**import** pandas **as** pd  
import numpy **as** np  
import matplotlib.pyplot **as** plt  
import seaborn **as** sns

# **2. Load the dataset.**

In [3]:

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

Out[3]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

# **3. Perform Below Visualizations.**

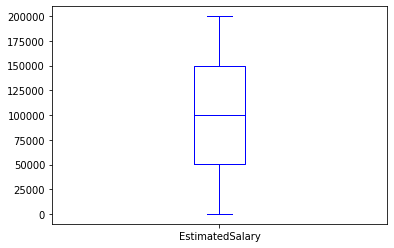
# Univariate Analysis

In [ ]:

data**.**boxplot(column**=**['EstimatedSalary'], grid**=False**, color**=**'blue')

Out[ ]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fe0c7bbd950>

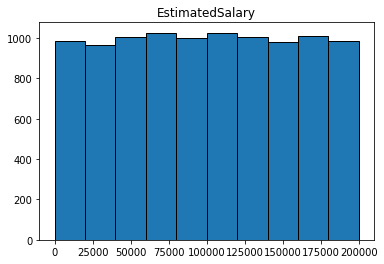


In [12]:

data**.**hist(column**=**'EstimatedSalary', grid**=False**, edgecolor**=**'black')

Out[12]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f4f02695090>]],  
 dtype=object)

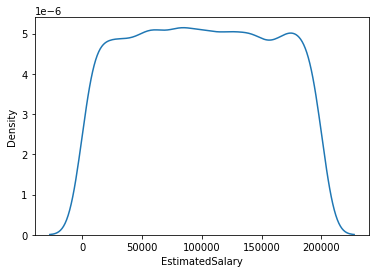


In [13]:

sns**.**kdeplot(data['EstimatedSalary'])

Out[13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4f0261cf90>

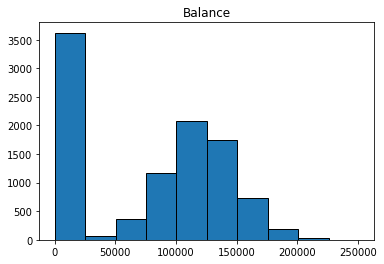


In [14]:

data**.**hist(column**=**'Balance', grid**=False**, edgecolor**=**'black')

Out[14]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f4f025ab890>]],  
 dtype=object)

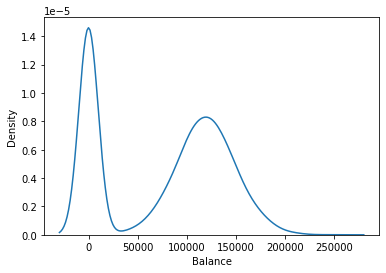


In [15]:

**import** seaborn **as** sns  
  
sns**.**kdeplot(data['Balance'])

Out[15]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4f02f08250>



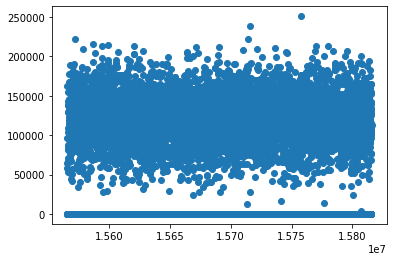
# Bi - Variate Analysis

In [16]:

plt**.**scatter(data**.**CustomerId, data**.**Balance )

Out[16]:

<matplotlib.collections.PathCollection at 0x7f4f02ec8e50>



In [17]:

data**.**corr()

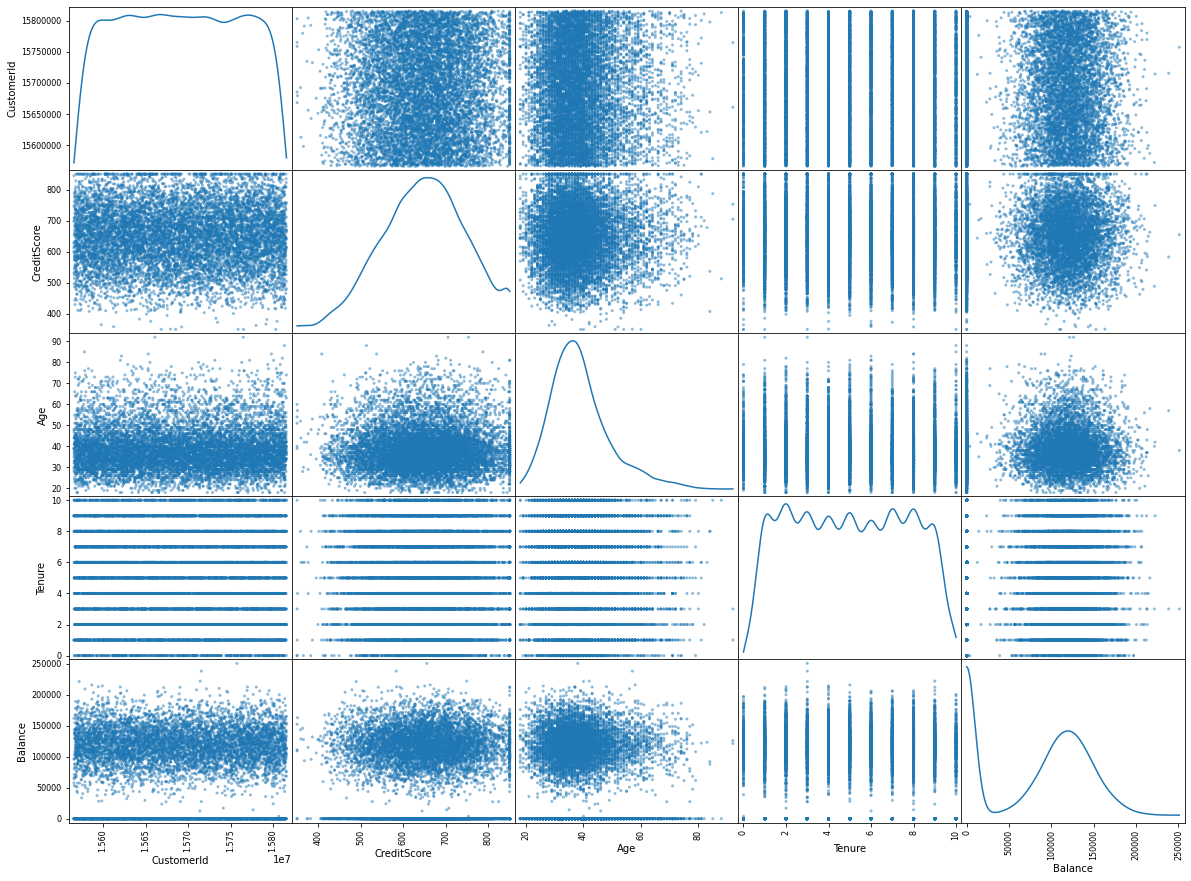
Out[17]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RowNumber** | **CustomerId** | **CreditScore** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| **RowNumber** | 1.000000 | 0.004202 | 0.005840 | 0.000783 | -0.006495 | -0.009067 | 0.007246 | 0.000599 | 0.012044 | -0.005988 | -0.016571 |
| **CustomerId** | 0.004202 | 1.000000 | 0.005308 | 0.009497 | -0.014883 | -0.012419 | 0.016972 | -0.014025 | 0.001665 | 0.015271 | -0.006248 |
| **CreditScore** | 0.005840 | 0.005308 | 1.000000 | -0.003965 | 0.000842 | 0.006268 | 0.012238 | -0.005458 | 0.025651 | -0.001384 | -0.027094 |
| **Age** | 0.000783 | 0.009497 | -0.003965 | 1.000000 | -0.009997 | 0.028308 | -0.030680 | -0.011721 | 0.085472 | -0.007201 | 0.285323 |
| **Tenure** | -0.006495 | -0.014883 | 0.000842 | -0.009997 | 1.000000 | -0.012254 | 0.013444 | 0.022583 | -0.028362 | 0.007784 | -0.014001 |
| **Balance** | -0.009067 | -0.012419 | 0.006268 | 0.028308 | -0.012254 | 1.000000 | -0.304180 | -0.014858 | -0.010084 | 0.012797 | 0.118533 |
| **NumOfProducts** | 0.007246 | 0.016972 | 0.012238 | -0.030680 | 0.013444 | -0.304180 | 1.000000 | 0.003183 | 0.009612 | 0.014204 | -0.047820 |
| **HasCrCard** | 0.000599 | -0.014025 | -0.005458 | -0.011721 | 0.022583 | -0.014858 | 0.003183 | 1.000000 | -0.011866 | -0.009933 | -0.007138 |
| **IsActiveMember** | 0.012044 | 0.001665 | 0.025651 | 0.085472 | -0.028362 | -0.010084 | 0.009612 | -0.011866 | 1.000000 | -0.011421 | -0.156128 |
| **EstimatedSalary** | -0.005988 | 0.015271 | -0.001384 | -0.007201 | 0.007784 | 0.012797 | 0.014204 | -0.009933 | -0.011421 | 1.000000 | 0.012097 |
| **Exited** | -0.016571 | -0.006248 | -0.027094 | 0.285323 | -0.014001 | 0.118533 | -0.047820 | -0.007138 | -0.156128 | 0.012097 | 1.000000 |

# Multi - Variate Analysis

In [19]:

pd**.**plotting**.**scatter\_matrix(data**.**loc[:, "CustomerId":"Balance"], diagonal**=**"kde",figsize**=**(20,15))  
plt**.**show()



# **4. Perform descriptive statistics on the dataset.**

In [51]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**mean()

Out[51]:

CreditScore 140.000000  
Balance 76485.889288  
EstimatedSalary 100090.239881  
dtype: float64

In [52]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**median()

Out[52]:

CreditScore 140.000  
Balance 97198.540  
EstimatedSalary 100193.915  
dtype: float64

In [53]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**mode()

Out[53]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CreditScore** | **Balance** | **EstimatedSalary** |
| **0** | 140 | 0.0 | 24924.92 |

In [54]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**quantile()

Out[54]:

CreditScore 140.000  
Balance 97198.540  
EstimatedSalary 100193.915  
Name: 0.5, dtype: float64

In [57]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**std()

Out[57]:

CreditScore 0.000000  
Balance 62397.405202  
EstimatedSalary 57510.492818  
dtype: float64

In [58]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**min()

Out[58]:

CreditScore 140.00  
Balance 0.00  
EstimatedSalary 11.58  
dtype: float64

In [59]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**max()

Out[59]:

CreditScore 140.00  
Balance 250898.09  
EstimatedSalary 199992.48  
dtype: float64

In [60]:

data[['CreditScore', 'Balance', 'EstimatedSalary']]**.**skew()

Out[60]:

CreditScore 0.000000  
Balance -0.141109  
EstimatedSalary 0.002085  
dtype: float64

In [20]:

data**.**info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10000 entries, 0 to 9999  
Data columns (total 14 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 RowNumber 10000 non-null int64   
 1 CustomerId 10000 non-null int64   
 2 Surname 10000 non-null object   
 3 CreditScore 10000 non-null int64   
 4 Geography 10000 non-null object   
 5 Gender 10000 non-null object   
 6 Age 10000 non-null int64   
 7 Tenure 10000 non-null int64   
 8 Balance 10000 non-null float64  
 9 NumOfProducts 10000 non-null int64   
 10 HasCrCard 10000 non-null int64   
 11 IsActiveMember 10000 non-null int64   
 12 EstimatedSalary 10000 non-null float64  
 13 Exited 10000 non-null int64   
dtypes: float64(2), int64(9), object(3)  
memory usage: 1.1+ MB

In [21]:

data**.**shape

Out[21]:

(10000, 14)

In [ ]:

data**.**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**.

There is no missing values

In [22]:

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

Out[22]:

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 [28]:

data**.**describe()

Out[28]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

In [26]:

numeric\_col **=** ['RowNumber CustomerId','CreditScore',' Age', 'Tenure', 'Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','Exited']  
categorical\_col **=** ['Surname', 'Geography', 'Gender']

In [29]:

print(data['CreditScore']**.**skew())  
data['CreditScore']**.**describe()

-0.07160660820092675

Out[29]:

count 10000.000000  
mean 650.528800  
std 96.653299  
min 350.000000  
25% 584.000000  
50% 652.000000  
75% 718.000000  
max 850.000000  
Name: CreditScore, dtype: float64

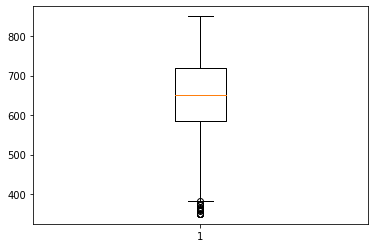
In [31]:

Q1 **=** data**.**quantile(0.25)  
Q3 **=** data**.**quantile(0.75)  
IQR **=** Q3 **-** Q1  
print(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

In [36]:

plt**.**boxplot(data["CreditScore"])  
plt**.**show()



In [37]:

print(data['CreditScore']**.**quantile(0.50))   
print(data['CreditScore']**.**quantile(0.95))   
data['CreditScore'] **=** np**.**where(data['CreditScore'] **>** 325, 140, data['CreditScore'])  
data**.**describe()

652.0  
812.0

Out[37]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RowNumber** | **CustomerId** | **CreditScore** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| **count** | 10000.00000 | 1.000000e+04 | 10000.0 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | 10000.000000 | 10000.000000 | 10000.000000 |
| **mean** | 5000.50000 | 1.569094e+07 | 140.0 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 | 0.203700 |
| **std** | 2886.89568 | 7.193619e+04 | 0.0 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 | 0.402769 |
| **min** | 1.00000 | 1.556570e+07 | 140.0 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 | 0.000000 |
| **25%** | 2500.75000 | 1.562853e+07 | 140.0 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 51002.110000 | 0.000000 |
| **50%** | 5000.50000 | 1.569074e+07 | 140.0 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.00000 | 1.000000 | 100193.915000 | 0.000000 |
| **75%** | 7500.25000 | 1.575323e+07 | 140.0 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.00000 | 1.000000 | 149388.247500 | 0.000000 |
| **max** | 10000.00000 | 1.581569e+07 | 140.0 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.00000 | 1.000000 | 199992.480000 | 1.000000 |

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

In [38]:

X **=** data**.**iloc[:, 3:13]**.**values  
y **=** data**.**iloc[:, 13]**.**values

In [39]:

**from** sklearn.preprocessing **import** LabelEncoder, OneHotEncoder  
from sklearn.compose **import** ColumnTransformer  
  
labelencoder\_X\_1 **=** LabelEncoder()  
X[:, 1] **=** labelencoder\_X\_1**.**fit\_transform(X[:, 1])  
labelencoder\_X\_2 **=** LabelEncoder()  
X[:, 2] **=** labelencoder\_X\_2**.**fit\_transform(X[:, 2])  
  
# remove categorical\_features, it works 100% perfectly  
onehotencoder **=** OneHotEncoder()  
X **=** onehotencoder**.**fit\_transform(X)**.**toarray()  
X **=** X[:, 1:]

# **8. Split the data into dependent and independent variables.**

In [40]:

X**=** data**.**iloc[:,3:**-**1]  
y**=**data**.**iloc[:,**-**1]  
X**.**head()

Out[40]:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** |
| **0** | 140 | France | Female | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 |
| **1** | 140 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 |
| **2** | 140 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 |
| **3** | 140 | France | Female | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 |
| **4** | 140 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 |

In [41]:

X **=** data**.**iloc[:, 3:13]**.**values  
y **=** data**.**iloc[:, 13]**.**values

In [42]:

print(X)

[[140 'France' 'Female' ... 1 1 101348.88]  
 [140 'Spain' 'Female' ... 0 1 112542.58]  
 [140 'France' 'Female' ... 1 0 113931.57]  
 ...  
 [140 'France' 'Female' ... 0 1 42085.58]  
 [140 'Germany' 'Male' ... 1 0 92888.52]  
 [140 'France' 'Female' ... 1 0 38190.78]]

In [43]:

print(y)

[1 0 1 ... 1 1 0]

In [49]:

**from** sklearn.model\_selection **import** train\_test\_split  
x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x, y, test\_size **=** 0.25, random\_state **=** 0)  
  
print(x\_train**.**shape)  
print(y\_train**.**shape)  
print(x\_test**.**shape)  
print(y\_test**.**shape)

(7500, 2)  
(7500,)  
(2500, 2)  
(2500,)

# **9. Scale the independant variables**

In [50]:

**from** sklearn.preprocessing **import** StandardScaler  
  
sc **=** StandardScaler()  
x\_train **=** sc**.**fit\_transform(x\_train)  
x\_test **=** sc**.**fit\_transform(x\_test)  
  
x\_train **=** pd**.**DataFrame(x\_train)  
x\_train**.**head()

Out[50]:

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **0** | 0.0 | -1.343330 |
| **1** | 0.0 | 1.558330 |
| **2** | 0.0 | -0.655156 |
| **3** | 0.0 | 1.200594 |
| **4** | 0.0 | 0.778798 |

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

In [63]:

**from** sklearn.model\_selection **import** train\_test\_split  
x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x, y, test\_size **=** 0.25, random\_state **=** 0)  
  
print(x\_train**.**shape)  
print(y\_train**.**shape)  
print(x\_test**.**shape)  
print(y\_test**.**shape)

(7500, 2)  
(7500,)  
(2500, 2)  
(2500,)