In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
```

loading the dataset

In [2]:
```python
df=pd.read_csv('Churn_Modelling.csv')
df.head()
```

Out[2]:

	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 [3]:
```python
df.shape
```

Out[3]:
```
(10000, 14)
```

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

In [5]:
```python
df.isnull().any()
```

Out[5]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

In [6]:
```python
df.describe()
```

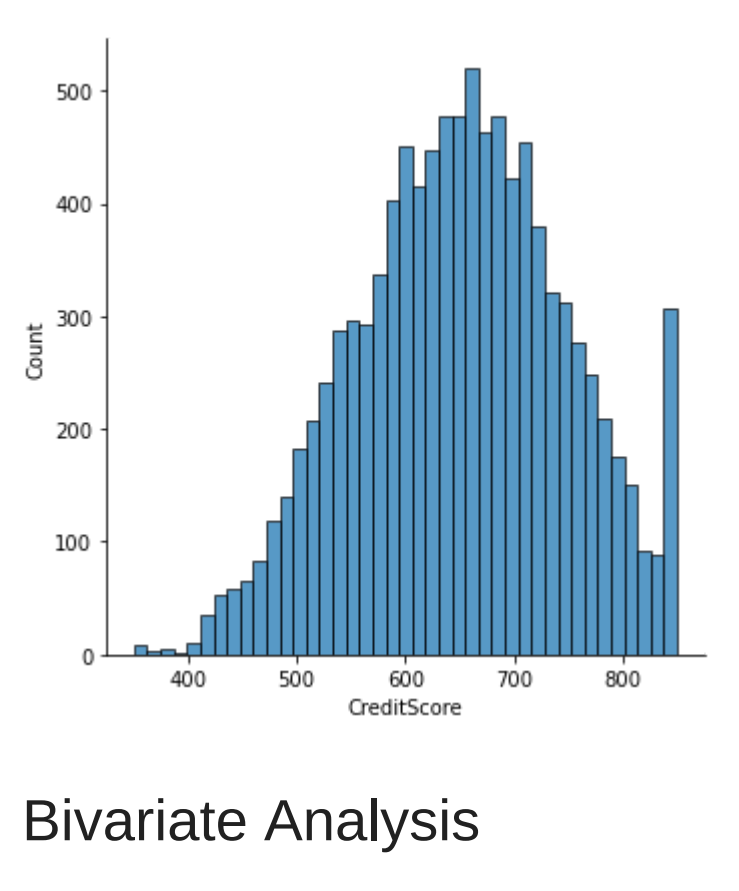
Out[6]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

Univariate anyalysis

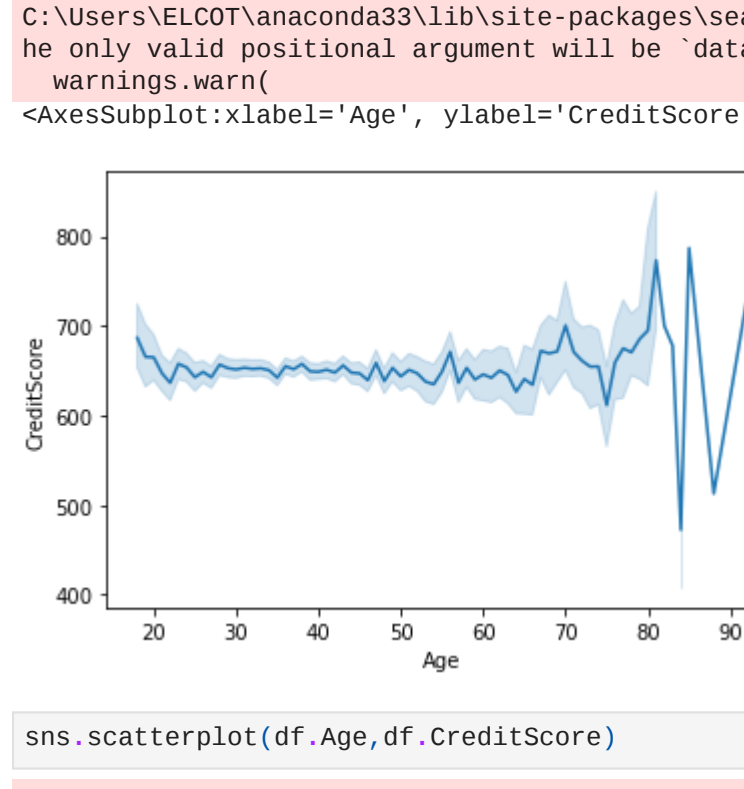
In [7]:
```python
sns.displot(df.Age)
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x2311f298d00>
```



In [8]:
```python
sns.displot(df.CreditScore)
```

Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x2311f37ca30>
```

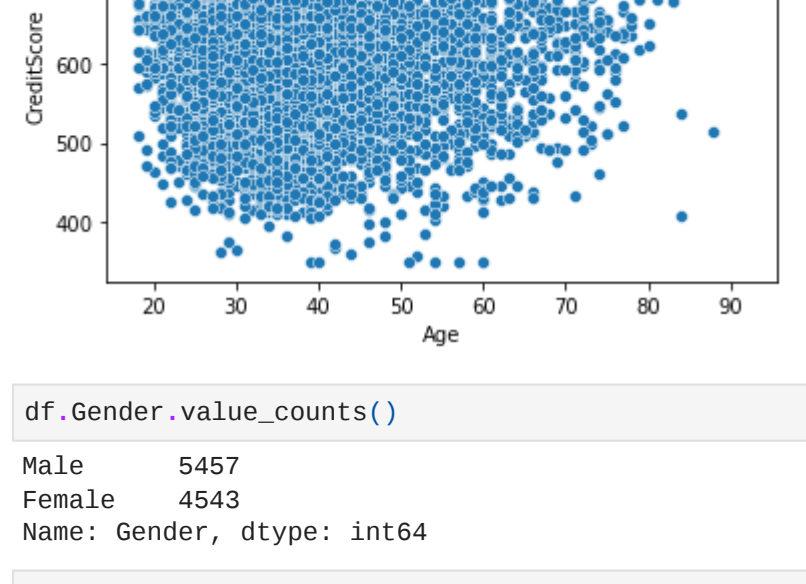


Bivariate Analysis

In [9]:
```python
sns.lineplot(df.Age,df.CreditScore)
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

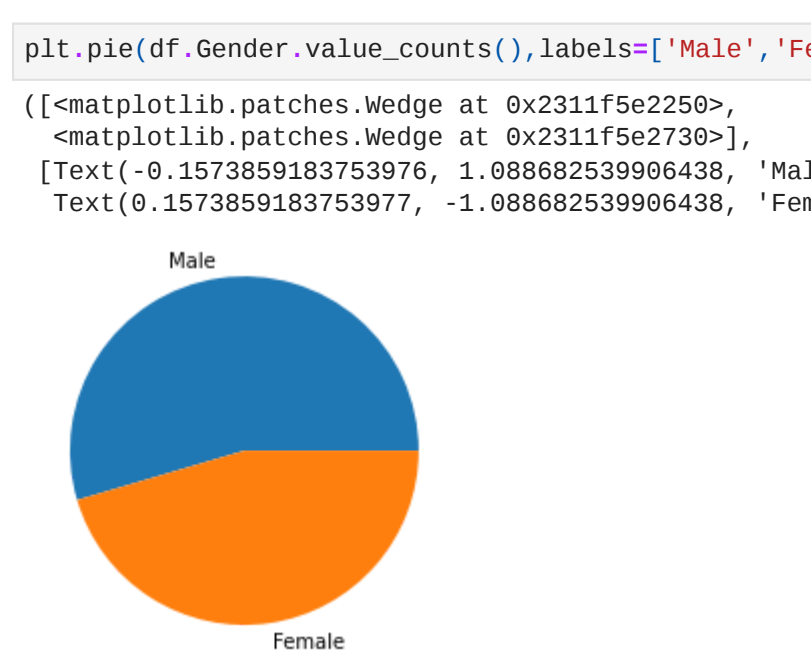
Out[9]:
```
<AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```



In [10]:
```python
sns.scatterplot(df.Age,df.CreditScore)
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[10]:
```
<AxesSubplot:xlabel='Age', ylabel='CreditScore'>
```



In [11]:
```python
df.Gender.value_counts()
```

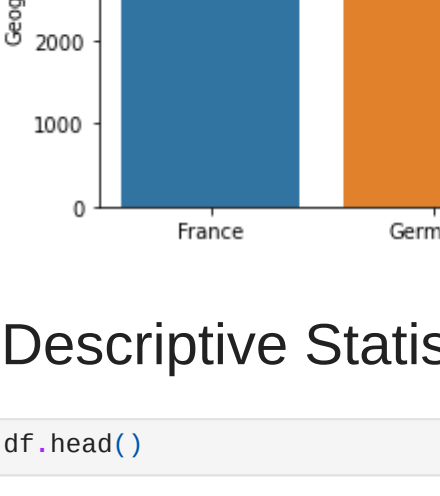
Out[11]:
```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

In [12]:
```python
df.Geography.value_counts()
```

Out[12]:
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

In [13]:
```python
plt.pie(df.Gender.value_counts(),labels=['Male','Female'])
```

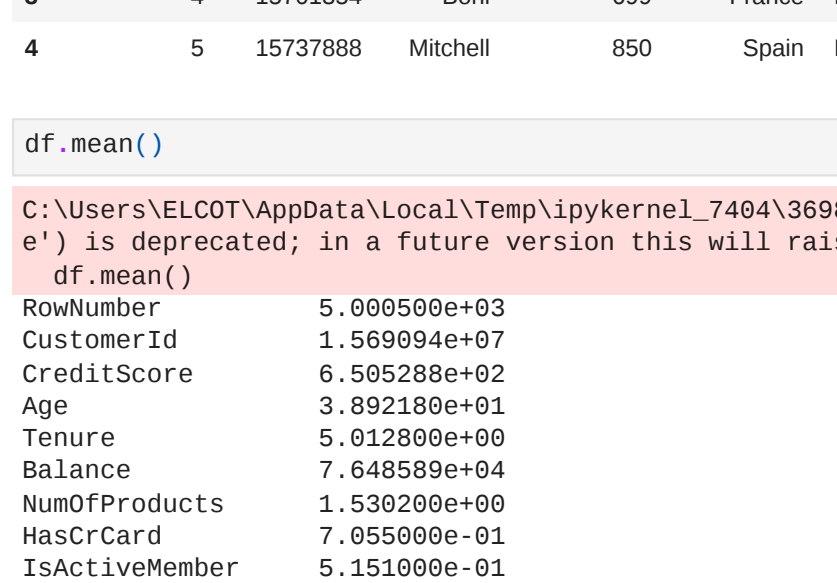
Out[13]:
```
([<matplotlib.patches.Wedge at 0x2311f5e2250>,
  <matplotlib.patches.Wedge at 0x2311f5e2730>],
 [Text(-0.1573859183753976, 1.088682539906438, 'Male'),
  Text(0.1573859183753977, -1.088682539906438, 'Female')])
```



In [14]:
```python
sns.barplot(df.Geography.value_counts().index,df.Geography.value_counts())
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[14]:
```
<AxesSubplot:ylabel='Geography'>
```



Descriptive Statistics

In [15]:
```python
df.head()
```

Out[15]:

	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 [16]:
```python
df.mean()
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_7404\3698961737.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=Non
e') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.mean()
```

Out[16]:
```
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 [17]:
```python
df.median()
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_7404\530051474.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=Non
e') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.median()
```

Out[17]:
```
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 [18]:
```python
df.mode()
```

Out[18]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.92	0.0
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10000 rows × 14 columns

Handle missing values

In [19]:
```python
df2=df.fillna(value=0)
```

In [20]:
```python
df2
```

Out[20]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

Find the outliers and replace the outliers

In [21]:
```python
sns.boxplot(df.CreditScore)
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: 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.
  warnings.warn(
```

Out[21]:
```
<AxesSubplot:xlabel='CreditScore'>
```

In [22]:
```python
q1=df.CreditScore.quantile(0.25)
q3=df.CreditScore.quantile(0.75)
```

In [23]:
```python
IQR=q3-q1
```

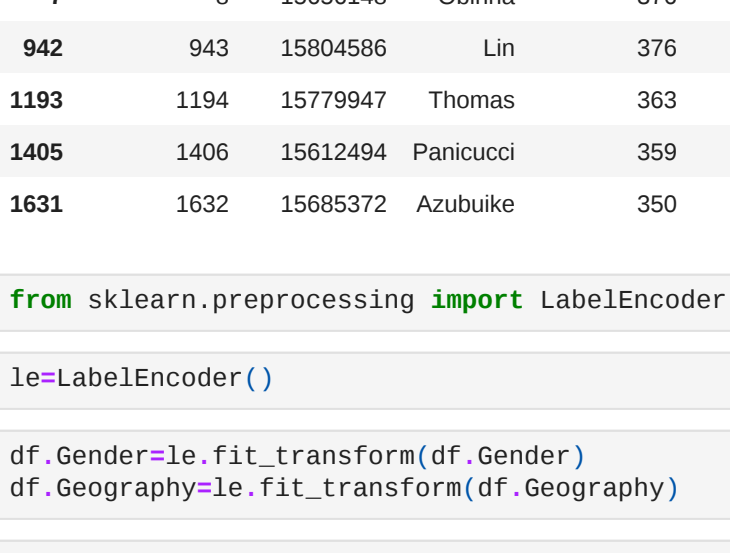
In [24]:
```python
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

In [25]:
```python
df=df[df.CreditScore<lower_limit]
```

In [26]:
```python
sns.boxplot(df.CreditScore)
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: 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.
  warnings.warn(
```

Out[26]:
```
<AxesSubplot:xlabel='CreditScore'>
```



Check for categorical column and perform encoding

In [27]:
```python
df.shape
```

Out[27]:
```
(15, 14)
```

In [28]:
```python
df.head()
```

Out[28]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
942	943	15804586	Lin	376	France	Female	46	6	0.00	1	1	0	157333.69	1
1193	1194	15779947	Thomas	363	Spain	Female	28	6	146098.43	3	1	0	100615.14	1
1405	1406	15612494	Panicucci	359	France	Female	44	6	128747.69	1	1	0	146955.71	1
1631	1632	15685372	Azubuike	350	Spain	Male	54	1	152677.48	1	1	1	191973.49	1

In [29]:
```python
from sklearn.preprocessing import LabelEncoder
```

In [30]:
```python
le=LabelEncoder()
```

In [31]:
```python
df.Gender=le.fit_transform(df.Gender)
df.Geography=le.fit_transform(df.Geography)
```

In [32]:
```python
df.head()
```

Out[32]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	8	15656148	Obinna	376	1	0	29	4	115046.74	4	1	0	119346.88	1
942	943	15804586	Lin	376	0	0	46	6	0.00	1	1	0	157333.69	1
1193	1194	15779947	Thomas	363	2	0	28	6	146098.43	3	1	0	100615.14	1
1405	1406	15612494	Panicucci	359	0	0	44	6	128747.69	1	1	0	146955.71	1
1631	1632	15685372	Azubuike	350	2	1	54	1	152677.48	1	1	1	191973.49	1

Split the data into independent and dependent variables

In [33]:
```python
df.head()
```

Out[33]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	8	15656148	Obinna	376	1	0	29	4	115046.74	4	1	0	119346.88	1
942	943	15804586	Lin	376	0	0	46	6	0.00	1	1	0	157333.69	1
1193	1194	15779947	Thomas	363	2	0	28	6	146098.43	3	1	0	100615.14	1
1405	1406	15612494	Panicucci	359	0	0	44	6	128747.69	1	1	0	146955.71	1
1631	1632	15685372	Azubuike	350	2	1	54	1	152677.48	1	1	1	191973.49	1

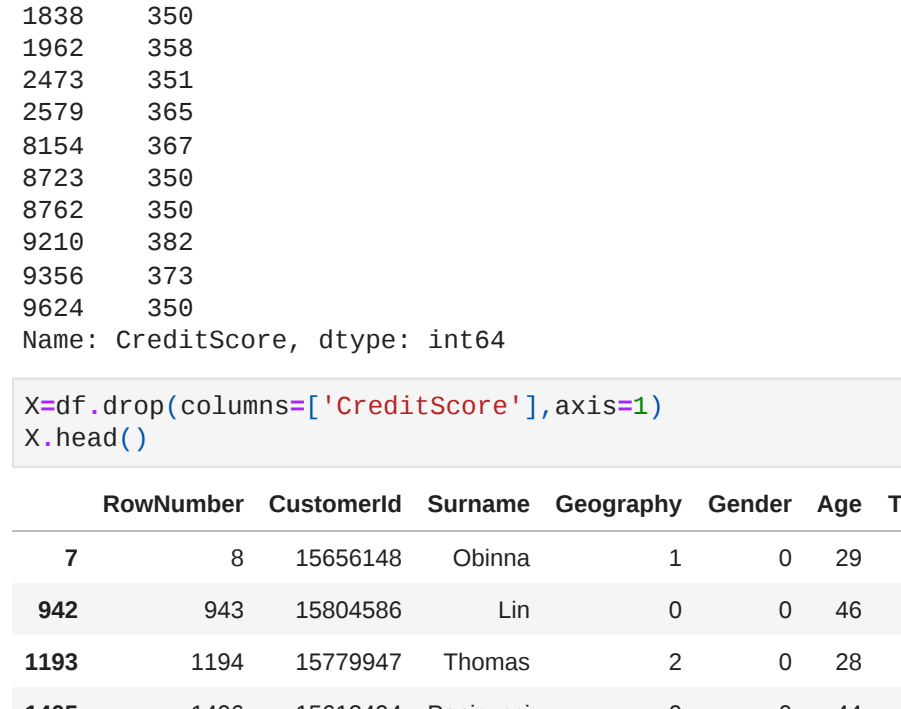
In [34]:
```python
df.corr()
```

Out[34]:

	RowNumber	CustomerId	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
RowNumber	1.000000	0.389145	-0.017874	-0.256838	0.355715	0.186891	0.002614	-0.625342	-0.529059	-0.082485	0.453133	0.034596	NaN
CustomerId	0.389145	1.000000	0.168988	0.086515	0.224575	0.137477	0.297238	-0.633616	-0.103019	-0.234804	0.141598	0.090157	NaN
CreditScore	-0.017874	0.168988	1.000000	0.165990	0.093334	-0.561783	0.060630	-0.361950	0.186189	0.404974	-0.183121	-0.126241	NaN
Geography	-0.256838	0.086515	0.165990	1.000000	0.229566	-0.221627	-0.170676	0.471992	0.369959	0.199083	0.047088	0.288301	NaN
Gender	0.355715	0.224575	0.093334	0.229566	1.000000	-0.127947	-0.188167	-0.251297	-0.381730	-0.026207	0.342475	-0.067618	NaN
Age	0.186891	0.137477	-0.561783	-0.221627	-0.127947	1.000000	0.247222	-0.148749	-0.433615	-0.248131	0.118568	0.373699	NaN
Tenure	0.002614	0.297238	0.060630	-0.170676	-0.188167	0.247222	1.000000	-0.208551	0.164001	0.317370	-0.250301	-0.215267	NaN
Balance	-0.625342	-0.633616	-0.361950	0.471992	-0.251297	-0.148749	-0.208551	1.000000	0.380584	0.193739	-0.192207	0.163385	NaN
NumOfProducts	-0.529059	-0.103019	0.186189	0.369959	-0.381730	-0.433615	0.164001	0.380584	1.000000	0.013664	-0.336117	-0.279123	NaN
HasCrCard	-0.082485	-0.234804	0.404974	0.199083	-0.026207	-0.248131	0.317370	0.193739	0.013664	1.000000	0.236525	0.220940	NaN
IsActiveMember	0.453133	0.141598	-0.183121	0.047088	0.342475	0.118568	-0.250301	-0.192207	-0.336117	0.236525	1.000000	0.514630	NaN
EstimatedSalary	0.034596	0.090157	-0.126241	0.288301	-0.067618	0.373699	-0.215267	0.163385	-0.279123	0.220940	0.514630	1.000000	NaN
Exited	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [35]:
```python
sns.heatmap(df.corr(),annot=True)
```

Out[35]:
```
<AxesSubplot:>
```



In [36]:
```python
y=df['CreditScore']
y
```

Out[36]:
```
7       376
942     376
1193    363
1405    359
1631    350
1838    350
1962    358
2473    351
2579    365
8154    367
8723    350
8762    350
9210    382
9356    373
9624    350
Name: CreditScore, dtype: int64
```

In [37]:
```python
X=df.drop(columns=['CreditScore'],axis=1)
X.head()
```

Out[37]:

	RowNumber	CustomerId	Surname	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	8	15656148	Obinna	1	0	29	4	115046.74	4	1	0	119346.88	1
942	943	15804586	Lin	0	0	46	6	0.00	1	1	0	157333.69	1
1193	1194	15779947	Thomas	2	0	28	6	146098.43	3	1	0	100615.14	1
1405	1406	15612494	Panicucci	0	0	44	6	128747.69	1	1	0	146955.71	1
1631	1632	15685372	Azubuike	2	1	54	1	152677.48	1	1	1	191973.49	1

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: