

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
df=pd.read_excel('/content/Churn_Modelling 1111.xlsx')
```

```
df.head()
```

	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	42	2	0.00	1	1	1	10134
1	41	1	83807.86	1	0	1	11254
2	42	8	159660.80	3	1	0	11396
3	39	1	0.00	2	0	0	9382
4	43	2	125510.82	1	1	1	7908

```
df.columns
```

```
Index(['Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
      'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

```
x=df[['Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
      'IsActiveMember', 'EstimatedSalary']]
y=df['Exited']
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense
```

```

model = Sequential()

model.add(Dense(units=64, activation='sigmoid', input_dim=x_train.shape[1]))
model.add(Dense(units=32, activation='sigmoid'))

model.add(Dense(units=1, activation='sigmoid'))

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

model.fit(x_train, y_train, epochs=12, batch_size=64, validation_split=0.2)

```

```

Epoch 1/12
100/100 [=====] - 1s 3ms/step - loss: 0.5052 - accuracy: 0.4948
Epoch 2/12
100/100 [=====] - 0s 2ms/step - loss: 0.4938 - accuracy: 0.5062
Epoch 3/12
100/100 [=====] - 0s 2ms/step - loss: 0.4932 - accuracy: 0.5068
Epoch 4/12
100/100 [=====] - 0s 2ms/step - loss: 0.4939 - accuracy: 0.5061
Epoch 5/12
100/100 [=====] - 0s 2ms/step - loss: 0.4942 - accuracy: 0.5058
Epoch 6/12
100/100 [=====] - 0s 2ms/step - loss: 0.4933 - accuracy: 0.5067
Epoch 7/12
100/100 [=====] - 0s 2ms/step - loss: 0.4929 - accuracy: 0.5071
Epoch 8/12
100/100 [=====] - 0s 2ms/step - loss: 0.4936 - accuracy: 0.5064
Epoch 9/12
100/100 [=====] - 0s 2ms/step - loss: 0.4934 - accuracy: 0.5066
Epoch 10/12
100/100 [=====] - 0s 2ms/step - loss: 0.4931 - accuracy: 0.5069
Epoch 11/12
100/100 [=====] - 0s 2ms/step - loss: 0.4931 - accuracy: 0.5069
Epoch 12/12
100/100 [=====] - 0s 2ms/step - loss: 0.4935 - accuracy: 0.5065
<keras.callbacks.History at 0x7afa6c1e61d0>

```

```

accuracy = model.evaluate(x_test, y_test)[1]
print(f'Test Accuracy: {accuracy:.4f}')

```

```

63/63 [=====] - 0s 990us/step - loss: 0.5071 - accuracy: 0.4929
Test Accuracy: 0.7895

```

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	512
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 1)	33
Total params: 2,625		
Trainable params: 2,625		
Non-trainable params: 0		

```
import numpy as np
predictions = model.predict(x_test)
rounded_predictions = np.round(predictions)
```

```
63/63 [=====] - 0s 783us/step
```

```
print(x_test.values[0])
print(rounded_predictions[0])
```

```
[3.0000000e+01 3.0000000e+00 9.8415370e+04 1.0000000e+00 1.0000000e+00
 1.0000000e+00 1.1052048e+05]
[0.]
```

```
print(x_test.values[1])
print(rounded_predictions[1])
```

```
[5.6000000e+01 9.0000000e+00 0.0000000e+00 1.0000000e+00 0.0000000e+00
 0.0000000e+00 8.554733e+04]
[0.]
```

x\_test

	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
2019	30	3	98415.37	1	1	1	111346.25
7834	56	9	0.00	1	0	0	83574.00
3619	33	0	0.00	2	1	0	113608.00
9836	39	10	0.00	2	1	1	173380.00
6715	67	10	124577.15	1	0	1	164984.00
...	...	...	...	...	...	...	...
2758	36	6	0.00	2	0	0	83574.00
3576	28	8	0.00	2	1	1	92580.00
747	46	5	177619.71	1	1	0	57653.00
6637	36	3	0.00	2	1	0	159354.00
8886	36	1	0.00	1	0	1	92580.00

2000 rows x 7 columns

```
print(x_test.values[3])
print(rounded_predictions[3])
```

```
[3.9000000e+01 1.0000000e+01 0.0000000e+00 2.0000000e+00 1.0000000e+00
 1.0000000e+00 1.7040945e+05]
[0.]
```

