```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
from scipy.stats import iqr
```

```
df = pd.read_csv(r"/content/Churn_Modelling.csv")
df
```

	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 × 14 columns

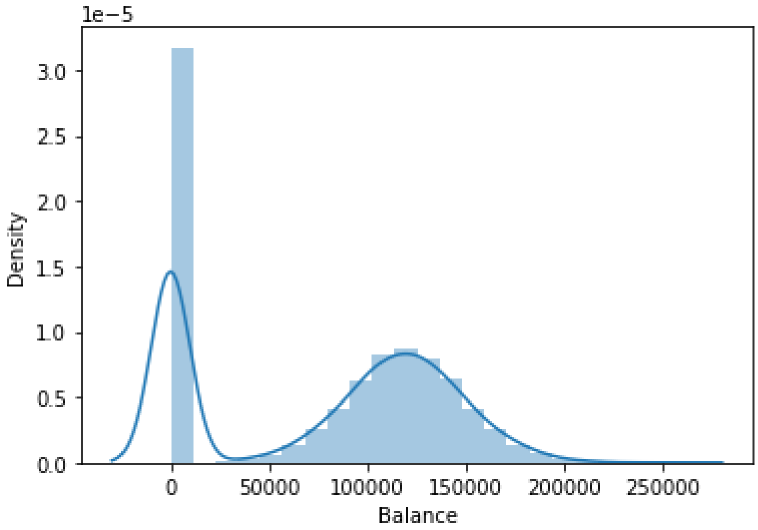
```
df.isnull().sum(1)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
9995    0
9996    0
9997    0
9998    0
9999    0
Length: 10000, dtype: int64
```

Perform Below Visualizations. ● Univariate Analysis ● Bi - Variate Analysis ● Multi - Variate Analysis

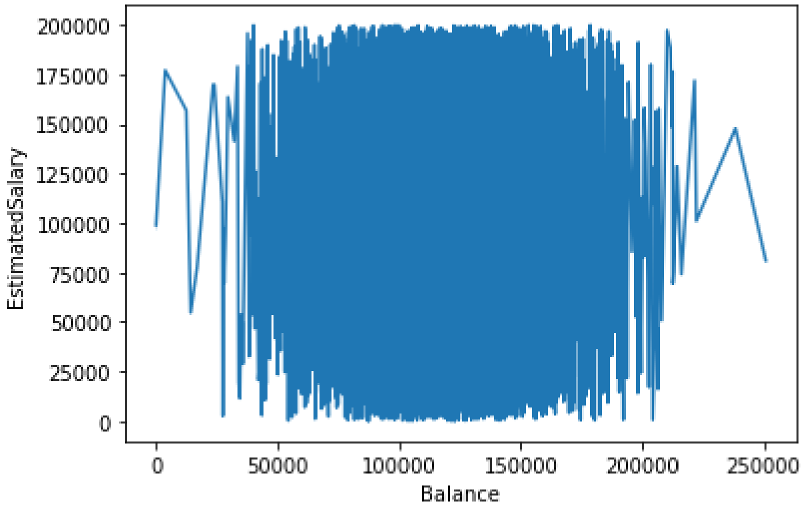
```
sns.distplot(df['Balance'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c77d9310>
```



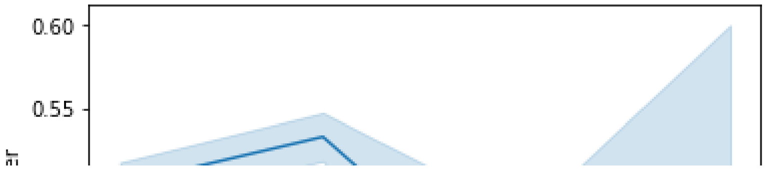
```
sns.lineplot(df['Balance'],df['EstimatedSalary'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c7780a90>
```



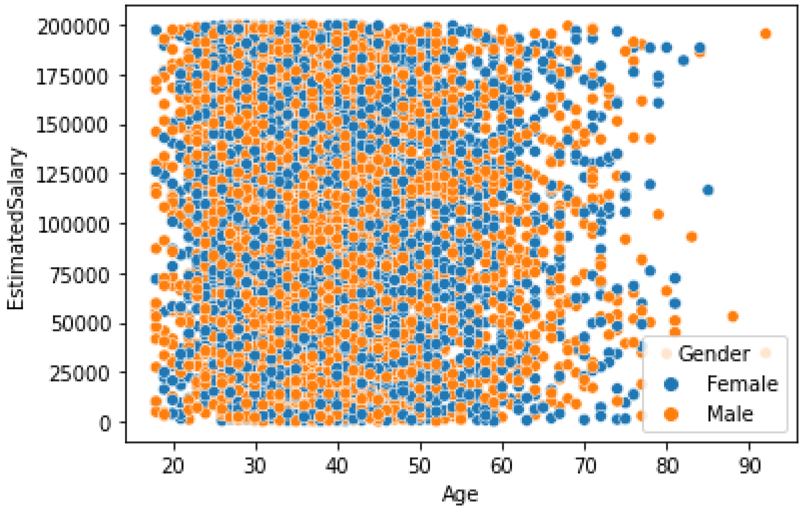
```
sns.lineplot(df['NumOfProducts'],df['IsActiveMember'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c833bc50>
```



```
sns.scatterplot(df['Age'],df['EstimatedSalary'],hue = df['Gender'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c9cc0990>
```



```
df.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
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
```

Perform descriptive statistics on the dataset.

```
df.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
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
```

```
df.mode()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	B
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	

10000 rows × 14 columns

```
df.var()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
RowNumber          8.334167e+06
CustomerId         5.174815e+09
CreditScore        9.341860e+03
Age                1.099941e+02
Tenure             8.364673e+00
Balance            3.893436e+09
NumOfProducts      3.383218e-01
HasCrCard          2.077905e-01
IsActiveMember     2.497970e-01
EstimatedSalary    3.307457e+09
Exited             1.622225e-01
dtype: float64
```

```
df.std()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
RowNumber           2886.895680
CustomerId         71936.186123
CreditScore           96.653299
Age                   10.487806
Tenure                 2.892174
Balance            62397.405202
NumOfProducts          0.581654
HasCrCard              0.455840
IsActiveMember         0.499797
EstimatedSalary    57510.492818
Exited                 0.402769
dtype: float64
```

```
df.min()
```

```
RowNumber                 1
CustomerId         15565701
Surname               Abazu
CreditScore             350
Geography            France
Gender               Female
Age                      18
Tenure                    0
Balance                 0.0
NumOfProducts             1
HasCrCard                 0
IsActiveMember            0
EstimatedSalary       11.58
Exited                    0
dtype: object
```

```
iqr(df['Age'])
```

```
12.0
```

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

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

```
print(df.skew())
```

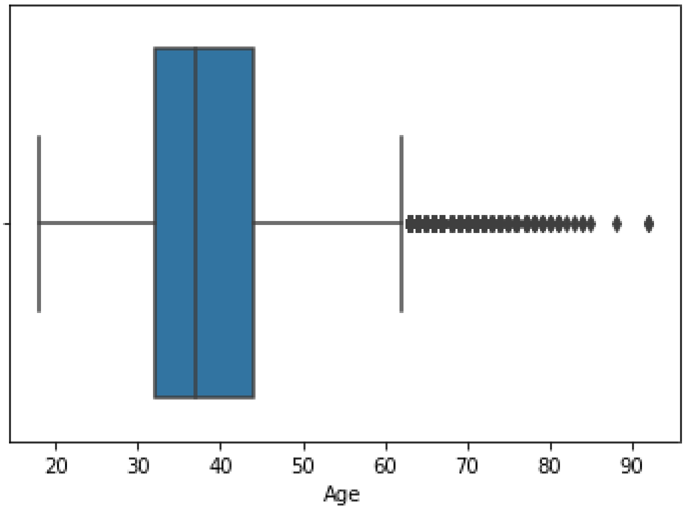
```
RowNumber          0.000000
CustomerId         0.001149
CreditScore       -0.071607
Age                1.011320
```

```
Tenure              0.010991
Balance            -0.141109
NumOfProducts       0.745568
HasCrCard          -0.901812
IsActiveMember     -0.060437
EstimatedSalary     0.002085
Exited              1.471611
dtype: float64
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Droppi
  """Entry point for launching an IPython kernel.
```

Find the outliers and replace the outliers

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c7c87c10>
```



```
q = df.quantile([0.60,0.30])
q
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	
0.6	6000.4	15715686.6	1790.4	678.0	1.0	1.0	37.0	6.0	11(
0.3	3000.7	15641363.9	929.0	598.7	0.0	0.0	30.0	3.0	

```
iqr = q.iloc[0] - q.iloc[1]
iqr
```

```
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
```

```
u = q.iloc[0] + (1.5 * iqr)
u
```

```
Age                5.500000e+01
Balance            3.016053e+05
CreditScore        8.790000e+02
CustomerId         1.590274e+07
EstimatedSalary    2.672892e+05
Exited             0.000000e+00
Gender                      NaN
Geography                   NaN
HasCrCard          2.500000e+00
IsActiveMember     2.500000e+00
NumOfProducts      3.500000e+00
RowNumber          1.349965e+04
Surname                     NaN
Tenure             1.200000e+01
dtype: float64
```

```
l = q.iloc[1] - (1.5*iqr)
l
```

```
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
```

```
df['Age'] = np.where(df['Age'] >50,20,df['Age'])
df['Age'].mean()
```

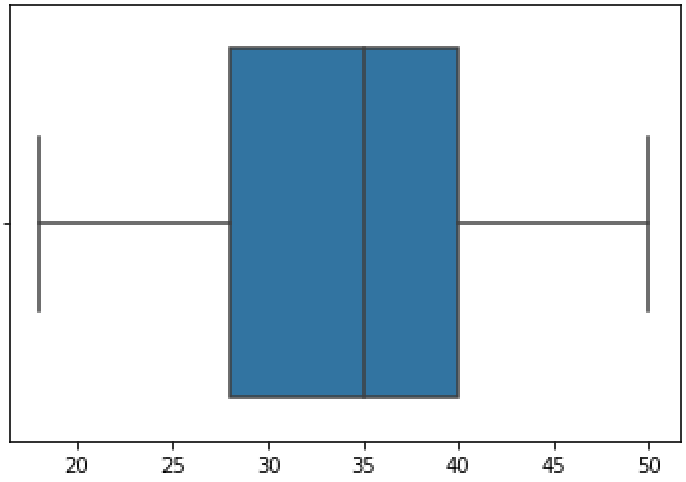
```
33.9313
```

```
df['Age'].median()
```

```
35.0
```

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1c7c072d0>
```



```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder

le = LabelEncoder()
oneh = OneHotEncoder()


df['Gender'] = le.fit_transform(df['Gender'])
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	0	42	2	
1	2	15647311	Hill	608	Spain	0	41	1	838
2	3	15619304	Onio	502	France	0	42	8	1596
3	4	15701354	Boni	699	France	0	39	1	
4	5	15737888	Mitchell	850	Spain	0	43	2	1255

```
df['Geography'] = le.fit_transform(df['Geography'])
df['Surname'] = le.fit_transform(df['Surname'])

df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	1115	619	0	0	42	2	
1	2	15647311	1177	608	2	0	41	1	8380
2	3	15619304	2040	502	0	0	42	8	15960
3	4	15701354	289	699	0	0	39	1	
4	5	15737888	1822	850	2	0	43	2	1255

Split the data into dependent and independent variables.

Independent variables

```
x = df.iloc[:,3:13:1]
x
```

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

10000 rows × 10 columns

Dependent varaiables

```
y=df['Exited']
y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
9995    0
9996    0
9997    1
9998    1
9999    0
Name: Exited, Length: 10000, dtype: int64
```

Scale the independent variables

```
from sklearn.preprocessing import StandardScaler,MinMaxScaler
sc = StandardScaler()
```

```
x_scaled = sc.fit_transform(x)
x_scaled
```

```
array([[-0.32622142, -0.90188624, -1.09598752, ...,  0.64609167,
         0.97024255,  0.02188649],
       [-0.44003595,  1.51506738, -1.09598752, ..., -1.54776799,
         0.97024255,  0.21653375],
       [-1.53679418, -0.90188624, -1.09598752, ...,  0.64609167,
        -1.03067011,  0.2406869 ],
       ...,
       [ 0.60498839, -0.90188624, -1.09598752, ..., -1.54776799,
         0.97024255, -1.00864308],
       [ 1.25683526,  0.30659057,  0.91241915, ...,  0.64609167,
        -1.03067011, -0.12523071],
       [ 1.46377078, -0.90188624, -1.09598752, ...,  0.64609167,
        -1.03067011, -1.07636976]])
```

Split the data into training and testing

```
from sklearn.model_selection import train_test_split
```

```
xtrain,xtest,ytrain,ytest = train_test_split(x_scaled,y,test_size = 0.3,random_state =0)
```

```
xtrain.shape
```

```
(7000, 10)
```

```
xtrain
```

```
array([[-0.09859236, -0.90188624,  0.91241915, ...,  0.64609167,
         0.97024255, -0.77021814],
       [-1.13326993, -0.90188624,  0.91241915, ...,  0.64609167,
        -1.03067011, -1.39576675],
       [-0.62627792, -0.90188624, -1.09598752, ..., -1.54776799,
         0.97024255, -1.49965629],
       ...,
       [ 0.90504489, -0.90188624,  0.91241915, ...,  0.64609167,
        -1.03067011,  1.41441489],
       [-0.62627792,  1.51506738, -1.09598752, ...,  0.64609167,
         0.97024255,  0.84614739],
       [-0.28483432,  0.30659057, -1.09598752, ...,  0.64609167,
        -1.03067011,  0.32630495]])
```

```
xtest
```

```
array([[-0.55385049,  0.30659057, -1.09598752, ...,  0.64609167,
         0.97024255,  1.61304597],
       [-1.31951189, -0.90188624, -1.09598752, ...,  0.64609167,
        -1.03067011,  0.49753166],
       [ 0.57394806,  1.51506738, -1.09598752, ...,  0.64609167,
         0.97024255, -0.4235611 ],
       ...,
       [ 0.35666577, -0.90188624,  0.91241915, ...,  0.64609167,
         0.97024255,  1.17045451],
       [ 0.4290932 , -0.90188624,  0.91241915, ...,  0.64609167,
          0.97024255, -0.50846777],
        [ 0.83261746,  0.30659057, -1.09598752, ...,  0.64609167,
          0.97024255, -1.15342685]])
```