**Load the data set**

In [1]:

**from** google.colab **import** drive

drive**.**mount('/content/drive')

Mounted at /content/drive

**IMPORT LIBRARIES**

In [2]:

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

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

**import** matplotlib.pyplot **as** plt

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

**from** sklearn.preprocessing **import** LabelEncoder,MinMaxScaler

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

In [5]:

data **=** pd**.**read\_csv('/content/drive/MyDrive/IBM PROJECT/assignment 2/Churn\_Modelling.csv')

data **=** data**.**iloc[:,3:]

data

Out[5]:

|  | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 619 | France | Female | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| **1** | 608 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| **2** | 502 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| **3** | 699 | France | Female | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| **4** | 850 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | 771 | France | Male | 39 | 5 | 0.00 | 2 | 1 | 0 | 96270.64 | 0 |
| **9996** | 516 | France | Male | 35 | 10 | 57369.61 | 1 | 1 | 1 | 101699.77 | 0 |
| **9997** | 709 | France | Female | 36 | 7 | 0.00 | 1 | 0 | 1 | 42085.58 | 1 |
| **9998** | 772 | Germany | Male | 42 | 3 | 75075.31 | 2 | 1 | 0 | 92888.52 | 1 |
| **9999** | 792 | France | Female | 28 | 4 | 130142.79 | 1 | 1 | 0 | 38190.78 | 0 |

10000 rows × 11 columns

**VISUALIZATIONS**

**1.Univariate Analysis**

In [6]:

**for** col **in** data**.**columns:

**if**(data**.**dtypes[col]**==**'int64' **or** data**.**dtypes[col]**==**'float64' ):

sns**.**boxplot(x**=**data[col])**.**set( xlabel**=**col)

plt**.**show()

**2.Bi-VARIATE ANALYSIS**

In [7]:

sns**.**FacetGrid(data,hue**=**'Exited',size**=**5)**.**map(plt**.**scatter,"Balance","CreditScore")**.**add\_legend()

plt**.**show()

/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning: The `size` parameter has been renamed to `height`; please update your code.

warnings.warn(msg, UserWarning)

**3.MULTI-VARIATE ANALYSIS**

In [8]:

sns**.**pairplot(data, hue**=**'Exited', height**=**2)

Out[8]:

**4.DESCRIPTIVE STATISTICS OF DATA SET**

In [9]:

data**.**describe()

Out[9]:

|  | **CreditScore** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | 10000.000000 | 10000.000000 | 10000.000000 |
| **mean** | 650.528800 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 | 0.203700 |
| **std** | 96.653299 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 | 0.402769 |
| **min** | 350.000000 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 | 0.000000 |
| **25%** | 584.000000 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 51002.110000 | 0.000000 |
| **50%** | 652.000000 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.00000 | 1.000000 | 100193.915000 | 0.000000 |
| **75%** | 718.000000 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.00000 | 1.000000 | 149388.247500 | 0.000000 |
| **max** | 850.000000 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.00000 | 1.000000 | 199992.480000 | 1.000000 |

**5.HANDLE THE MISSING VALUES**

In [10]:

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

Out[10]:

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 [11]:

CreditsMedian **=** data**.**loc[data['CreditScore']**<**400, 'CreditScore']**.**median()

ProdMedian **=** data**.**loc[data['NumOfProducts']**>=**3.5,'NumOfProducts']**.**median()

data**.**loc[data**.**CreditScore **<** 400, 'CreditScore'] **=** np**.**nan

data**.**fillna(CreditsMedian,inplace**=True**)

data**.**loc[data**.**NumOfProducts **>** 3, 'NumOfProducts'] **=** np**.**nan

data**.**fillna(ProdMedian,inplace**=True**)

**7.CHECK FOR CATEGORICAL COLUMNS AND PERFORM ENCODING**

In [12]:

labelencoder **=** LabelEncoder()

data['Geography']**=** labelencoder**.**fit\_transform(data['Geography'])

data['Gender'] **=** labelencoder**.**fit\_transform(data['Gender'])

**8.SPLITTING OF DATA**

In [13]:

independent **=** data**.**iloc[:, :**-**1]

dependent **=** data**.**iloc[:,**-**1:]

**9.SCALE THE INDEPENDENT VALUES**

In [14]:

nm **=**MinMaxScaler()

N\_independent **=** nm**.**fit\_transform(independent)

**10.SPLIT THE DATA INTO TRAINING AND TESTING**

In [15]:

xtrain,xtest,ytrain,ytest**=**train\_test\_split(N\_independent,dependent,test\_size**=**0.3)

print(xtrain,xtest,ytrain,ytest)

[[0.47628866 0. 0. ... 1. 0. 0.42704558]

[0.82061856 0. 0. ... 0. 1. 0.1790851 ]

[0.70721649 0. 1. ... 1. 0. 0.64200651]

...

[0.74020619 0. 0. ... 1. 1. 0.31137639]

[0.40206186 0. 1. ... 1. 1. 0.50036538]

[0.48247423 0.5 1. ... 0. 1. 0.88073666]] [[0.58556701 0. 1. ... 0. 1. 0.44481188]

[0.4371134 1. 1. ... 1. 1. 0.54896608]

[0.74020619 0.5 1. ... 1. 1. 0.68451752]

...

[0.81649485 0.5 1. ... 0. 0. 0.43458845]

[0.83298969 1. 1. ... 1. 1. 0.23857883]

[0.88659794 1. 1. ... 0. 0. 0.77857465]] Exited

2410 0

2562 0

8386 0

4816 0

8864 0

... ...

8673 1

4824 0

8930 0

8880 1

4091 0

[7000 rows x 1 columns] Exited

2715 0

8245 0

6568 0

290 0

9704 0

... ...

5906 0

6311 0

8768 1

1737 0

8373 0

[3000 rows x 1 columns]