - val_accuracy: 0.6957
Epoch 51/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4697 - accuracy: 0.7762 -
val_loss: 0.5462 - val_accuracy: 0.7101
Epoch 52/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4661 - accuracy: 0.7746 -
val_loss: 0.5389 - val_accuracy: 0.6957
Epoch 53/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4655 - accuracy: 0.7681 -
val_loss: 0.5428 - val_accuracy: 0.6957
Epoch 54/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4685 - accuracy: 0.7762 -
val_loss: 0.5394 - val_accuracy: 0.7246
Epoch 55/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4690 - accuracy: 0.7762 -
val_loss: 0.5459 - val_accuracy: 0.6957
Epoch 56/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4665 - accuracy: 0.7681 -
val_loss: 0.5417 - val_accuracy: 0.6957
Epoch 57/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4711 - accuracy: 0.7762 -
val_loss: 0.5378 - val_accuracy: 0.7246
Epoch 58/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4769 - accuracy: 0.7729 -
val_loss: 0.5359 - val_accuracy: 0.7246
Epoch 59/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4710 - accuracy: 0.7810 -
val_loss: 0.5385 - val_accuracy: 0.6957
Epoch 60/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4650 - accuracy: 0.7713 -
val_loss: 0.5351 - val_accuracy: 0.7101
Epoch 61/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4646 - accuracy: 0.7665 -
val_loss: 0.5370 - val_accuracy: 0.6957
```

```
Epoch 62/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4634 - accuracy: 0.7729 -
val_loss: 0.5359 - val_accuracy: 0.6957
Epoch 63/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4626 - accuracy: 0.7713 -
val_loss: 0.5362 - val_accuracy: 0.6957
Epoch 64/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4643 - accuracy: 0.7729 -
val_loss: 0.5466 - val_accuracy: 0.7246
Epoch 65/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4701 - accuracy: 0.7778 -
val_loss: 0.5324 - val_accuracy: 0.7246
Epoch 66/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4670 - accuracy: 0.7874 -
val_loss: 0.5451 - val_accuracy: 0.7246
Epoch 67/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4648 - accuracy: 0.7762 -
val_loss: 0.5333 - val_accuracy: 0.6957
Epoch 68/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4639 - accuracy: 0.7697 -
val_loss: 0.5333 - val_accuracy: 0.6957
Epoch 69/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4637 - accuracy: 0.7729 -
val_loss: 0.5320 - val_accuracy: 0.6957
Epoch 70/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4676 - accuracy: 0.7762 -
val_loss: 0.5344 - val_accuracy: 0.6957
Epoch 71/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4643 - accuracy: 0.7746 -
val_loss: 0.5332 - val_accuracy: 0.6957
Epoch 72/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4624 - accuracy: 0.7713 -
val_loss: 0.5384 - val_accuracy: 0.6957
Epoch 73/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4641 - accuracy: 0.7762 -
val_loss: 0.5317 - val_accuracy: 0.7101
Epoch 74/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4645 - accuracy: 0.7794 -
val_loss: 0.5373 - val_accuracy: 0.6957
Epoch 75/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4633 - accuracy: 0.7762 -
val_loss: 0.5337 - val_accuracy: 0.7101
Epoch 76/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4610 - accuracy: 0.7746 -
val_loss: 0.5353 - val_accuracy: 0.6957
Epoch 77/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4621 - accuracy: 0.7746 -
val_loss: 0.5337 - val_accuracy: 0.6957
Epoch 78/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4621 - accuracy: 0.7729 -
val_loss: 0.5305 - val_accuracy: 0.7246
Epoch 79/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4645 - accuracy: 0.7810 -
val_loss: 0.5398 - val_accuracy: 0.7101
Epoch 80/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4632 - accuracy: 0.7826 -
val_loss: 0.5284 - val_accuracy: 0.7246
Epoch 81/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4675 - accuracy: 0.7794 -
val_loss: 0.5423 - val_accuracy: 0.7246
Epoch 82/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4663 - accuracy: 0.7713 -
val_loss: 0.5271 - val_accuracy: 0.7246
Epoch 83/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4632 - accuracy: 0.7633 -
val_loss: 0.5314 - val_accuracy: 0.7101
Epoch 84/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4610 - accuracy: 0.7729 -
val_loss: 0.5294 - val_accuracy: 0.7246
Epoch 85/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4606 - accuracy: 0.7778 -
val_loss: 0.5336 - val_accuracy: 0.6957
```

```
Epoch 86/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4667 - accuracy: 0.7778 -
val_loss: 0.5272 - val_accuracy: 0.7246
Epoch 87/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4658 - accuracy: 0.7810 -
val_loss: 0.5349 - val_accuracy: 0.6957
Epoch 88/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4607 - accuracy: 0.7762 -
val_loss: 0.5336 - val_accuracy: 0.6957
Epoch 89/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4609 - accuracy: 0.7681 -
val_loss: 0.5299 - val_accuracy: 0.7246
Epoch 90/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4615 - accuracy: 0.7778 -
val_loss: 0.5311 - val_accuracy: 0.7101
Epoch 91/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4600 - accuracy: 0.7713 -
val_loss: 0.5301 - val_accuracy: 0.7246
Epoch 92/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4596 - accuracy: 0.7729 -
val_loss: 0.5295 - val_accuracy: 0.7246
Epoch 93/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4591 - accuracy: 0.7762 -
val_loss: 0.5324 - val_accuracy: 0.6957
Epoch 94/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4598 - accuracy: 0.7810 -
val_loss: 0.5273 - val_accuracy: 0.7246
Epoch 95/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4623 - accuracy: 0.7713 -
val_loss: 0.5401 - val_accuracy: 0.7101
Epoch 96/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4597 - accuracy: 0.7826 -
val_loss: 0.5253 - val_accuracy: 0.7246
Epoch 97/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4621 - accuracy: 0.7713 -
val_loss: 0.5329 - val_accuracy: 0.7101
Epoch 98/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4611 - accuracy: 0.7729 -
val_loss: 0.5293 - val_accuracy: 0.7101
Epoch 99/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4595 - accuracy: 0.7729 -
val_loss: 0.5370 - val_accuracy: 0.7101
Epoch 100/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4629 - accuracy: 0.7697 -
val_loss: 0.5293 - val_accuracy: 0.7246
```

In [ ]:

```
history
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7fa68f1af128>
```

In [ ]:

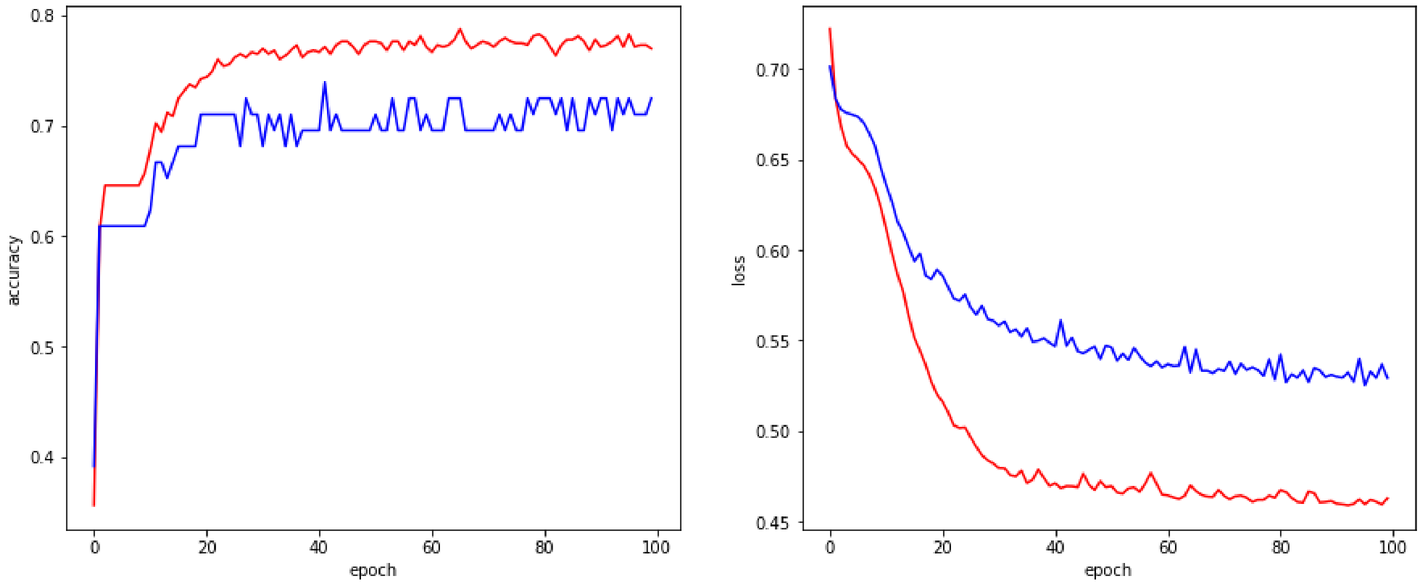
```
history.history  # hash table
```

In [ ]:

```
plt.rcParams['figure.figsize'] = [15, 6]

plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], 'r', history.history['val_accuracy'], 'b')
plt.xlabel('epoch')
plt.ylabel('accuracy')

plt.subplot(1,2,2)
plt.plot(history.history['loss'], 'r', history.history['val_loss'], 'b')
plt.xlabel('epoch')
plt.ylabel('loss')
```

Out[ ]:

```
Text(0, 0.5, 'loss')
```



In [ ]:

```
model.evaluate(X_test,Y_test)
```

```
3/3 [==============================] - 0s 2ms/step - loss: 0.9825 - accuracy: 0.7403
```

Out[ ]:

```
[0.9824575185775757, 0.7402597665786743]
```

different way

In [ ]:

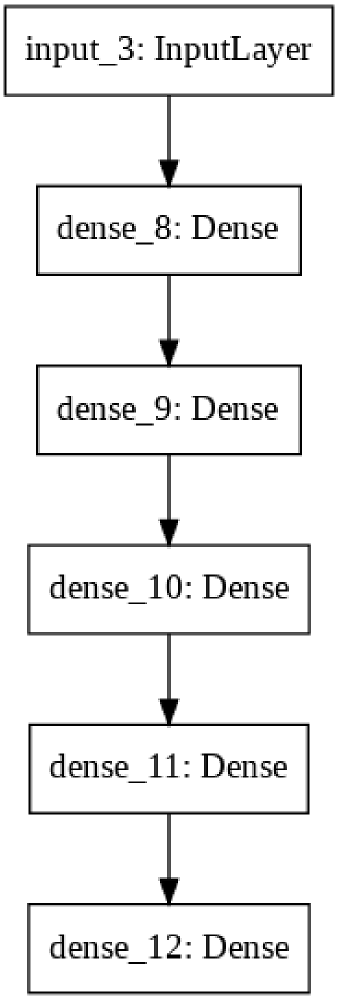
```
model = tf.keras.Sequential()
inp = tf.keras.layers.Input(shape = (8,))
mid = tf.keras.layers.Dense(12, input_dim = 8, activation = 'elu')(inp)
mid = tf.keras.layers.Dense(8,  activation = 'elu')(mid)
mid = tf.keras.layers.Dense(8, activation = 'elu')(mid)
mid = tf.keras.layers.Dense(8, activation = 'sigmoid')(mid)
out = tf.keras.layers.Dense(1, activation = 'sigmoid')(mid)

model = tf.keras.Model(inputs = inp, outputs = out)
print(model.summary())

model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_3 (InputLayer)         [(None, 8)]               0         
_________________________________________________________________
dense_8 (Dense)              (None, 12)                108       
_________________________________________________________________
dense_9 (Dense)              (None, 8)                 104       
_________________________________________________________________
dense_10 (Dense)             (None, 8)                 72        
_________________________________________________________________
dense_11 (Dense)             (None, 8)                 72        
_________________________________________________________________
dense_12 (Dense)             (None, 1)                 9         
=================================================================
Total params: 365
Trainable params: 365
Non-trainable params: 0
_________________________________________________________________
None
```

In [ ]:

```
tf.keras.utils.plot_model(model)
```

Out[ ]:



In [ ]:

```
history = model.fit(X_train, Y_train, epochs = 100, verbose = 1, validation_data = (X_va
lid,Y_valid))

plt.rcParams['figure.figsize'] = [15, 6]

plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], 'r', history.history['val_accuracy'], 'b')
plt.xlabel('epoch')
plt.ylabel('accuracy')

plt.subplot(1,2,2)
plt.plot(history.history['loss'], 'r', history.history['val_loss'], 'b')
plt.xlabel('epoch')
plt.ylabel('loss')

model.evaluate(X_test,Y_test)
```

```
Epoch 1/100
20/20 [==============================] - 0s 8ms/step - loss: 0.7796 - accuracy: 0.3543 -
val_loss: 0.7343 - val_accuracy: 0.3913
Epoch 2/100
20/20 [==============================] - 0s 2ms/step - loss: 0.7209 - accuracy: 0.3543 -
val_loss: 0.6974 - val_accuracy: 0.4203
Epoch 3/100
20/20 [==============================] - 0s 3ms/step - loss: 0.6821 - accuracy: 0.6248 -
val_loss: 0.6783 - val_accuracy: 0.6087
Epoch 4/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6623 - accuracy: 0.6457 -
val_loss: 0.6732 - val_accuracy: 0.6087
Epoch 5/100
```

```
20/20 [==============================] - 0s 2ms/step - loss: 0.6548 - accuracy: 0.6457 -
val_loss: 0.6729 - val_accuracy: 0.6087
Epoch 6/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6517 - accuracy: 0.6457 -
val_loss: 0.6728 - val_accuracy: 0.6087
Epoch 7/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6511 - accuracy: 0.6457 -
val_loss: 0.6690 - val_accuracy: 0.6087
Epoch 8/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6456 - accuracy: 0.6457 -
val_loss: 0.6694 - val_accuracy: 0.6087
Epoch 9/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6418 - accuracy: 0.6457 -
val_loss: 0.6660 - val_accuracy: 0.6087
Epoch 10/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6363 - accuracy: 0.6457 -
val_loss: 0.6603 - val_accuracy: 0.6087
Epoch 11/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6304 - accuracy: 0.6457 -
val_loss: 0.6573 - val_accuracy: 0.6087
Epoch 12/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6243 - accuracy: 0.6602 -
val_loss: 0.6494 - val_accuracy: 0.6087
Epoch 13/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6151 - accuracy: 0.6747 -
val_loss: 0.6450 - val_accuracy: 0.5942
Epoch 14/100
20/20 [==============================] - 0s 2ms/step - loss: 0.6065 - accuracy: 0.6908 -
val_loss: 0.6383 - val_accuracy: 0.6377
Epoch 15/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5971 - accuracy: 0.7037 -
val_loss: 0.6321 - val_accuracy: 0.6232
Epoch 16/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5881 - accuracy: 0.6860 -
val_loss: 0.6246 - val_accuracy: 0.6377
Epoch 17/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5799 - accuracy: 0.7005 -
val_loss: 0.6226 - val_accuracy: 0.6522
Epoch 18/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5725 - accuracy: 0.7005 -
val_loss: 0.6160 - val_accuracy: 0.6667
Epoch 19/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5654 - accuracy: 0.7037 -
val_loss: 0.6102 - val_accuracy: 0.6667
Epoch 20/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5581 - accuracy: 0.7134 -
val_loss: 0.6089 - val_accuracy: 0.6667
Epoch 21/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5516 - accuracy: 0.7198 -
val_loss: 0.6024 - val_accuracy: 0.6812
Epoch 22/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5474 - accuracy: 0.7166 -
val_loss: 0.6012 - val_accuracy: 0.6812
Epoch 23/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5406 - accuracy: 0.7279 -
val_loss: 0.5968 - val_accuracy: 0.6667
Epoch 24/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5353 - accuracy: 0.7343 -
val_loss: 0.5946 - val_accuracy: 0.6667
Epoch 25/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5315 - accuracy: 0.7359 -
val_loss: 0.5915 - val_accuracy: 0.6522
Epoch 26/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5279 - accuracy: 0.7327 -
val_loss: 0.5892 - val_accuracy: 0.6522
Epoch 27/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5219 - accuracy: 0.7440 -
val_loss: 0.5866 - val_accuracy: 0.6522
Epoch 28/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5202 - accuracy: 0.7440 -
val_loss: 0.5827 - val_accuracy: 0.6522
Epoch 29/100
```

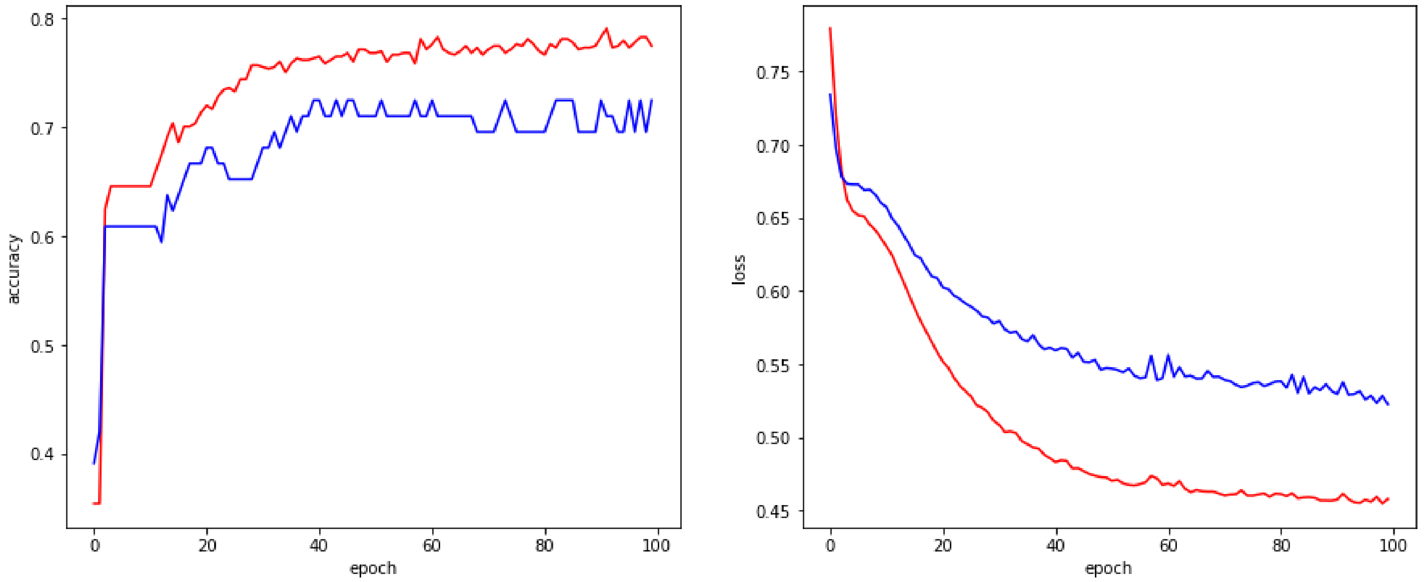
```
20/20 [==============================] - 0s 2ms/step - loss: 0.5169 - accuracy: 0.7568 -
val_loss: 0.5819 - val_accuracy: 0.6522
Epoch 30/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5113 - accuracy: 0.7568 -
val_loss: 0.5778 - val_accuracy: 0.6667
Epoch 31/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5085 - accuracy: 0.7552 -
val_loss: 0.5794 - val_accuracy: 0.6812
Epoch 32/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5035 - accuracy: 0.7536 -
val_loss: 0.5737 - val_accuracy: 0.6812
Epoch 33/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5041 - accuracy: 0.7552 -
val_loss: 0.5714 - val_accuracy: 0.6957
Epoch 34/100
20/20 [==============================] - 0s 2ms/step - loss: 0.5028 - accuracy: 0.7601 -
val_loss: 0.5724 - val_accuracy: 0.6812
Epoch 35/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4974 - accuracy: 0.7504 -
val_loss: 0.5672 - val_accuracy: 0.6957
Epoch 36/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4953 - accuracy: 0.7585 -
val_loss: 0.5656 - val_accuracy: 0.7101
Epoch 37/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4931 - accuracy: 0.7633 -
val_loss: 0.5697 - val_accuracy: 0.6957
Epoch 38/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4923 - accuracy: 0.7617 -
val_loss: 0.5636 - val_accuracy: 0.7101
Epoch 39/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4881 - accuracy: 0.7617 -
val_loss: 0.5601 - val_accuracy: 0.7101
Epoch 40/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4856 - accuracy: 0.7633 -
val_loss: 0.5613 - val_accuracy: 0.7246
Epoch 41/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4830 - accuracy: 0.7649 -
val_loss: 0.5593 - val_accuracy: 0.7246
Epoch 42/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4843 - accuracy: 0.7585 -
val_loss: 0.5610 - val_accuracy: 0.7101
Epoch 43/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4838 - accuracy: 0.7617 -
val_loss: 0.5604 - val_accuracy: 0.7101
Epoch 44/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4787 - accuracy: 0.7649 -
val_loss: 0.5545 - val_accuracy: 0.7246
Epoch 45/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4788 - accuracy: 0.7649 -
val_loss: 0.5576 - val_accuracy: 0.7101
Epoch 46/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4769 - accuracy: 0.7681 -
val_loss: 0.5516 - val_accuracy: 0.7246
Epoch 47/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4748 - accuracy: 0.7601 -
val_loss: 0.5513 - val_accuracy: 0.7246
Epoch 48/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4737 - accuracy: 0.7713 -
val_loss: 0.5530 - val_accuracy: 0.7101
Epoch 49/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4727 - accuracy: 0.7713 -
val_loss: 0.5461 - val_accuracy: 0.7101
Epoch 50/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4724 - accuracy: 0.7681 -
val_loss: 0.5473 - val_accuracy: 0.7101
Epoch 51/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4702 - accuracy: 0.7681 -
val_loss: 0.5469 - val_accuracy: 0.7101
Epoch 52/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4709 - accuracy: 0.7697 -
val_loss: 0.5460 - val_accuracy: 0.7246
Epoch 53/100
```

```
20/20 [==============================] - 0s 2ms/step - loss: 0.4685 - accuracy: 0.7601 -
val_loss: 0.5444 - val_accuracy: 0.7101
Epoch 54/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4675 - accuracy: 0.7665 -
val_loss: 0.5470 - val_accuracy: 0.7101
Epoch 55/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4671 - accuracy: 0.7665 -
val_loss: 0.5421 - val_accuracy: 0.7101
Epoch 56/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4680 - accuracy: 0.7681 -
val_loss: 0.5403 - val_accuracy: 0.7101
Epoch 57/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4693 - accuracy: 0.7681 -
val_loss: 0.5410 - val_accuracy: 0.7101
Epoch 58/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4735 - accuracy: 0.7585 -
val_loss: 0.5558 - val_accuracy: 0.7246
Epoch 59/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4718 - accuracy: 0.7810 -
val_loss: 0.5391 - val_accuracy: 0.7101
Epoch 60/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4674 - accuracy: 0.7713 -
val_loss: 0.5403 - val_accuracy: 0.7101
Epoch 61/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4685 - accuracy: 0.7762 -
val_loss: 0.5563 - val_accuracy: 0.7246
Epoch 62/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4668 - accuracy: 0.7826 -
val_loss: 0.5413 - val_accuracy: 0.7101
Epoch 63/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4699 - accuracy: 0.7713 -
val_loss: 0.5477 - val_accuracy: 0.7101
Epoch 64/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4647 - accuracy: 0.7681 -
val_loss: 0.5415 - val_accuracy: 0.7101
Epoch 65/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4624 - accuracy: 0.7665 -
val_loss: 0.5421 - val_accuracy: 0.7101
Epoch 66/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4640 - accuracy: 0.7697 -
val_loss: 0.5400 - val_accuracy: 0.7101
Epoch 67/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4631 - accuracy: 0.7746 -
val_loss: 0.5402 - val_accuracy: 0.7101
Epoch 68/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4628 - accuracy: 0.7681 -
val_loss: 0.5453 - val_accuracy: 0.7101
Epoch 69/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4629 - accuracy: 0.7729 -
val_loss: 0.5413 - val_accuracy: 0.6957
Epoch 70/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4616 - accuracy: 0.7665 -
val_loss: 0.5414 - val_accuracy: 0.6957
Epoch 71/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4602 - accuracy: 0.7713 -
val_loss: 0.5392 - val_accuracy: 0.6957
Epoch 72/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4608 - accuracy: 0.7746 -
val_loss: 0.5383 - val_accuracy: 0.6957
Epoch 73/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4611 - accuracy: 0.7746 -
val_loss: 0.5358 - val_accuracy: 0.7101
Epoch 74/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4637 - accuracy: 0.7681 -
val_loss: 0.5342 - val_accuracy: 0.7246
Epoch 75/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4602 - accuracy: 0.7713 -
val_loss: 0.5352 - val_accuracy: 0.7101
Epoch 76/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4601 - accuracy: 0.7762 -
val_loss: 0.5370 - val_accuracy: 0.6957
Epoch 77/100
```

```
20/20 [==============================] - 0s 2ms/step - loss: 0.4609 - accuracy: 0.7746 -
val_loss: 0.5376 - val_accuracy: 0.6957
Epoch 78/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4617 - accuracy: 0.7810 -
val_loss: 0.5349 - val_accuracy: 0.6957
Epoch 79/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4592 - accuracy: 0.7762 -
val_loss: 0.5362 - val_accuracy: 0.6957
Epoch 80/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4615 - accuracy: 0.7697 -
val_loss: 0.5380 - val_accuracy: 0.6957
Epoch 81/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4612 - accuracy: 0.7665 -
val_loss: 0.5383 - val_accuracy: 0.6957
Epoch 82/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4598 - accuracy: 0.7762 -
val_loss: 0.5339 - val_accuracy: 0.7101
Epoch 83/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4617 - accuracy: 0.7729 -
val_loss: 0.5427 - val_accuracy: 0.7246
Epoch 84/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4582 - accuracy: 0.7810 -
val_loss: 0.5303 - val_accuracy: 0.7246
Epoch 85/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4588 - accuracy: 0.7810 -
val_loss: 0.5413 - val_accuracy: 0.7246
Epoch 86/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4590 - accuracy: 0.7778 -
val_loss: 0.5299 - val_accuracy: 0.7246
Epoch 87/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4586 - accuracy: 0.7713 -
val_loss: 0.5341 - val_accuracy: 0.6957
Epoch 88/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4568 - accuracy: 0.7729 -
val_loss: 0.5321 - val_accuracy: 0.6957
Epoch 89/100
20/20 [==============================] - 0s 3ms/step - loss: 0.4568 - accuracy: 0.7729 -
val_loss: 0.5363 - val_accuracy: 0.6957
Epoch 90/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4567 - accuracy: 0.7746 -
val_loss: 0.5315 - val_accuracy: 0.6957
Epoch 91/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4573 - accuracy: 0.7826 -
val_loss: 0.5295 - val_accuracy: 0.7246
Epoch 92/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4614 - accuracy: 0.7907 -
val_loss: 0.5376 - val_accuracy: 0.7101
Epoch 93/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4577 - accuracy: 0.7729 -
val_loss: 0.5291 - val_accuracy: 0.7101
Epoch 94/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4555 - accuracy: 0.7746 -
val_loss: 0.5295 - val_accuracy: 0.6957
Epoch 95/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4551 - accuracy: 0.7794 -
val_loss: 0.5314 - val_accuracy: 0.6957
Epoch 96/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4573 - accuracy: 0.7729 -
val_loss: 0.5259 - val_accuracy: 0.7246
Epoch 97/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4559 - accuracy: 0.7778 -
val_loss: 0.5284 - val_accuracy: 0.6957
Epoch 98/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4593 - accuracy: 0.7826 -
val_loss: 0.5233 - val_accuracy: 0.7246
Epoch 99/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4547 - accuracy: 0.7826 -
val_loss: 0.5284 - val_accuracy: 0.6957
Epoch 100/100
20/20 [==============================] - 0s 2ms/step - loss: 0.4578 - accuracy: 0.7746 -
val_loss: 0.5225 - val_accuracy: 0.7246
3/3 [==============================] - 0s 2ms/step - loss: 0.9351 - accuracy: 0.7403
```

Out[ ]:

```
[0.9350751638412476, 0.7402597665786743]
```



Credit card fraud

In [5]:

```
path = '/content/gdrive/My Drive/' + 'sampledatasets/creditcard.csv'
raw_data = pd.read_csv(path)
raw_data
```

Out[5]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	0.55
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.61
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.62
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	0.22
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	0.82
...	...	...	...	...	...	...	...	...	...	...	...	
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	4.356170	1.59
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	-0.975926	0.15
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	-0.484782	0.41
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	-0.399126	1.93
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	-0.915427	1.04

284807 rows × 31 columns

In [6]:

```
ratio = 0.8
```

```
point_to = int(len(raw_data) * ratio)
```

In [7]:

```
X_train_df = raw_data.iloc[:point_to,1:-1]
X_test_df = raw_data.iloc[point_to:,1:-1]
Y_train_df = raw_data.iloc[:point_to,-1]
Y_test_df = raw_data.iloc[point_to:,-1]

X_train_df
```

Out[7]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V
0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	0.6178
1	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.0652
2	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.0660
3	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.1782
4	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.5381
...	...	...	...	...	...	...	...	...	...	...	...	
227840	2.028950	-0.374089	-1.268051	0.349127	-0.057977	-0.515489	-0.087045	-0.146316	1.001341	0.007773	-1.307879	0.0032
227841	0.306600	1.116021	-0.047348	3.593785	2.079047	5.748707	-2.059246	-1.352120	-1.849240	1.000154	-0.330464	0.2023
227842	-1.781954	2.062680	-3.758871	-1.801001	-0.084365	1.919610	-1.454364	-9.825473	-2.073119	-0.005816	-0.644403	0.9820
227843	-0.061507	1.024900	-0.170060	-0.263220	0.982164	-1.162749	1.468942	-0.648407	-0.157586	-0.764370	-0.442168	0.0242
227844	2.050034	-0.103557	-1.204713	0.207198	0.108850	-0.665621	0.073598	-0.164464	0.303531	0.242471	0.614559	1.0168

227845 rows × 29 columns

In [8]:

```
X_train = X_train_df.values
X_test = X_test_df.values
Y_train = Y_train_df.values
Y_test = Y_test_df.values

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

Out[8]:

```
((227845, 29), (56962, 29), (227845,), (56962,))
```

In [9]:

```
print(X_train[0])
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print(X_train[0])
```

```
[-1.35980713e+00 -7.27811733e-02  2.53634674e+00  1.37815522e+00
 -3.38320770e-01  4.62387778e-01  2.39598554e-01  9.86979013e-02
  3.63786970e-01  9.07941720e-02 -5.51599533e-01 -6.17800856e-01
 -9.91389847e-01 -3.11169354e-01  1.46817697e+00 -4.70400525e-01
```

```
  2.07971242e-01  2.57905802e-02  4.03992960e-01  2.51412098e-01
 -1.83067779e-02  2.77837576e-01 -1.10473910e-01  6.69280749e-02
  1.28539358e-01 -1.89114844e-01  1.33558377e-01 -2.10530535e-02
  1.49620000e+02]
[0.93519234 0.76649042 0.8410207  0.31302266 0.54333742 0.5467858
 0.54450243 0.7864442  0.47531173 0.51060048 0.25248432 0.68090763
 0.46332808 0.63559053 0.5779519  0.60926003 0.73717255 0.65506586
 0.59486323 0.41911347 0.56118439 0.52299212 0.70046681 0.42329004
 0.58512179 0.39455679 0.65381604 0.25658857 0.00761172]
```

In [10]:

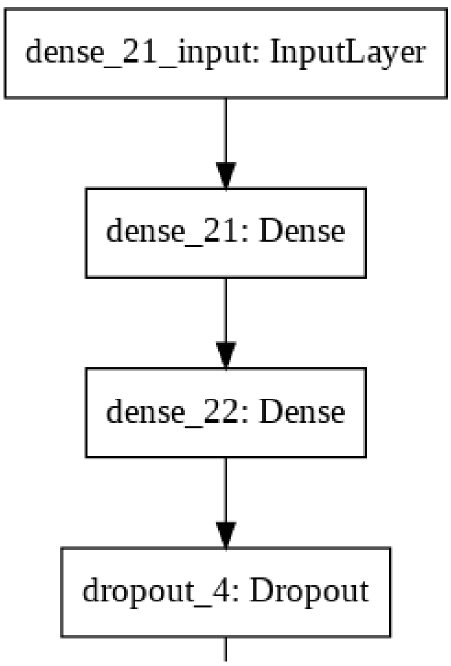
```
model = tf.keras.Sequential(
    [
        tf.keras.layers.Dense(256, activation="relu", input_shape=(X_train.shape[1],)),
        tf.keras.layers.Dense(256, activation="relu"),
        tf.keras.layers.Dropout(0.3),
        tf.keras.layers.Dense(256, activation="relu"),
        tf.keras.layers.Dropout(0.3),
        tf.keras.layers.Dense(1, activation="sigmoid"),
    ]
)
model.summary()
```

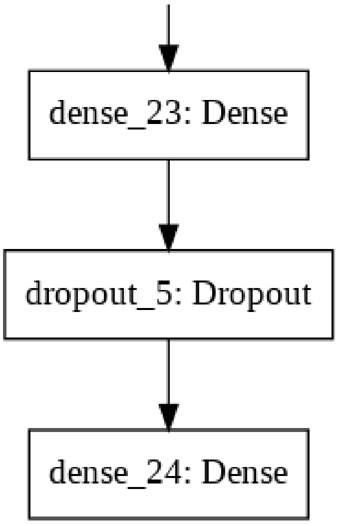
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense (Dense)                (None, 256)               7680      
_________________________________________________________________
dense_1 (Dense)              (None, 256)               65792     
_________________________________________________________________
dropout (Dropout)            (None, 256)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792     
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 257       
=================================================================
Total params: 139,521
Trainable params: 139,521
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
tf.keras.utils.plot_model(model)
```

Out[ ]:





In [11]:

```
metrics = [
    tf.keras.metrics.FalseNegatives(name="fn"),
    tf.keras.metrics.FalsePositives(name="fp"),
    tf.keras.metrics.TrueNegatives(name="tn"),
    tf.keras.metrics.TruePositives(name="tp"),
    tf.keras.metrics.Precision(name="precision"),
    tf.keras.metrics.Recall(name="recall"),
]
```

In [12]:

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-2), loss="binary_crossentropy", metrics=metric
s
)
```

In [13]:

```
history = model.fit(X_train, Y_train, batch_size = 2048, epochs = 30, verbose = 2, valid
ation_data = (X_test, Y_test))
```

```
Epoch 1/30
112/112 - 6s - loss: 0.0184 - fn: 314.0000 - fp: 1068.0000 - tn: 226360.0000 - tp: 103.00
00 - precision: 0.0880 - recall: 0.2470 - val_loss: 0.0035 - val_fn: 35.0000 - val_fp: 1.
0000 - val_tn: 56886.0000 - val_tp: 40.0000 - val_precision: 0.9756 - val_recall: 0.5333
Epoch 2/30
112/112 - 6s - loss: 0.0055 - fn: 152.0000 - fp: 54.0000 - tn: 227374.0000 - tp: 265.0000
- precision: 0.8307 - recall: 0.6355 - val_loss: 0.0029 - val_fn: 24.0000 - val_fp: 5.000
0 - val_tn: 56882.0000 - val_tp: 51.0000 - val_precision: 0.9107 - val_recall: 0.6800
Epoch 3/30
112/112 - 6s - loss: 0.0039 - fn: 115.0000 - fp: 57.0000 - tn: 227371.0000 - tp: 302.0000
- precision: 0.8412 - recall: 0.7242 - val_loss: 0.0043 - val_fn: 34.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 41.0000 - val_precision: 1.0000 - val_recall: 0.5467
Epoch 4/30
112/112 - 6s - loss: 0.0041 - fn: 109.0000 - fp: 60.0000 - tn: 227368.0000 - tp: 308.0000
- precision: 0.8370 - recall: 0.7386 - val_loss: 0.0034 - val_fn: 31.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 44.0000 - val_precision: 1.0000 - val_recall: 0.5867
Epoch 5/30
112/112 - 6s - loss: 0.0039 - fn: 97.0000 - fp: 66.0000 - tn: 227362.0000 - tp: 320.0000
- precision: 0.8290 - recall: 0.7674 - val_loss: 0.0055 - val_fn: 46.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 29.0000 - val_precision: 1.0000 - val_recall: 0.3867
Epoch 6/30
112/112 - 6s - loss: 0.0043 - fn: 111.0000 - fp: 58.0000 - tn: 227370.0000 - tp: 306.0000
- precision: 0.8407 - recall: 0.7338 - val_loss: 0.0028 - val_fn: 23.0000 - val_fp: 8.000
0 - val_tn: 56879.0000 - val_tp: 52.0000 - val_precision: 0.8667 - val_recall: 0.6933
Epoch 7/30
112/112 - 6s - loss: 0.0040 - fn: 96.0000 - fp: 67.0000 - tn: 227361.0000 - tp: 321.0000
- precision: 0.8273 - recall: 0.7698 - val_loss: 0.0030 - val_fn: 26.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 49.0000 - val_precision: 1.0000 - val_recall: 0.6533
Epoch 8/30
112/112 - 6s - loss: 0.0042 - fn: 103.0000 - fp: 55.0000 - tn: 227373.0000 - tp: 314.0000
- precision: 0.8509 - recall: 0.7530 - val_loss: 0.0030 - val_fn: 25.0000 - val_fp: 3.000
```

```
0 - val_tn: 56884.0000 - val_tp: 50.0000 - val_precision: 0.9434 - val_recall: 0.6667
Epoch 9/30
112/112 - 6s - loss: 0.0041 - fn: 99.0000 - fp: 63.0000 - tn: 227365.0000 - tp: 318.0000
- precision: 0.8346 - recall: 0.7626 - val_loss: 0.0028 - val_fn: 24.0000 - val_fp: 2.000
0 - val_tn: 56885.0000 - val_tp: 51.0000 - val_precision: 0.9623 - val_recall: 0.6800
Epoch 10/30
112/112 - 6s - loss: 0.0040 - fn: 95.0000 - fp: 68.0000 - tn: 227360.0000 - tp: 322.0000
- precision: 0.8256 - recall: 0.7722 - val_loss: 0.0177 - val_fn: 68.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 7.0000 - val_precision: 1.0000 - val_recall: 0.0933
Epoch 11/30
112/112 - 6s - loss: 0.0070 - fn: 196.0000 - fp: 39.0000 - tn: 227389.0000 - tp: 221.0000
- precision: 0.8500 - recall: 0.5300 - val_loss: 0.0041 - val_fn: 37.0000 - val_fp: 1.000
0 - val_tn: 56886.0000 - val_tp: 38.0000 - val_precision: 0.9744 - val_recall: 0.5067
Epoch 12/30
112/112 - 6s - loss: 0.0040 - fn: 110.0000 - fp: 54.0000 - tn: 227374.0000 - tp: 307.0000
- precision: 0.8504 - recall: 0.7362 - val_loss: 0.0029 - val_fn: 25.0000 - val_fp: 3.000
0 - val_tn: 56884.0000 - val_tp: 50.0000 - val_precision: 0.9434 - val_recall: 0.6667
Epoch 13/30
112/112 - 6s - loss: 0.0041 - fn: 113.0000 - fp: 67.0000 - tn: 227361.0000 - tp: 304.0000
- precision: 0.8194 - recall: 0.7290 - val_loss: 0.0036 - val_fn: 33.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 42.0000 - val_precision: 1.0000 - val_recall: 0.5600
Epoch 14/30
112/112 - 6s - loss: 0.0041 - fn: 99.0000 - fp: 62.0000 - tn: 227366.0000 - tp: 318.0000
- precision: 0.8368 - recall: 0.7626 - val_loss: 0.0060 - val_fn: 37.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 38.0000 - val_precision: 1.0000 - val_recall: 0.5067
Epoch 15/30
112/112 - 6s - loss: 0.0043 - fn: 105.0000 - fp: 67.0000 - tn: 227361.0000 - tp: 312.0000
- precision: 0.8232 - recall: 0.7482 - val_loss: 0.0032 - val_fn: 30.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 45.0000 - val_precision: 1.0000 - val_recall: 0.6000
Epoch 16/30
112/112 - 6s - loss: 0.0039 - fn: 95.0000 - fp: 58.0000 - tn: 227370.0000 - tp: 322.0000
- precision: 0.8474 - recall: 0.7722 - val_loss: 0.0041 - val_fn: 31.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 44.0000 - val_precision: 1.0000 - val_recall: 0.5867
Epoch 17/30
112/112 - 6s - loss: 0.0038 - fn: 94.0000 - fp: 62.0000 - tn: 227366.0000 - tp: 323.0000
- precision: 0.8390 - recall: 0.7746 - val_loss: 0.0027 - val_fn: 22.0000 - val_fp: 9.000
0 - val_tn: 56878.0000 - val_tp: 53.0000 - val_precision: 0.8548 - val_recall: 0.7067
Epoch 18/30
112/112 - 6s - loss: 0.0039 - fn: 92.0000 - fp: 67.0000 - tn: 227361.0000 - tp: 325.0000
- precision: 0.8291 - recall: 0.7794 - val_loss: 0.0045 - val_fn: 33.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 42.0000 - val_precision: 1.0000 - val_recall: 0.5600
Epoch 19/30
112/112 - 6s - loss: 0.0039 - fn: 90.0000 - fp: 64.0000 - tn: 227364.0000 - tp: 327.0000
- precision: 0.8363 - recall: 0.7842 - val_loss: 0.0027 - val_fn: 24.0000 - val_fp: 5.000
0 - val_tn: 56882.0000 - val_tp: 51.0000 - val_precision: 0.9107 - val_recall: 0.6800
Epoch 20/30
112/112 - 6s - loss: 0.0039 - fn: 96.0000 - fp: 62.0000 - tn: 227366.0000 - tp: 321.0000
- precision: 0.8381 - recall: 0.7698 - val_loss: 0.0035 - val_fn: 19.0000 - val_fp: 15.00
00 - val_tn: 56872.0000 - val_tp: 56.0000 - val_precision: 0.7887 - val_recall: 0.7467
Epoch 21/30
112/112 - 6s - loss: 0.0039 - fn: 91.0000 - fp: 69.0000 - tn: 227359.0000 - tp: 326.0000
- precision: 0.8253 - recall: 0.7818 - val_loss: 0.0028 - val_fn: 24.0000 - val_fp: 6.000
0 - val_tn: 56881.0000 - val_tp: 51.0000 - val_precision: 0.8947 - val_recall: 0.6800
Epoch 22/30
112/112 - 6s - loss: 0.0041 - fn: 97.0000 - fp: 64.0000 - tn: 227364.0000 - tp: 320.0000
- precision: 0.8333 - recall: 0.7674 - val_loss: 0.0031 - val_fn: 21.0000 - val_fp: 9.000
0 - val_tn: 56878.0000 - val_tp: 54.0000 - val_precision: 0.8571 - val_recall: 0.7200
Epoch 23/30
112/112 - 6s - loss: 0.0039 - fn: 93.0000 - fp: 64.0000 - tn: 227364.0000 - tp: 324.0000
- precision: 0.8351 - recall: 0.7770 - val_loss: 0.0026 - val_fn: 24.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 51.0000 - val_precision: 1.0000 - val_recall: 0.6800
Epoch 24/30
112/112 - 6s - loss: 0.0037 - fn: 91.0000 - fp: 59.0000 - tn: 227369.0000 - tp: 326.0000
- precision: 0.8468 - recall: 0.7818 - val_loss: 0.0027 - val_fn: 21.0000 - val_fp: 7.000
0 - val_tn: 56880.0000 - val_tp: 54.0000 - val_precision: 0.8852 - val_recall: 0.7200
Epoch 25/30
112/112 - 6s - loss: 0.0039 - fn: 93.0000 - fp: 67.0000 - tn: 227361.0000 - tp: 324.0000
- precision: 0.8286 - recall: 0.7770 - val_loss: 0.0028 - val_fn: 25.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 50.0000 - val_precision: 1.0000 - val_recall: 0.6667
Epoch 26/30
112/112 - 6s - loss: 0.0039 - fn: 99.0000 - fp: 63.0000 - tn: 227365.0000 - tp: 318.0000
- precision: 0.8346 - recall: 0.7626 - val_loss: 0.0027 - val_fn: 21.0000 - val_fp: 9.000
```

```
- precision: 0.8310 - recall: 0.7620 - val_loss: 0.0027 - val_fn: 21.0000 - val_fp: 9.000
0 - val_tn: 56878.0000 - val_tp: 54.0000 - val_precision: 0.8571 - val_recall: 0.7200
Epoch 27/30
112/112 - 6s - loss: 0.0041 - fn: 92.0000 - fp: 66.0000 - tn: 227362.0000 - tp: 325.0000
- precision: 0.8312 - recall: 0.7794 - val_loss: 0.0029 - val_fn: 24.0000 - val_fp: 2.000
0 - val_tn: 56885.0000 - val_tp: 51.0000 - val_precision: 0.9623 - val_recall: 0.6800
Epoch 28/30
112/112 - 6s - loss: 0.0040 - fn: 85.0000 - fp: 66.0000 - tn: 227362.0000 - tp: 332.0000
- precision: 0.8342 - recall: 0.7962 - val_loss: 0.0038 - val_fn: 28.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 47.0000 - val_precision: 1.0000 - val_recall: 0.6267
Epoch 29/30
112/112 - 6s - loss: 0.0040 - fn: 92.0000 - fp: 61.0000 - tn: 227367.0000 - tp: 325.0000
- precision: 0.8420 - recall: 0.7794 - val_loss: 0.0037 - val_fn: 25.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 50.0000 - val_precision: 1.0000 - val_recall: 0.6667
Epoch 30/30
112/112 - 6s - loss: 0.0036 - fn: 87.0000 - fp: 63.0000 - tn: 227365.0000 - tp: 330.0000
- precision: 0.8397 - recall: 0.7914 - val_loss: 0.0034 - val_fn: 25.0000 - val_fp: 0.000
0e+00 - val_tn: 56887.0000 - val_tp: 50.0000 - val_precision: 1.0000 - val_recall: 0.6667
```

In [14]:

```
X = raw_data.iloc[:,1:-1].values
Y = raw_data.iloc[:,-1].values

scaler = MinMaxScaler()
X = scaler.fit_transform(X)

X.shape, Y.shape
```

Out[14]:

```
((284807, 29), (284807,))
```

In [15]:

```
model.evaluate(X,Y)
```

```
8901/8901 [==============================] - 14s 2ms/step - loss: 0.0033 - fn: 92.0000 -
fp: 113.0000 - tn: 284202.0000 - tp: 400.0000 - precision: 0.7797 - recall: 0.8130
```

Out[15]:

```
[0.003279717406257987,
 92.0,
 113.0,
 284202.0,
 400.0,
 0.7797271013259888,
 0.8130081295967102]
```

In [16]:

```
Y_pred = model.predict(X)
Y_pred.shape, type(Y_pred)
```

Out[16]:

```
((284807, 1), numpy.ndarray)
```

In [17]:

```
Y_pred = Y_pred.reshape((-1))
Y_pred.shape
```

Out[17]:

```
(284807,)
```

In [18]:

```
Y_pred
```

Out[18]:

```
array([0.00119984, 0.00036457, 0.00056165, ..., 0.00043574, 0.00019965,
```

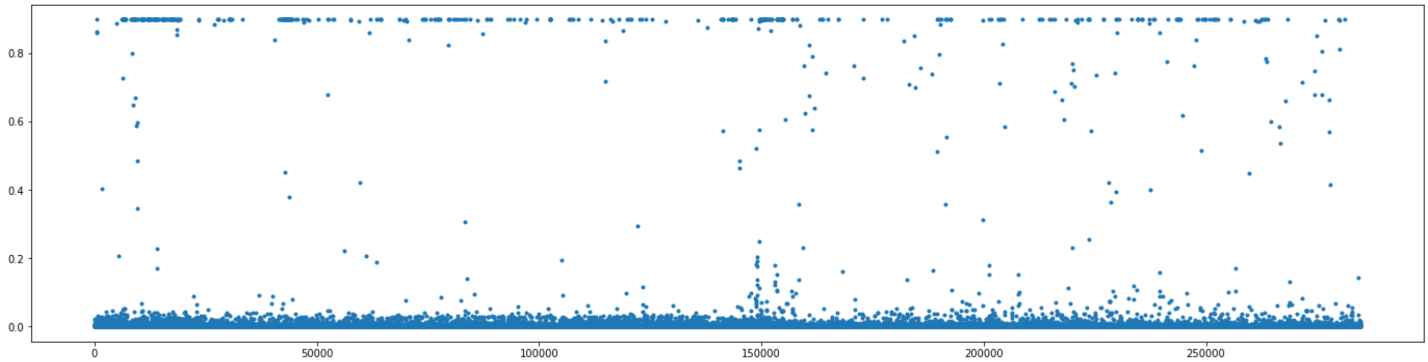
```
array([0.00119984, 0.00036457, 0.00036165, ..., 0.00043574, 0.00019965,
       0.00017482], dtype=float32)
```

In [19]:

```python
plt.rcParams['figure.figsize'] = [24, 6]
plt.plot(Y_pred,'.')
```

Out[19]:

```
[<matplotlib.lines.Line2D at 0x7fbe552b5940>]
```



In [20]:

```python
Y_pred[Y_pred > 0.5] = 1
Y_pred[Y_pred <= 0.5] = 0
Y_pred
```

Out[20]:

```
array([0., 0., 0., ..., 0., 0., 0.], dtype=float32)
```

In [21]:

```python
from sklearn.metrics import confusion_matrix
```

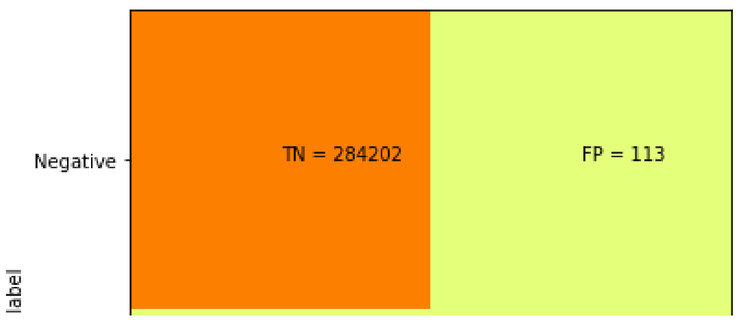
In [22]:

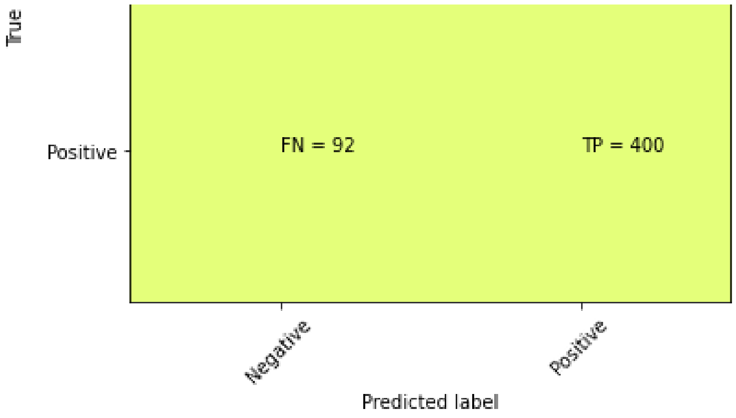
```python
cm = confusion_matrix(Y, Y_pred)
print(cm)
```

```
[[284202    113]
 [    92    400]]
```

In [23]:

```python
plt.clf()
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Wistia)
classNames = ['Negative','Positive']
plt.ylabel('True label')
plt.xlabel('Predicted label')
tick_marks = np.arange(len(classNames))
plt.xticks(tick_marks, classNames, rotation=45)
plt.yticks(tick_marks, classNames)
s = [['TN','FP'], ['FN', 'TP']]
for i in range(2):
    for j in range(2):
        plt.text(j,i, str(s[i][j])+" = "+str(cm[i][j]))
plt.show()
```





why this happend?

In [24]:

```
label = raw_data.iloc[:,-1]

print('label 0 : ', len(label[label == 0]) )
print('label 1 : ', len(label[label == 1]))
```

```
label 0 :  284315
label 1 :  492
```

HW : class_weight 를 이용해 FP 줄이기

In [ ]:

```
```