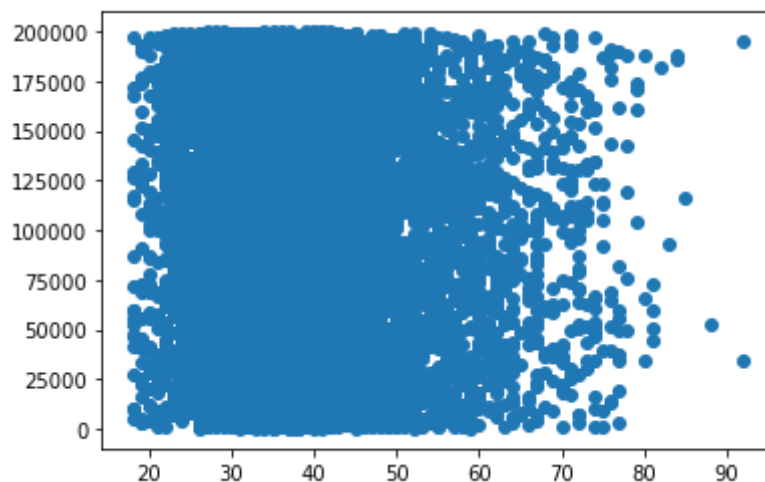


```
In [1]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: # load the dataset
df = pd.read_csv("Churn_Modelling.csv")
```

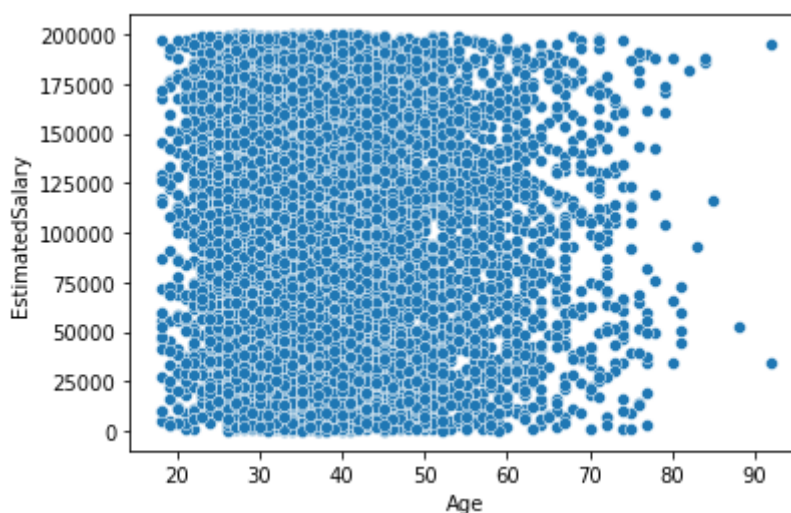
```
In [4]: import matplotlib.pyplot as plt
plt.scatter(df.Age, df.EstimatedSalary)
```

Out[4]: <matplotlib.collections.PathCollection at 0x214a67dbbe0>



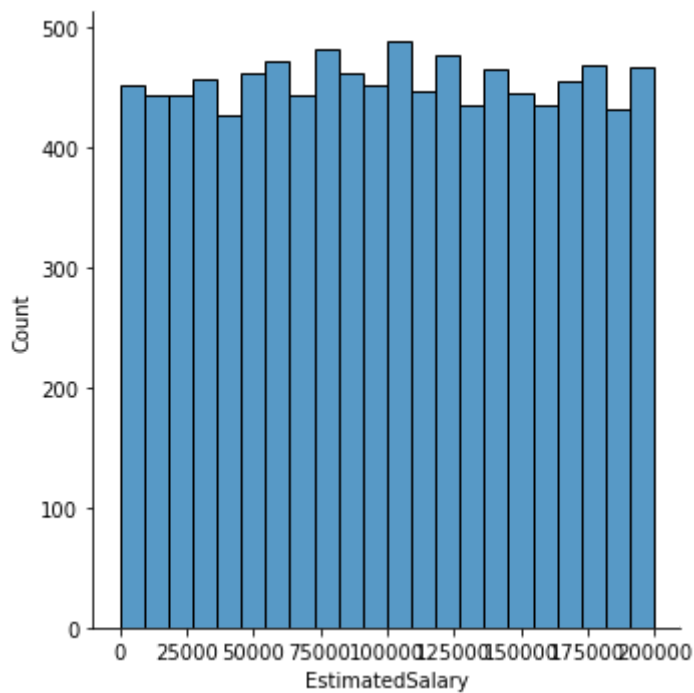
```
In [5]: import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(x = df.Age, y = df.EstimatedSalary)
```

Out[5]: <AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>



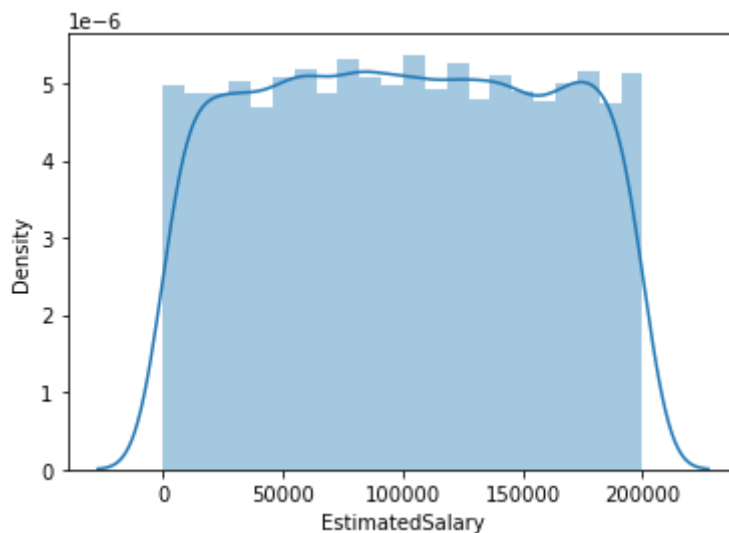
```
In [6]: import matplotlib.pyplot as plt
import seaborn as sns
sns.displot(df["EstimatedSalary"])
```

Out[6]: <seaborn.axisgrid.FacetGrid at 0x214a6b87760>



```
In [8]: import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot(df["EstimatedSalary"])
```

```
Out[8]: <AxesSubplot:xlabel='EstimatedSalary', ylabel='Density'>
```

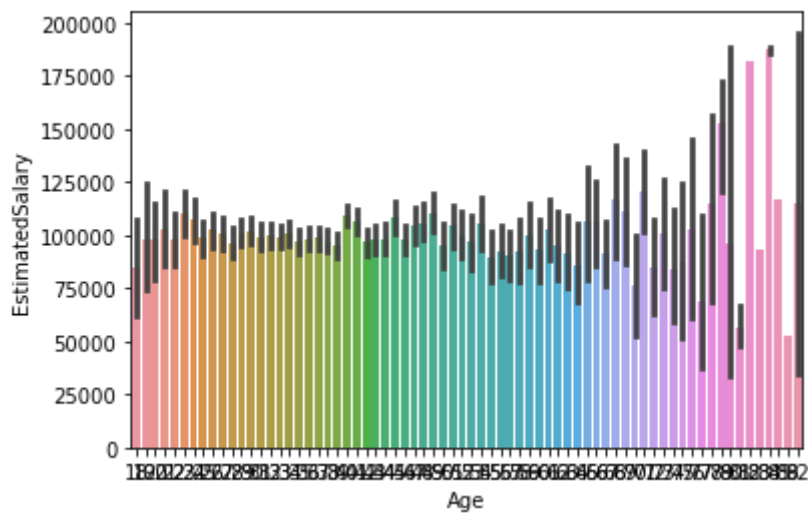


```
In [9]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [21]: # load the dataset
df = pd.read_csv("Churn_Modelling.csv")
```

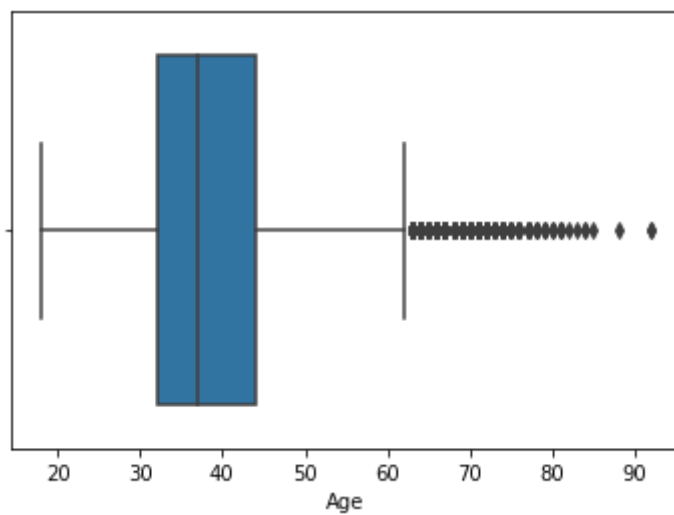
```
In [11]: import matplotlib.pyplot as plt
import seaborn as sns
sns.barplot(df["Age"],df["EstimatedSalary"])
```

```
Out[11]: <AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>
```



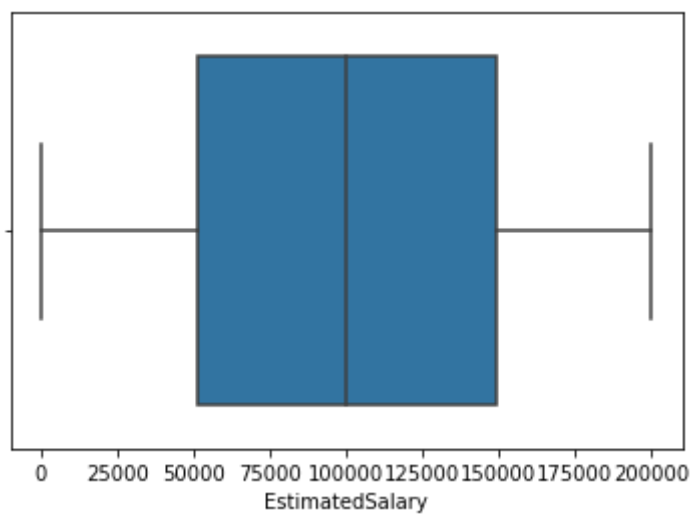
```
In [12]: sns.boxplot(df["Age"])
```

```
Out[12]: <AxesSubplot:xlabel='Age'>
```



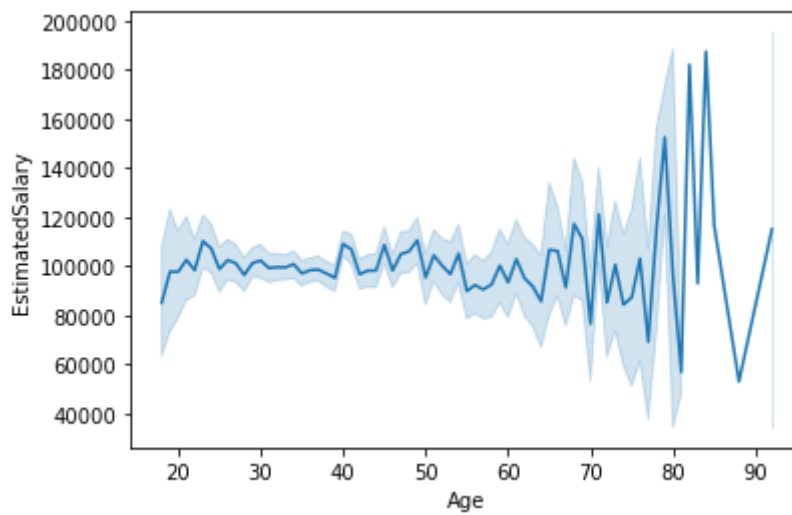
```
In [13]: sns.boxplot(df["EstimatedSalary"])
```

```
Out[13]: <AxesSubplot:xlabel='EstimatedSalary'>
```



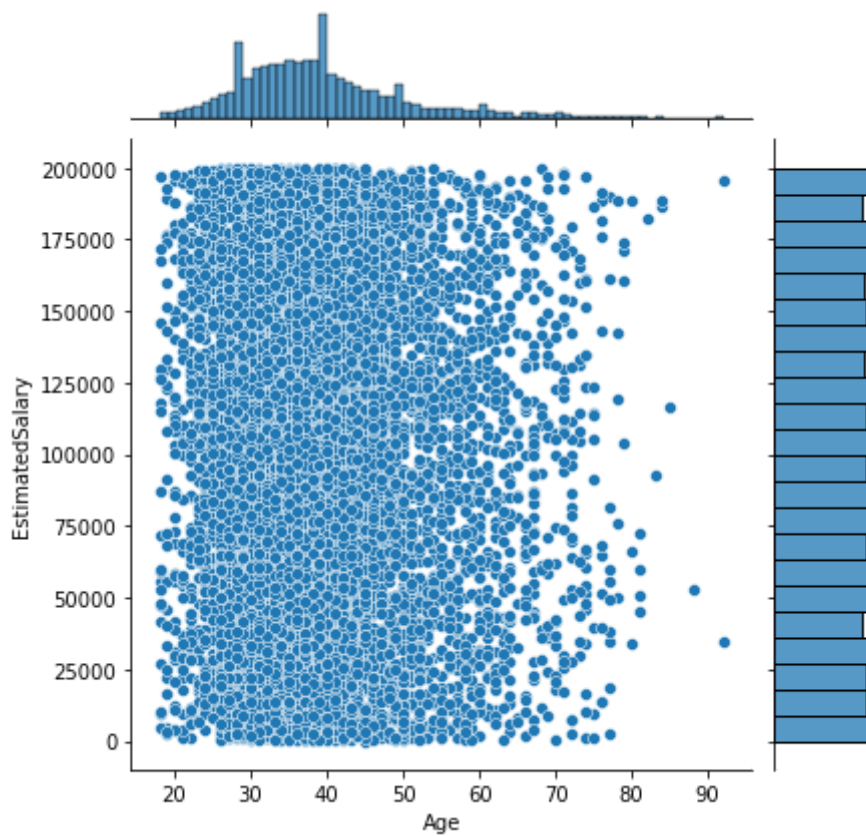
```
In [14]: sns.lineplot(df["Age"],df["EstimatedSalary"])
```

```
Out[14]: <AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>
```



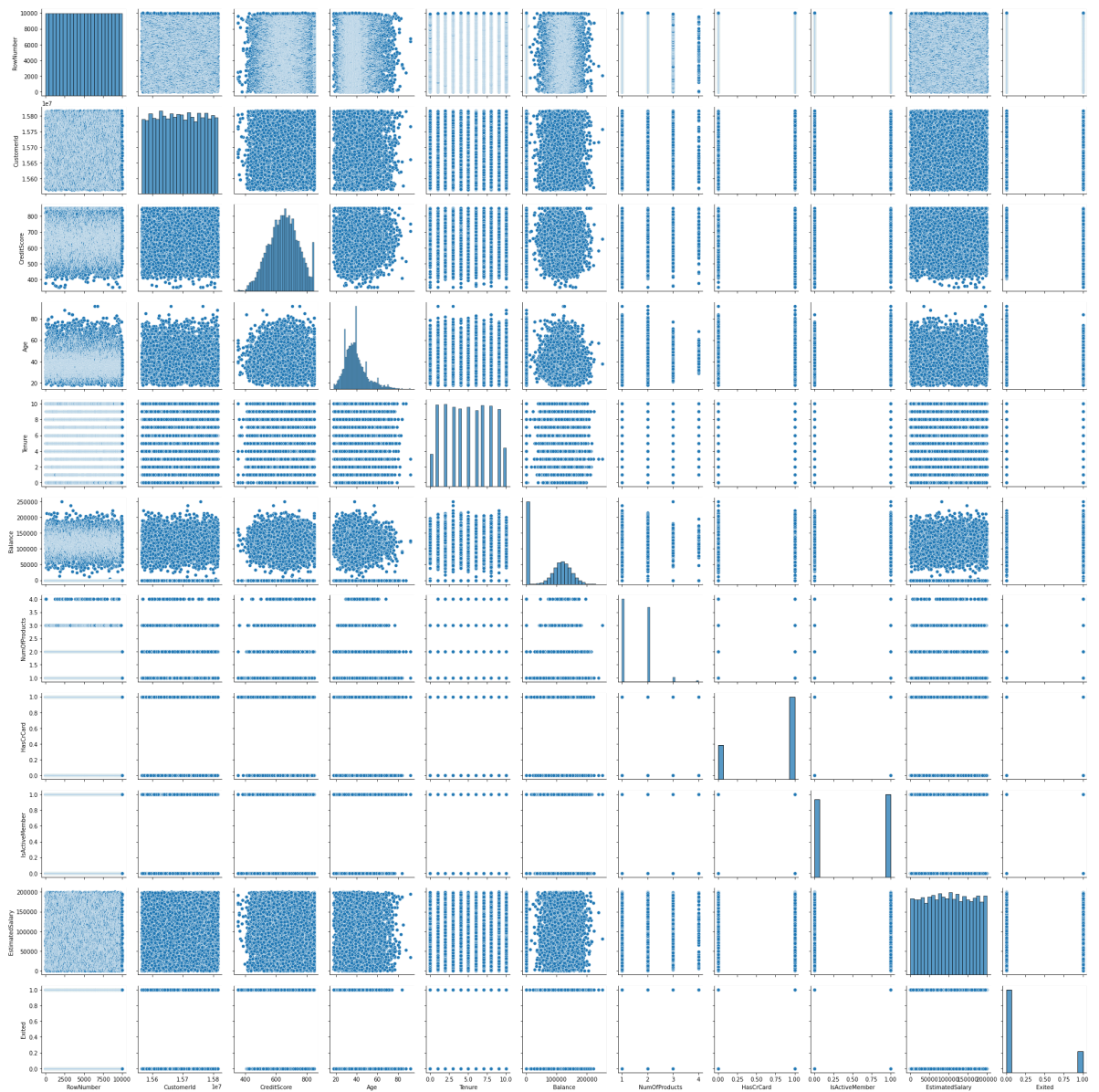
```
In [15]: sns.jointplot(df["Age"],df["EstimatedSalary"])
```

```
Out[15]: <seaborn.axisgrid.JointGrid at 0x214a6fd7ca0>
```



```
In [16]: sns.pairplot(df)
```

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x214a71e4be0>
```



```
In [17]: # descriptive statistics
df.describe()
```

Out[17]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	Num
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	1
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	

```
In [18]: # handling missing values
df = pd.DataFrame({"Gender": [1, 2, np.nan], "Geography": [1, np.nan, np.nan], "Balance": [1, np.nan, np.nan]})
```

Out[18]:

	Gender	Geography	Balance
0	1.0	1.0	1
1	2.0	NaN	2
2	NaN	NaN	3

	Gender	Geography	Balance
0	1.0	1.0	1
1	2.0	NaN	2
2	NaN	NaN	3

In [22]: `df.isnull().any()`

Out[22]:

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 [23]: `qnt = df.quantile(q = (0.25,0.75))`
`qnt`

Out[23]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard
0.25	2500.75	15628528.25	584.0	32.0	3.0	0.00	1.0	0.0
0.75	7500.25	15753233.75	718.0	44.0	7.0	127644.24	2.0	1.0

In [24]: `iqr = qnt.loc[0.75] - qnt.loc[0.25]`
`iqr`

Out[24]:

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 [25]: `lower = qnt.loc [0.25] - 1.5*iqr`
`lower`

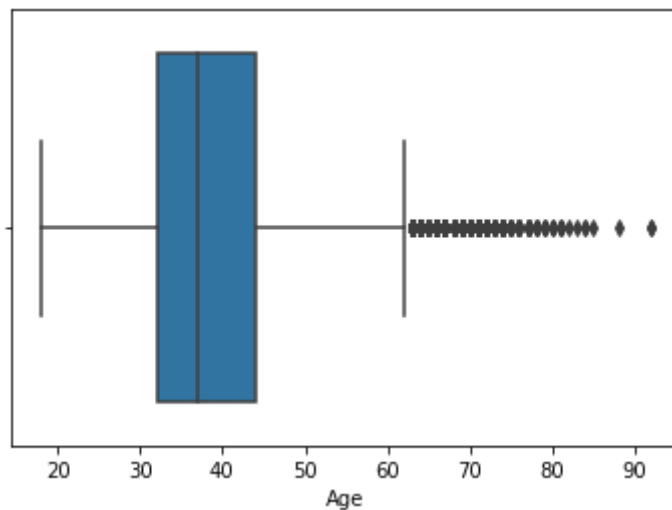
```
Out[25]: RowNumber      -4.998500e+03
CustomerId    1.544147e+07
CreditScore   3.830000e+02
Age           1.400000e+01
Tenure        -3.000000e+00
Balance       -1.914664e+05
NumOfProducts -5.000000e-01
HasCrCard     -1.500000e+00
IsActiveMember -1.500000e+00
EstimatedSalary -9.657710e+04
Exited        0.000000e+00
dtype: float64
```

```
In [26]: upper =qnt.loc[0.75] + 1.5*iqr
upper
```

```
Out[26]: 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 [27]: sns.boxplot(df["Age"])
```

```
Out[27]: <AxesSubplot:xlabel='Age'>
```



```
In [28]: df["Age"] = np.where(df["Age"]>87,40,df["Age"])
df["EstimatedSalary"] = np.where(df["EstimatedSalary"]>45,31,df["EstimatedSalary"])
```

```
In [29]: df
```

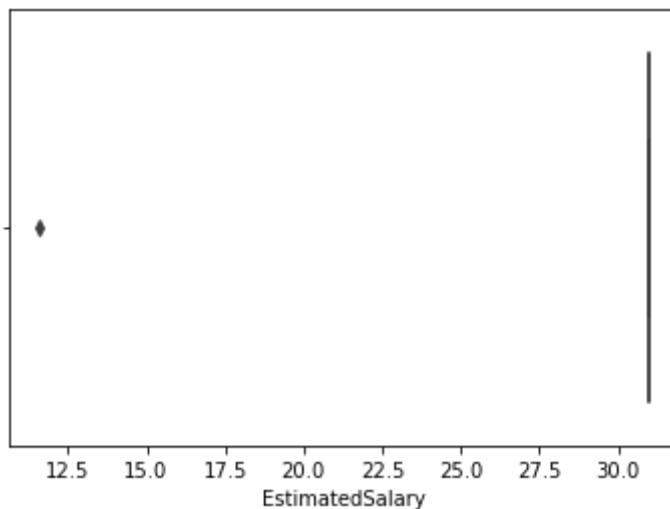
```
Out[29]:
```

	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	8380
2	3	15619304	Onio	502	France	Female	42	8	15960
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	12557
...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	5736
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	7507
9999	10000	15628319	Walker	792	France	Female	28	4	13014

10000 rows × 14 columns

```
In [30]: sns.boxplot(df["EstimatedSalary"])
```

```
Out[30]: <AxesSubplot:xlabel='EstimatedSalary'>
```



```
In [31]: df.head(2)
```

```
Out[31]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86

```
In [32]: df_main = pd.get_dummies(df, columns=["EstimatedSalary"])
df_main
```


Out[32]:

	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	8380
2	3	15619304	Onio	502	France	Female	42	8	15960
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	12550
...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	5730
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	7500
9999	10000	15628319	Walker	792	France	Female	28	4	13010

10000 rows × 15 columns



In [33]:

```
# split x & y
x = df.iloc[:,0:1]
x
```

Out[33]:

	RowNumber
0	1
1	2
2	3
3	4
4	5
...	...
9995	9996
9996	9997
9997	9998
9998	9999
9999	10000

10000 rows × 1 columns

In [34]:

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

Out[34]:

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfI
0	15634602	Hargrave	619	France	Female	42	2	0.00	
1	15647311	Hill	608	Spain	Female	41	1	83807.86	
2	15619304	Onio	502	France	Female	42	8	159660.80	
3	15701354	Boni	699	France	Female	39	1	0.00	
4	15737888	Mitchell	850	Spain	Female	43	2	125510.82	
...	
9995	15606229	Obijiaku	771	France	Male	39	5	0.00	
9996	15569892	Johnstone	516	France	Male	35	10	57369.61	
9997	15584532	Liu	709	France	Female	36	7	0.00	
9998	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	
9999	15628319	Walker	792	France	Female	28	4	130142.79	

10000 rows × 13 columns



In [35]:

```
# train test split
from sklearn.model_selection import train_test_split
```

In [36]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_st
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[36]:

```
((8000, 1), (2000, 1), (8000, 13), (2000, 13))
```

In [37]:

```
x_test
```

Out[37]:

	RowNumber
9394	9395
898	899
2398	2399
5906	5907
2343	2344
...	...
1037	1038
2899	2900
9549	9550
2740	2741
6690	6691

2000 rows × 1 columns

In [38]:

```
x_train
```

Out[38]:

RowNumber	
7389	7390
9275	9276
2995	2996
5316	5317
356	357
...	...
9225	9226
4859	4860
3264	3265
9845	9846
2732	2733

8000 rows × 1 columns

In [39]:

y_test

Out[39]:

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfI
9394	15615753	Upchurch	597	Germany	Female	35	8	131101.04	
898	15654700	Fallaci	523	France	Female	40	2	102967.41	
2398	15633877	Morrison	706	Spain	Female	42	8	95386.82	
5906	15745623	Worsnop	788	France	Male	32	4	112079.58	
2343	15765902	Gibson	706	Germany	Male	38	5	163034.82	
...	
1037	15631054	Volkova	625	France	Female	24	1	0.00	
2899	15810944	Bryant	586	France	Female	35	7	0.00	
9549	15772604	Chiemezie	578	Spain	Male	36	1	157267.95	
2740	15787699	Burke	650	Germany	Male	34	4	142393.11	
6690	15579223	Niu	573	Germany	Male	30	8	127406.50	

2000 rows × 13 columns



In [40]:

y_train

Out[40]:

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
7389	15676909	Mishin	667	Spain	Female	34	5	0.00
9275	15749265	Carslaw	427	Germany	Male	42	1	75681.52
2995	15582492	Moore	535	France	Female	29	2	112367.34
5316	15780386	Ferri	654	Spain	Male	40	5	105683.63
356	15611759	Simmons	850	Spain	Female	57	8	126776.30
...
9225	15584928	Ugochukwutubelum	594	Germany	Female	32	4	120074.97
4859	15647111	White	794	Spain	Female	22	4	114440.24
3264	15574372	Hoolan	738	France	Male	35	5	161274.05
9845	15664035	Parsons	590	Spain	Female	38	9	0.00
2732	15592816	Udokamma	623	Germany	Female	48	1	108076.33

8000 rows × 13 columns



In []: