In [14]:

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

In [15]:

```
df = pd.read_csv('Churn_Modelling.csv')
df.head()
```

Out[15]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bala
0	1	15634602	Hargrave	619	France	Female	42	2	(
1	2	15647311	Hill	608	Spain	Female	41	1	83807
2	3	15619304	Onio	502	France	Female	42	8	15966(
3	4	15701354	Boni	699	France	Female	39	1	(
4	5	15737888	Mitchell	850	Spain	Female	43	2	12551(

In [16]:

```
df.drop('RowNumber', axis = 'columns', inplace = True)
df.drop('CustomerId', axis = 'columns', inplace = True)
df.drop('Surname', axis = 'columns', inplace = True)
```

In [17]:

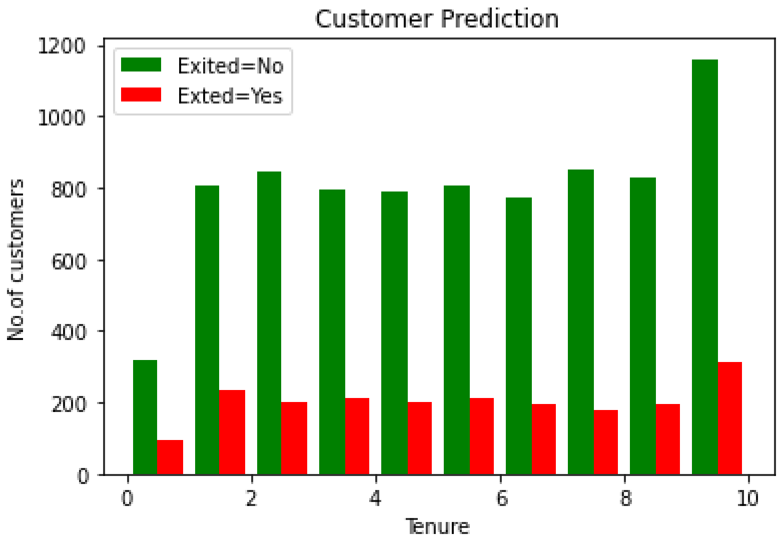
```
Ext_NO = df[df.Exited == 0].Tenure
Ext_YES = df[df.Exited == 1].Tenure
```

In [18]:

```
plt.xlabel('Tenure')
plt.ylabel('No.of customers')
plt.title('Customer Prediction')
plt.hist([Ext_NO,Ext_YES], color = ['green', 'red'], label = ['Exited=N
plt.legend()
```

Out[18]:

```
<matplotlib.legend.Legend at 0x26c3d83fb50>
```



In [19]:

```
df['Gender'].replace({'Female':1, 'Male':0}, inplace = True)
```

In [20]:

```python
df.head()
```

Out[20]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	Is
0	619	France	1	42	2	0.00	1	1	
1	608	Spain	1	41	1	83807.86	1	0	
2	502	France	1	42	8	159660.80	3	1	
3	699	France	1	39	1	0.00	2	0	
4	850	Spain	1	43	2	125510.82	1	1	

In [21]:

```python
one_hot = pd.get_dummies(df['Geography'])
df = df.join(one_hot)
df.drop('Geography', axis = 1, inplace = True)
```

In [22]:

```python
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
```

In [23]:

```python
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,
```

In [24]:

```python
import tensorflow as tf
from tensorflow import keras
model = keras.Sequential([keras.layers.Dense(30, activation = 'relu'),
                          keras.layers.Dense(15, activation = 'relu'),
                          keras.layers.Dense(1, activation = 'sigmoid')
```

In [25]:

```python
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics
model.fit(x_train, y_train, epochs = 100)
```

```
Epoch 1/100
250/250 [==============================] - 1s 2ms/step - loss: 645.4672
- accuracy: 0.6267
Epoch 2/100
250/250 [==============================] - 0s 2ms/step - loss: 39.7789 -
accuracy: 0.6378
Epoch 3/100
250/250 [==============================] - 0s 2ms/step - loss: 45.1663 -
accuracy: 0.6471
Epoch 4/100
250/250 [==============================] - 0s 2ms/step - loss: 41.0481 -
accuracy: 0.6392
Epoch 5/100
250/250 [==============================] - 0s 2ms/step - loss: 39.6550 -
accuracy: 0.6446
Epoch 6/100
250/250 [==============================] - 0s 2ms/step - loss: 39.5323 -
accuracy: 0.6409
Epoch 7/100
250/250 [==============================] - 0s 2ms/step - loss: 40.4845
```

In [ ]:

```python

```