

1. The Dataset was successfully downloaded.

Import Libraries

In [3]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split
```

2. Load the Dataset

In [4]:

```
data = pd.read_csv("Churn_Modelling.csv")
```

In [5]:

```
data.head()
```

Out[5]:

	Row Number	Customer Id	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	Michell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

3. Perform Below Visualizations

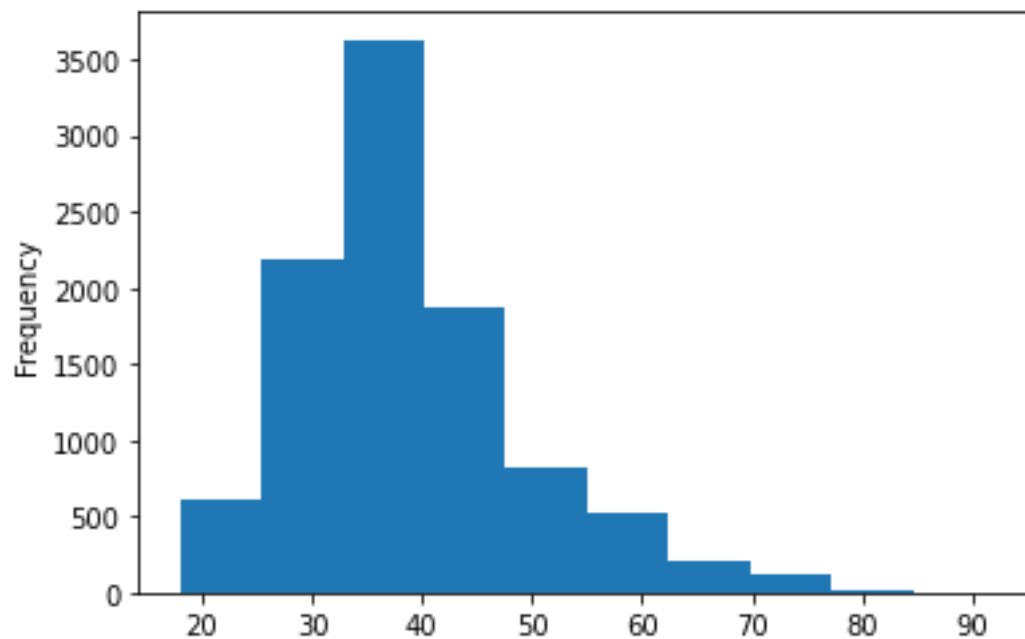
3.1 Univariate Analysis

In [6]:

```
#Histogram
data['Age'].plot(kind='hist')
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcfc78bbfd0>
```

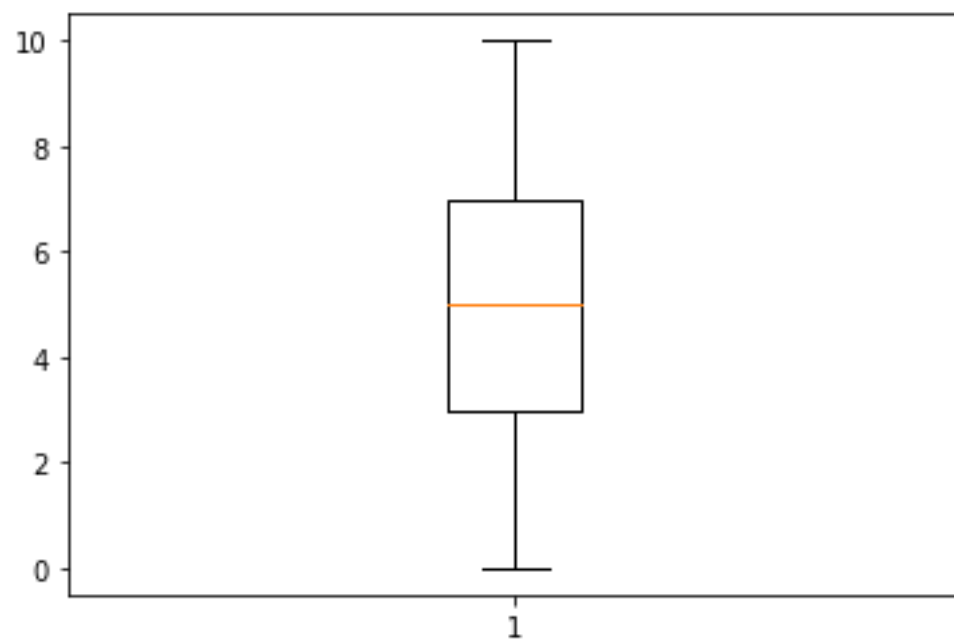


In [7]:

```
#Boxplot  
plt.boxplot(data['Tenure'])
```

Out[7]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcfc72d4050>,  
             <matplotlib.lines.Line2D at 0x7fcfc72d4590>],  
 'caps': [<matplotlib.lines.Line2D at 0x7fcfc72d4ad0>,  
         <matplotlib.lines.Line2D at 0x7fcfc72db050>],  
 'boxes': [<matplotlib.lines.Line2D at 0x7fcfc7349a90>],  
 'medians': [<matplotlib.lines.Line2D at 0x7fcfc72db5d0>],  
 'fliers': [<matplotlib.lines.Line2D at 0x7fcfc72dbb10>],  
 'means': []}
```

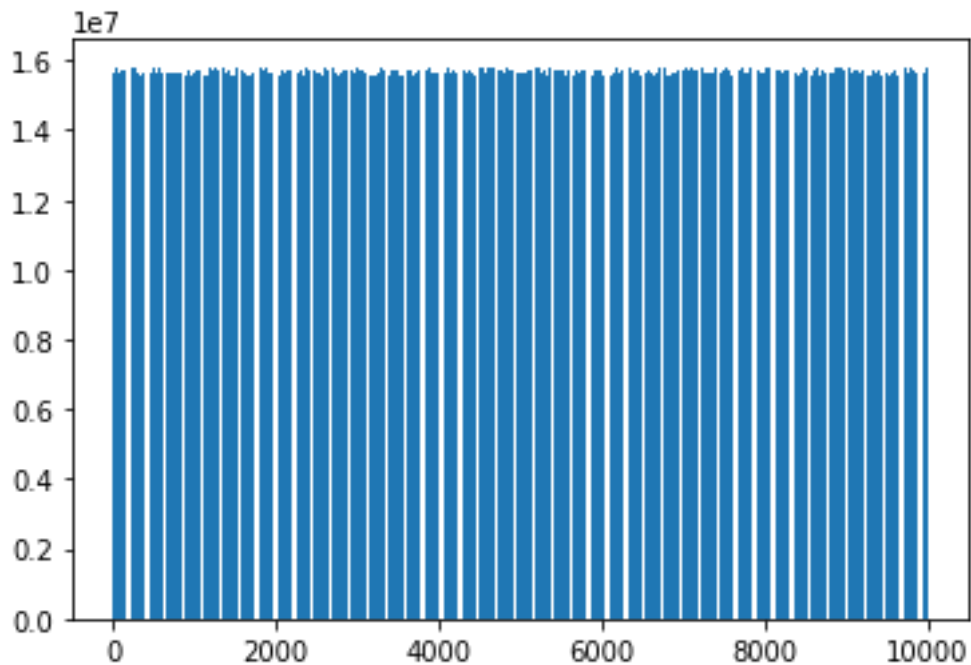


In [8]:

```
#Bar chart
df = pd.DataFrame(data)
X = list(df.iloc[:,0])
Y = list(df.iloc[:,1])
plt.bar(X,Y)
```

Out[8]:

<BarContainer object of 10000 artists>



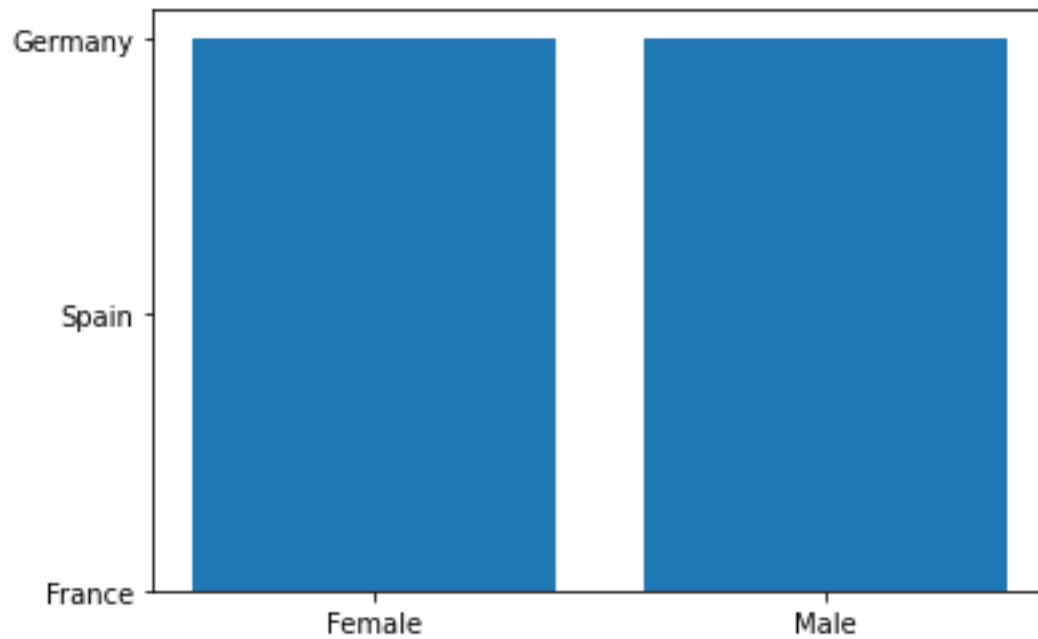
3.2 Bivariate Analysis

```
#Stacked Bar Chart
plt.bar(data['Gender'],data['Geography'])
```

In [9]:

<BarContainer object of 10000 artists>

Out[9]:

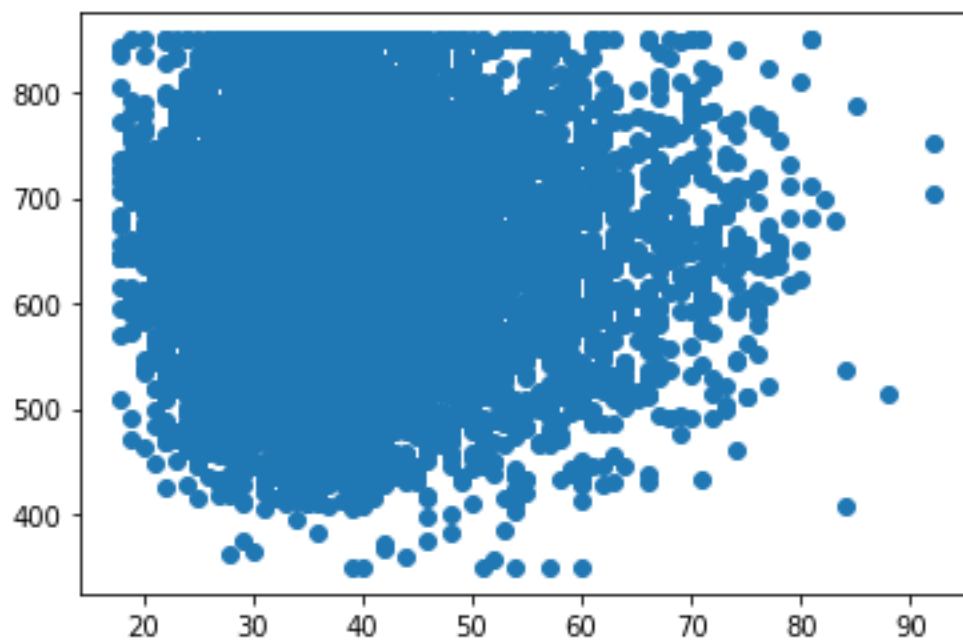


In [10]:

```
#Scatter plot
plt.scatter(data['Age'],data['CreditScore'])
```

Out[10]:

```
<matplotlib.collections.PathCollection at 0x7fcfb9647f90>
```



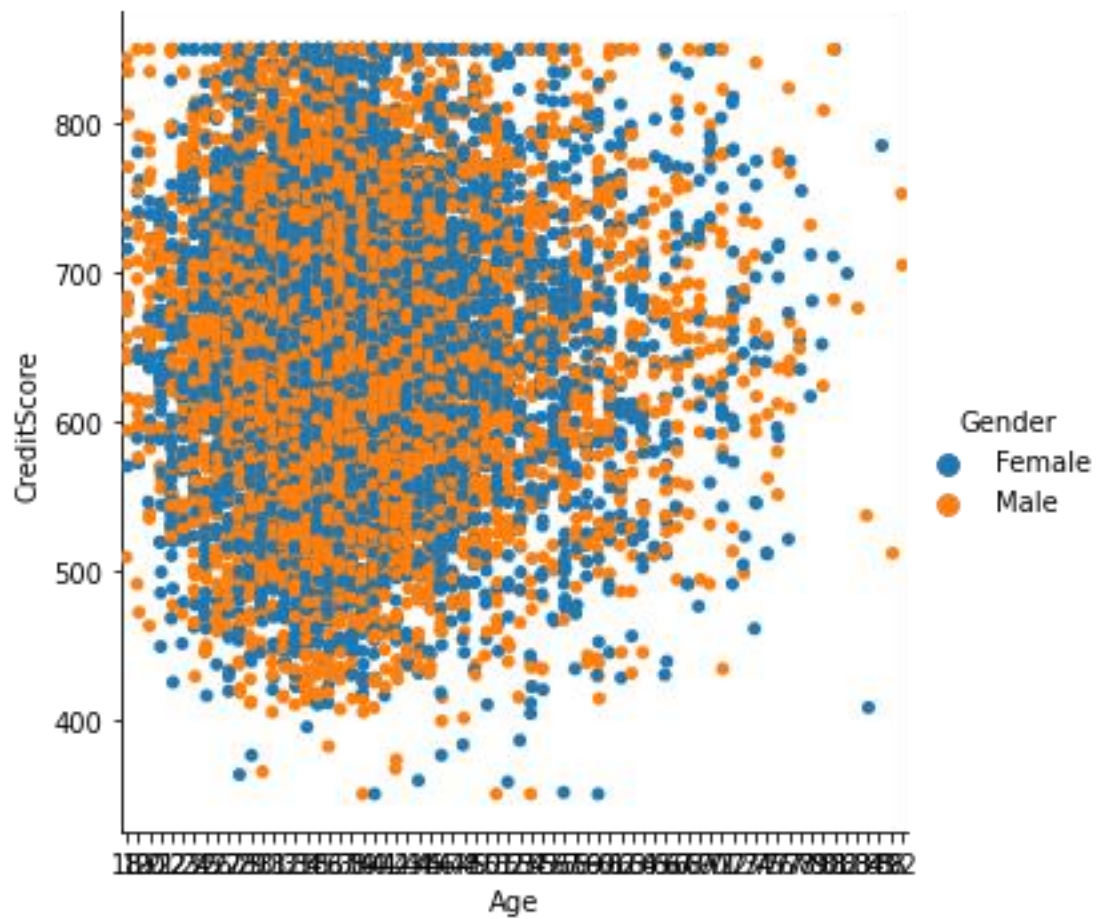
3.3 Multivariate Analysis

In [11]:

```
sns.catplot(data=data,x='Age',y='CreditScore',hue='Gender')
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x7fcfbfd54a10>
```

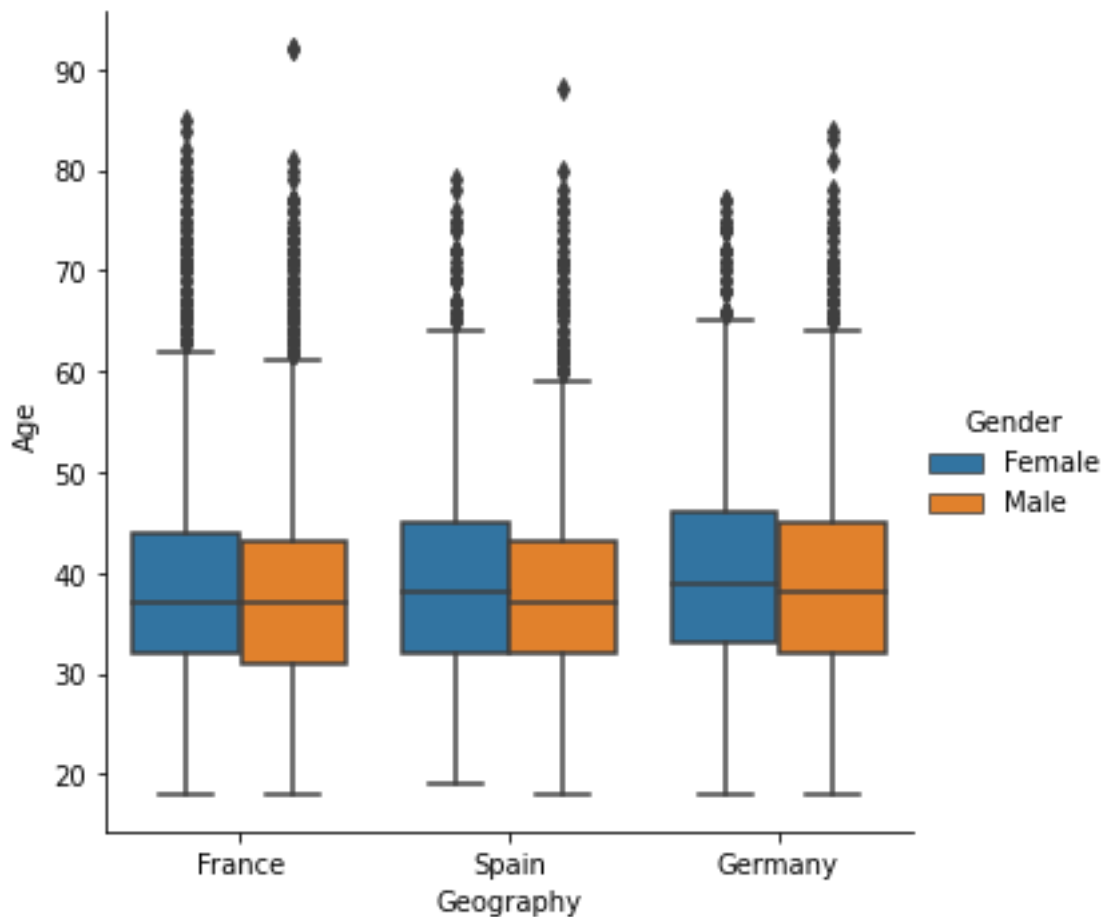


In [12]:

```
#Box Plot  
sns.catplot(data=data,x='Geography',y='Age',hue='Gender',kind='box')
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x7fcfb93821d0>
```



4. Perform Descriptive Statistics on the dataset

In [13]:

```
data.mean()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
  """Entry point for launching an IPython kernel.
```

Out[13]:

```
RowNumber          5.000500e+03
CustomerId          1.569094e+07
CreditScore        6.505288e+02
Age                3.892180e+01
Tenure             5.012800e+00
Balance            7.648589e+04
NumOfProducts     1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```

In [14]:

```
data.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
"""Entry point for launching an IPython kernel.
```

Out[14]:

```
RowNumber      5.000500e+03
CustomerId      1.569074e+07
CreditScore     6.520000e+02
Age             3.700000e+01
Tenure          5.000000e+00
Balance         9.719854e+04
NumOfProducts  1.000000e+00
HasCrCard       1.000000e+00
IsActiveMember  1.000000e+00
EstimatedSalary 1.001939e+05
Exited          0.000000e+00
dtype: float64
```

In [15]:

```
data.describe()
```

Out[15]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.500000	15690.794000	650.528800	38.921800	5.012800	76485.889288	1.530200	0.705500	0.515100	100090.239881	0.203700
std	2886.895680	71936.190000	96.653299	10.487806	2.892174	62397.405202	0.581654	0.455840	0.499797	57510.492818	0.402769
min	1.000000	1556570.000000	350.000000	18.000000	0.000000	0.000000	1.000000	0.000000	0.000000	11.580000	0.000000
25%	2500.750000	1562853.000000	584.000000	32.000000	3.000000	0.000000	1.000000	0.000000	0.000000	51002.100000	0.000000
50%	5000.500000	15690.794000	652.000000	37.000000	5.000000	97198.540000	1.000000	1.000000	1.000000	100193.915000	0.000000

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
75%	7500.25000	1.575323e+07	718.00000	44.00000	7.00000	127644.240000	2.000000	1.000000	1.000000	149388.247500	0.000000
max	10000.000000	1.581569e+07	850.00000	92.00000	10.00000	250898.090000	4.000000	1.000000	1.000000	199992.480000	1.000000

In [16]:

```
data.shape
```

Out[16]:

```
(10000, 14)
```

5. Handle the missing values

In [17]:

```
data.isnull().sum()
```

Out[17]:

```
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
```

6. Find the Outliners and replace the Outliners

In [18]:

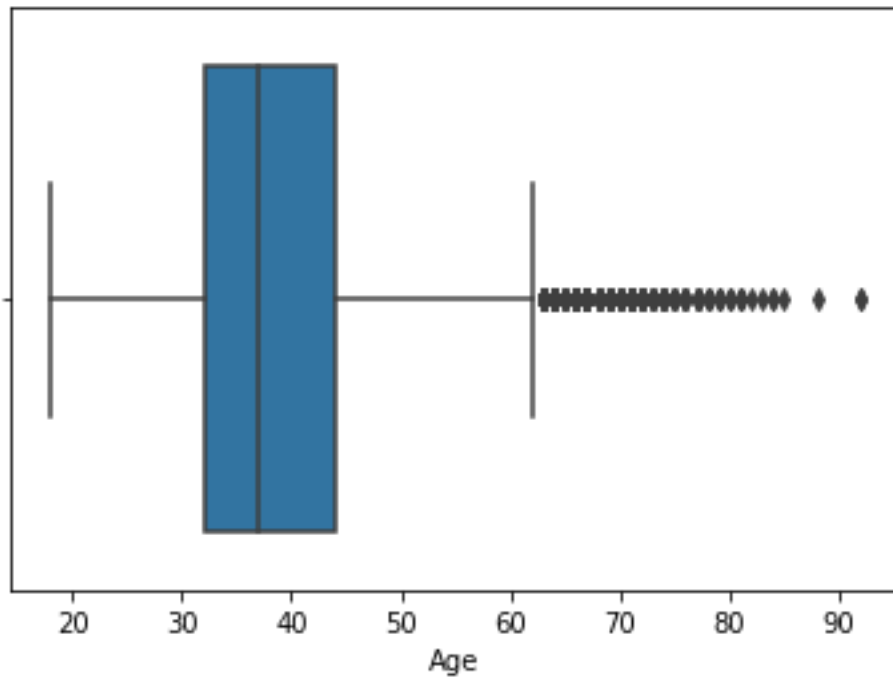
```
sns.boxplot(data['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
FutureWarning
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcfb4080c50>
```

In [19]:

```
qnt = data.quantile(q=[0.25,0.75])
qnt
```

Out[19]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.25	2500.75	156285.28.25	584.0	32.0	3.0	0.00	1.0	0.0	0.0	51002.1100	0.0
0.75	7500.25	157532.33.75	718.0	44.0	7.0	12764.4.24	2.0	1.0	1.0	149388.2475	0.0

In [20]:

```
IQR = qnt.loc[0.75] - qnt.loc[0.25]
IQR
```

Out[20]:

```
RowNumber      4999.5000
CustomerId     124705.5000
CreditScore    134.0000
Age            12.0000
Tenure         4.0000
Balance        127644.2400
NumOfProducts  1.0000
HasCrCard      1.0000
IsActiveMember 1.0000
EstimatedSalary 98386.1375
Exited         0.0000
dtype: float64
```

In [21]:

```
upper_extreme = qnt.loc[0.75]+1.5*IQR
upper_extreme
```

Out[21]:

```
RowNumber      1.499950e+04
CustomerId      1.594029e+07
CreditScore     9.190000e+02
Age             6.200000e+01
Tenure          1.300000e+01
Balance         3.191106e+05
NumOfProducts   3.500000e+00
HasCrCard       2.500000e+00
IsActiveMember  2.500000e+00
EstimatedSalary 2.969675e+05
Exited          0.000000e+00
dtype: float64
```

In [22]:

```
lower_extreme = qnt.loc[0.25]-1.5*IQR
lower_extreme
```

Out[22]:

```
RowNumber      -4.998500e+03
CustomerId      1.544147e+07
CreditScore     3.830000e+02
Age             1.400000e+01
Tenure          -3.000000e+00
Balance         -1.914664e+05
NumOfProducts   -5.000000e-01
HasCrCard       -1.500000e+00
IsActiveMember  -1.500000e+00
EstimatedSalary -9.657710e+04
Exited          0.000000e+00
dtype: float64
```

In [23]:

```
df2 = data[(data['Age']<upper_extreme['Age']) &
            (data['Age']>lower_extreme['Age'])]
df2.shape
```

Out[23]:

```
(10000, 14)
```

In [24]:

```
df2.shape
```

Out[24]:

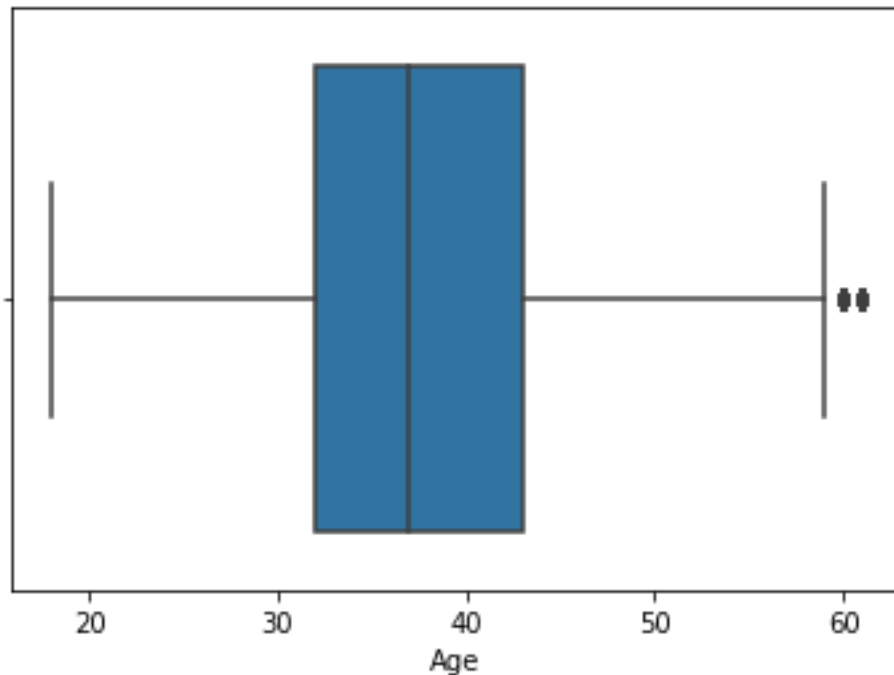
```
(9589, 14)
```

In [25]:

```
sns.boxplot(df2['Age'])
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation
FutureWarning
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcfb593ecd0>
```



7. Check the Categorical columns and perform Encoding

In [26]:

```
le = LabelEncoder()
df2['Geography'] = le.fit_transform(df2['Geography'])
df2['Gender'] = le.fit_transform(df2['Gender'])
df2.head()
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

This is separate from the ipykernel package so we can avoid doing imports until

Out[26]:

	Row Num ber	Cust omer Id	Sur na me	Cred itSco re	Geo grap hy	Ge nd er	A g e	Te nu re	Bal anc e	NumO fProdu cts	Has CrC ard	IsActiv eMemb er	Estima tedSala ry	Ex ite d
0	1	1563 4602	Hargrave	619	0	0	42	2	0.00	1	1	1	101348.88	1
1	2	1564 7311	Hill	608	2	0	41	1	838 07.86	1	0	1	112542.58	0
2	3	1561 9304	Onio	502	0	0	42	8	159 660.80	3	1	0	113931.57	1
3	4	1570 1354	Boni	699	0	0	39	1	0.00	2	0	0	93826.63	0
4	5	1573 7888	Mitchell	850	2	0	43	2	125 510.82	1	1	1	79084.10	0

In [26]:

8. Split the data into dependent and independent variables

In [27]:

```
x = df.iloc[:, :-1].values
x
```

Out[27]:

```
array([[1, 15634602, 'Hargrave', ..., 1, 1, 101348.88],
       [2, 15647311, 'Hill', ..., 0, 1, 112542.58],
       [3, 15619304, 'Onio', ..., 1, 0, 113931.57],
       ...,
       [9998, 15584532, 'Liu', ..., 0, 1, 42085.58],
       [9999, 15682355, 'Sabbatini', ..., 1, 0, 92888.52],
       [10000, 15628319, 'Walker', ..., 1, 0, 38190.78]], dtype=object)
```

In [28]:

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

Out[28]:

```
array([1, 0, 1, ..., 1, 1, 0])
```

9. Scale the independent variables

In [29]:

```
scaler = MinMaxScaler()
df[['CustomerId']] = scaler.fit_transform(df[['CustomerId']])
df
```

														Out[29]:	
	Row Num ber	Cust omer Id	Sur na me	Cred itSco re	Geo grap hy	Ge nd er	A g e	Te nu re	Bal anc e	NumO fProdu cts	Has CrC ard	IsActiv eMem ber	Estima tedSal ary	Ex ite d	
0	1	0.275 616	Har gra ve	619	Fran ce	Fe ma le	4 2	2	0.00	1	1	1	101348 .88	1	
1	2	0.326 454	Hill	608	Spai n	Fe ma le	4 1	1	838 07.8 6	1	0	1	112542 .58	0	
2	3	0.214 421	Oni o	502	Fran ce	Fe ma le	4 2	8	159 660. 80	3	1	0	113931 .57	1	
3	4	0.542 636	Bon i	699	Fran ce	Fe ma le	3 9	1	0.00	2	0	0	93826. 63	0	
4	5	0.688 778	Mit chel l	850	Spai n	Fe ma le	4 3	2	125 510. 82	1	1	1	79084. 10	0	
...	
9995	9996	0.162 119	Obi jiak u	771	Fran ce	Ma le	3 9	5	0.00	2	1	0	96270. 64	0	
9996	9997	0.016 765	Joh nsto ne	516	Fran ce	Ma le	3 5	10	573 69.6 1	1	1	1	101699 .77	0	
9997	9998	0.075 327	Liu	709	Fran ce	Fe ma le	3 6	7	0.00	1	0	1	42085. 58	1	
9998	9999	0.466 637	Sab bati ni	772	Ger man y	Ma le	4 2	3	750 75.3 1	2	1	0	92888. 52	1	

	Row Number	Customer Id	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9999	1000	0.250483	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

10. Split the data into training and testing

In [30]:

```
train_size=0.8
X = df.drop(columns=['Tenure']).copy()
Y = df['Tenure']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,train_size=0.8)
print(X_train)
print(Y_train)
print(X_test)
print(Y_test)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
\						
3893	3894	0.396421	Chinweuba	543	France	Female
3109	3110	0.680738	Law	850	Germany	Female
603	604	0.106749	Burke	566	France	Male
904	905	0.915556	Ch'en	599	France	Male
3469	3470	0.361596	Cumbrae-Stewart	679	Spain	Female
...
5935	5936	0.849589	Stevenson	544	Spain	Male
6210	6211	0.105989	Simmons	522	Spain	Male
7570	7571	0.905012	Harker	697	France	Male
6644	6645	0.008768	Lei	556	Germany	Male
8260	8261	0.248527	Nikitina	640	Germany	Female

	Age	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
3893	42	0.00	2	0	0	
3109	47	134381.52	1	0	0	
603	30	0.00	1	1	0	
904	58	0.00	1	0	0	
3469	26	76554.06	1	1	1	
...	
5935	37	0.00	2	0	0	
6210	30	0.00	2	1	0	
7570	32	175464.85	3	1	0	
6644	33	124213.36	2	1	0	
8260	30	32197.64	1	0	1	

	EstimatedSalary	Exited
3893	101905.34	0
3109	26812.89	1
603	54926.51	1
904	176407.15	1
3469	184800.27	0

```

...
5935      135067.02      0
6210      145490.85      0
7570      116442.42      1
6644        62627.55      0
8260      141446.01      0

```

[8000 rows x 13 columns]

```

3893      5
3109     10
603       5
904       4
3469      3

```

```

..
5935      2
6210      3
7570      7
6644      3
8260      5

```

Name: Tenure, Length: 8000, dtype: int64

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
\							
1041	1042	0.506486	Craig	722	France	Male	30
5208	5209	0.396161	Lei	779	Spain	Female	38
3628	3629	0.030473	Azubuike	493	Germany	Female	35
4635	4636	0.781326	Long	619	France	Female	33
8221	8222	0.834665	Robertson	443	Germany	Male	59
...
449	450	0.369888	Cook	778	Spain	Female	47
4162	4163	0.431551	Bell	652	France	Female	74
9585	9586	0.179432	McCarthy	695	Spain	Female	35
4494	4495	0.605775	Watson	850	Spain	Male	31
7207	7208	0.021157	Begley	520	Spain	Female	30

	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
\					
1041	0.00	2	1	0	166376.54
5208	0.00	2	1	1	138542.87
3628	178317.60	1	0	0	197428.64
4635	167733.51	2	1	1	65222.48
8221	110939.30	1	1	0	72846.58
...
449	127299.34	2	1	0	124694.99
4162	0.00	2	1	1	937.15
9585	79858.13	2	1	1	127977.66
4494	82613.56	2	1	0	149170.92
7207	145222.99	2	0	0	145160.96

	Exited
1041	0
5208	0
3628	0
4635	0
8221	1
...	...
449	0
4162	0

9585	0
4494	0
7207	0

[2000 rows x 13 columns]

1041	5
------	---

5208	7
------	---

3628	8
------	---

4635	2
------	---

8221	4
------	---

..	
----	--

449	6
-----	---

4162	5
------	---

9585	7
------	---

4494	6
------	---

7207	4
------	---

Name: Tenure, Length: 2000, dtype: int64