# Data Visualization and Pre-processing

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

## 2. Load the dataset

```
#1. Loading DataSet
df=pd.read_csv('C:/Users/Pavi/Downloads/Churn_Modelling.csv')
df
```

| | 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 |
| **999** | 9997 | 15569892 | Johnsto | 516 | France | Male | 35 | 10 | 57369.6 | 1 | 1 | 1 | 101699.77 | 0 |

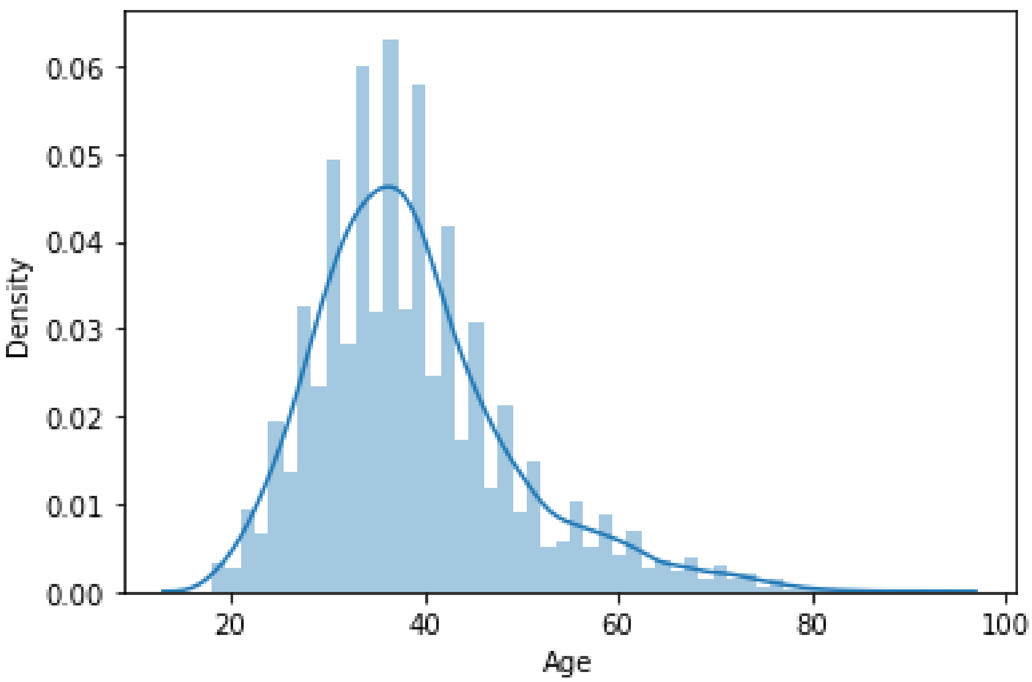
| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **6** | | | ne | | | | | | 1 | | | | | |
| **9997** | 9998 | 15584532 | 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

# 2. Perform Univariate, Bi - Variate and Multi - Variate Analysis Visualizations

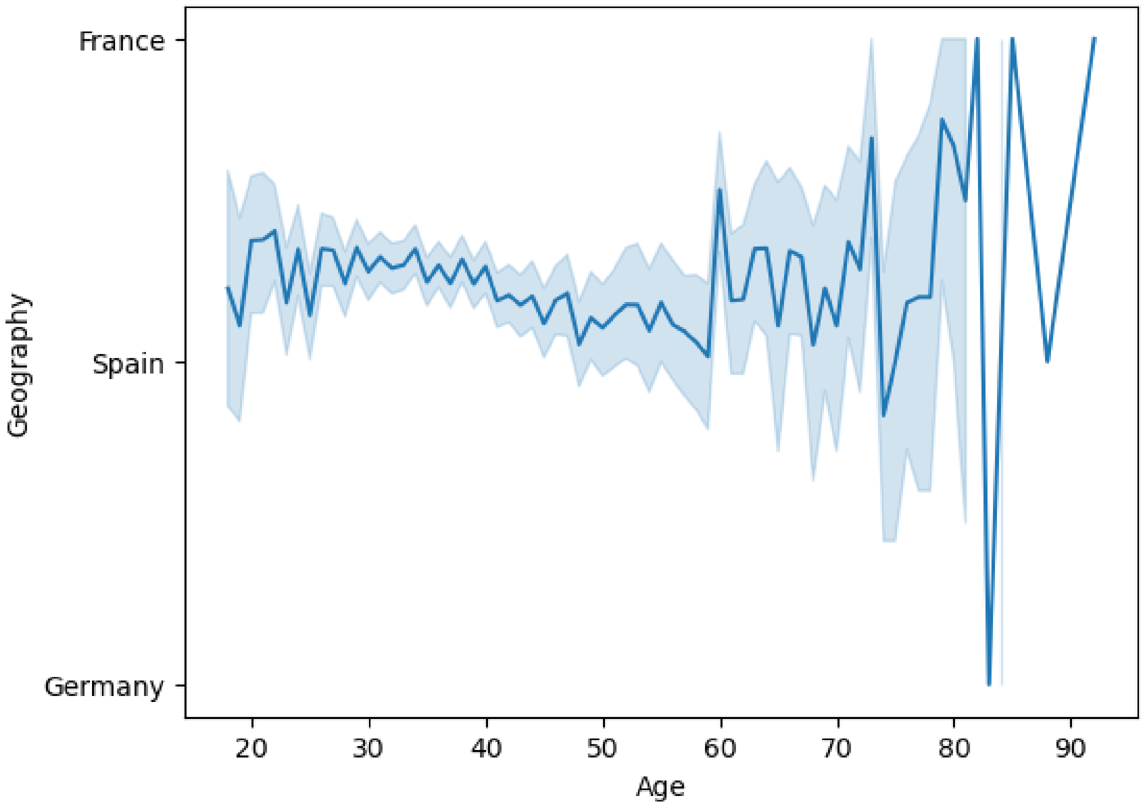
```
sns.distplot(df.Age) #Univariate Analysis
```

<AxesSubplot:xlabel='Age', ylabel='Density'>



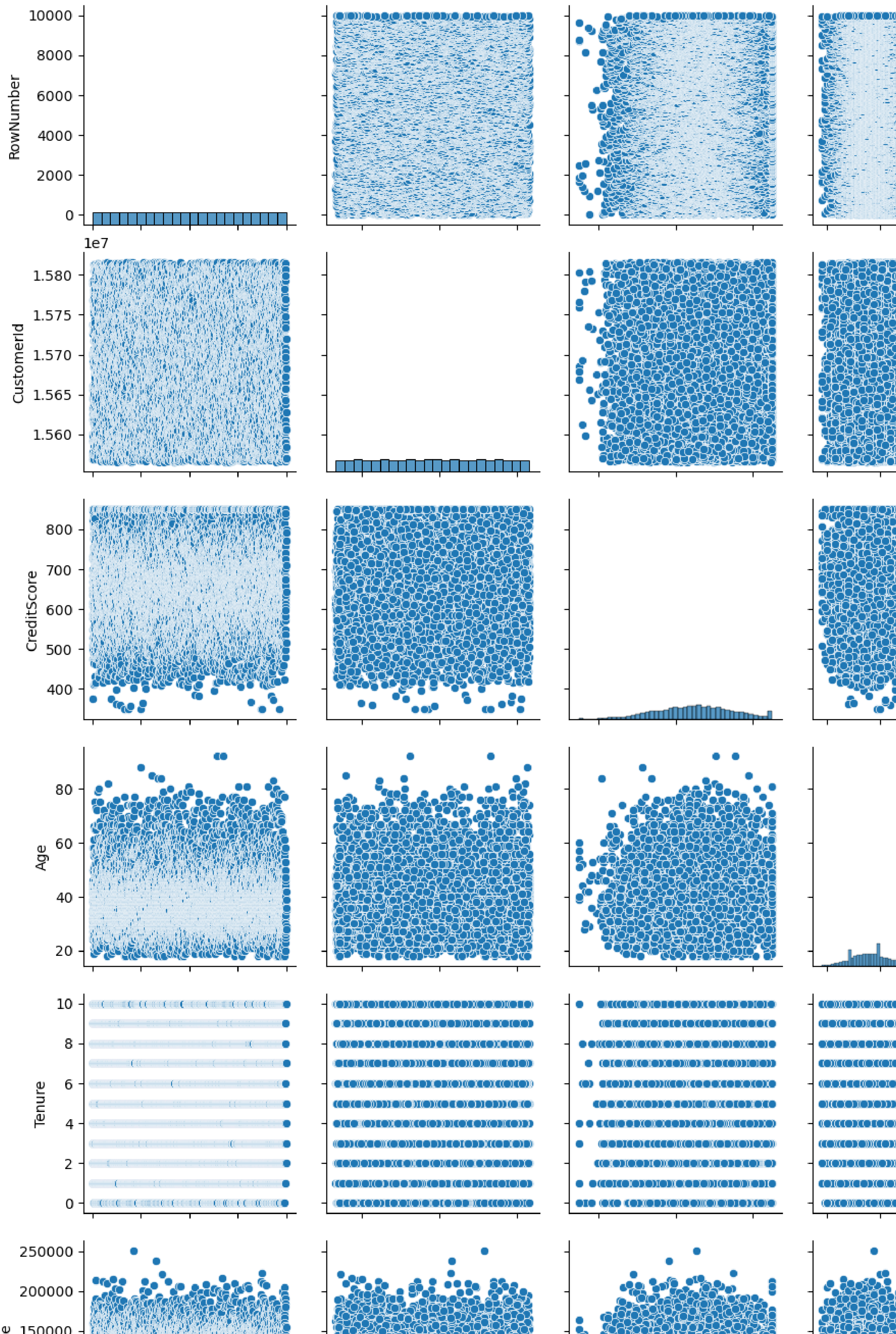
```
sns.lineplot(df.Age,df.Geography) # Bi - Variate Analysis
```

<AxesSubplot:xlabel='Age', ylabel='Geography'>




```
sns.pairplot(df) #Multi - Variate Analysis
```

<seaborn.axisgrid.PairGrid at 0x2055b3070a0>



RowNumber
CustomerId
CreditScore
Age
Tenure
1e7
10000
8000
6000
4000
2000
0
1.580
1.575
1.570
1.565
1.560
800
700
600
500
400
80
60
40
20
10
8
6
4
2
0
250000
200000
150000

# 4. Perform descriptive statistics on the dataset.

df.describe()

| | 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 |

# 5. Handle the Missing values.

df.isnull().any()

```
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
```

```
df['CreditScore'].fillna(df['CreditScore'].mean(),inplace=True)
df
```

| 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 |

| **RowNumber** | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 |

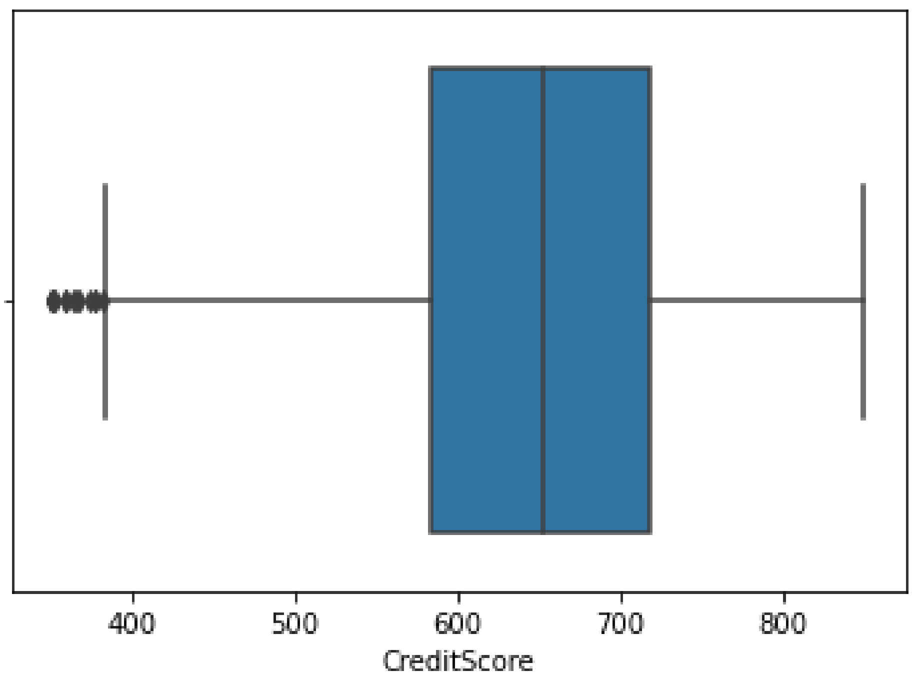
10000 rows × 14 columns

df['Gender'].fillna(df['Gender'].mode(),inplace=**True**)

# 5. outlier detection

sns.boxplot(df.CreditScore)

<AxesSubplot:xlabel='CreditScore'>




# outlier removal- IQR method

Q1= df.CreditScore.quantile(0.25)
Q3=df.CreditScore.quantile(0.75)
IQR=Q3-Q1

IQR

134.0

upper_limit =Q3 + 1.5*IQR

lower_limit =Q1 - 1.5*IQR

upper_limit

919.0

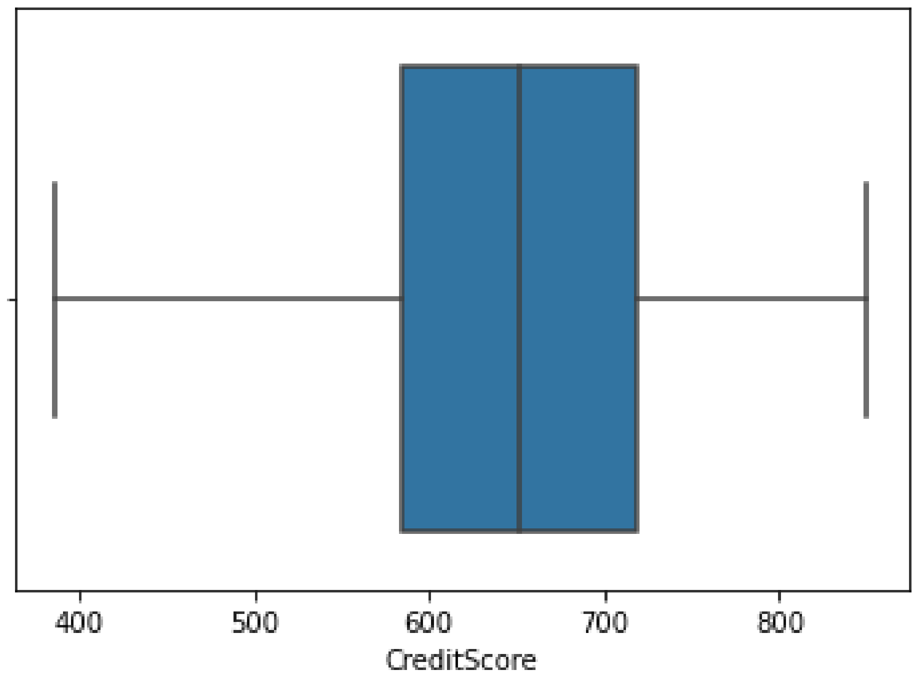
lower_limit

383.0

df=df[df.CreditScore>lower_limit]

sns.boxplot(df.CreditScore)

<AxesSubplot:xlabel='CreditScore'>




df.shape

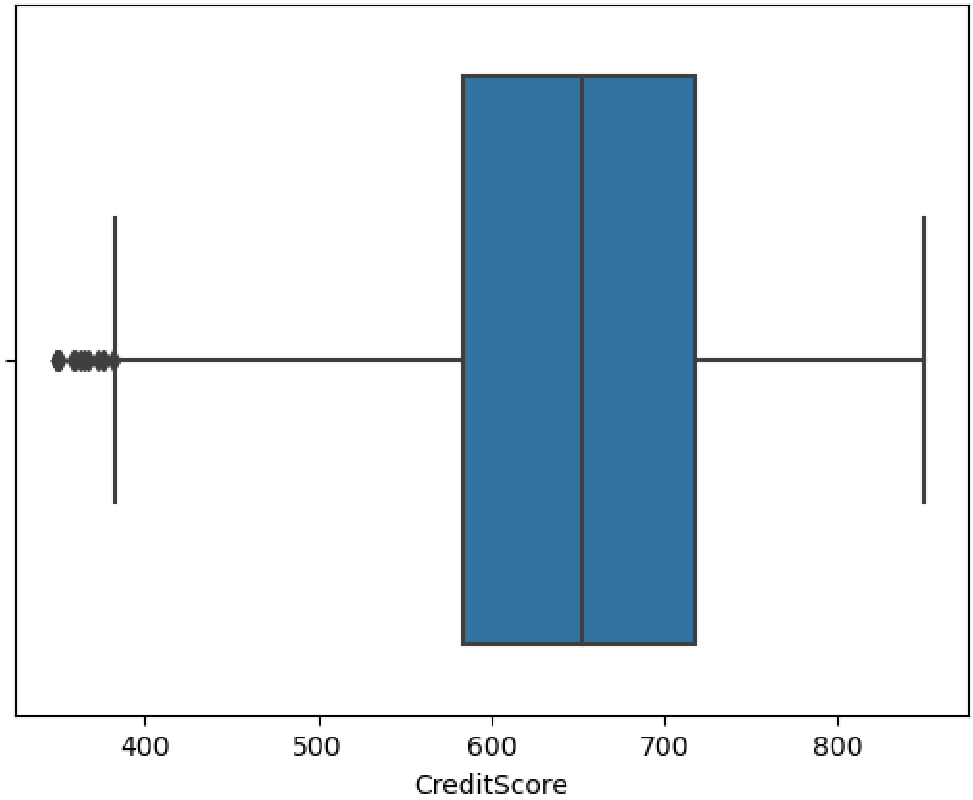
Out[11]:

(9984, 14)

## Outlier replacement using median

In [23]:

sns.boxplot(df.CreditScore)




```
df.shape
```

Out[11]:

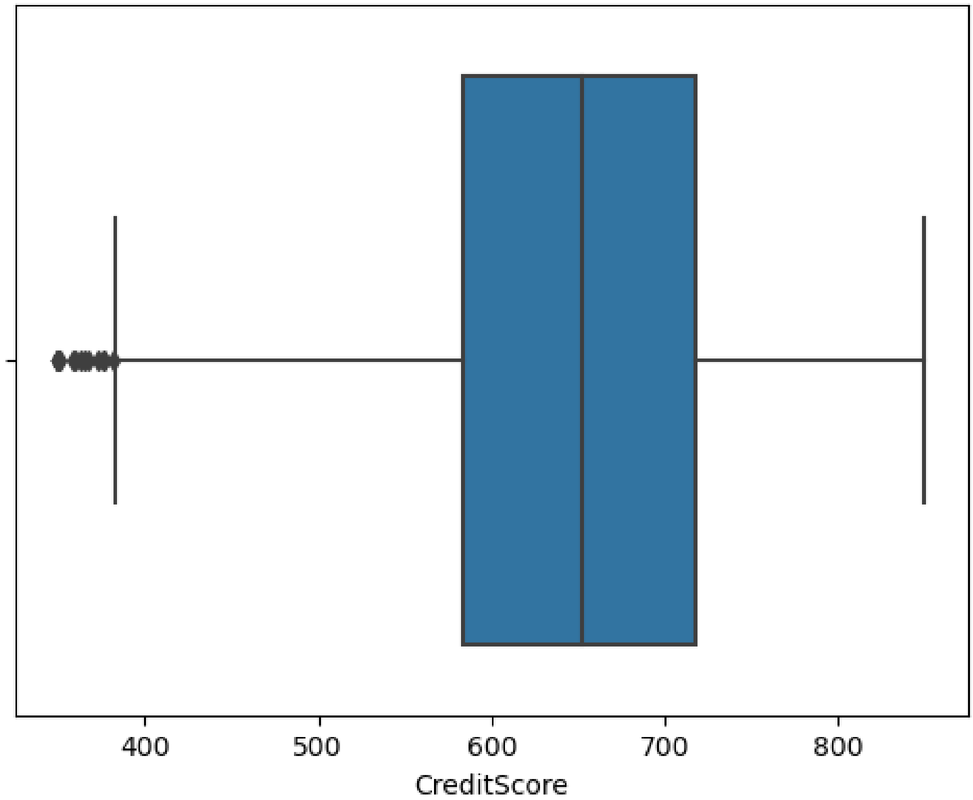
```
(9984, 14)
```

## Outlier replacement using median

In [23]:

```
sns.boxplot(df.CreditScore)
```

```
<AxesSubplot:xlabel='CreditScore'>
```




df.median()

```
RowNumber          5.001500e+03
CustomerId         1.569073e+07
CreditScore        6.520000e+02
Age                3.700000e+01
Tenure             5.000000e+00
Balance            9.717329e+04
NumOfProducts      1.000000e+00
HasCrCard          1.000000e+00
IsActiveMember     1.000000e+00
EstimatedSalary    1.001144e+05
Exited             0.000000e+00
dtype: float64
```

Q1= df.CreditScore.quantile(0.25)
Q3=df.CreditScore.quantile(0.75)
IQR=Q3-Q1
IQR

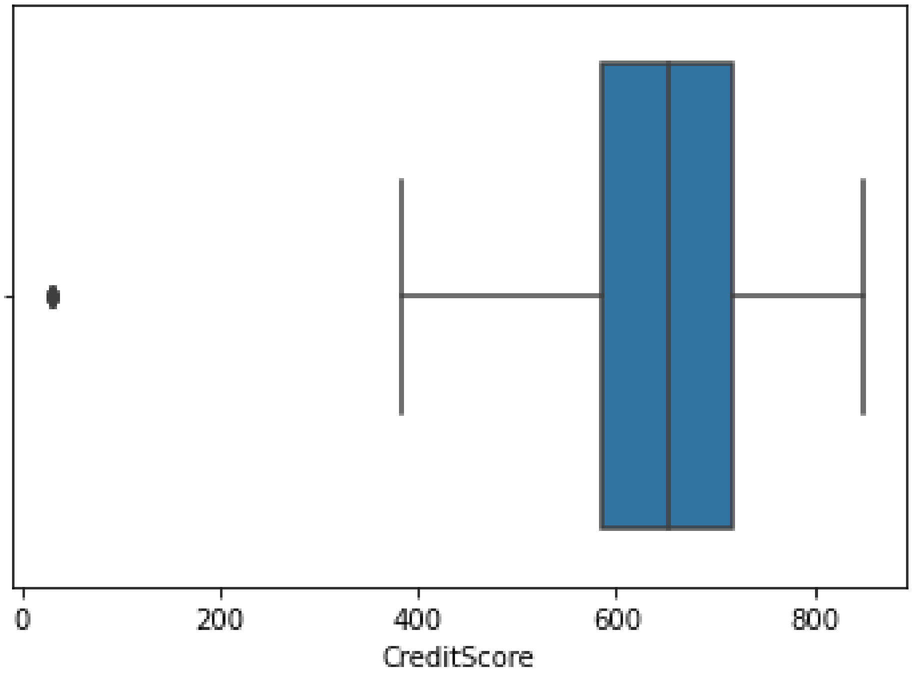
134.0

upper_limit =Q3 + 1.5*IQR
lower_limit =Q1 - 1.5*IQR

df['CreditScore'] = np.where(df['CreditScore']<lower_limit,30,df['CreditScore'])

sns.boxplot(df.CreditScore)

<AxesSubplot:xlabel='CreditScore'>




# outlier removal- Percentile method

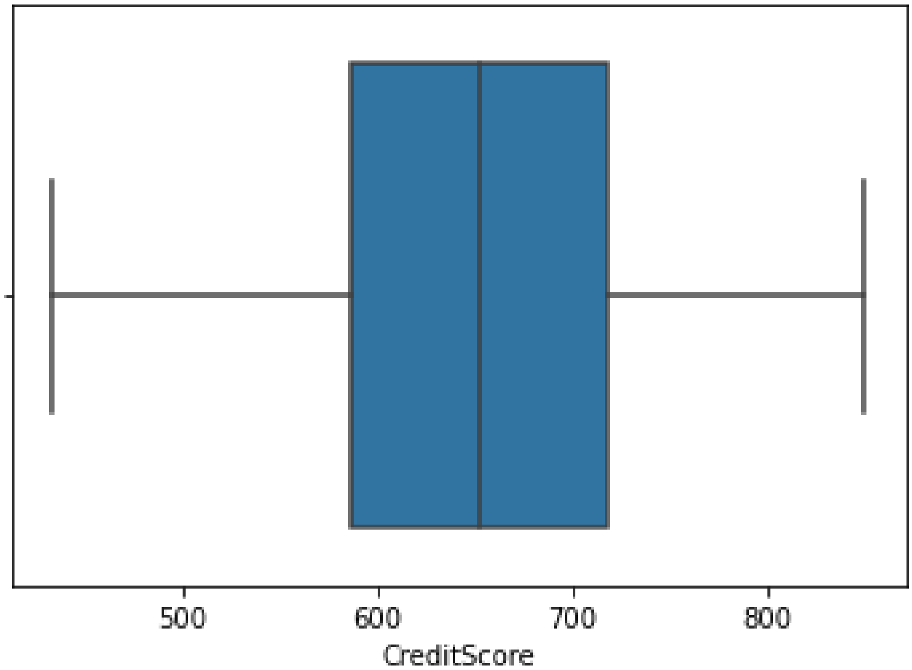
p99= df.CreditScore.quantile(0.01)
p99

432.0
df = df[df.CreditScore>p99]
df.CreditScore

```
0       619
1       608
2       502
3       699
4       850
       ...
9995    771
9996    516
9997    709
9998    772
9999    792
Name: CreditScore, Length: 9896, dtype: int64
```

sns.boxplot(df.CreditScore)

<AxesSubplot:xlabel='CreditScore'>




# outlier removal- z-score

```
from scipy import stats
```

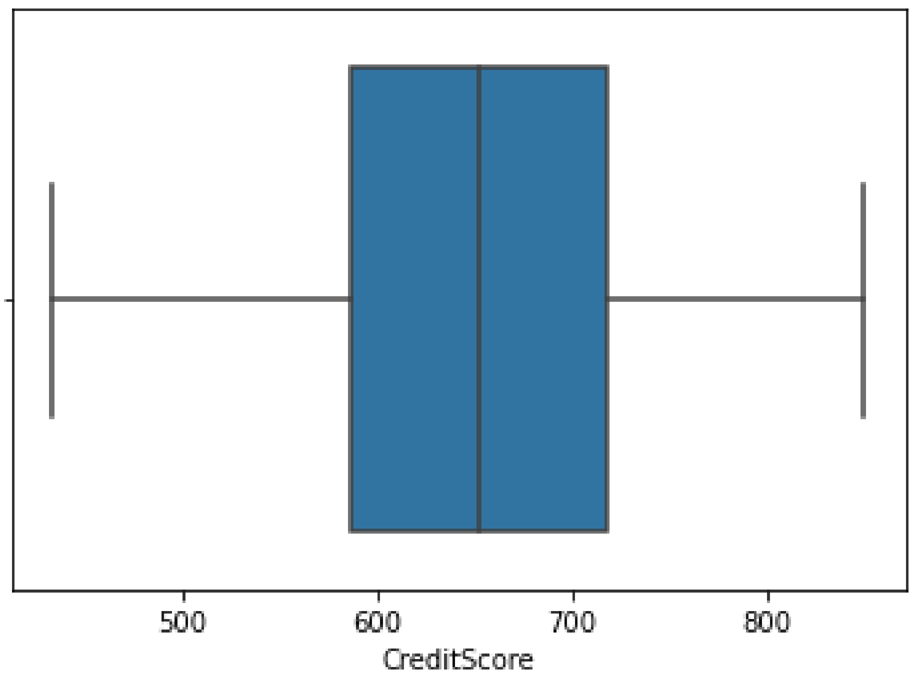
```
CreditScore_zscore = stats.zscore(df.CreditScore)
CreditScore_zscore
```

```
0      -0.362351
1      -0.479437
2      -1.607724
3       0.489187
4       2.096463
          ...
9995    1.255570
9996   -1.458705
9997    0.595629
9998    1.266214
9999    1.479099
Name: CreditScore, Length: 9896, dtype: float64
```

```
df_z = df[np.abs(CreditScore_zscore)<=3]
```

```
sns.boxplot(df_z.CreditScore)
```

```
<AxesSubplot:xlabel='CreditScore'>
```




df.head()

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 15634602 | 1115 | 619 | France | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| **1** | 2 | 15647311 | 1177 | 608 | Spain | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| **2** | 3 | 15619304 | 2040 | 502 | France | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| **3** | 4 | 15701354 | 289 | 699 | France | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| **4** | 5 | 15737888 | 1822 | 850 | Spain | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |

# 7. Check for Categorical columns and perform encoding.

## Encoding Techniques

# Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.Gender = le.fit_transform(df.Gender)
df.Surname = le.fit_transform(df.Surname)
```

df

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 15634602 | 1115 | 619 | France | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| **1** | 2 | 15647311 | 1177 | 608 | Spain | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| **2** | 3 | 15619304 | 2040 | 502 | France | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| **3** | 4 | 15701354 | 289 | 699 | France | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| **4** | 5 | 15737888 | 1822 | 850 | Spain | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | 9996 | 15606229 | 1999 | 771 | France | 1 | 39 | 5 | 0.00 | 2 | 1 | 0 | 96270.64 | 0 |
| **9996** | 9997 | 15569892 | 1336 | 516 | France | 1 | 35 | 10 | 57369.61 | 1 | 1 | 1 | 101699.77 | 0 |
| **9997** | 9998 | 15584532 | 1570 | 709 | France | 0 | 36 | 7 | 0.00 | 1 | 0 | 1 | 42085.58 | 1 |
| **999** | 9999 | 15682355 | 2345 | 772 | Germany | 1 | 42 | 3 | 75075.31 | 2 | 1 | 0 | 92888.52 | 1 |

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **8** | | | | | | | | | | | | | | |
| **9999** | 10000 | 15628319 | 2751 | 792 | France | 0 | 28 | 4 | 130142.79 | 1 | 1 | 0 | 38190.78 | 0 |

10000 rows × 14 columns

# One hot Encoding

```
df_main=pd.get_dummies(df,columns=['Geography'])
df_main.head()
```

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 15634602 | 1115 | 619 | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 | 1 | 0 | 0 |
| **1** | 2 | 15647311 | 1177 | 608 | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 | 0 | 0 | 1 |
| **2** | 3 | 15619304 | 2040 | 502 | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 | 1 | 0 | 0 |
| **3** | 4 | 15701354 | 289 | 699 | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 | 1 | 0 | 0 |
| **4** | 5 | 15737888 | 1822 | 850 | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 | 0 | 0 | 1 |

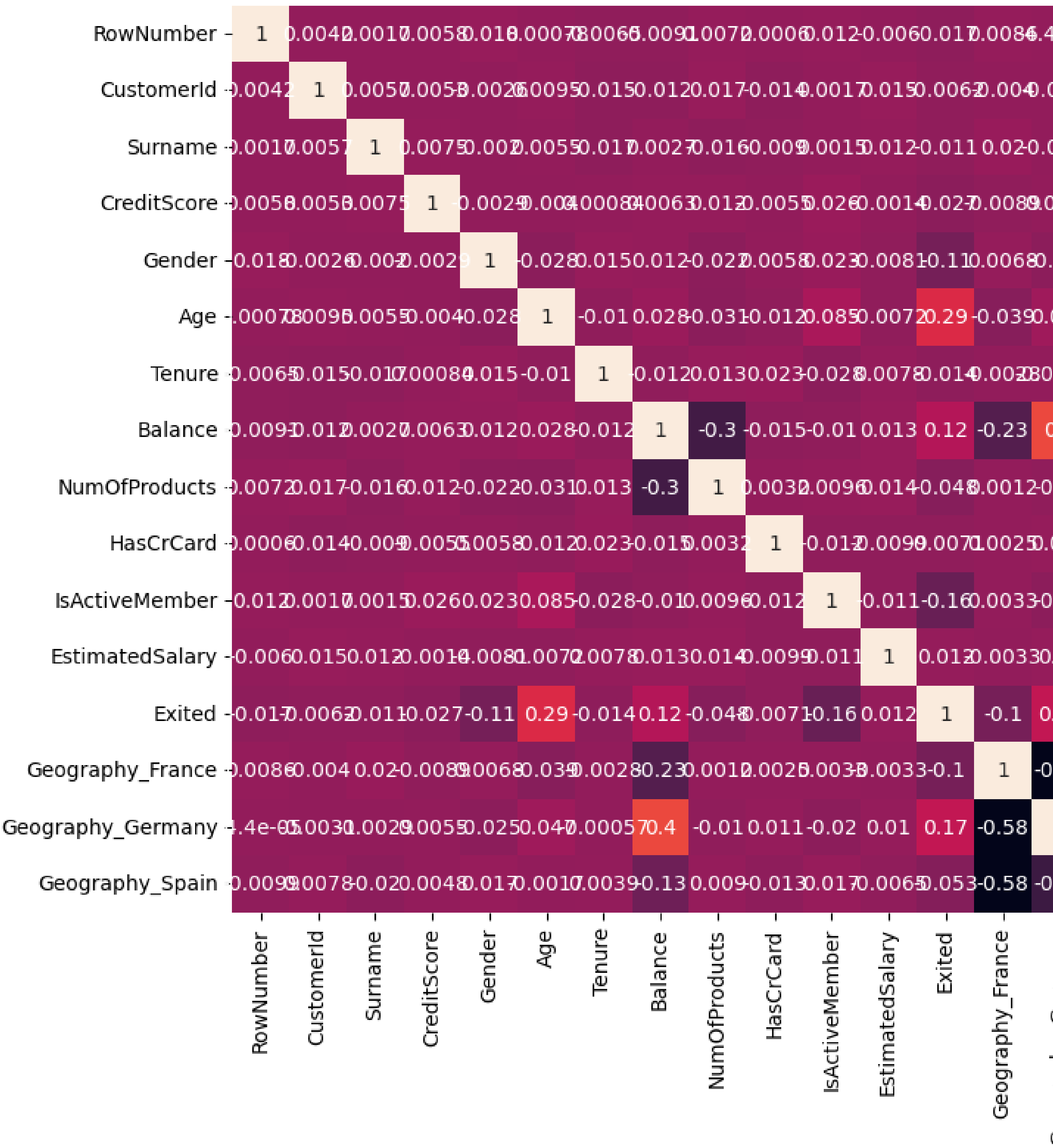
```
df_main.describe()
```

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 10000.000000 | 1.000000e+04 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| **mean** | 5000.500000 | 1.569094e+07 | 1507.774200 | 650.528800 | 0.545700 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 | 0.203700 | 0.501400 | 0.250900 | 0.247700 |
| **std** | 2886.895680 | 7.193619e+04 | 846.204311 | 96.653299 | 0.497932 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 | 0.402769 | 0.500023 | 0.433553 | 0.431698 |
| **min** | 1.000000 | 1.556570e+07 | 0.000000 | 350.000000 | 0.000000 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 2500.750000 | 1.562853e+07 | 773.750000 | 584.000000 | 0.000000 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 51002.110000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 5000.500000 | 1.569074e+07 | 1542.000000 | 652.000000 | 1.000000 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.00000 | 1.000000 | 100193.915000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 |
| **75%** | 7500.250000 | 1.575323e+07 | 2238.250000 | 718.000000 | 1.000000 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.00000 | 1.000000 | 149388.247500 | 0.000000 | 1.000000 | 1.000000 | 0.000000 |
| **max** | 10000.000000 | 1.581569e+07 | 2931.000000 | 850.000000 | 1.000000 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.00000 | 1.000000 | 199992.480000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

# 8. Split the data into dependent and independent variables

```
plt.figure(figsize=(10,8))
sns.heatmap(df_main.corr(),annot=True)
```

<AxesSubplot:>



# X and y split

```
# y target-dependent variable

y=df_main.Balance
y
```

```
0            0.00
1        83807.86
2       159660.80
3            0.00
4       125510.82
          ...
9995         0.00
9996     57369.61
9997         0.00
9998     75075.31
9999    130142.79
Name: Balance, Length: 10000, dtype: float64
```

```
# independent varibles-X

X=df_main.drop(columns=['Balance'],axis=1)
X.head()
```

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 15634602 | 1115 | 619 | 0 | 42 | 2 | 1 | 1 | 1 | 101348.88 | 1 | 1 | 0 | 0 |
| **1** | 2 | 15647311 | 1177 | 608 | 0 | 41 | 1 | 1 | 0 | 1 | 112542.58 | 0 | 0 | 0 | 1 |
| **2** | 3 | 15619304 | 2040 | 502 | 0 | 42 | 8 | 3 | 1 | 0 | 113931.57 | 1 | 1 | 0 | 0 |
| **3** | 4 | 15701354 | 289 | 699 | 0 | 39 | 1 | 2 | 0 | 0 | 93826.63 | 0 | 1 | 0 | 0 |
| **4** | 5 | 15737888 | 1822 | 850 | 0 | 43 | 2 | 1 | 1 | 1 | 79084.10 | 0 | 0 | 0 | 1 |

# 9. Scale the independent variables

```
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()
```

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | -1.731878 | -0.783213 | -0.464183 | -0.326221 | -1.095988 | 0.293517 | -1.041760 | -0.911583 | 0.646092 | 0.970243 | 0.021886 | 1.977165 | 0.997204 | -0.578736 | -0.573809 |
| **1** | -1.731531 | -0.606534 | -0.390911 | -0.440036 | -1.095988 | 0.198164 | -1.387538 | -0.911583 | -1.547768 | 0.970243 | 0.216534 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |
| **2** | -1.731185 | -0.995885 | 0.628988 | -1.536794 | -1.095988 | 0.293517 | 1.032908 | 2.527057 | 0.646092 | -1.030670 | 0.240687 | 1.977165 | 0.997204 | -0.578736 | -0.573809 |
| **3** | -1.730838 | 0.144767 | -1.440356 | 0.501521 | -1.095988 | 0.007457 | -1.387538 | 0.807737 | -1.547768 | -1.030670 | -0.108918 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **4** | -1.730492 | 0.652659 | 0.371354 | 2.063884 | -1.095988 | 0.388871 | -1.041760 | -0.911583 | 0.646092 | 0.970243 | -0.365276 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |

# 10. Split the data into training and testing

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =train_test_split(X_scaled,y, test_size=0.3,random_state=0)
```

X_train

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **7681** | 0.928899 | -0.797032 | -1.475810 | -0.098592 | 0.912419 | -0.564665 | -1.041760 | 0.807737 | 0.646092 | 0.970243 | -0.770218 | 1.977165 | 0.997204 | -0.578736 | -0.573809 |
| **9003** | 1.39655 | 0.71431 | -1.588 | -1.13327 | 0.91241 | 0.00745 | 0.68713 | 0.807 | 0.6460 | -1.030 | -1.395 | -0.505 | 0.9972 | -0.5787 | -0.573 |

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 3 | 4 | 081 | 0 | 9 | 7 | 0 | 737 | 92 | 670 | 767 | 775 | 04 | 36 | 809 |
| **3691** | -0.453278 | 0.963450 | -0.240822 | -0.626278 | -1.095988 | 3.535540 | -0.004426 | -0.911583 | -1.547768 | 0.970243 | -1.499656 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **202** | -1.661903 | -1.250707 | -0.427547 | -1.391939 | 0.912419 | 1.056346 | -0.004426 | -0.911583 | -1.547768 | 0.970243 | 0.800862 | 1.977165 | -1.002804 | -0.578736 | 1.742740 |
| **5625** | 0.216680 | -0.385174 | -1.478173 | -1.474714 | -1.095988 | 2.009882 | 0.687130 | -0.911583 | 0.646092 | 0.970243 | 0.512497 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9225** | 1.463756 | -1.473777 | 1.385344 | -0.584891 | -1.095988 | -0.660018 | -0.350204 | 0.807737 | 0.646092 | 0.970243 | 1.093273 | -0.505775 | -1.002804 | 1.727904 | -0.573809 |
| **4859** | -0.048671 | -0.609314 | 1.530707 | 1.484464 | -1.095988 | -1.613554 | -0.350204 | -0.911583 | 0.646092 | 0.970243 | 0.133249 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |
| **3264** | -0.601195 | -1.620525 | -0.361366 | 0.905045 | 0.912419 | -0.373958 | -0.004426 | 0.807737 | 0.646092 | -1.030670 | 1.414415 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **9845** | 1.678530 | -0.374039 | 0.725896 | -0.626278 | -1.095988 | -0.087897 | 1.378686 | 0.807737 | 0.646092 | 0.970243 | 0.846147 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2732** | -0.785485 | -1.364118 | 1.382981 | -0.284834 | -1.095988 | 0.865639 | -1.387538 | -0.911583 | 0.646092 | -1.030670 | 0.326305 | 1.977165 | -1.002804 | 1.727904 | -0.573809 |

7000 rows × 15 columns

X_train.shape

(7000, 15)
y_train.shape

(7000,)
X_test

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **9394** | 1.522299 | -1.045250 | 1.398344 | -0.553850 | -1.095988 | -0.373958 | 1.032908 | -0.911583 | 0.646092 | 0.970243 | 1.613046 | -0.505775 | -1.002804 | 1.727904 | -0.573809 |
| **898** | -1.420801 | -0.503813 | -0.782089 | -1.319512 | -1.095988 | 0.102810 | -1.041760 | -0.911583 | 0.646092 | -1.030670 | 0.497532 | 1.977165 | 0.997204 | -0.578736 | -0.573809 |
| **2398** | -0.901186 | -0.793292 | 0.412717 | 0.573948 | -1.095988 | 0.293517 | 1.032908 | -0.911583 | 0.646092 | 0.970243 | -0.423561 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |
| **5906** | 0.314021 | 0.760190 | 1.590979 | 1.422384 | 0.912419 | -0.660018 | -0.350204 | -0.911583 | -1.547768 | -1.030670 | -0.186439 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **2343** | -0.920239 | 1.042107 | -0.603636 | 0.573948 | 0.912419 | -0.087897 | -0.004426 | 0.807737 | 0.646092 | 0.970243 | 0.618560 | -0.505775 | -1.002804 | 1.727904 | -0.573809 |

| | RowNumber | CustomerId | Surname | CreditScore | Gender | Age | Tenure | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited | Geography_France | Geography_Germany | Geography_Spain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4004** | -0.344851 | 0.661806 | -0.626091 | -1.247084 | -1.095988 | 0.198164 | -0.350204 | 0.807737 | -1.547768 | 0.970243 | 0.826264 | -0.505775 | -1.002804 | -0.578736 | 1.742740 |
| **7375** | 0.822897 | -0.723866 | -1.266630 | -0.119286 | 0.912419 | 3.630893 | 0.341352 | 0.807737 | 0.646092 | 0.970243 | -0.769654 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **9307** | 1.492162 | -0.146464 | 0.686897 | 0.356666 | 0.912419 | 0.102810 | -1.041760 | 0.807737 | 0.646092 | 0.970243 | 1.170455 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **8394** | 1.175889 | -1.292287 | -1.384811 | 0.429093 | 0.912419 | 2.868064 | 1.724464 | -0.911583 | 0.646092 | 0.970243 | -0.508468 | -0.505775 | 0.997204 | -0.578736 | -0.573809 |
| **5233** | 0.080887 | -1.385388 | 1.117074 | 0.832617 | -1.095988 | 0.960993 | -0.350204 | -0.911583 | 0.646092 | 0.970243 | -1.153427 | 1.977165 | -1.002804 | 1.727904 | -0.573809 |

3000 rows × 15 columns

X_test.shape

(3000, 15)