{

"cells": [

{

"cell\_type": "markdown",

"id": "fd15e517",

"metadata": {},

"source": [

"## import the necessary liberaries"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"id": "cb82500b",

"metadata": {},

"outputs": [],

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"import sklearn\n",

"from sklearn.preprocessing import OneHotEncoder, LabelEncoder\n",

"from sklearn.preprocessing import StandardScaler, MinMaxScaler\n",

"from sklearn.model\_selection import train\_test\_split\n",

"import os"

]

},

{

"cell\_type": "markdown",

"id": "89174964",

"metadata": {},

"source": [

"## Load Dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "af913005",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 10000 entries, 0 to 9999\n",

"Data columns (total 14 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 RowNumber 10000 non-null int64 \n",

" 1 CustomerId 10000 non-null int64 \n",

" 2 Surname 10000 non-null object \n",

" 3 CreditScore 10000 non-null int64 \n",

" 4 Geography 10000 non-null object \n",

" 5 Gender 10000 non-null object \n",

" 6 Age 10000 non-null int64 \n",

" 7 Tenure 10000 non-null int64 \n",

" 8 Balance 10000 non-null float64\n",

" 9 NumOfProducts 10000 non-null int64 \n",

" 10 HasCrCard 10000 non-null int64 \n",

" 11 IsActiveMember 10000 non-null int64 \n",

" 12 EstimatedSalary 10000 non-null float64\n",

" 13 Exited 10000 non-null int64 \n",

"dtypes: float64(2), int64(9), object(3)\n",

"memory usage: 1.1+ MB\n"

]

}

],

"source": [

"data = pd.read\_csv(os.path.join(os.getcwd(),\"Churn\_Modelling.csv\"))\n",

"data.info()"

]

},

{

"cell\_type": "markdown",

"id": "60185579",

"metadata": {},

"source": [

"## Visualizations"

]

},

{

"cell\_type": "markdown",

"id": "b7fdfc45",

"metadata": {},

"source": [

"### Univariate Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "bef470d4",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age', ylabel='Count'>"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "2f5c6519",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x1f2f1f2bcd0>"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.displot(data['CreditScore'])"

]

},

{

"cell\_type": "markdown",

"id": "4a1c9597",

"metadata": {},

"source": [

"### Bi - Variate Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "9fe001c0",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>"

]

},

"execution\_count": 5,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.lineplot(x=data['Age'],y=data['EstimatedSalary'])"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "1dd67004",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age', ylabel='Balance'>"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.barplot(x=data['Age'],y=data['Balance'])"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "4b8bd175",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age', ylabel='Balance'>"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.scatterplot(x=data['Age'],y=data['Balance'])"

]

},

{

"cell\_type": "markdown",

"id": "96f9f1de",

"metadata": {},

"source": [

"### Multi - Variate Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "1d1a2b72",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age', ylabel='Balance'>"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.scatterplot(x=data['Age'],y=data['Balance'],hue=data['HasCrCard'])"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "0d4080b7",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x1f2f34f08e0>"

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1980x1980 with 132 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.pairplot(data)"

]

},

{

"cell\_type": "markdown",

"id": "659ed624",

"metadata": {},

"source": [

"## Descriptive Statistics"

]

},

{

"cell\_type": "markdown",

"id": "6af2a783",

"metadata": {},

"source": [

"### Measure of central tendency"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "b4129263",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>RowNumber</th>\n",

" <th>CustomerId</th>\n",

" <th>CreditScore</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>EstimatedSalary</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>10000.00000</td>\n",

" <td>1.000000e+04</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.00000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" <td>10000.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>5000.50000</td>\n",

" <td>1.569094e+07</td>\n",

" <td>650.528800</td>\n",

" <td>38.921800</td>\n",

" <td>5.012800</td>\n",

" <td>76485.889288</td>\n",

" <td>1.530200</td>\n",

" <td>0.70550</td>\n",

" <td>0.515100</td>\n",

" <td>100090.239881</td>\n",

" <td>0.203700</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>2886.89568</td>\n",

" <td>7.193619e+04</td>\n",

" <td>96.653299</td>\n",

" <td>10.487806</td>\n",

" <td>2.892174</td>\n",

" <td>62397.405202</td>\n",

" <td>0.581654</td>\n",

" <td>0.45584</td>\n",

" <td>0.499797</td>\n",

" <td>57510.492818</td>\n",

" <td>0.402769</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>1.00000</td>\n",

" <td>1.556570e+07</td>\n",

" <td>350.000000</td>\n",

" <td>18.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>0.00000</td>\n",

" <td>0.000000</td>\n",

" <td>11.580000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>2500.75000</td>\n",

" <td>1.562853e+07</td>\n",

" <td>584.000000</td>\n",

" <td>32.000000</td>\n",

" <td>3.000000</td>\n",

" <td>0.000000</td>\n",

" <td>1.000000</td>\n",

" <td>0.00000</td>\n",

" <td>0.000000</td>\n",

" <td>51002.110000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>5000.50000</td>\n",

" <td>1.569074e+07</td>\n",

" <td>652.000000</td>\n",

" <td>37.000000</td>\n",

" <td>5.000000</td>\n",

" <td>97198.540000</td>\n",

" <td>1.000000</td>\n",

" <td>1.00000</td>\n",

" <td>1.000000</td>\n",

" <td>100193.915000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>7500.25000</td>\n",

" <td>1.575323e+07</td>\n",

" <td>718.000000</td>\n",

" <td>44.000000</td>\n",

" <td>7.000000</td>\n",

" <td>127644.240000</td>\n",

" <td>2.000000</td>\n",

" <td>1.00000</td>\n",

" <td>1.000000</td>\n",

" <td>149388.247500</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>10000.00000</td>\n",

" <td>1.581569e+07</td>\n",

" <td>850.000000</td>\n",

" <td>92.000000</td>\n",

" <td>10.000000</td>\n",

" <td>250898.090000</td>\n",

" <td>4.000000</td>\n",

" <td>1.00000</td>\n",

" <td>1.000000</td>\n",

" <td>199992.480000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" RowNumber CustomerId CreditScore Age Tenure \\\n",

"count 10000.00000 1.000000e+04 10000.000000 10000.000000 10000.000000 \n",

"mean 5000.50000 1.569094e+07 650.528800 38.921800 5.012800 \n",

"std 2886.89568 7.193619e+04 96.653299 10.487806 2.892174 \n",

"min 1.00000 1.556570e+07 350.000000 18.000000 0.000000 \n",

"25% 2500.75000 1.562853e+07 584.000000 32.000000 3.000000 \n",

"50% 5000.50000 1.569074e+07 652.000000 37.000000 5.000000 \n",

"75% 7500.25000 1.575323e+07 718.000000 44.000000 7.000000 \n",

"max 10000.00000 1.581569e+07 850.000000 92.000000 10.000000 \n",

"\n",

" Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"count 10000.000000 10000.000000 10000.00000 10000.000000 \n",

"mean 76485.889288 1.530200 0.70550 0.515100 \n",

"std 62397.405202 0.581654 0.45584 0.499797 \n",

"min 0.000000 1.000000 0.00000 0.000000 \n",

"25% 0.000000 1.000000 0.00000 0.000000 \n",

"50% 97198.540000 1.000000 1.00000 1.000000 \n",

"75% 127644.240000 2.000000 1.00000 1.000000 \n",

"max 250898.090000 4.000000 1.00000 1.000000 \n",

"\n",

" EstimatedSalary Exited \n",

"count 10000.000000 10000.000000 \n",

"mean 100090.239881 0.203700 \n",

"std 57510.492818 0.402769 \n",

"min 11.580000 0.000000 \n",

"25% 51002.110000 0.000000 \n",

"50% 100193.915000 0.000000 \n",

"75% 149388.247500 0.000000 \n",

"max 199992.480000 1.000000 "

]

},

"execution\_count": 10,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.describe()"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "067090c6",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 5.000500e+03\n",

"CustomerId 1.569074e+07\n",

"CreditScore 6.520000e+02\n",

"Age 3.700000e+01\n",

"Tenure 5.000000e+00\n",

"Balance 9.719854e+04\n",

"NumOfProducts 1.000000e+00\n",

"HasCrCard 1.000000e+00\n",

"IsActiveMember 1.000000e+00\n",

"EstimatedSalary 1.001939e+05\n",

"Exited 0.000000e+00\n",

"dtype: float64"

]

},

"execution\_count": 11,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.median(numeric\_only=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "1a2b0bb7",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>RowNumber</th>\n",

" <th>CustomerId</th>\n",

" <th>Surname</th>\n",

" <th>CreditScore</th>\n",

" <th>Geography</th>\n",

" <th>Gender</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>EstimatedSalary</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>15565701</td>\n",

" <td>Smith</td>\n",

" <td>850.0</td>\n",

" <td>France</td>\n",

" <td>Male</td>\n",

" <td>37.0</td>\n",

" <td>2.0</td>\n",

" <td>0.0</td>\n",

" <td>1.0</td>\n",

" <td>1.0</td>\n",

" <td>1.0</td>\n",

" <td>24924.92</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>15565706</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>3</td>\n",

" <td>15565714</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>4</td>\n",

" <td>15565779</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>5</td>\n",

" <td>15565796</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9995</th>\n",

" <td>9996</td>\n",

" <td>15815628</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9996</th>\n",

" <td>9997</td>\n",

" <td>15815645</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9997</th>\n",

" <td>9998</td>\n",

" <td>15815656</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9998</th>\n",

" <td>9999</td>\n",

" <td>15815660</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9999</th>\n",

" <td>10000</td>\n",

" <td>15815690</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 14 columns</p>\n",

"</div>"

],

"text/plain": [

" RowNumber CustomerId Surname CreditScore Geography Gender Age \\\n",

"0 1 15565701 Smith 850.0 France Male 37.0 \n",

"1 2 15565706 NaN NaN NaN NaN NaN \n",

"2 3 15565714 NaN NaN NaN NaN NaN \n",

"3 4 15565779 NaN NaN NaN NaN NaN \n",

"4 5 15565796 NaN NaN NaN NaN NaN \n",

"... ... ... ... ... ... ... ... \n",

"9995 9996 15815628 NaN NaN NaN NaN NaN \n",

"9996 9997 15815645 NaN NaN NaN NaN NaN \n",

"9997 9998 15815656 NaN NaN NaN NaN NaN \n",

"9998 9999 15815660 NaN NaN NaN NaN NaN \n",

"9999 10000 15815690 NaN NaN NaN NaN NaN \n",

"\n",

" Tenure Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"0 2.0 0.0 1.0 1.0 1.0 \n",

"1 NaN NaN NaN NaN NaN \n",

"2 NaN NaN NaN NaN NaN \n",

"3 NaN NaN NaN NaN NaN \n",

"4 NaN NaN NaN NaN NaN \n",

"... ... ... ... ... ... \n",

"9995 NaN NaN NaN NaN NaN \n",

"9996 NaN NaN NaN NaN NaN \n",

"9997 NaN NaN NaN NaN NaN \n",

"9998 NaN NaN NaN NaN NaN \n",

"9999 NaN NaN NaN NaN NaN \n",

"\n",

" EstimatedSalary Exited \n",

"0 24924.92 0.0 \n",

"1 NaN NaN \n",

"2 NaN NaN \n",

"3 NaN NaN \n",

"4 NaN NaN \n",

"... ... ... \n",

"9995 NaN NaN \n",

"9996 NaN NaN \n",

"9997 NaN NaN \n",

"9998 NaN NaN \n",

"9999 NaN NaN \n",

"\n",

"[10000 rows x 14 columns]"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.mode()"

]

},

{

"cell\_type": "markdown",

"id": "e0127b60",

"metadata": {},

"source": [

"### Measures of variability"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "0c94c1cf",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 8.334167e+06\n",

"CustomerId 5.174815e+09\n",

"CreditScore 9.341860e+03\n",

"Age 1.099941e+02\n",

"Tenure 8.364673e+00\n",

"Balance 3.893436e+09\n",

"NumOfProducts 3.383218e-01\n",

"HasCrCard 2.077905e-01\n",

"IsActiveMember 2.497970e-01\n",

"EstimatedSalary 3.307457e+09\n",

"Exited 1.622225e-01\n",

"dtype: float64"

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.var(numeric\_only=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"id": "0c3b0cb8",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"74"

]

},

"execution\_count": 14,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"range = max(data['Age']) - min(data['Age'])\n",

"range"

]

},

{

"cell\_type": "markdown",

"id": "61cca593",

"metadata": {},

"source": [

"## Handle the Missing values"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "35729a83",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber False\n",

"CustomerId False\n",

"Surname False\n",

"CreditScore False\n",

"Geography False\n",

"Gender False\n",

"Age False\n",

"Tenure False\n",

"Balance False\n",

"NumOfProducts False\n",

"HasCrCard False\n",

"IsActiveMember False\n",

"EstimatedSalary False\n",

"Exited False\n",

"dtype: bool"

]

},

"execution\_count": 15,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.isnull().any() # There in No null value in the dataset"

]

},

{

"cell\_type": "markdown",

"id": "51d55723",

"metadata": {},

"source": [

"## Find the outliers and replace the outliers"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "a3b45558",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>RowNumber</th>\n",

" <th>CustomerId</th>\n",

" <th>CreditScore</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>EstimatedSalary</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0.75</th>\n",

" <td>7500.25</td>\n",

" <td>15753233.75</td>\n",

" <td>718.0</td>\n",

" <td>44.0</td>\n",

" <td>7.0</td>\n",

" <td>127644.24</td>\n",

" <td>2.0</td>\n",

" <td>1.0</td>\n",

" <td>1.0</td>\n",

" <td>149388.2475</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>0.25</th>\n",

" <td>2500.75</td>\n",

" <td>15628528.25</td>\n",

" <td>584.0</td>\n",

" <td>32.0</td>\n",

" <td>3.0</td>\n",

" <td>0.00</td>\n",

" <td>1.0</td>\n",

" <td>0.0</td>\n",

" <td>0.0</td>\n",

" <td>51002.1100</td>\n",

" <td>0.0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" RowNumber CustomerId CreditScore Age Tenure Balance \\\n",

"0.75 7500.25 15753233.75 718.0 44.0 7.0 127644.24 \n",

"0.25 2500.75 15628528.25 584.0 32.0 3.0 0.00 \n",

"\n",

" NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited \n",

"0.75 2.0 1.0 1.0 149388.2475 0.0 \n",

"0.25 1.0 0.0 0.0 51002.1100 0.0 "

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"q = data.quantile([0.75,0.25])\n",

"q"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"id": "5b4b71c2",

"metadata": {},

"outputs": [],

"source": [

"iqr = q.iloc[0]-q.iloc[1]\n",

"u = q.iloc[0]+(1.5\*iqr) # upper bound\n",

"l = q.iloc[1]-(1.5\*iqr) # lower bound"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "d524bae4",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 4999.5000\n",

"CustomerId 124705.5000\n",

"CreditScore 134.0000\n",

"Age 12.0000\n",

"Tenure 4.0000\n",

"Balance 127644.2400\n",

"NumOfProducts 1.0000\n",

"HasCrCard 1.0000\n",

"IsActiveMember 1.0000\n",

"EstimatedSalary 98386.1375\n",

"Exited 0.0000\n",

"dtype: float64"

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"iqr"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"id": "beaddfaf",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 1.499950e+04\n",

"CustomerId 1.594029e+07\n",

"CreditScore 9.190000e+02\n",

"Age 6.200000e+01\n",

"Tenure 1.300000e+01\n",

"Balance 3.191106e+05\n",

"NumOfProducts 3.500000e+00\n",

"HasCrCard 2.500000e+00\n",

"IsActiveMember 2.500000e+00\n",

"EstimatedSalary 2.969675e+05\n",

"Exited 0.000000e+00\n",

"dtype: float64"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"u"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"id": "9fc3532c",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber -4.998500e+03\n",

"CustomerId 1.544147e+07\n",

"CreditScore 3.830000e+02\n",

"Age 1.400000e+01\n",

"Tenure -3.000000e+00\n",

"Balance -1.914664e+05\n",

"NumOfProducts -5.000000e-01\n",

"HasCrCard -1.500000e+00\n",

"IsActiveMember -1.500000e+00\n",

"EstimatedSalary -9.657710e+04\n",

"Exited 0.000000e+00\n",

"dtype: float64"

]

},

"execution\_count": 20,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"l"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"id": "75e80431",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age'>"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(x = data['Age'])"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "71db955e",

"metadata": {},

"outputs": [],

"source": [

"data['Age'] = np.where(data['Age']>56,int(data['Age'].mean()),data['Age'])"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "cf6d864a",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age'>"

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(x = data['Age'])"

]

},

{

"cell\_type": "markdown",

"id": "b0ad126a",

"metadata": {},

"source": [

"## Check for Categorical columns and perform encoding"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "57ed3dc5",

"metadata": {},

"outputs": [],

"source": [

"le = LabelEncoder()\n",

"oneh = OneHotEncoder()"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "453359f3",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 10000 entries, 0 to 9999\n",

"Data columns (total 14 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 RowNumber 10000 non-null int64 \n",

" 1 CustomerId 10000 non-null int64 \n",

" 2 Surname 10000 non-null object \n",

" 3 CreditScore 10000 non-null int64 \n",

" 4 Geography 10000 non-null object \n",

" 5 Gender 10000 non-null object \n",

" 6 Age 10000 non-null int64 \n",

" 7 Tenure 10000 non-null int64 \n",

" 8 Balance 10000 non-null float64\n",

" 9 NumOfProducts 10000 non-null int64 \n",

" 10 HasCrCard 10000 non-null int64 \n",

" 11 IsActiveMember 10000 non-null int64 \n",

" 12 EstimatedSalary 10000 non-null float64\n",

" 13 Exited 10000 non-null int64 \n",

"dtypes: float64(2), int64(9), object(3)\n",

"memory usage: 1.1+ MB\n"

]

}

],

"source": [

"data.info() # 3 categorical columns (Surname, Geography, Gender)"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "041da651",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>RowNumber</th>\n",

" <th>CustomerId</th>\n",

" <th>Surname</th>\n",

" <th>CreditScore</th>\n",

" <th>Geography</th>\n",

" <th>Gender</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>EstimatedSalary</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>15634602</td>\n",

" <td>Hargrave</td>\n",

" <td>619</td>\n",

" <td>France</td>\n",

" <td>Female</td>\n",

" <td>42</td>\n",

" <td>2</td>\n",

" <td>0.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>101348.88</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>15647311</td>\n",

" <td>Hill</td>\n",

" <td>608</td>\n",

" <td>Spain</td>\n",

" <td>Female</td>\n",

" <td>41</td>\n",

" <td>1</td>\n",

" <td>83807.86</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>112542.58</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>3</td>\n",

" <td>15619304</td>\n",

" <td>Onio</td>\n",

" <td>502</td>\n",

" <td>France</td>\n",

" <td>Female</td>\n",

" <td>42</td>\n",

" <td>8</td>\n",

" <td>159660.80</td>\n",

" <td>3</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>113931.57</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>4</td>\n",

" <td>15701354</td>\n",

" <td>Boni</td>\n",

" <td>699</td>\n",

" <td>France</td>\n",

" <td>Female</td>\n",

" <td>39</td>\n",

" <td>1</td>\n",

" <td>0.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>93826.63</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>5</td>\n",

" <td>15737888</td>\n",

" <td>Mitchell</td>\n",

" <td>850</td>\n",

" <td>Spain</td>\n",

" <td>Female</td>\n",

" <td>43</td>\n",

" <td>2</td>\n",

" <td>125510.82</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>79084.10</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" RowNumber CustomerId Surname CreditScore Geography Gender Age \\\n",

"0 1 15634602 Hargrave 619 France Female 42 \n",

"1 2 15647311 Hill 608 Spain Female 41 \n",

"2 3 15619304 Onio 502 France Female 42 \n",

"3 4 15701354 Boni 699 France Female 39 \n",

"4 5 15737888 Mitchell 850 Spain Female 43 \n",

"\n",

" Tenure Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"2 8 159660.80 3 1 0 \n",

"3 1 0.00 2 0 0 \n",

"4 2 125510.82 1 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 "

]

},

"execution\_count": 26,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "6d5165ef",

"metadata": {},

"outputs": [],

"source": [

"data['Surname'] = le.fit\_transform(data['Surname'])\n",

"data['Geography'] = le.fit\_transform(data['Geography'])\n",

"data['Gender'] = le.fit\_transform(data['Gender'])"

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"id": "929d4c8e",

"metadata": {

"scrolled": false

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 10000 entries, 0 to 9999\n",

"Data columns (total 14 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 RowNumber 10000 non-null int64 \n",

" 1 CustomerId 10000 non-null int64 \n",

" 2 Surname 10000 non-null int32 \n",

" 3 CreditScore 10000 non-null int64 \n",

" 4 Geography 10000 non-null int32 \n",

" 5 Gender 10000 non-null int32 \n",

" 6 Age 10000 non-null int64 \n",

" 7 Tenure 10000 non-null int64 \n",

" 8 Balance 10000 non-null float64\n",

" 9 NumOfProducts 10000 non-null int64 \n",

" 10 HasCrCard 10000 non-null int64 \n",

" 11 IsActiveMember 10000 non-null int64 \n",

" 12 EstimatedSalary 10000 non-null float64\n",

" 13 Exited 10000 non-null int64 \n",

"dtypes: float64(2), int32(3), int64(9)\n",

"memory usage: 976.7 KB\n"

]

}

],

"source": [

"data.info() # categorical columns converted into int32"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"id": "dbfbf922",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>RowNumber</th>\n",

" <th>CustomerId</th>\n",

" <th>Surname</th>\n",

" <th>CreditScore</th>\n",

" <th>Geography</th>\n",

" <th>Gender</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>EstimatedSalary</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1</td>\n",

" <td>15634602</td>\n",

" <td>1115</td>\n",

" <td>619</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>42</td>\n",

" <td>2</td>\n",

" <td>0.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>101348.88</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>15647311</td>\n",

" <td>1177</td>\n",

" <td>608</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>41</td>\n",

" <td>1</td>\n",

" <td>83807.86</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>112542.58</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>3</td>\n",

" <td>15619304</td>\n",

" <td>2040</td>\n",

" <td>502</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>42</td>\n",

" <td>8</td>\n",

" <td>159660.80</td>\n",

" <td>3</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>113931.57</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>4</td>\n",

" <td>15701354</td>\n",

" <td>289</td>\n",

" <td>699</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>39</td>\n",

" <td>1</td>\n",

" <td>0.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>93826.63</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>5</td>\n",

" <td>15737888</td>\n",

" <td>1822</td>\n",

" <td>850</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>43</td>\n",

" <td>2</td>\n",

" <td>125510.82</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>79084.10</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" RowNumber CustomerId Surname CreditScore Geography Gender Age \\\n",

"0 1 15634602 1115 619 0 0 42 \n",

"1 2 15647311 1177 608 2 0 41 \n",

"2 3 15619304 2040 502 0 0 42 \n",

"3 4 15701354 289 699 0 0 39 \n",

"4 5 15737888 1822 850 2 0 43 \n",

"\n",

" Tenure Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"2 8 159660.80 3 1 0 \n",

"3 1 0.00 2 0 0 \n",

"4 2 125510.82 1 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 "

]

},

"execution\_count": 29,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.head()"

]

},

{

"cell\_type": "markdown",

"id": "279465c3",

"metadata": {},

"source": [

"## Split the data into dependent and independent variables"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"id": "d435fe5c",

"metadata": {},

"outputs": [],

"source": [

"x = data.iloc[:,[1,3,4,5,6,7,8,9,10,11,13]] # Independent Columns\n",

"y=data['EstimatedSalary']"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"id": "29bdcce7",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>CustomerId</th>\n",

" <th>CreditScore</th>\n",

" <th>Geography</th>\n",

" <th>Gender</th>\n",

" <th>Age</th>\n",

" <th>Tenure</th>\n",

" <th>Balance</th>\n",

" <th>NumOfProducts</th>\n",

" <th>HasCrCard</th>\n",

" <th>IsActiveMember</th>\n",

" <th>Exited</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>15634602</td>\n",

" <td>619</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>42</td>\n",

" <td>2</td>\n",

" <td>0.00</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>15647311</td>\n",

" <td>608</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>41</td>\n",

" <td>1</td>\n",

" <td>83807.86</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>15619304</td>\n",

" <td>502</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>42</td>\n",

" <td>8</td>\n",

" <td>159660.80</td>\n",

" <td>3</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>15701354</td>\n",

" <td>699</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>39</td>\n",

" <td>1</td>\n",

" <td>0.00</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>15737888</td>\n",

" <td>850</td>\n",

" <td>2</td>\n",

" <td>0</td>\n",

" <td>43</td>\n",

" <td>2</td>\n",

" <td>125510.82</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9995</th>\n",

" <td>15606229</td>\n",

" <td>771</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>39</td>\n",

" <td>5</td>\n",

" <td>0.00</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9996</th>\n",

" <td>15569892</td>\n",

" <td>516</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>35</td>\n",

" <td>10</td>\n",

" <td>57369.61</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9997</th>\n",

" <td>15584532</td>\n",

" <td>709</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>36</td>\n",

" <td>7</td>\n",

" <td>0.00</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9998</th>\n",

" <td>15682355</td>\n",

" <td>772</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>42</td>\n",

" <td>3</td>\n",

" <td>75075.31</td>\n",

" <td>2</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9999</th>\n",

" <td>15628319</td>\n",

" <td>792</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>28</td>\n",

" <td>4</td>\n",

" <td>130142.79</td>\n",

" <td>1</td>\n",

" <td>1</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 11 columns</p>\n",

"</div>"

],

"text/plain": [

" CustomerId CreditScore Geography Gender Age Tenure Balance \\\n",

"0 15634602 619 0 0 42 2 0.00 \n",

"1 15647311 608 2 0 41 1 83807.86 \n",

"2 15619304 502 0 0 42 8 159660.80 \n",

"3 15701354 699 0 0 39 1 0.00 \n",

"4 15737888 850 2 0 43 2 125510.82 \n",

"... ... ... ... ... ... ... ... \n",

"9995 15606229 771 0 1 39 5 0.00 \n",

"9996 15569892 516 0 1 35 10 57369.61 \n",

"9997 15584532 709 0 0 36 7 0.00 \n",

"9998 15682355 772 1 1 42 3 75075.31 \n",

"9999 15628319 792 0 0 28 4 130142.79 \n",

"\n",

" NumOfProducts HasCrCard IsActiveMember Exited \n",

"0 1 1 1 1 \n",

"1 1 0 1 0 \n",

"2 3 1 0 1 \n",

"3 2 0 0 0 \n",

"4 1 1 1 0 \n",

"... ... ... ... ... \n",

"9995 2 1 0 0 \n",

"9996 1 1 1 0 \n",

"9997 1 0 1 1 \n",

"9998 2 1 0 1 \n",

"9999 1 1 0 0 \n",

"\n",

"[10000 rows x 11 columns]"

]

},

"execution\_count": 31,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"id": "35e16180",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0 101348.88\n",

"1 112542.58\n",

"2 113931.57\n",

"3 93826.63\n",

"4 79084.10\n",

" ... \n",

"9995 96270.64\n",

"9996 101699.77\n",

"9997 42085.58\n",

"9998 92888.52\n",

"9999 38190.78\n",

"Name: EstimatedSalary, Length: 10000, dtype: float64"

]

},

"execution\_count": 32,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y"

]

},

{

"cell\_type": "markdown",

"id": "88a6c2e3",

"metadata": {},

"source": [

"## Scale the independent variables"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"id": "b9a5878c",

"metadata": {

"scrolled": true

},

"outputs": [

{

"data": {

"text/plain": [

"array([[-0.78321342, -0.32622142, -0.90188624, ..., 0.64609167,\n",

" 0.97024255, 1.97716468],\n",

" [-0.60653412, -0.44003595, 1.51506738, ..., -1.54776799,\n",

" 0.97024255, -0.50577476],\n",

" [-0.99588476, -1.53679418, -0.90188624, ..., 0.64609167,\n",

" -1.03067011, 1.97716468],\n",

" ...,\n",

" [-1.47928179, 0.60498839, -0.90188624, ..., -1.54776799,\n",

" 0.97024255, 1.97716468],\n",

" [-0.11935577, 1.25683526, 0.30659057, ..., 0.64609167,\n",

" -1.03067011, 1.97716468],\n",

" [-0.87055909, 1.46377078, -0.90188624, ..., 0.64609167,\n",

" -1.03067011, -0.50577476]])"

]

},

"execution\_count": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"sc = StandardScaler()\n",

"x\_scaled = sc.fit\_transform(x)\n",

"x\_scaled"

]

},

{

"cell\_type": "markdown",

"id": "2755546b",

"metadata": {},

"source": [

"## Split the data into training and testing"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"id": "8aaeea33",

"metadata": {},

"outputs": [],

"source": [

"x\_train, x\_test, y\_train, y\_test = train\_test\_split(x\_scaled, y, test\_size=0.3)"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"id": "ae85e85b",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[ 1.27328144, 0.346319 , 1.51506738, ..., 0.64609167,\n",

" -1.03067011, -0.50577476],\n",

" [ 1.23234033, 1.77417405, 0.30659057, ..., 0.64609167,\n",

" -1.03067011, -0.50577476],\n",

" [ 0.03088212, 0.44978676, -0.90188624, ..., -1.54776799,\n",

" -1.03067011, -0.50577476],\n",

" ...,\n",

" [-1.08399503, -1.09188282, 0.30659057, ..., -1.54776799,\n",

" 0.97024255, 1.97716468],\n",

" [ 0.12884884, -1.70234259, -0.90188624, ..., 0.64609167,\n",

" 0.97024255, -0.50577476],\n",

" [ 1.09871522, 0.29458512, 1.51506738, ..., -1.54776799,\n",

" -1.03067011, -0.50577476]])"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"id": "845bf153",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(7000, 11)"

]

},

"execution\_count": 36,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "651b875a",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"4455 188637.05\n",

"7626 26249.86\n",

"5462 112317.89\n",

"8481 189727.12\n",

"5360 65675.47\n",

" ... \n",

"4159 39934.41\n",

"2028 116343.09\n",

"734 176924.21\n",

"2615 85101.00\n",

"779 121210.09\n",

"Name: EstimatedSalary, Length: 7000, dtype: float64"

]

},

"execution\_count": 37,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_train"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "036c95e5",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(7000,)"

]

},

"execution\_count": 38,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_train.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "cd60dad9",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[-0.80804222, -1.34020544, 0.30659057, ..., 0.64609167,\n",

" 0.97024255, -0.50577476],\n",

" [-0.70735073, -0.46072951, -0.90188624, ..., 0.64609167,\n",

" 0.97024255, -0.50577476],\n",

" [ 0.13506299, 0.11868993, -0.90188624, ..., -1.54776799,\n",

" -1.03067011, -0.50577476],\n",

" ...,\n",

" [ 1.24880018, -2.1265604 , -0.90188624, ..., 0.64609167,\n",

" 0.97024255, -0.50577476],\n",

" [ 0.44556203, -0.86425376, -0.90188624, ..., -1.54776799,\n",

" -1.03067011, 1.97716468],\n",

" [ 0.52663794, 1.91902891, 1.51506738, ..., 0.64609167,\n",

" -1.03067011, -0.50577476]])"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_test"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"id": "d4eaf076",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(3000, 11)"

]

},

"execution\_count": 40,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_test.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"id": "1c7b58a3",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"9064 182765.14\n",

"745 70899.27\n",

"6802 65089.38\n",

"3801 138396.32\n",

"2696 143681.83\n",

" ... \n",

"8387 107181.22\n",

"4421 169802.73\n",

"4668 175893.53\n",

"8936 52035.29\n",

"6529 111324.41\n",

"Name: EstimatedSalary, Length: 3000, dtype: float64"

]

},

"execution\_count": 41,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_test"

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"id": "773edccc",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(3000,)"

]

},

"execution\_count": 42,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_test.shape"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.12"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}