In [11]:

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

```

In [12]:

```python
df = pd.read_csv("bank.csv")
```

In [13]:

```python
df
```

Out[13]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure
0	1	15634602	Hargrave	619	France	Female	42	2
1	2	15647311	Hill	608	Spain	Female	41	1
2	3	15619304	Onio	502	France	Female	42	8
3	4	15701354	Boni	699	France	Female	39	1
4	5	15737888	Mitchell	850	Spain	Female	43	2
...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5
9996	9997	15569892	Johnstone	516	France	Male	35	10
9997	9998	15584532	Liu	709	France	Female	36	7
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3
9999	10000	15628319	Walker	792	France	Female	28	4

10000 rows × 14 columns

In [14]:

```
df.isnull()
```

Out[14]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	E
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
9995	False	False	False	False	False	False	False	False	
9996	False	False	False	False	False	False	False	False	
9997	False	False	False	False	False	False	False	False	
9998	False	False	False	False	False	False	False	False	
9999	False	False	False	False	False	False	False	False	

10000 rows × 14 columns

In [15]:

```
df.isnull().sum()
```

Out[15]:

```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

In [16]:

```
df.info
```

Out[16]:

```
<bound method DataFrame.info of       RowNumber  CustomerId    Surname  Cr
editScore Geography  Gender  Age  \
0             1    15634602   Hargrave          619    France  Female   42
1             2    15647311       Hill          608     Spain  Female   41
2             3    15619304       Onio          502    France  Female   42
3             4    15701354       Boni          699    France  Female   39
4             5    15737888   Mitchell          850     Spain  Female   43
...         ...         ...        ...          ...       ...     ...  ...
9995       9996    15606229   Obijiaku          771    France    Male   39
9996       9997    15569892  Johnstone          516    France    Male   35
9997       9998    15584532        Liu          709    France  Female   36
9998       9999    15682355  Sabbatini          772   Germany    Male   42
9999      10000    15628319     Walker          792    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
...               ...     ...
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]>
```

In [17]:

```
df.dtypes
```

Out[17]:

```
RowNumber            int64
CustomerId           int64
Surname             object
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 [18]:

```
df.columns
```

Out[18]:

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

In [19]:

```
df = df.drop(['RowNumber','CustomerId','Surname'], axis = 1)
```

In [20]:

```
df.head()
```

Out[20]:

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

In [21]:

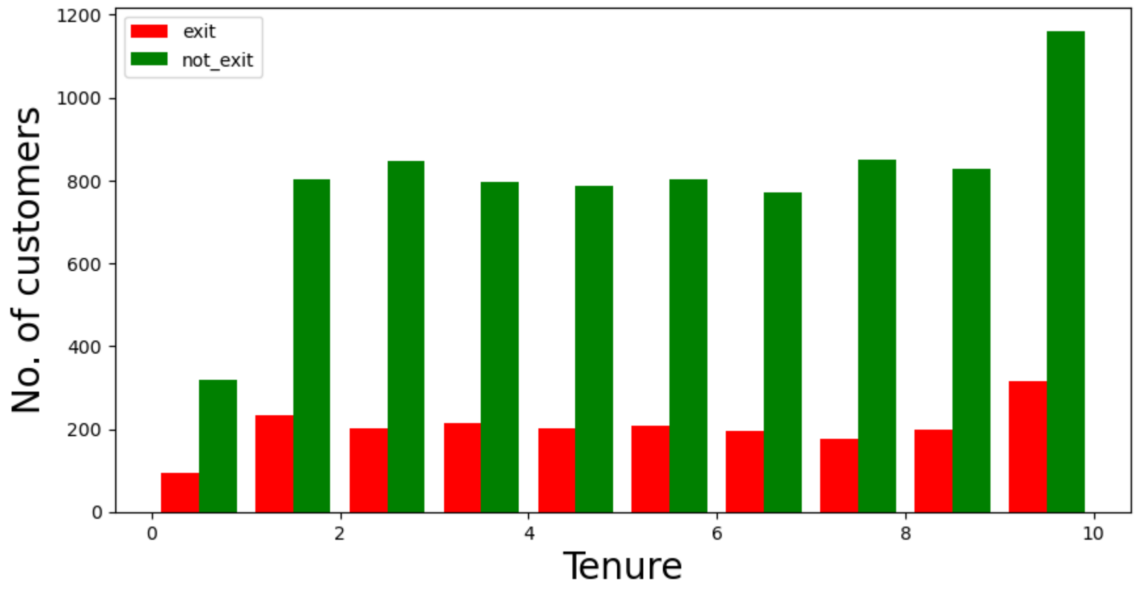
```
def visualization (x,y,xlabel):
    plt.figure(figsize=(10,5))
    plt.hist([x,y], color=['red','green'], label=['exit', 'not_exit'])
    plt.xlabel(xlabel,fontsize=20)
    plt.ylabel("No. of customers", fontsize = 20)
    plt.legend()
```

In [22]:

```
df_churn_exited = df [df['Exited']==1]['Tenure']
df_churn_not_exited = df[df['Exited']==0]['Tenure']
```

In [23]:

```
visualization(df_churn_exited,df_churn_not_exited,"Tenure")
```

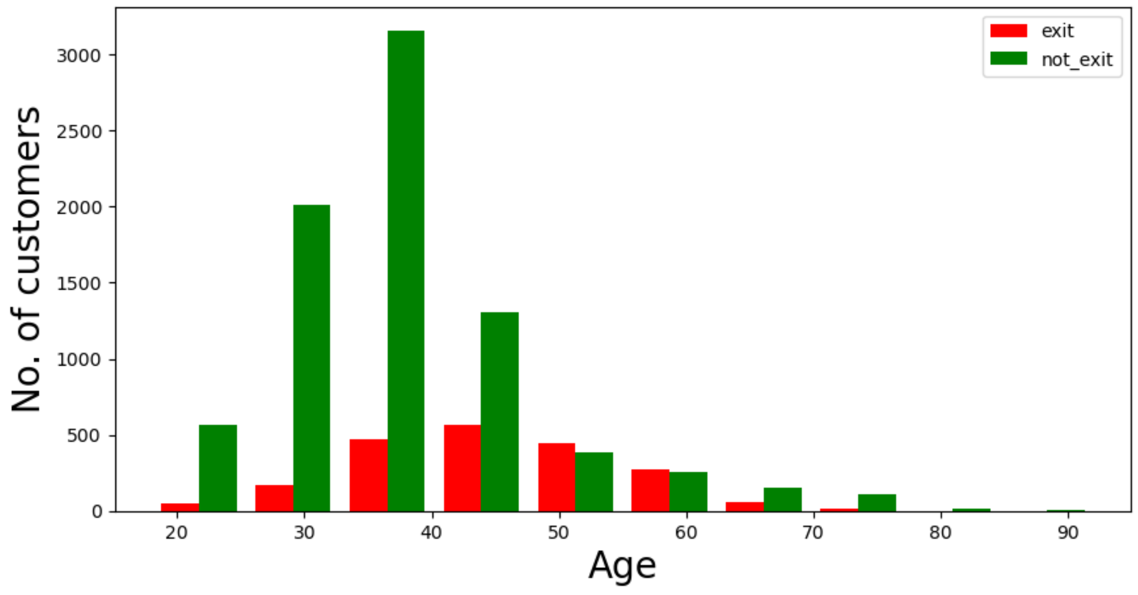


In [24]:

```
df_churn_exited = df [df['Exited']==1]['Age']
df_churn_not_exited = df[df['Exited']==0]['Age']
```

In [25]:

```
visualization(df_churn_exited,df_churn_not_exited,"Age")
```



In [26]:

```
df.dtypes
```

Out[26]:

```
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 [27]:

```
x = df[['CreditScore','Gender','Age','Tenure','Balance', 'NumOfProducts','HasCrCard'
states = pd.get_dummies(df['Geography'],drop_first= True)
gender = pd.get_dummies(df['Gender'],drop_first= True)
```

In [28]:

```
df = pd.concat([df,gender,states], axis=1)
```

In [29]:

```
df.head()
```

Out[29]:

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

In [30]:

```
x = df[['CreditScore','Age','Tenure','Balance', 'NumOfProducts','HasCrCard','IsActiv
```

In [31]:

```
y = df['Exited']
```

In [ ]:

```

```

In [32]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0, test_size=0.25)
```

In [33]:

```
x.shape
```

Out[33]:

```
(10000, 9)
```

In [34]:

```
x_test.shape
```

Out[34]:

```
(2500, 9)
```

In [43]:

```
x_train
```

Out[43]:

```
array([[-1.13575957, -0.46944165, -0.70387817, ..., -1.02664045,
         0.60325767, -0.50290013],
       [-0.42644403,  1.73021217,  0.67859859, ...,  0.97405085,
         0.2269567 ,  1.98846639],
       [ 1.29030518,  0.39129245, -1.39511655, ..., -1.02664045,
        -1.36445846, -0.50290013],
       ...,
       [ 2.04074047,  0.5825667 , -0.01263979, ...,  0.97405085,
         1.61079312,  1.98846639],
       [-0.87876119, -0.2781674 , -1.39511655, ..., -1.02664045,
        -0.35801334, -0.50290013],
       [-0.38532429,  0.20001821, -0.70387817, ...,  0.97405085,
         1.37555676, -0.50290013]])
```

In [35]:

```
x_test.shape
```

Out[35]:

```
(2500, 9)
```

In [36]:

```
x_train.shape
```

Out[36]:

```
(7500, 9)
```

In [37]:

```
from sklearn.neural_network import MLPClassifier
```

In [39]:

```
ann = MLPClassifier(hidden_layer_sizes=(100,100,100),
                   random_state =0,
                   max_iter=100, activation='relu')
```

In [40]:

```
ann.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (100) reached and the optimization hasn't converged yet.
  warnings.warn(
```

Out[40]:

```
MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=100, random_sta
te=0)
```

In [41]:

```
y_pred =ann.predict(x_test)
```

In [42]:

```
y_pred
```

Out[42]:

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [45]:

```
import sklearn
```

In [48]:

```
from sklearn.metrics import ConfusionMatrixDisplay, classification_report
from sklearn.metrics import accuracy_score
```

In [49]:

```
y_test.value_counts()
```

Out[49]:

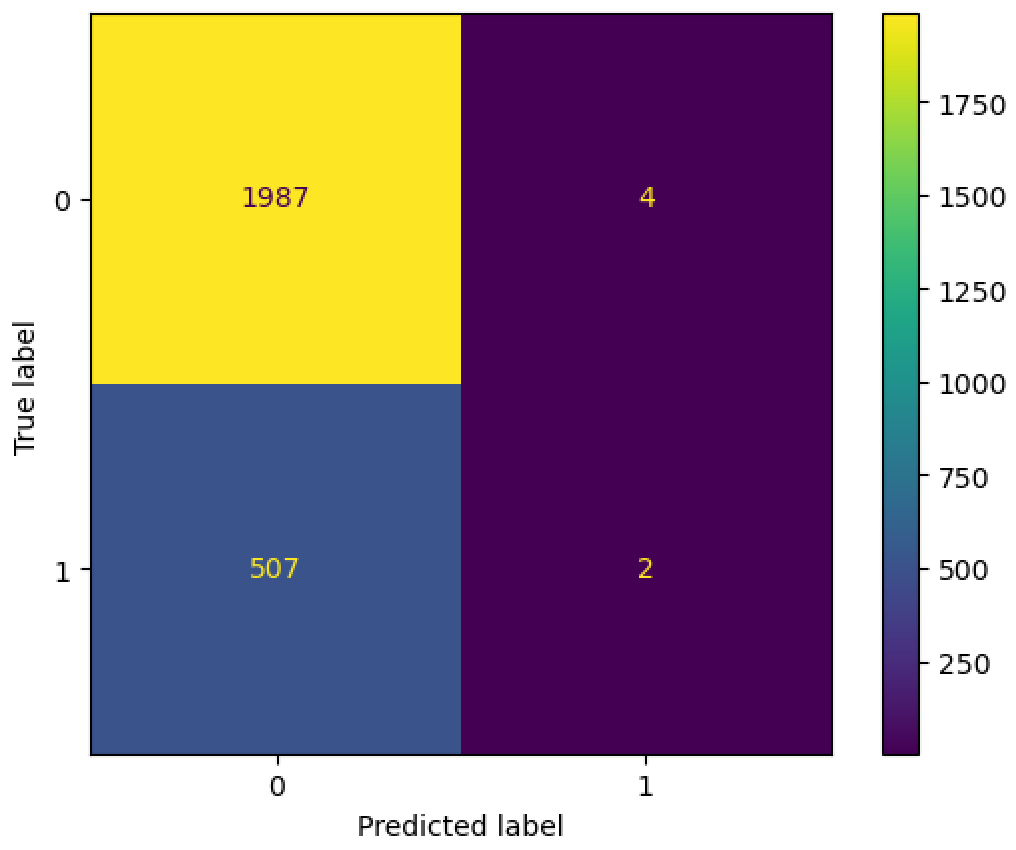
```
0    1991
1     509
Name: Exited, dtype: int64
```

In [50]:

```
ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

Out[50]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2257a6
4b3d0>
```



In [51]:

```
accuracy_score(y_test,y_pred)
```

Out[51]:

```
0.7956
```

In [52]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      1.00      0.89      1991
           1       0.33      0.00      0.01       509

    accuracy                           0.80      2500
   macro avg       0.57      0.50      0.45      2500
weighted avg       0.70      0.80      0.71      2500
```

In [ ]:

```

```