In [44]:
```
import numpy as np
import pandas as pd
```

In [45]:
```
df =pd.read_csv(r'C:\Users\kishore\Downloads\archive (6)\Churn_Modelling.csv')
```

In [46]:
```
df.head()
```

Out[46]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [47]:
```
df.drop(['CustomerId','RowNumber','Surname'],axis=1,inplace=True)
```

In [48]:
```
df.head()
```

Out[48]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

In [49]:
```
df.dtypes
```

Out[49]:
```
CreditScore          int64
Geography           object
Gender              object
Age                  int64
Tenure               int64
Balance            float64
NumOfProducts        int64
HasCrCard            int64
IsActiveMember       int64
EstimatedSalary    float64
Exited               int64
dtype: object
```

In [70]:
```
df['Geography'].unique()
dummy=pd.get_dummies(df['Geography'],drop_first=True)
df1=pd.concat([df,dummy],axis=1)
df1.head()
df1['Gender'].replace({'Female':0,'Male':1},inplace=True)
df1.drop('Geography',axis=1,inplace=True)
```

In [71]:
```
df1.head()
```

Out[71]:

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Germany	Spain
0	619	0	42	2	0.00	1	1	1	101348.88	1	0	0
1	608	0	41	1	83807.86	1	0	1	112542.58	0	0	1
2	502	0	42	8	159660.80	3	1	0	113931.57	1	0	0
3	699	0	39	1	0.00	2	0	0	93826.63	0	0	0
4	850	0	43	2	125510.82	1	1	1	79084.10	0	0	1

In [72]:
```
df1['Gender'].unique()
```

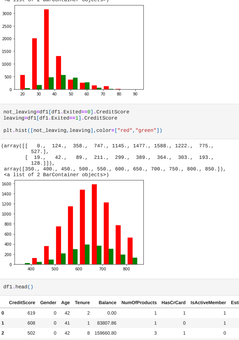
Out[72]:
```
array([0, 1], dtype=int64)
```

In [73]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [74]:
```
not_leaving=df1[df1.Exited==0].Age
leaving=df1[df1.Exited==1].Age

plt.hist([not_leaving,leaving],color=["red","green"])
```

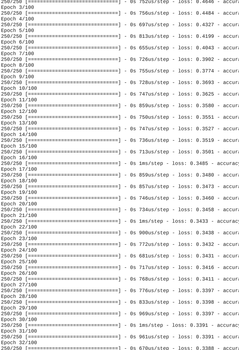
Out[74]:
```
(array([[5.650e+02, 2.012e+03, 3.156e+03, 3.309e+03, 3.818e+02, 2.540e+02,
         1.560e+02, 1.130e+02, 1.360e+01, 4.000e+00],
        [4.600e+01, 1.670e+02, 2.730e+02, 5.628e+02, 4.470e+02, 2.690e+02,
         5.800e+01, 1.400e+01, 1.600e+00, 0.000e+00]]),
 array([18. , 25.4, 32.8, 40.2, 47.6, 55. , 62.4, 69.8, 77.2, 84.6, 92. ]),
 <a list of 2 BarContainer objects>)
```



In [75]:
```
not_leaving=df1[df1.Exited==0].CreditScore
leaving=df1[df1.Exited==1].CreditScore

plt.hist([not_leaving,leaving],color=["red","green"])
```

Out[75]:
```
(array([[  9.,   2.,  124.,  358.,  747., 1145., 1477., 1588., 1222.,  775.,
          [ 19.,   42.,   89.,  211.,  299.,  389.,  364.,  303.,  193.,
           128.]]),
 array([350., 400., 450., 500., 550., 600., 650., 700., 750., 800., 850.]),
 <a list of 2 BarContainer objects>)
```



In [76]:
```
df1.head()
```

Out[76]:

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Germany	Spain
0	619	0	42	2	0.00	1	1	1	101348.88	1	0	0
1	608	0	41	1	83807.86	1	0	1	112542.58	0	0	1
2	502	0	42	8	159660.80	3	1	0	113931.57	1	0	0
3	699	0	39	1	0.00	2	0	0	93826.63	0	0	0
4	850	0	43	2	125510.82	1	1	1	79084.10	0	0	1

In [77]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [107]:
```
scaler=MinMaxScaler()
cols=['CreditScore','Age','Tenure','EstimatedSalary','NumOfProducts','Balance']
df1[cols]=scaler.fit_transform(df1[cols])
X=df1.drop('Exited',axis=1)
y=df1.Exited
```

In [108]:
```
from sklearn.model_selection import train_test_split
Xtrain,Xtest,Ytrain,Ytest=train_test_split(X,Y,test_size=0.2)
```

In [109]:
```
from tensorflow import keras
```

In [110]:
```
model=keras.Sequential([keras.layers.Dense(20,activation='relu'),
                        keras.layers.Dense(10,activation='relu'),
                        keras.layers.Dense(1,activation='sigmoid')])
```

In [111]:
```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

In [112]:
```
model.fit(Xtrain,Ytrain,epochs=100)
```

```
Epoch 1/100
250/250 [==============================] - 0s 744us/step - loss: 0.4912 - accuracy: 0.7940
Epoch 2/100
250/250 [==============================] - 0s 752us/step - loss: 0.4646 - accuracy: 0.7940
Epoch 3/100
250/250 [==============================] - 0s 756us/step - loss: 0.4484 - accuracy: 0.7994
Epoch 4/100
250/250 [==============================] - 0s 697us/step - loss: 0.4327 - accuracy: 0.8314
Epoch 5/100
250/250 [==============================] - 0s 813us/step - loss: 0.4199 - accuracy: 0.8198
Epoch 6/100
250/250 [==============================] - 0s 655us/step - loss: 0.4043 - accuracy: 0.8263
Epoch 7/100
250/250 [==============================] - 0s 726us/step - loss: 0.3902 - accuracy: 0.8326
Epoch 8/100
250/250 [==============================] - 0s 755us/step - loss: 0.3774 - accuracy: 0.8395
Epoch 9/100
250/250 [==============================] - 0s 728us/step - loss: 0.3693 - accuracy: 0.8438
Epoch 10/100
250/250 [==============================] - 0s 747us/step - loss: 0.3625 - accuracy: 0.8481
Epoch 11/100
250/250 [==============================] - 0s 859us/step - loss: 0.3580 - accuracy: 0.8533
Epoch 12/100
250/250 [==============================] - 0s 755us/step - loss: 0.3551 - accuracy: 0.8546
Epoch 13/100
250/250 [==============================] - 0s 747us/step - loss: 0.3527 - accuracy: 0.8546
Epoch 14/100
250/250 [==============================] - 0s 736us/step - loss: 0.3519 - accuracy: 0.8569
Epoch 15/100
250/250 [==============================] - 0s 713us/step - loss: 0.3501 - accuracy: 0.8550
Epoch 16/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3485 - accuracy: 0.8564
Epoch 17/100
250/250 [==============================] - 0s 859us/step - loss: 0.3480 - accuracy: 0.8586
Epoch 18/100
250/250 [==============================] - 0s 857us/step - loss: 0.3473 - accuracy: 0.8572
Epoch 19/100
250/250 [==============================] - 0s 746us/step - loss: 0.3460 - accuracy: 0.8564
Epoch 20/100
250/250 [==============================] - 0s 734us/step - loss: 0.3458 - accuracy: 0.8569
Epoch 21/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3433 - accuracy: 0.8591
Epoch 22/100
250/250 [==============================] - 0s 900us/step - loss: 0.3439 - accuracy: 0.8590
Epoch 23/100
250/250 [==============================] - 0s 772us/step - loss: 0.3432 - accuracy: 0.8586
Epoch 24/100
250/250 [==============================] - 0s 681us/step - loss: 0.3431 - accuracy: 0.8591
Epoch 25/100
250/250 [==============================] - 0s 717us/step - loss: 0.3416 - accuracy: 0.8596
Epoch 26/100
250/250 [==============================] - 0s 768us/step - loss: 0.3411 - accuracy: 0.8589Os - loss: 0.3463 - accuracy: 0.
Epoch 27/100
250/250 [==============================] - 0s 776us/step - loss: 0.3397 - accuracy: 0.8608
Epoch 28/100
250/250 [==============================] - 0s 833us/step - loss: 0.3398 - accuracy: 0.8609Os - loss: 0.3398 - accuracy
Epoch 29/100
250/250 [==============================] - 0s 908us/step - loss: 0.3397 - accuracy: 0.8602
Epoch 30/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3391 - accuracy: 0.8596
Epoch 31/100
250/250 [==============================] - 0s 961us/step - loss: 0.3391 - accuracy: 0.8593
Epoch 32/100
250/250 [==============================] - 0s 678us/step - loss: 0.3388 - accuracy: 0.8609
Epoch 33/100
250/250 [==============================] - 0s 713us/step - loss: 0.3378 - accuracy: 0.8614
Epoch 34/100
250/250 [==============================] - 0s 897us/step - loss: 0.3370 - accuracy: 0.8611
Epoch 35/100
250/250 [==============================] - 0s 877us/step - loss: 0.3371 - accuracy: 0.8583Os - loss: 0.3352 - accuracy: 0.85
Epoch 36/100
250/250 [==============================] - 0s 805us/step - loss: 0.3367 - accuracy: 0.8609
Epoch 37/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3358 - accuracy: 0.8605
Epoch 38/100
250/250 [==============================] - 0s 821us/step - loss: 0.3359 - accuracy: 0.8608
Epoch 39/100
250/250 [==============================] - 0s 805us/step - loss: 0.3357 - accuracy: 0.8614
Epoch 40/100
250/250 [==============================] - 0s 795us/step - loss: 0.3350 - accuracy: 0.8616
Epoch 41/100
250/250 [==============================] - 0s 900us/step - loss: 0.3347 - accuracy: 0.8634
Epoch 42/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3357 - accuracy: 0.8615
Epoch 43/100
250/250 [==============================] - 0s 937us/step - loss: 0.3340 - accuracy: 0.8635
Epoch 44/100
250/250 [==============================] - 0s 781us/step - loss: 0.3346 - accuracy: 0.8618
Epoch 45/100
250/250 [==============================] - 0s 763us/step - loss: 0.3346 - accuracy: 0.8597
Epoch 46/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3335 - accuracy: 0.8631
Epoch 47/100
250/250 [==============================] - 0s 918us/step - loss: 0.3341 - accuracy: 0.8608
Epoch 48/100
250/250 [==============================] - 0s 752us/step - loss: 0.3334 - accuracy: 0.8616
Epoch 49/100
250/250 [==============================] - 0s 746us/step - loss: 0.3341 - accuracy: 0.8602
Epoch 50/100
250/250 [==============================] - 0s 745us/step - loss: 0.3332 - accuracy: 0.8620Os - loss: 0.3299 - accuracy: 0.86
Epoch 51/100
250/250 [==============================] - 0s 871us/step - loss: 0.3327 - accuracy: 0.8636
Epoch 52/100
250/250 [==============================] - 0s 933us/step - loss: 0.3322 - accuracy: 0.8629
Epoch 53/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3329 - accuracy: 0.8610
Epoch 54/100
250/250 [==============================] - 0s 986us/step - loss: 0.3320 - accuracy: 0.8629
Epoch 55/100
250/250 [==============================] - 0s 989us/step - loss: 0.3314 - accuracy: 0.8636
Epoch 56/100
250/250 [==============================] - 0s 889us/step - loss: 0.3314 - accuracy: 0.8630
Epoch 57/100
250/250 [==============================] - 0s 906us/step - loss: 0.3312 - accuracy: 0.8644
Epoch 58/100
250/250 [==============================] - 0s 872us/step - loss: 0.3320 - accuracy: 0.8652
Epoch 59/100
250/250 [==============================] - 0s 876us/step - loss: 0.3312 - accuracy: 0.8636
Epoch 60/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3314 - accuracy: 0.8621
Epoch 61/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3305 - accuracy: 0.8616
Epoch 62/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3307 - accuracy: 0.8627
Epoch 63/100
250/250 [==============================] - 0s 976us/step - loss: 0.3305 - accuracy: 0.8631
Epoch 64/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3314 - accuracy: 0.8640
Epoch 65/100
250/250 [==============================] - 0s 929us/step - loss: 0.3304 - accuracy: 0.8641
Epoch 66/100
250/250 [==============================] - 0s 993us/step - loss: 0.3312 - accuracy: 0.8619
Epoch 67/100
250/250 [==============================] - 0s 901us/step - loss: 0.3298 - accuracy: 0.8633
Epoch 68/100
250/250 [==============================] - 0s 945us/step - loss: 0.3296 - accuracy: 0.8639
Epoch 69/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3296 - accuracy: 0.8625
Epoch 70/100
250/250 [==============================] - 0s 978us/step - loss: 0.3298 - accuracy: 0.8644
Epoch 71/100
250/250 [==============================] - 0s 878us/step - loss: 0.3301 - accuracy: 0.8627
Epoch 72/100
250/250 [==============================] - 0s 854us/step - loss: 0.3299 - accuracy: 0.8641
Epoch 73/100
250/250 [==============================] - 0s 948us/step - loss: 0.3287 - accuracy: 0.8646
Epoch 74/100
250/250 [==============================] - 0s 923us/step - loss: 0.3296 - accuracy: 0.8635
Epoch 75/100
250/250 [==============================] - 0s 901us/step - loss: 0.3288 - accuracy: 0.8631
Epoch 76/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3298 - accuracy: 0.8652
Epoch 77/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3293 - accuracy: 0.8634
Epoch 78/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3299 - accuracy: 0.8631
Epoch 79/100
250/250 [==============================] - 0s 990us/step - loss: 0.3299 - accuracy: 0.8635
Epoch 80/100
250/250 [==============================] - 0s 826us/step - loss: 0.3305 - accuracy: 0.8629
Epoch 81/100
250/250 [==============================] - 0s 951us/step - loss: 0.3291 - accuracy: 0.8651
Epoch 82/100
250/250 [==============================] - 0s 941us/step - loss: 0.3289 - accuracy: 0.8643
Epoch 83/100
250/250 [==============================] - 0s 883us/step - loss: 0.3283 - accuracy: 0.8666
Epoch 84/100
250/250 [==============================] - 0s 839us/step - loss: 0.3279 - accuracy: 0.8645
Epoch 85/100
250/250 [==============================] - 0s 945us/step - loss: 0.3278 - accuracy: 0.8649
Epoch 86/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3282 - accuracy: 0.8643
Epoch 87/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3284 - accuracy: 0.8644
Epoch 88/100
250/250 [==============================] - 0s 936us/step - loss: 0.3282 - accuracy: 0.8645
Epoch 89/100
250/250 [==============================] - 0s 876us/step - loss: 0.3280 - accuracy: 0.8650
Epoch 90/100
250/250 [==============================] - 0s 836us/step - loss: 0.3282 - accuracy: 0.8637
Epoch 91/100
250/250 [==============================] - 0s 964us/step - loss: 0.3284 - accuracy: 0.8668
Epoch 92/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3278 - accuracy: 0.8646
Epoch 93/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3281 - accuracy: 0.8625
Epoch 94/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3273 - accuracy: 0.8649
Epoch 95/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3272 - accuracy: 0.8659
Epoch 96/100
250/250 [==============================] - 0s 982us/step - loss: 0.3272 - accuracy: 0.8661
Epoch 97/100
250/250 [==============================] - 0s 956us/step - loss: 0.3280 - accuracy: 0.8639
Epoch 98/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3288 - accuracy: 0.8639
Epoch 99/100
250/250 [==============================] - 0s 940us/step - loss: 0.3263 - accuracy: 0.8650
Epoch 100/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3277 - accuracy: 0.8654
```

Out[112]:
```
<keras.callbacks.History at 0x14aa2e53820>
```

In [113]:
```
model.evaluate(Xtest,Ytest)
```

```
63/63 [==============================] - 0s 720us/step - loss: 0.3271 - accuracy: 0.8620
[0.3271419405937195, 0.8619999885559082]
```

In [ ]:
```

```