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

```
import warnings
warnings.filterwarnings('ignore')
```

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

```
df.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Es
0	619	42	2	0.00	1.0	1	1	
1	608	41	1	83807.86	1.0	0	1	
2	502	42	8	159660.80	3.0	1	0	
3	699	39	1	0.00	2.0	0	0	
4	850	43	2	125510.82	1.0	1	1	

```
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Bala
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090

```
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
```

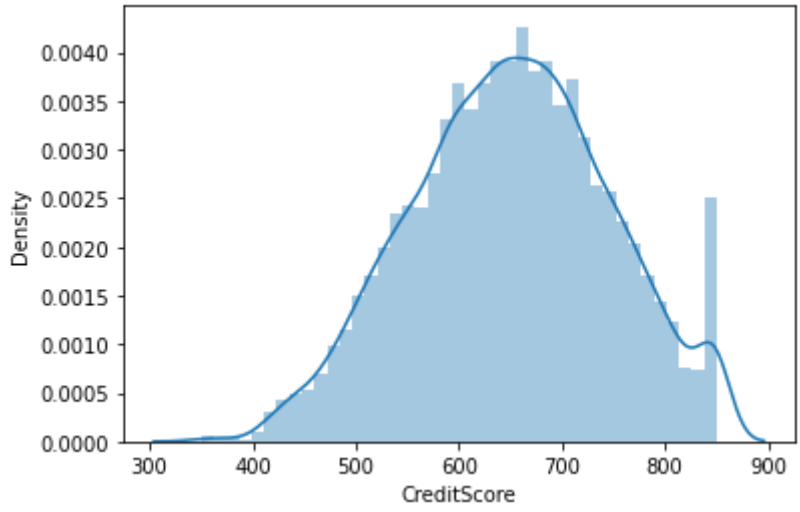
```
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  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
```

```
df.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838(

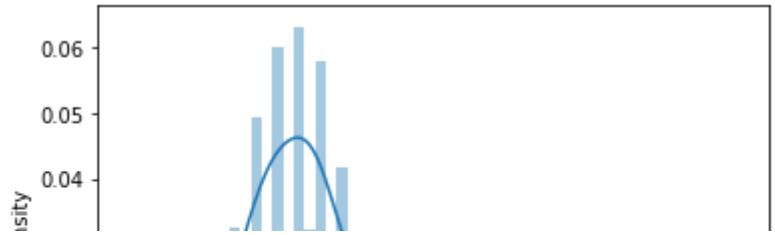
```
sns.distplot(df.CreditScore)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78f236090>
```




```
sns.distplot(df.Age)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78dce3a10>
```



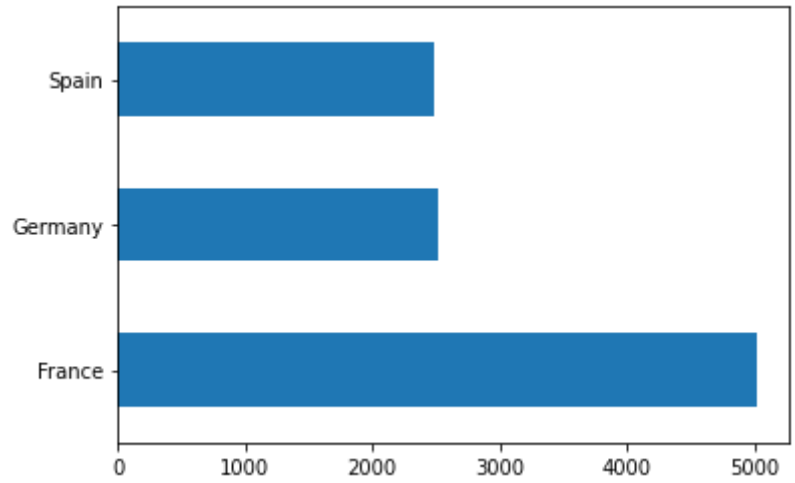

```
df.Gender.value_counts().plot(kind='barh')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-60-863b90bae51d> in <module>
----> 1 df.Gender.value_counts().plot(kind='barh')

/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py in __getattr__(self, n
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'Gender'
```

SEARCH STACK OVERFLOW

```
df.Geography.value_counts().plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d7012d0>
```

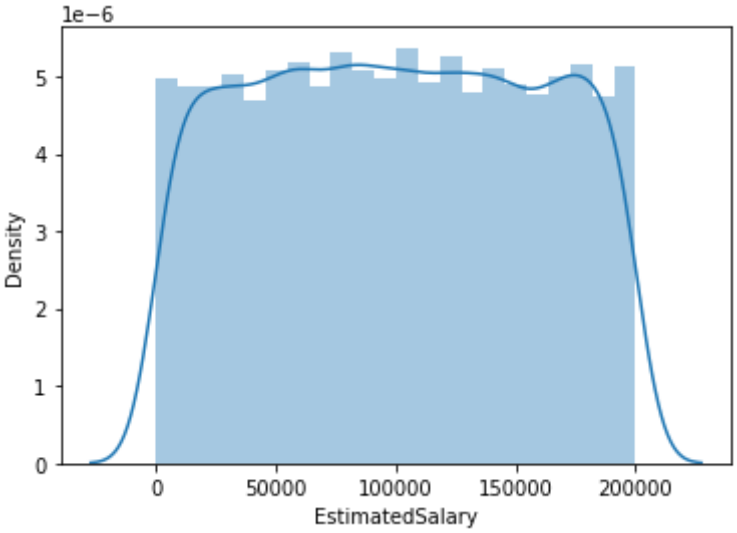



```
df.Tenure.value_counts().plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d6e4290>
```

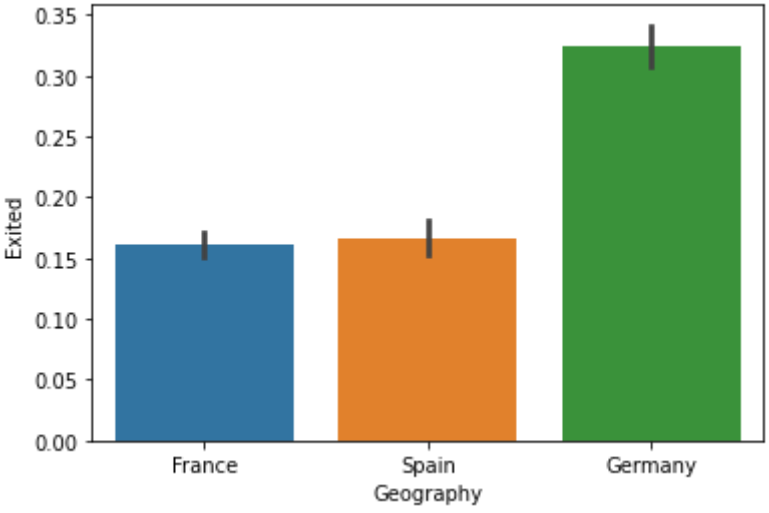

```
sns.distplot(df.EstimatedSalary)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d611810>
```




```
sns.barplot(df.Geography, df.Exited)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d548750>
```

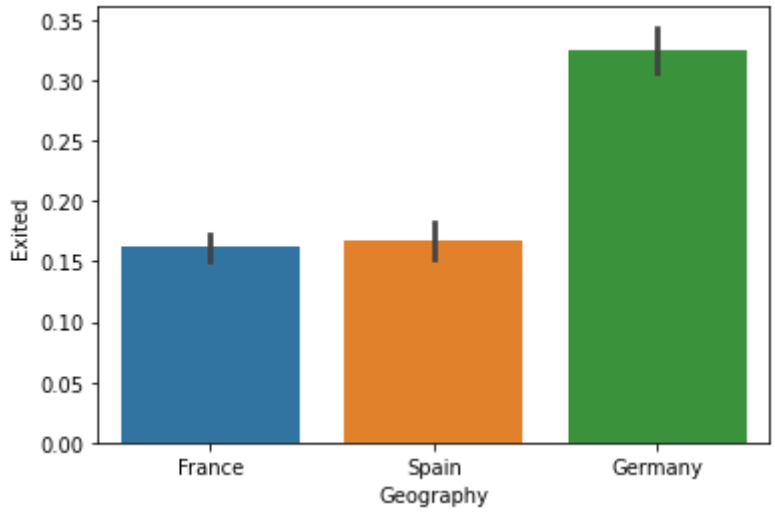



```
df.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838(

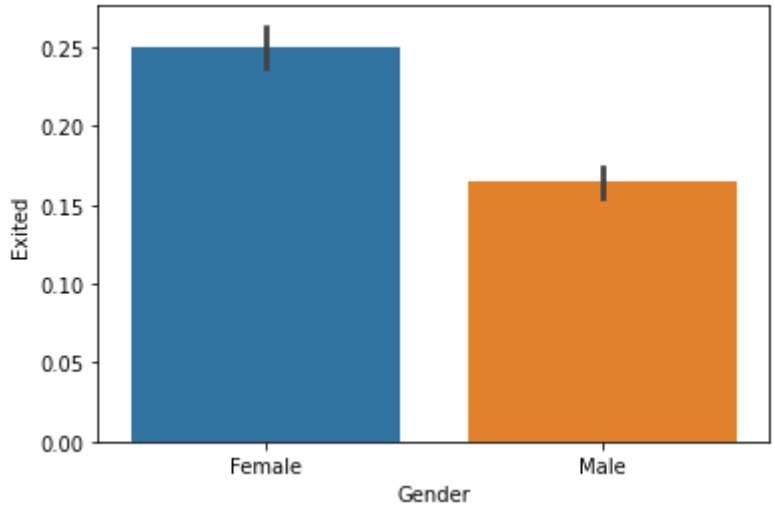
```
sns.barplot(x='Geography',y='Exited',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d4aa850>
```




```
sns.barplot(x='Gender',y='Exited',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d50ee50>
```

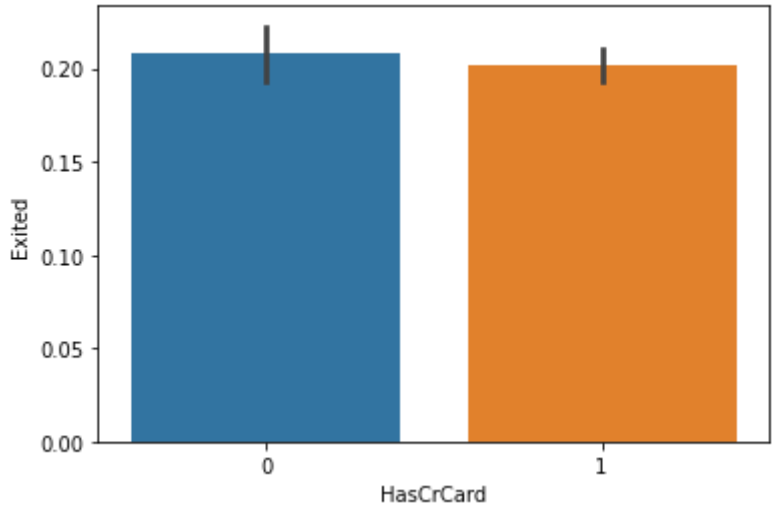



```
sns.barplot(x='NumOfProducts',y='Exited',data=df)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe785fd8650>

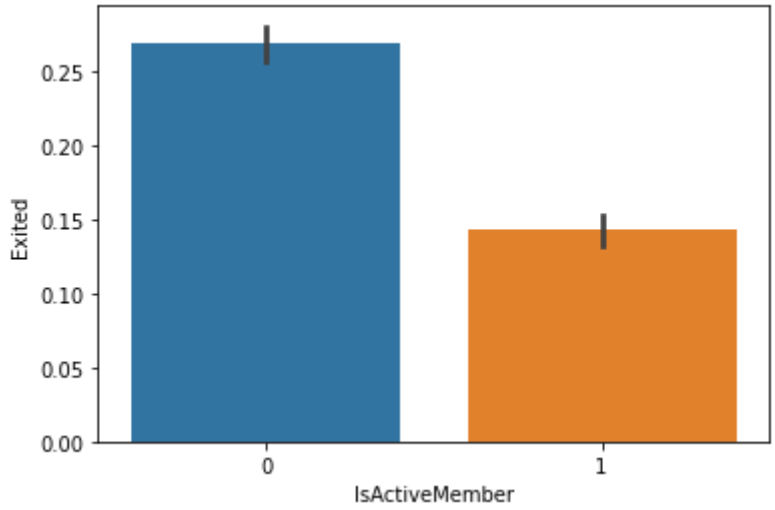
```
sns.barplot(x='HasCrCard',y='Exited',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785fb4b50>
```



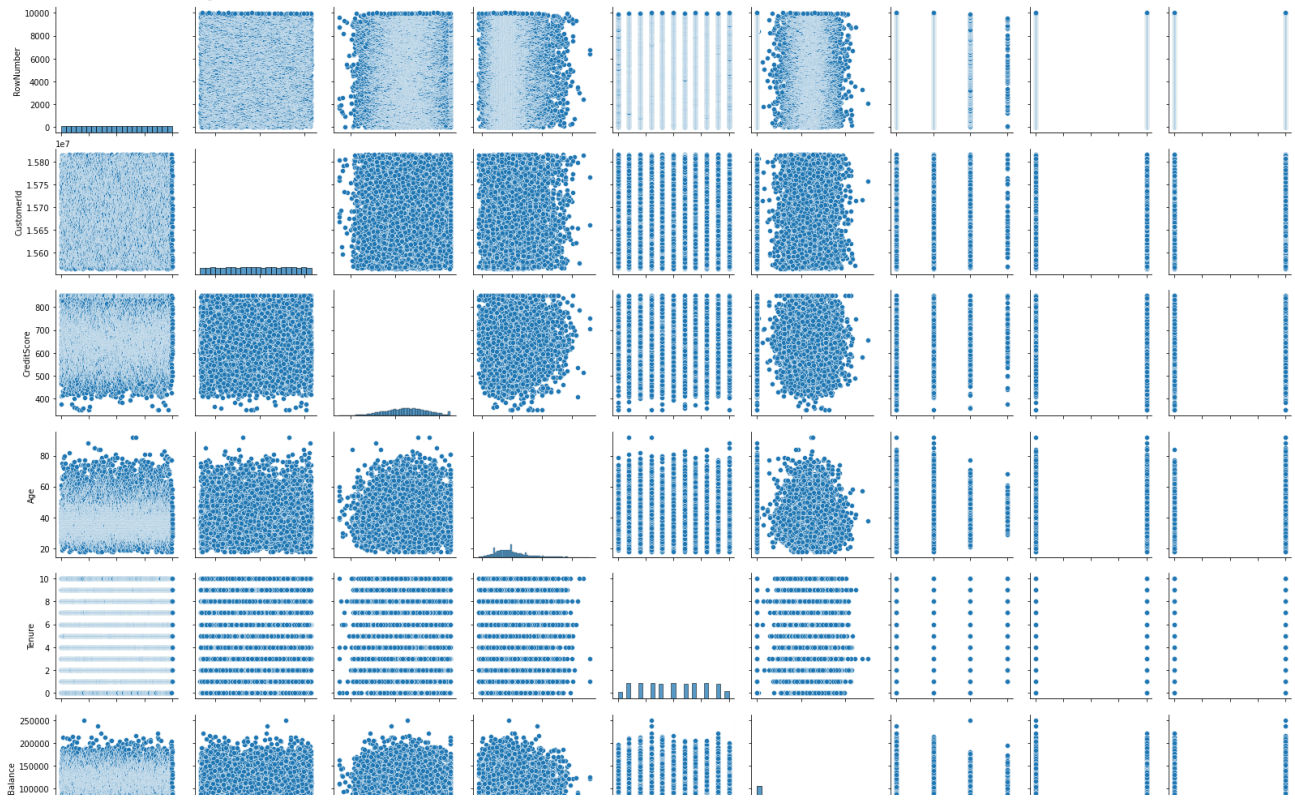

```
sns.barplot(x='IsActiveMember',y='Exited',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78603dcd0>
```



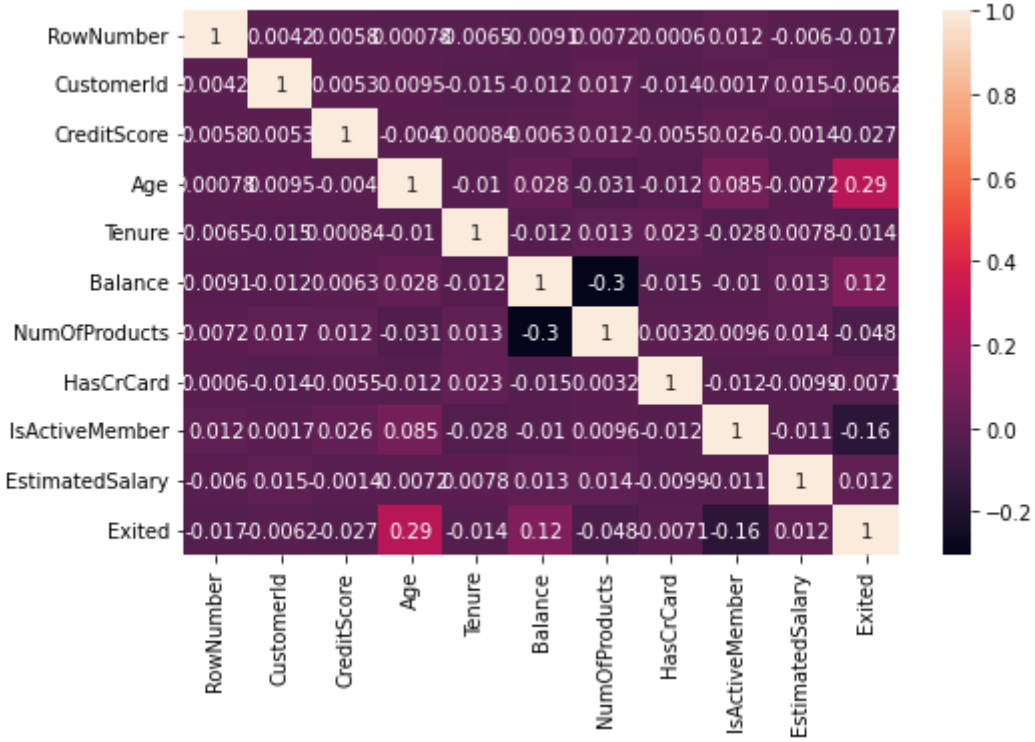

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7fe78ab4de90>
```




```
plt.figure(figsize=(8,5))
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78d49c9d0>
```




```
df.Exited.value_counts()
```

```
0    7963
1    2037
Name: Exited, dtype: int64
```

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

```
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
```

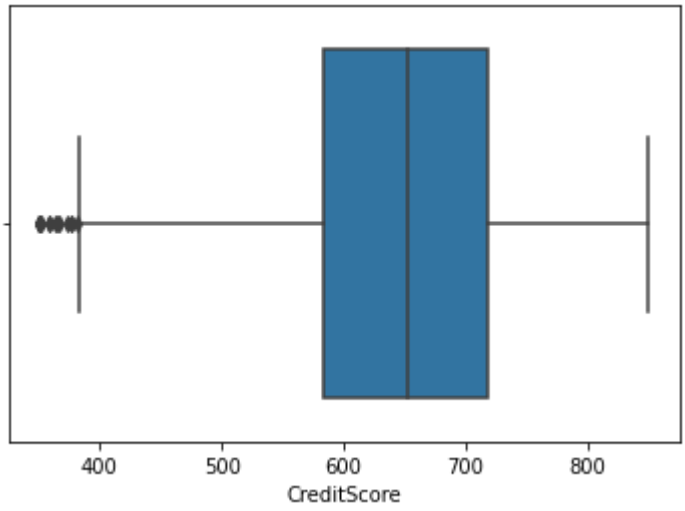
```
#No missing values
```

```
df.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838(

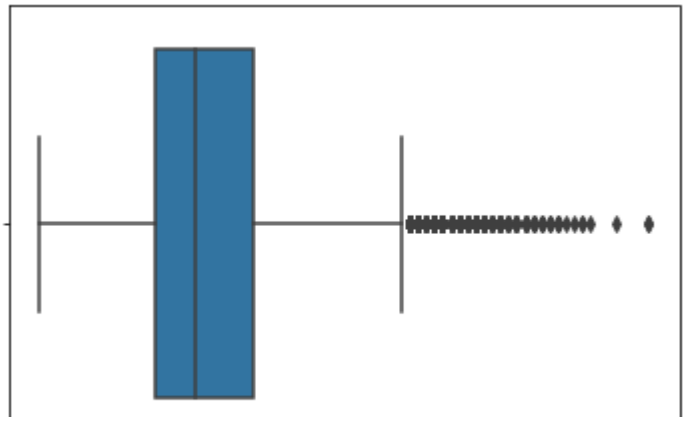
```
sns.boxplot(df.CreditScore)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785f1f9d0>
```



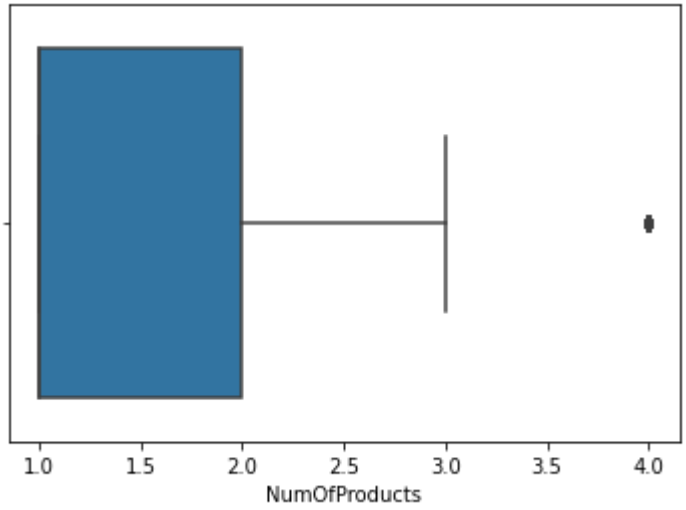

```
sns.boxplot(df.Age)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785ee05d0>
```



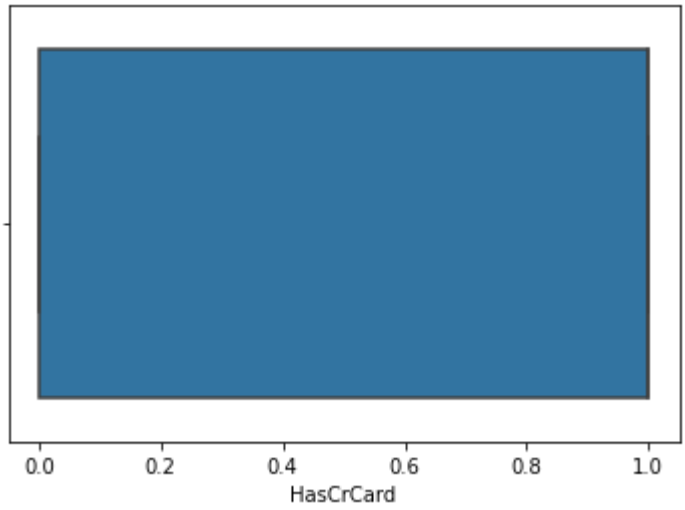
```
sns.boxplot(df.NumOfProducts)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785e5d2d0>
```




```
sns.boxplot(df.HasCrCard)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785dd1650>
```

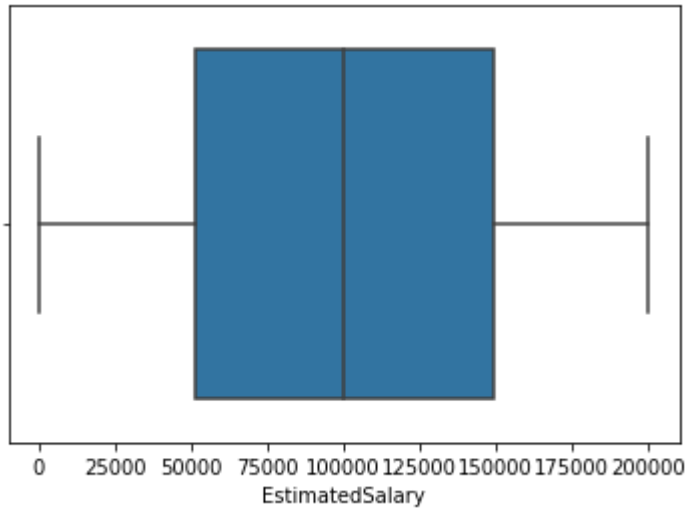



```
sns.boxplot(df.IsActiveMember)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785d37d10>
```

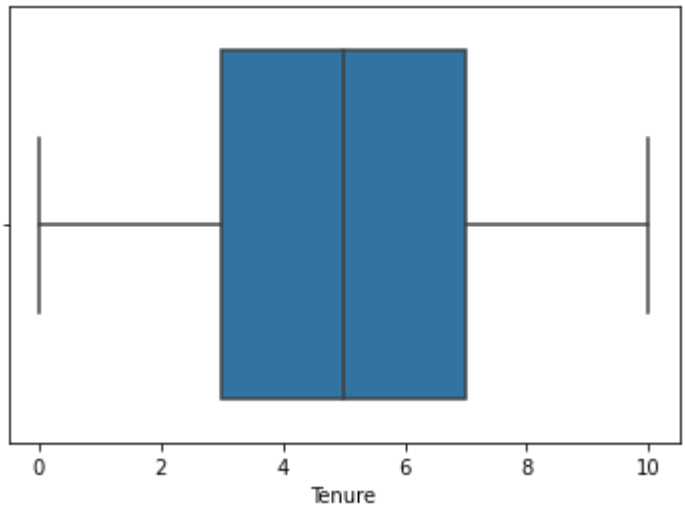
```
sns.boxplot(df.EstimatedSalary)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785cb58d0>
```



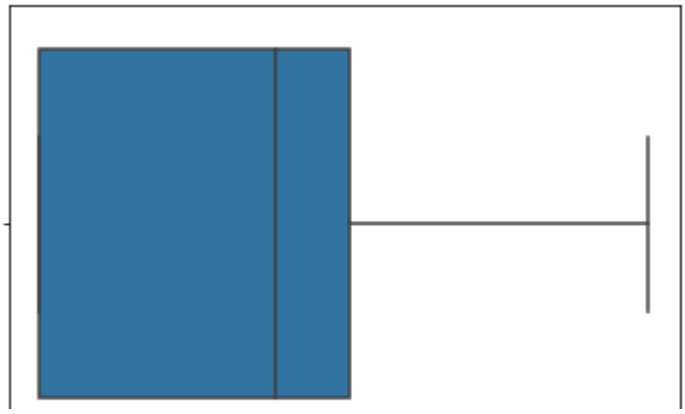

```
sns.boxplot(df.Tenure)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785c6d210>
```



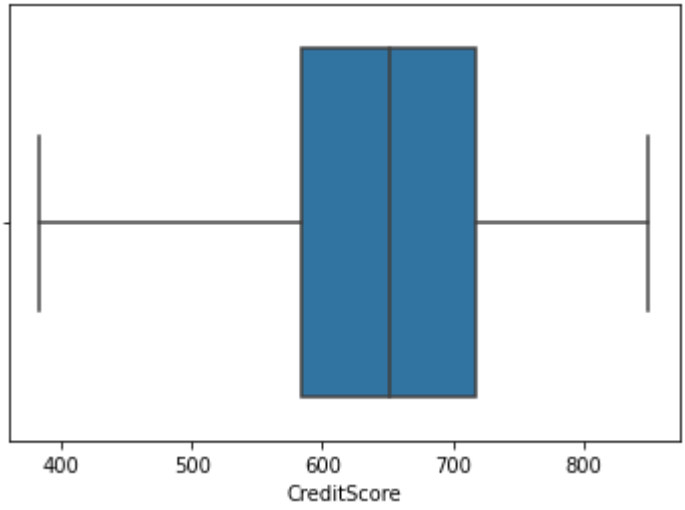

```
sns.boxplot(df.Balance)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe786b4be10>
```



```
#Outlier Removal


def outlier_credit_score(df):
    IQR = df['CreditScore'].quantile(0.75) - df['CreditScore'].quantile(0.25)

    lower_range = df['CreditScore'].quantile(0.25) - (1.5 * IQR)
    upper_range = df['CreditScore'].quantile(0.75) + (1.5 * IQR)

    df.loc[df['CreditScore'] <= lower_range, 'CreditScore'] = lower_range
    df.loc[df['CreditScore'] >= upper_range, 'CreditScore'] = upper_range

outlier_credit_score(df)


sns.boxplot(df.CreditScore)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785d3f790>
```

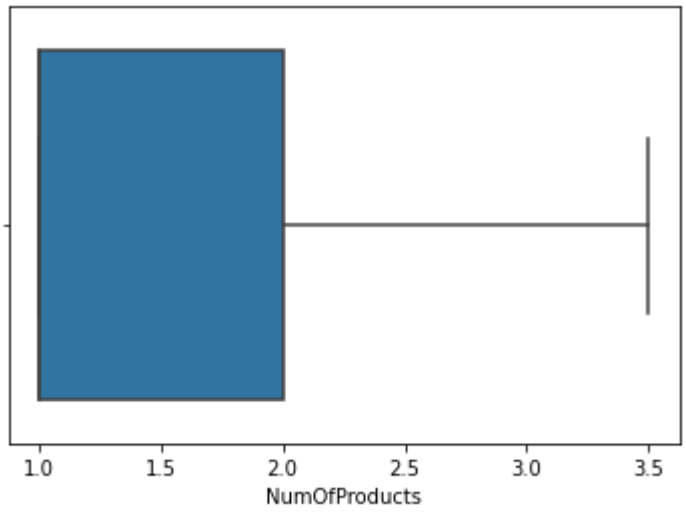



```
def outlier_NOP(df):
    IQR = df['NumOfProducts'].quantile(0.75) - df['NumOfProducts'].quantile(0.25)

    lower_range = df['NumOfProducts'].quantile(0.25) - (1.5 * IQR)
    upper_range = df['NumOfProducts'].quantile(0.75) + (1.5 * IQR)

    df.loc[df['NumOfProducts'] <= lower_range, 'NumOfProducts'] = lower_range
    df.loc[df['NumOfProducts'] >= upper_range, 'NumOfProducts'] = upper_range
```

```
outlier_NOP(df)
```

```
sns.boxplot(df.NumOfProducts)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe785b5a0d0>
```

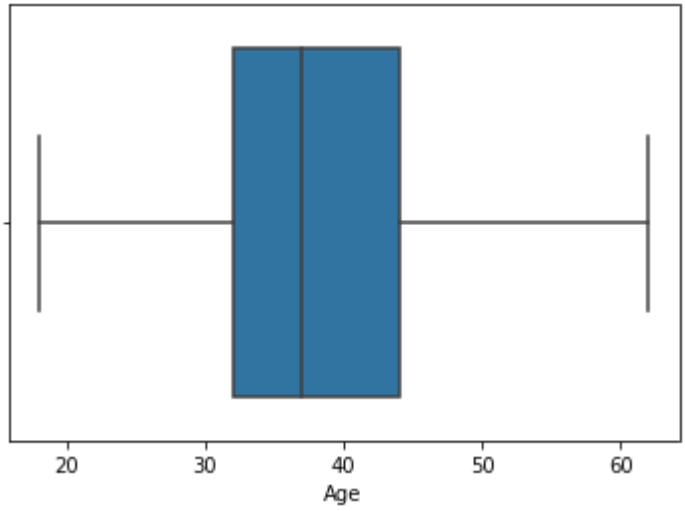



```
def outlier_age(df):
    IQR = df['Age'].quantile(0.75) - df['Age'].quantile(0.25)

    lower_range = df['Age'].quantile(0.25) - (1.5 * IQR)
    upper_range = df['Age'].quantile(0.75) + (1.5 * IQR)

    df.loc[df['Age'] <= lower_range, 'Age'] = lower_range
    df.loc[df['Age'] >= upper_range, 'Age'] = upper_range
```

```
outlier_age(df)
```

```
sns.boxplot(df.Age)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe78509fbd0>
```




```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   CreditScore      10000 non-null  int64
 1   Geography        10000 non-null  object
 2   Gender           10000 non-null  object
 3   Age              10000 non-null  int64
 4   Tenure           10000 non-null  int64
 5   Balance          10000 non-null  float64
 6   NumOfProducts    10000 non-null  float64
 7   HasCrCard        10000 non-null  int64
 8   IsActiveMember   10000 non-null  int64
 9   EstimatedSalary  10000 non-null  float64
 10  Exited           10000 non-null  int64
dtypes: float64(3), int64(6), object(2)
memory usage: 859.5+ KB
```

```
df.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	838

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

```
df.head(2)
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	619	France	Female	42	2	0.00	1.0	1
1	608	Spain	Female	41	1	83807.86	1.0	0

```
from sklearn.preprocessing import LabelEncoder
le_geo = LabelEncoder()
le_gen = LabelEncoder()
df['Sex']=le_gen.fit_transform(df.Gender)
df['Country']=le_geo.fit_transform(df.Geography)
df.drop(['Geography','Gender'],axis=1,inplace=True)
```

```
df.head(2)
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	619	France	Female	42	2	0.00	1.0	1
1	608	Spain	Female	41	1	83807.86	1.0	0

```
X=df.drop('Exited',axis=1)
y=df.Exited
```

```
X
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	619	42	2	0.00	1.0	1	1
1	608	41	1	83807.86	1.0	0	1
2	502	42	8	159660.80	3.0	1	0
3	699	39	1	0.00	2.0	0	0
4	850	43	2	125510.82	1.0	1	1
...	...	...	...	...	...	...	...
9995	771	39	5	0.00	2.0	1	0
9996	516	35	10	57369.61	1.0	1	1
9997	709	36	7	0.00	1.0	0	1
9998	772	42	3	75075.31	2.0	1	0
9999	792	28	4	130142.79	1.0	1	0

10000 rows × 10 columns

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X = sc.fit_transform(X)
```

```
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,
                                               random_state=42)
```

```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
((8000, 10), (2000, 10), (8000,), (2000,))
```