**Importing the dataset**

In [1]:

*#Importing Required Modules*

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

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

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

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

**%**matplotlib inline

dataset **=** pd**.**read\_csv('/content/Churn\_Modelling.csv')

In [2]:

dataset**.**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 [3]:

dataset**.**describe()

Out[3]:

|  | **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 |

**Uni-variate Analysis**

In [4]:

*#Boxplot*

sns**.**boxplot(dataset['CreditScore'])

/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 0x7f491c424fd0>

In [5]:

*#Displot*

sns**.**displot(dataset['Age'],color**=**'red')

Out[5]:

<seaborn.axisgrid.FacetGrid at 0x7f491c30d250>

In [6]:

*#Histogram*

sns**.**histplot(y**=**'Balance',data**=**dataset,color**=**'blue')

Out[6]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f491959ce10>

In [7]:

*#Histogram for categorical values*

sns**.**histplot(y**=**'CreditScore',data**=**dataset,hue**=**dataset['Gender'])

Out[7]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4919473890>

In [8]:

*#This is distplot along with histogram*

sns**.**distplot(dataset['Age'])

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

Out[8]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f49192b3990>

In [9]:

*#Count plot*

sns**.**countplot(dataset['Tenure'])

/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[9]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f49191fb510>

**Bi-Variate Analysis**

In [10]:

*#Barplot*

sns**.**barplot(dataset['Geography'],dataset['NumOfProducts'])

/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[10]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f491915c1d0>

In [11]:

*#Barplot using mathplotlib*

plt**.**bar(dataset['Gender'],dataset['Age'])

Out[11]:

<BarContainer object of 10000 artists>

In [12]:

*#Linear plot*

sns**.**lineplot(dataset['Age'],dataset['Tenure'],color**=**'green')

/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[12]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f491263e750>

In [13]:

*#Scatterplot*

sns**.**scatterplot(dataset['Age'],dataset['CreditScore'],color**=**'blue')

/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[13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4912459490>

In [14]:

*#Point plot*

sns**.**pointplot(dataset['Age'],dataset['Balance'],color**=**'yellow')

/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[14]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f491221b290>

In [15]:

*#Regplot*

sns**.**regplot(dataset['Balance'],dataset['Tenure'],color**=**'red')

/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[15]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4918fde310>

**Multi-Variate Analysis**

In [16]:

*#Pair plot*

sns**.**pairplot(data**=**dataset[['Age','Balance','Surname','NumOfProducts']], kind**=**'kde')

Out[16]:

<seaborn.axisgrid.PairGrid at 0x7f4919542a10>

In [17]:

sns**.**pairplot(data**=**dataset[["RowNumber","Age","Tenure","Balance","NumOfProducts"]], hue**=**"Age", diag\_kind**=**"hist")

Out[17]:

<seaborn.axisgrid.PairGrid at 0x7f49180659d0>

In [18]:

sns**.**pairplot(data**=**dataset[["RowNumber","Age","Tenure","Balance","NumOfProducts"]], hue**=**"Age")

Out[18]:

<seaborn.axisgrid.PairGrid at 0x7f490c78de90>

**Performing descriptive statistics on the dataset**

In [19]:

dataset**.**describe()

Out[19]:

|  | **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 [20]:

dataset**.**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

In [21]:

dataset['Tenure']**.**mean()

Out[21]:

5.0128

In [22]:

dataset['Age']**.**median()

Out[22]:

37.0

In [23]:

dataset['EstimatedSalary']**.**mode()

Out[23]:

0 24924.92

dtype: float64

**Handling the Missing values**

In [24]:

dataset['EstimatedSalary']**.**isnull()**.**any()

Out[24]:

False

In [25]:

*#If there is any missing value in EstimatedSalary column, it will be replaced by the mean using the below statement*

dataset['EstimatedSalary']**=**dataset['EstimatedSalary']**.**fillna(dataset['EstimatedSalary']**.**mean())

In [26]:

dataset['EstimatedSalary']**.**isnull()**.**sum()**/**len(dataset)

Out[26]:

0.0

**Finding the outliers and replacing the outliers**

In [27]:

outliers **=** dataset**.**quantile(q**=**(0.3,0.85))

outliers

Out[27]:

|  | **RowNumber** | **CustomerId** | **CreditScore** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0.30** | 3000.70 | 15641363.90 | 598.7 | 33.0 | 3.0 | 0.0000 | 1.0 | 1.0 | 0.0 | 60736.0790 | 0.0 |
| **0.85** | 8500.15 | 15777904.45 | 754.0 | 49.0 | 8.0 | 140895.0965 | 2.0 | 1.0 | 1.0 | 170322.3935 | 1.0 |

In [28]:

outliers\_removed **=** outliers**.**loc[0.85]**-**outliers**.**loc[0.3]

outliers\_removed

Out[28]:

RowNumber 5499.4500

CustomerId 136540.5500

CreditScore 155.3000

Age 16.0000

Tenure 5.0000

Balance 140895.0965

NumOfProducts 1.0000

HasCrCard 0.0000

IsActiveMember 1.0000

EstimatedSalary 109586.3145

Exited 1.0000

dtype: float64

In [29]:

low\_val **=** outliers**.**loc[0.3] **-** 1.5 **\*** outliers\_removed

low\_val

Out[29]:

RowNumber -5.248475e+03

CustomerId 1.543655e+07

CreditScore 3.657500e+02

Age 9.000000e+00

Tenure -4.500000e+00

Balance -2.113426e+05

NumOfProducts -5.000000e-01

HasCrCard 1.000000e+00

IsActiveMember -1.500000e+00

EstimatedSalary -1.036434e+05

Exited -1.500000e+00

dtype: float64

In [30]:

high\_val **=** outliers**.**loc[0.85] **+** 1.5 **\*** outliers\_removed

high\_val

Out[30]:

RowNumber 1.674932e+04

CustomerId 1.598272e+07

CreditScore 9.869500e+02

Age 7.300000e+01

Tenure 1.550000e+01

Balance 3.522377e+05

NumOfProducts 3.500000e+00

HasCrCard 1.000000e+00

IsActiveMember 2.500000e+00

EstimatedSalary 3.347019e+05

Exited 2.500000e+00

dtype: float64

In [31]:

sns**.**boxplot(dataset["Age"],color**=**'blue')

/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[31]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f49091ed610>

In [32]:

dataset["Age"]**=**np**.**where(dataset["Age"]**<**29,50,dataset["Age"])

In [33]:

sns**.**boxplot(dataset["Age"],color**=**'red')

/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[33]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f49191c2e10>

**Checking for categorical data and encoding them**

In [34]:

dataset**.**head(5)

Out[34]:

|  | **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 |

In [39]:

**from** sklearn.preprocessing **import** LabelEncoder

Encoder **=** LabelEncoder()

dataset['Gender'] **=** Encoder**.**fit\_transform(dataset['Gender'])

dataset['Geography'] **=** Encoder**.**fit\_transform(dataset['Geography'])

dataset['Surname'] **=** Encoder**.**fit\_transform(dataset['Surname'])

In [40]:

dataset**.**head()

Out[40]:

|  | **RowNumber** | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** | **Exited** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 15634602 | 1115 | 619 | 0 | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| **1** | 2 | 15647311 | 1177 | 608 | 2 | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| **2** | 3 | 15619304 | 2040 | 502 | 0 | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| **3** | 4 | 15701354 | 289 | 699 | 0 | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| **4** | 5 | 15737888 | 1822 | 850 | 2 | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |

**Splitting the data into independent and dependent data**

In [45]:

x **=** dataset**.**drop(columns**=**['Exited','RowNumber'],axis**=**1)

y **=** dataset['Exited']

In [46]:

x**.**head()

Out[46]:

|  | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 15634602 | 1115 | 619 | 0 | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 |
| **1** | 15647311 | 1177 | 608 | 2 | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 |
| **2** | 15619304 | 2040 | 502 | 0 | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 |
| **3** | 15701354 | 289 | 699 | 0 | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 |
| **4** | 15737888 | 1822 | 850 | 2 | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 |

**Scaling the values**

In [47]:

**from** sklearn.preprocessing **import** scale

X **=** scale(x)

X

Out[47]:

array([[-0.78321342, -0.46418322, -0.32622142, ..., 0.64609167,

0.97024255, 0.02188649],

[-0.60653412, -0.3909112 , -0.44003595, ..., -1.54776799,

0.97024255, 0.21653375],

[-0.99588476, 0.62898807, -1.53679418, ..., 0.64609167,

-1.03067011, 0.2406869 ],

...,

[-1.47928179, 0.07353887, 0.60498839, ..., -1.54776799,

0.97024255, -1.00864308],

[-0.11935577, 0.98943914, 1.25683526, ..., 0.64609167,

-1.03067011, -0.12523071],

[-0.87055909, 1.4692527 , 1.46377078, ..., 0.64609167,

-1.03067011, -1.07636976]])

In [48]:

x **=** pd**.**DataFrame(X,columns **=** x**.**columns)

x

Out[48]:

|  | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | -0.783213 | -0.464183 | -0.326221 | -0.901886 | -1.095988 | -0.014933 | -1.041760 | -1.225848 | -0.911583 | 0.646092 | 0.970243 | 0.021886 |
| **1** | -0.606534 | -0.390911 | -0.440036 | 1.515067 | -1.095988 | -0.120395 | -1.387538 | 0.117350 | -0.911583 | -1.547768 | 0.970243 | 0.216534 |
| **2** | -0.995885 | 0.628988 | -1.536794 | -0.901886 | -1.095988 | -0.014933 | 1.032908 | 1.333053 | 2.527057 | 0.646092 | -1.030670 | 0.240687 |
| **3** | 0.144767 | -1.440356 | 0.501521 | -0.901886 | -1.095988 | -0.331319 | -1.387538 | -1.225848 | 0.807737 | -1.547768 | -1.030670 | -0.108918 |
| **4** | 0.652659 | 0.371354 | 2.063884 | 1.515067 | -1.095988 | 0.090529 | -1.041760 | 0.785728 | -0.911583 | 0.646092 | 0.970243 | -0.365276 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | -1.177652 | 0.580534 | 1.246488 | -0.901886 | 0.912419 | -0.331319 | -0.004426 | -1.225848 | 0.807737 | 0.646092 | -1.030670 | -0.066419 |
| **9996** | -1.682806 | -0.203004 | -1.391939 | -0.901886 | 0.912419 | -0.753167 | 1.724464 | -0.306379 | -0.911583 | 0.646092 | 0.970243 | 0.027988 |
| **9997** | -1.479282 | 0.073539 | 0.604988 | -0.901886 | -1.095988 | -0.647705 | 0.687130 | -1.225848 | -0.911583 | -1.547768 | 0.970243 | -1.008643 |
| **9998** | -0.119356 | 0.989439 | 1.256835 | 0.306591 | 0.912419 | -0.014933 | -0.695982 | -0.022608 | 0.807737 | 0.646092 | -1.030670 | -0.125231 |
| **9999** | -0.870559 | 1.469253 | 1.463771 | -0.901886 | -1.095988 | 0.828762 | -0.350204 | 0.859965 | -0.911583 | 0.646092 | -1.030670 | -1.076370 |

10000 rows × 12 columns

**Splitting the data into training and testing data**

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**=**0.2,random\_state**=**42)

In [50]:

X\_train**.**head()

Out[50]:

|  | **CustomerId** | **Surname** | **CreditScore** | **Geography** | **Gender** | **Age** | **Tenure** | **Balance** | **NumOfProducts** | **HasCrCard** | **IsActiveMember** | **EstimatedSalary** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **9254** | -1.248733 | 0.685715 | 0.367013 | -0.901886 | 0.912419 | -1.069553 | 0.341352 | -1.225848 | 0.807737 | 0.646092 | 0.970243 | 1.373784 |
| **1561** | 1.048668 | 0.022721 | -0.191713 | 0.306591 | 0.912419 | -0.014933 | -0.350204 | 0.691389 | 0.807737 | 0.646092 | 0.970243 | 1.667407 |
| **1670** | 0.362192 | -0.533910 | -0.947028 | 1.515067 | 0.912419 | 0.828762 | -0.695982 | 0.613102 | -0.911583 | 0.646092 | -1.030670 | -0.246910 |
| **6087** | 0.553552 | -1.141359 | -0.926334 | -0.901886 | -1.095988 | 0.828762 | 1.378686 | 0.948021 | -0.911583 | 0.646092 | -1.030670 | 0.921446 |
| **6669** | 1.486940 | -1.168540 | -1.381593 | -0.901886 | 0.912419 | 1.461533 | 1.378686 | 1.052363 | -0.911583 | -1.547768 | -1.030670 | -1.053812 |

In [51]:

X\_train**.**shape, Y\_train**.**shape, X\_test**.**shape, Y\_test**.**shape

Out[51]:

((8000, 12), (8000,), (2000, 12), (2000,))