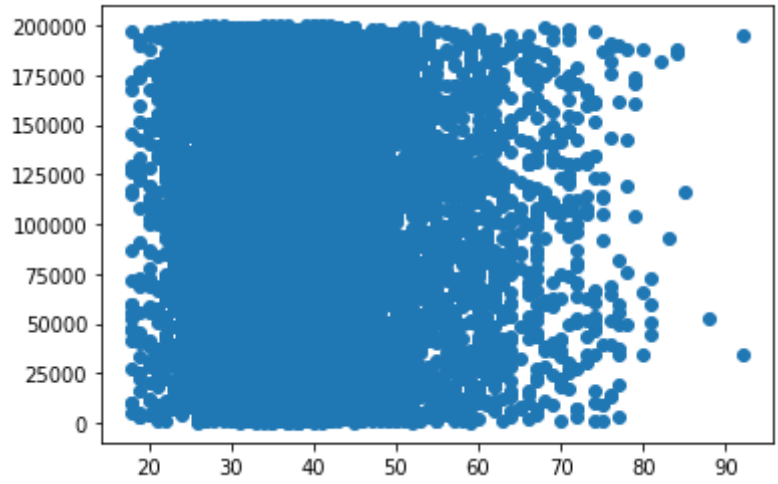
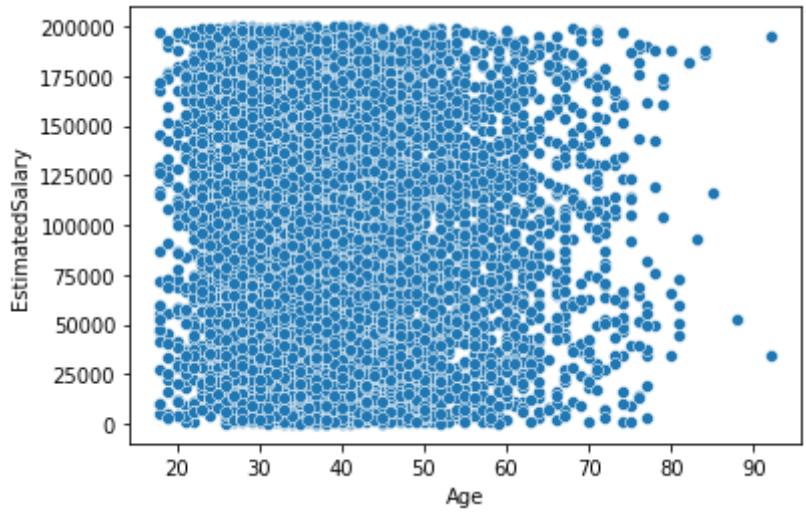
```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")


# load the dataset
df = pd.read_csv("Churn_Modelling.csv")


import matplotlib.pyplot as plt
plt.scatter(df.Age,df.EstimatedSalary)
```

```
<matplotlib.collections.PathCollection at 0x7f656e791590>
```




```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(x = df.Age,y = df.EstimatedSalary)
```

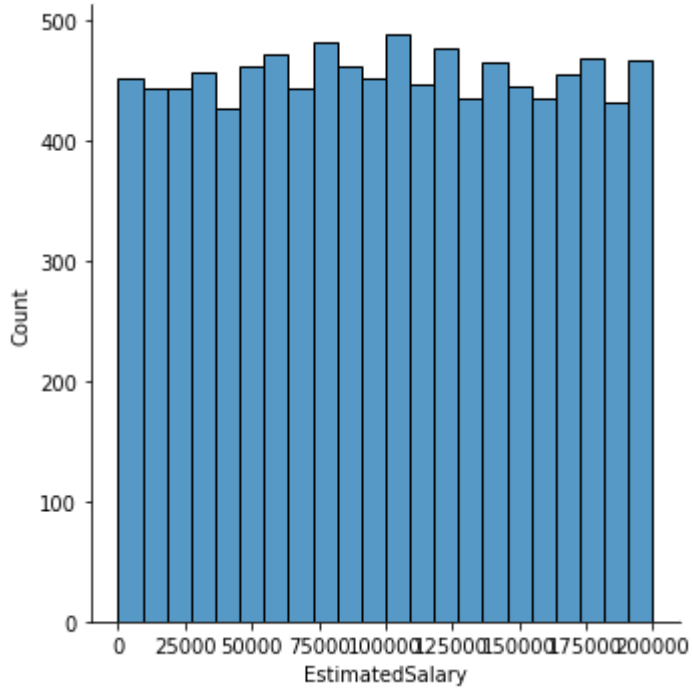
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f656e25c410>
```




```
import matplotlib.pyplot as plt
import seaborn as sns
```

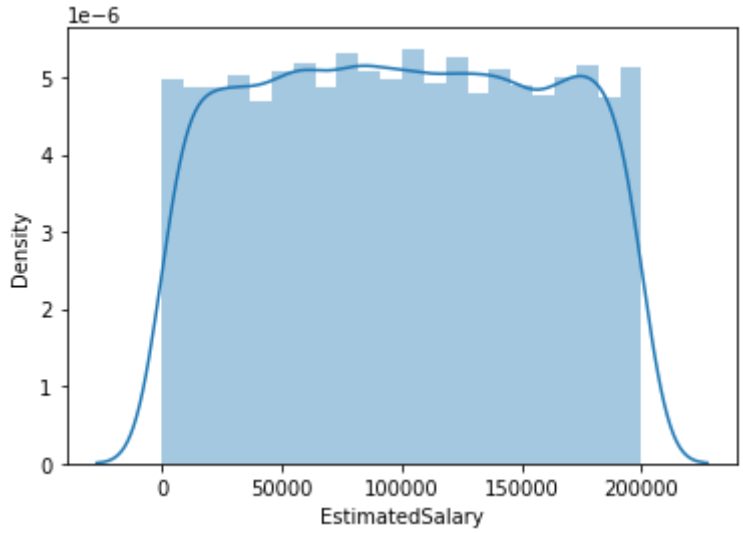
```
sns.displot(df["EstimatedSalary"])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f656e6ec350>
```




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

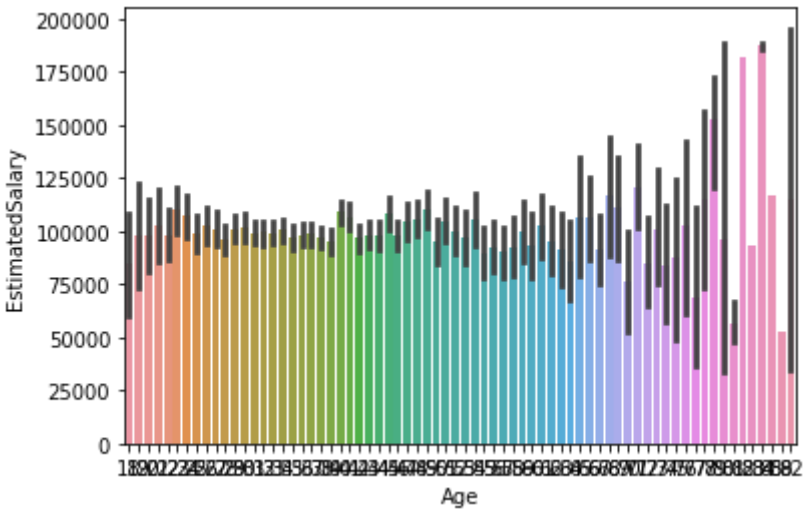
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f656a074590>
```




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


# load the dataset
df = pd.read_csv("Churn_Modelling.csv")
```

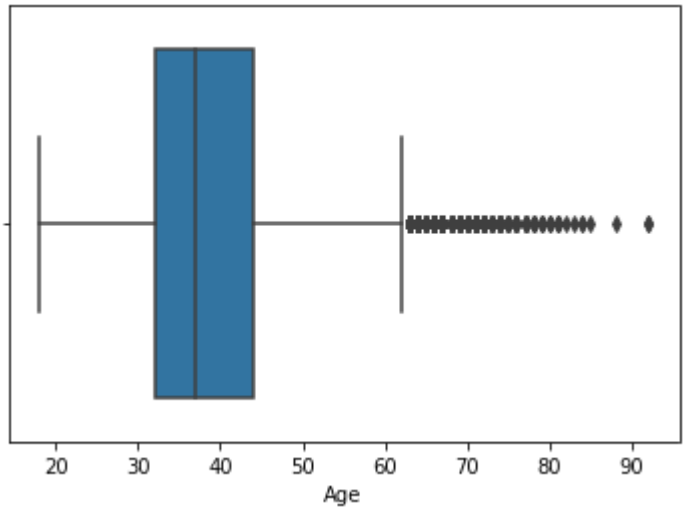
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.barplot(df["Age"],df["EstimatedSalary"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6566542a90>
```




```
sns.boxplot(df["Age"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6569c1f2d0>
```

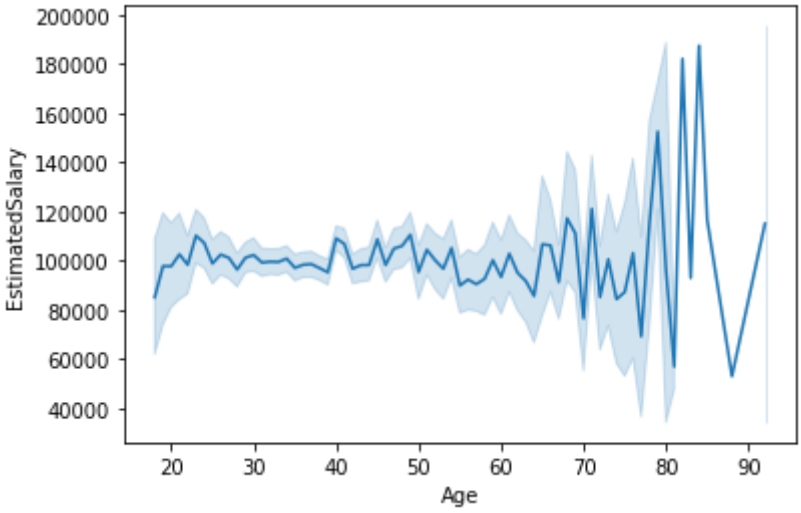



```
sns.boxplot(df["EstimatedSalary"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f65662807d0>
```

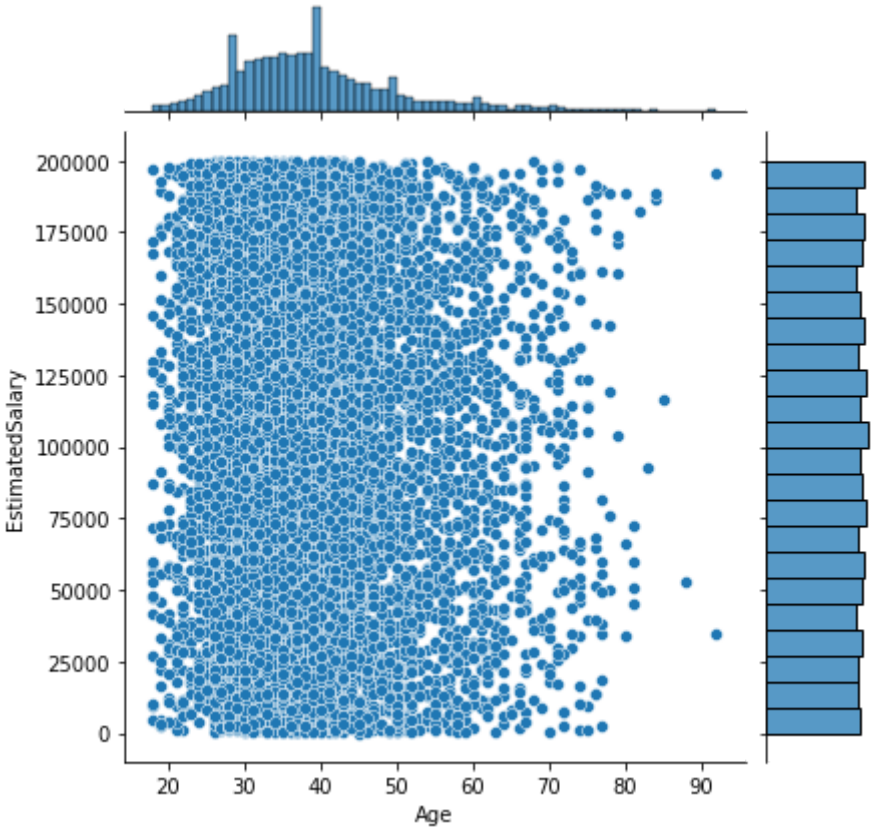
```
sns.lineplot(df["Age"],df["EstimatedSalary"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f656626f590>
```



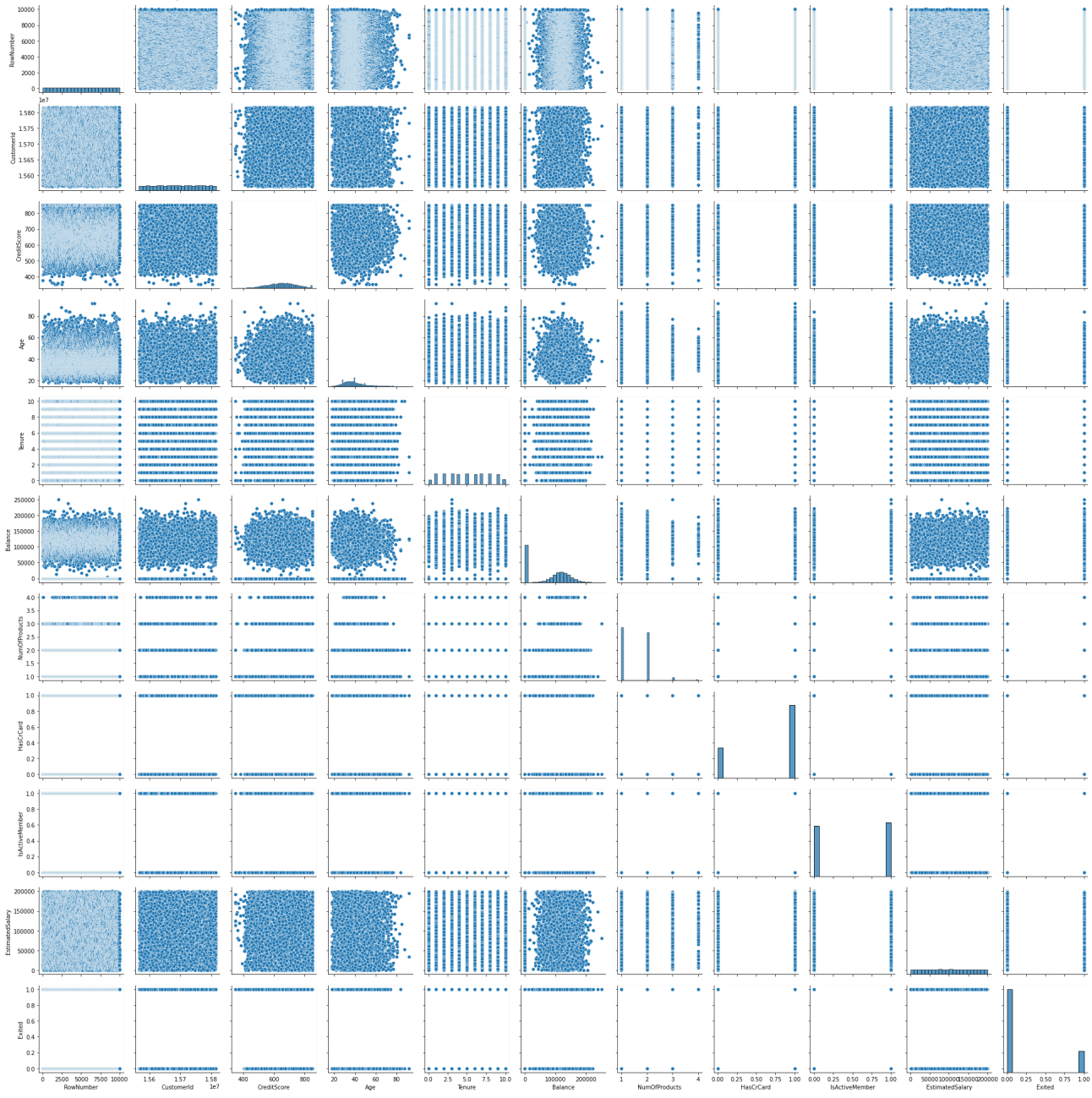

```
sns.jointplot(df["Age"],df["EstimatedSalary"])
```

```
<seaborn.axisgrid.JointGrid at 0x7f6569ae7610>
```




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

```
<seaborn.axisgrid.PairGrid at 0x7f6569901950>
```



```
# descriptive statistics
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

```
# handling missing values
df = pd.DataFrame({"Gender":[1,2,np.nan],"Geography":[1,np.nan,np.nan],"Balance":[1,2,3]})
df
```

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

```
df.isnull().any()
```

```
Gender        True
Geography     True
Balance      False
dtype: bool
```

```
qnt = df.quantile(q = (0.25,0.75))
```

```
qnt
```

	Gender	Geography	Balance
0.25	1.25	1.0	1.5
0.75	1.75	1.0	2.5

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

iqr
```

```
Gender       0.5
Geography    0.0
Balance      1.0
dtype: float64
```

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

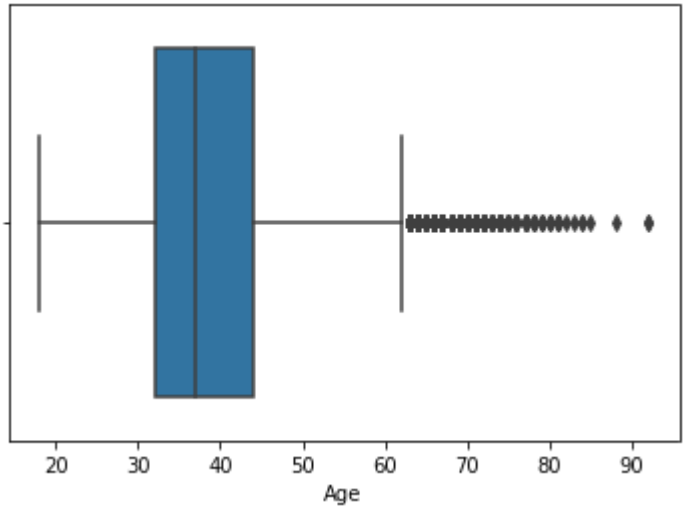
```
Gender       0.5
Geography    1.0
Balance      0.0
dtype: float64
```

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


upper
```

```
Gender       2.5
Geography    1.0
Balance      4.0
dtype: float64
```

```
sns.boxplot(df["Age"])
```

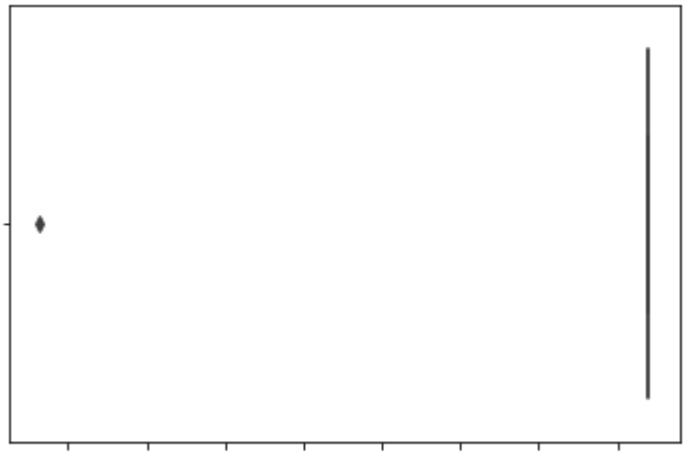
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f656610e2d0>
```




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

```
sns.boxplot(df["EstimatedSalary"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f656608ff10>
```



```
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["Age"].replace({"40":0,"32":1},inplace = True)
df["EstimatedSalary"].replace({"31.0":1,"40.0":0},inplace = True)
```

```
df.head(10)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
5	6	15574012	Chu	645	Spain	Male	44	8	113
6	7	15592531	Bartlett	822	France	Male	50	7	
7	8	15656148	Obinna	376	Germany	Female	29	4	115
8	9	15792365	He	501	France	Male	44	4	142
9	10	15592389	H?	684	France	Male	27	2	134

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

```
df_main
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure
0	1	15634602	Hargrave	619	France	Female	42	2
1	2	15647311	Hill	608	Spain	Female	41	1
2	3	15619304	Onio	502	France	Female	42	8
3	4	15701354	Boni	699	France	Female	39	1
4	5	15737888	Mitchell	850	Spain	Female	43	2
...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5
9996	9997	15569892	Johnstone	516	France	Male	35	10
9997	9998	15584532	Liu	709	France	Female	36	7
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3
9999	10000	15628319	Walker	792	France	Female	28	4

10000 rows × 15 columns

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

	RowNumber
0	1

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

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
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

```
# train test split
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=0)

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

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

```
x_test
```

	RowNumber
9394	9395
898	899
2398	2399
5906	5907
2343	2344
...	...
1037	1038

```
x_train
```

	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

```
y_test
```

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
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
...	...	...	...	...	...	...	...	...

y_train

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	B
7389	15676909	Mishin	667	Spain	Female	34	5	
9275	15749265	Carslaw	427	Germany	Male	42	1	75
2995	15582492	Moore	535	France	Female	29	2	112
5316	15780386	Ferri	654	Spain	Male	40	5	105
356	15611759	Simmons	850	Spain	Female	57	8	126
...	...	...	...	...	...	...	...	
9225	15584928	Ugochukwutubelum	594	Germany	Female	32	4	120
4859	15647111	White	794	Spain	Female	22	4	114
3264	15574372	Hoolan	738	France	Male	35	5	161
9845	15664035	Parsons	590	Spain	Female	38	9	
2732	15592816	Udokamma	623	Germany	Female	48	1	108

8000 rows × 13 columns

Colab paid products - Cancel contracts here

✓ 0s completed at 12:02 PM