**Load the Dataset**

In [1]:

**import** pandas **as** pd

**import** seaborn **as** sns

**import** numpy **as** np

**from** matplotlib **import** pyplot **as** plt

**%**matplotlib inline

df **=** pd**.**read\_csv("Churn\_Modelling.csv")

df

Out[1]:

|  | **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 |
| **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

**3.1 Univariate Analysis**

In [ ]:

sns**.**displot(df**.**Gender)

Out[ ]:

<seaborn.axisgrid.FacetGrid at 0x7f1ecbb27b90>

In [23]:

sns**.**histplot(data['Age'])

Out[23]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f466da262d0>

**3.2 Bi-Variate Analysis**

In [24]:

sns**.**lineplot(data['Age'], data['EstimatedSalary'])

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[24]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f466d8ebbd0>

**3.3 Multi-variate Analysis**

In [27]:

sns**.**scatterplot(data['Age'], data['EstimatedSalary'], hue **=** data['Gender'])

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[27]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f466662f390>

In [26]:

sns**.**pairplot(data)

Out[26]:

<seaborn.axisgrid.PairGrid at 0x7f466d80f110>

**4. Descriptive statistic on the dataset**

In [ ]:

df**.**describe()

Out[ ]:

|  | **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**

In [19]:

data **=** pd**.**read\_csv("Churn\_Modelling.csv")

data**.**isnull()**.**any()

Out[19]:

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 [20]:

data**.**isnull()**.**sum()

Out[20]:

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

**6. Find the outliers and replace the outliers**

In [2]:

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

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[2]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f466f79a9d0>

In [3]:

df['Age']**=**np**.**where(df['Age']**>**50,40,df['Age'])

df['Age']

Out[3]:

0 42

1 41

2 42

3 39

4 43

..

9995 39

9996 35

9997 36

9998 42

9999 28

Name: Age, Length: 10000, dtype: int64

In [4]:

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

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[4]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f466f6f7290>

In [5]:

df['Age']**=**np**.**where(df['Age']**<**20,35,df['Age'])

df['Age']

Out[5]:

0 42

1 41

2 42

3 39

4 43

..

9995 39

9996 35

9997 36

9998 42

9999 28

Name: Age, Length: 10000, dtype: int64

**7. Check for Categorical columns and perform encoding.**

In [8]:

pd**.**get\_dummies(df, columns**=**["Gender", "Age"], prefix**=**["Age", "Gender"])**.**head()

Out[8]:

|  | **RowNumber** | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **...** | **Gender\_41** | **Gender\_42** | **Gender\_43** | **Gender\_44** | **Gender\_45** | **Gender\_46** | **Gender\_47** | **Gender\_48** | **Gender\_49** | **Gender\_50** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 15634602 | Hargrave | 619 | France | 2 | 0.00 | 1 | 1 | 1 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 2 | 15647311 | Hill | 608 | Spain | 1 | 83807.86 | 1 | 0 | 1 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 3 | 15619304 | Onio | 502 | France | 8 | 159660.80 | 3 | 1 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 4 | 15701354 | Boni | 699 | France | 1 | 0.00 | 2 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 5 | 15737888 | Mitchell | 850 | Spain | 2 | 125510.82 | 1 | 1 | 1 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 45 columns

**8. Split the data into indepentent variables**

In [9]:

X **=** df**.**iloc[:, :**-**1]**.**values

print(X)

[[1 15634602 'Hargrave' ... 1 1 101348.88]

[2 15647311 'Hill' ... 0 1 112542.58]

[3 15619304 'Onio' ... 1 0 113931.57]

...

[9998 15584532 'Liu' ... 0 1 42085.58]

[9999 15682355 'Sabbatini' ... 1 0 92888.52]

[10000 15628319 'Walker' ... 1 0 38190.78]]

**8.1 Split the data into dependent variables**

In [10]:

Y **=** df**.**iloc[:, **-**1]**.**values

print(Y)

[1 0 1 ... 1 1 0]

**9. Scale the independent variables**

In [13]:

**import** pandas **as** pd

**from** sklearn.preprocessing **import** MinMaxScaler

scaler **=** MinMaxScaler()

df[["CustomerId"]] **=** scaler**.**fit\_transform(df[["CustomerId"]])

print(df)

RowNumber CustomerId Surname CreditScore Geography Gender Age \

0 1 0.275616 Hargrave 619 France Female 42

1 2 0.326454 Hill 608 Spain Female 41

2 3 0.214421 Onio 502 France Female 42

3 4 0.542636 Boni 699 France Female 39

4 5 0.688778 Mitchell 850 Spain Female 43

... ... ... ... ... ... ... ...

9995 9996 0.162119 Obijiaku 771 France Male 39

9996 9997 0.016765 Johnstone 516 France Male 35

9997 9998 0.075327 Liu 709 France Female 36

9998 9999 0.466637 Sabbatini 772 Germany Male 42

9999 10000 0.250483 Walker 792 France Female 28

Tenure Balance NumOfProducts HasCrCard IsActiveMember \

0 2 0.00 1 1 1

1 1 83807.86 1 0 1

2 8 159660.80 3 1 0

3 1 0.00 2 0 0

4 2 125510.82 1 1 1

... ... ... ... ... ...

9995 5 0.00 2 1 0

9996 10 57369.61 1 1 1

9997 7 0.00 1 0 1

9998 3 75075.31 2 1 0

9999 4 130142.79 1 1 0

EstimatedSalary Exited

0 101348.88 1

1 112542.58 0

2 113931.57 1

3 93826.63 0

4 79084.10 0

... ... ...

9995 96270.64 0

9996 101699.77 0

9997 42085.58 1

9998 92888.52 1

9999 38190.78 0

[10000 rows x 14 columns]

**10. Split the data into training and testing**

In [17]:

**from** sklearn.model\_selection **import** train\_test\_split

x **=** df**.**drop(columns **=** ['Tenure'])**.**copy()

y **=** df['Tenure']

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,test\_size**=**0.2,random\_state**=**0)

x\_train**.**head()

Out[17]:

|  | **RowNumber** | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **7389** | 7390 | 0.444852 | Mishin | 667 | Spain | Female | 34 | 0.00 | 2 | 1 | 0 | 163830.64 | 0 |
| **9275** | 9276 | 0.734288 | Carslaw | 427 | Germany | Male | 42 | 75681.52 | 1 | 1 | 1 | 57098.00 | 0 |
| **2995** | 2996 | 0.067167 | Moore | 535 | France | Female | 29 | 112367.34 | 1 | 1 | 0 | 185630.76 | 0 |
| **5316** | 5317 | 0.858778 | Ferri | 654 | Spain | Male | 40 | 105683.63 | 1 | 1 | 0 | 173617.09 | 0 |
| **356** | 357 | 0.184240 | Simmons | 850 | Spain | Female | 40 | 126776.30 | 2 | 1 | 1 | 132298.49 | 0 |

In [18]:

x\_train**.**shape,y\_train**.**shape,x\_test**.**shape,y\_test**.**shape

Out[18]:

((8000, 13), (8000,), (2000, 13), (2000,))