ASSIGNMENT-2

{

"cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"assignment 2\n",

"student name : sit"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"1. Download the dataset: Dataset"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"my dataset is downloaded in the path : C:\\Users\\STUDENT\\IBM"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"2. Load the dataset.\n"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"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": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset = pd.read\_csv(r\"C:\\Users\\STUDENT\\Desktop\\Downloads\\Churn\_Modelling.csv\")\n",

"dataset.head()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"3. Perform Below Visualizations.\n",

"● Univariate Analysis\n",

"● Bi - Variate Analysis\n",

"● Multi - Variate Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {},

"outputs": [],

"source": [

"import seaborn as sns"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x18555db1250>"

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x1855101bb50>"

]

},

"execution\_count": 14,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.relplot(x=\"CreditScore\",y='Age',data=dataset)"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x18556340700>"

]

},

"execution\_count": 17,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 448x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.relplot(x=\"CreditScore\",y='Age',hue=\"IsActiveMember\",data=dataset)"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x185565264f0>"

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.catplot(x=\"Gender\",y='Age',data=dataset)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"4. Perform descriptive statistics on the dataset."

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"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",

" </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",

"\n",

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

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 "

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"ds = pd.read\_csv(r\"C:\\Users\\STUDENT\\Desktop\\Downloads\\Churn\_Modelling.csv\")\n",

"ds.head(2)"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"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": 31,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"ds.isnull().any()"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"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": 32,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"ds.describe()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"5. Handle the Missing values."

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"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": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 0\n",

"CustomerId 0\n",

"Surname 0\n",

"CreditScore 0\n",

"Geography 0\n",

"Gender 0\n",

"Age 0\n",

"Tenure 0\n",

"Balance 0\n",

"NumOfProducts 0\n",

"HasCrCard 0\n",

"IsActiveMember 0\n",

"EstimatedSalary 0\n",

"Exited 0\n",

"dtype: int64"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.isnull().sum()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"Thus,the dataset is not having any missing or null values.\n",

"if an dataset will have any missing values,we can handle it in following ways\n",

"1) lot of missimg values----remove\n",

"2) less missing values ----replace\n",

"function used---fillna()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"6. Find the outliers and replace the outliers"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber 0.000000\n",

"CustomerId 0.001149\n",

"CreditScore -0.071607\n",

"Age 1.011320\n",

"Tenure 0.010991\n",

"Balance -0.141109\n",

"NumOfProducts 0.745568\n",

"HasCrCard -0.901812\n",

"IsActiveMember -0.060437\n",

"EstimatedSalary 0.002085\n",

"Exited 1.471611\n",

"dtype: float64"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.skew()"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\ProgramData\\Anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: 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.\n",

" warnings.warn(\n"

]

},

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age'>"

]

},

"execution\_count": 37,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(dataset[\"Age\"])"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"metadata": {},

"outputs": [],

"source": [

"q1= dataset[\"Age\"].describe()[\"25%\"]\n",

"q3= dataset[\"Age\"].describe()[\"75%\"]"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"32.0"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"q1"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"44.0"

]

},

"execution\_count": 40,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"q3"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"12.0"

]

},

"execution\_count": 41,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"iqr=q3-q1\n",

"iqr"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"metadata": {},

"outputs": [],

"source": [

"l\_b=q1-(1.5\*iqr)\n",

"u\_b=q3+(1.5\*iqr)"

]

},

{

"cell\_type": "code",

"execution\_count": 48,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"14.0"

]

},

"execution\_count": 48,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"l\_b"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"metadata": {},

"outputs": [],

"source": [

"l\_b=q1-(1.5\*iqr)\n",

"u\_b=q3+(1.5\*iqr)"

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"14.0"

]

},

"execution\_count": 51,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"l\_b"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"62.0"

]

},

"execution\_count": 52,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"u\_b"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"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",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

"Empty DataFrame\n",

"Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited]\n",

"Index: []"

]

},

"execution\_count": 53,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset[dataset[\"Age\"]<l\_b]"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"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>58</th>\n",

" <td>59</td>\n",

" <td>15623944</td>\n",

" <td>T'ien</td>\n",

" <td>511</td>\n",

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

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

" <td>66</td>\n",

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

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

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

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

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

" <td>1643.11</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>85</th>\n",

" <td>86</td>\n",

" <td>15805254</td>\n",

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

" <td>652</td>\n",

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

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

" <td>75</td>\n",

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

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

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

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

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

" <td>114675.75</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>104</th>\n",

" <td>105</td>\n",

" <td>15804919</td>\n",

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

" <td>670</td>\n",

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

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

" <td>65</td>\n",

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

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

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

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

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

" <td>177655.68</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>158</th>\n",

" <td>159</td>\n",

" <td>15589975</td>\n",

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

" <td>646</td>\n",

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

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

" <td>73</td>\n",

" <td>6</td>\n",

" <td>97259.25</td>\n",

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

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

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

" <td>104719.66</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>181</th>\n",

" <td>182</td>\n",

" <td>15789669</td>\n",

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

" <td>510</td>\n",

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

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

" <td>65</td>\n",

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

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

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

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

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

" <td>48071.61</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"58 59 15623944 T'ien 511 Spain Female 66 \n",

"85 86 15805254 Ndukaku 652 Spain Female 75 \n",

"104 105 15804919 Dunbabin 670 Spain Female 65 \n",

"158 159 15589975 Maclean 646 France Female 73 \n",

"181 182 15789669 Hsia 510 France Male 65 \n",

"\n",

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

"58 4 0.00 1 1 0 \n",

"85 10 0.00 2 1 1 \n",

"104 1 0.00 1 1 1 \n",

"158 6 97259.25 1 0 1 \n",

"181 2 0.00 2 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"58 1643.11 1 \n",

"85 114675.75 0 \n",

"104 177655.68 1 \n",

"158 104719.66 0 \n",

"181 48071.61 0 "

]

},

"execution\_count": 54,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset[dataset[\"Age\"]>u\_b].head()"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RowNumber int64\n",

"CustomerId int64\n",

"Surname object\n",

"CreditScore int64\n",

"Geography object\n",

"Gender object\n",

"Age int64\n",

"Tenure int64\n",

"Balance float64\n",

"NumOfProducts int64\n",

"HasCrCard int64\n",

"IsActiveMember int64\n",

"EstimatedSalary float64\n",

"Exited int64\n",

"dtype: object"

]

},

"execution\_count": 55,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.dtypes"

]

},

{

"cell\_type": "code",

"execution\_count": 56,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"[66,\n",

" 75,\n",

" 65,\n",

" 73,\n",

" 65,\n",

" 72,\n",

" 67,\n",

" 67,\n",

" 79,\n",

" 80,\n",

" 68,\n",

" 75,\n",

" 66,\n",

" 66,\n",

" 70,\n",

" 63,\n",

" 72,\n",

" 64,\n",

" 64,\n",

" 70,\n",

" 67,\n",

" 82,\n",

" 63,\n",

" 69,\n",

" 65,\n",

" 69,\n",

" 64,\n",

" 65,\n",

" 74,\n",

" 67,\n",

" 66,\n",

" 67,\n",

" 63,\n",

" 70,\n",

" 71,\n",

" 72,\n",

" 67,\n",

" 74,\n",

" 76,\n",

" 66,\n",

" 63,\n",

" 66,\n",

" 68,\n",

" 67,\n",

" 63,\n",

" 71,\n",

" 66,\n",

" 69,\n",

" 73,\n",

" 65,\n",

" 66,\n",

" 64,\n",

" 69,\n",

" 64,\n",

" 77,\n",

" 74,\n",

" 65,\n",

" 70,\n",

" 67,\n",

" 69,\n",

" 67,\n",

" 74,\n",

" 69,\n",

" 74,\n",

" 74,\n",

" 64,\n",

" 63,\n",

" 63,\n",

" 70,\n",

" 74,\n",

" 65,\n",

" 72,\n",

" 77,\n",

" 66,\n",

" 65,\n",

" 74,\n",

" 88,\n",

" 63,\n",

" 71,\n",

" 63,\n",

" 64,\n",

" 67,\n",

" 70,\n",

" 68,\n",

" 72,\n",

" 71,\n",

" 66,\n",

" 75,\n",

" 67,\n",

" 73,\n",

" 69,\n",

" 76,\n",

" 63,\n",

" 85,\n",

" 67,\n",

" 74,\n",

" 76,\n",

" 66,\n",

" 69,\n",

" 66,\n",

" 72,\n",

" 63,\n",

" 71,\n",

" 63,\n",

" 74,\n",

" 67,\n",

" 72,\n",

" 72,\n",

" 66,\n",

" 84,\n",

" 71,\n",

" 66,\n",

" 63,\n",

" 74,\n",

" 69,\n",

" 84,\n",

" 67,\n",

" 64,\n",

" 68,\n",

" 66,\n",

" 77,\n",

" 70,\n",

" 67,\n",

" 79,\n",

" 67,\n",

" 76,\n",

" 73,\n",

" 66,\n",

" 67,\n",

" 64,\n",

" 73,\n",

" 76,\n",

" 72,\n",

" 64,\n",

" 71,\n",

" 63,\n",

" 70,\n",

" 65,\n",

" 66,\n",

" 65,\n",

" 80,\n",

" 66,\n",

" 63,\n",

" 63,\n",

" 63,\n",

" 63,\n",

" 66,\n",

" 74,\n",

" 69,\n",

" 63,\n",

" 64,\n",

" 76,\n",

" 75,\n",

" 68,\n",

" 69,\n",

" 77,\n",

" 64,\n",

" 66,\n",

" 74,\n",

" 71,\n",

" 67,\n",

" 68,\n",

" 64,\n",

" 68,\n",

" 70,\n",

" 64,\n",

" 75,\n",

" 66,\n",

" 64,\n",

" 78,\n",

" 65,\n",

" 74,\n",

" 64,\n",

" 64,\n",

" 71,\n",

" 77,\n",

" 79,\n",

" 70,\n",

" 81,\n",

" 64,\n",

" 68,\n",

" 68,\n",

" 63,\n",

" 79,\n",

" 66,\n",

" 64,\n",

" 70,\n",

" 69,\n",

" 71,\n",

" 72,\n",

" 66,\n",

" 68,\n",

" 63,\n",

" 71,\n",

" 72,\n",

" 72,\n",

" 64,\n",

" 78,\n",

" 75,\n",

" 65,\n",

" 65,\n",

" 67,\n",

" 63,\n",

" 68,\n",

" 71,\n",

" 73,\n",

" 64,\n",

" 66,\n",

" 71,\n",

" 69,\n",

" 71,\n",

" 66,\n",

" 76,\n",

" 69,\n",

" 73,\n",

" 64,\n",

" 64,\n",

" 75,\n",

" 73,\n",

" 71,\n",

" 72,\n",

" 63,\n",

" 67,\n",

" 68,\n",

" 73,\n",

" 67,\n",

" 64,\n",

" 63,\n",

" 92,\n",

" 65,\n",

" 75,\n",

" 67,\n",

" 71,\n",

" 64,\n",

" 66,\n",

" 64,\n",

" 66,\n",

" 67,\n",

" 77,\n",

" 92,\n",

" 67,\n",

" 63,\n",

" 66,\n",

" 66,\n",

" 68,\n",

" 65,\n",

" 72,\n",

" 71,\n",

" 76,\n",

" 63,\n",

" 67,\n",

" 67,\n",

" 66,\n",

" 67,\n",

" 63,\n",

" 65,\n",

" 70,\n",

" 72,\n",

" 77,\n",

" 74,\n",

" 72,\n",

" 73,\n",

" 77,\n",

" 67,\n",

" 71,\n",

" 64,\n",

" 72,\n",

" 81,\n",

" 76,\n",

" 69,\n",

" 68,\n",

" 74,\n",

" 64,\n",

" 64,\n",

" 71,\n",

" 68,\n",

" 63,\n",

" 67,\n",

" 63,\n",

" 64,\n",

" 76,\n",

" 63,\n",

" 63,\n",

" 68,\n",

" 67,\n",

" 72,\n",

" 70,\n",

" 81,\n",

" 67,\n",

" 73,\n",

" 66,\n",

" 68,\n",

" 71,\n",

" 66,\n",

" 63,\n",

" 75,\n",

" 69,\n",

" 64,\n",

" 69,\n",

" 70,\n",

" 71,\n",

" 71,\n",

" 66,\n",

" 70,\n",

" 63,\n",

" 64,\n",

" 65,\n",

" 63,\n",

" 67,\n",

" 71,\n",

" 67,\n",

" 65,\n",

" 66,\n",

" 63,\n",

" 73,\n",

" 66,\n",

" 64,\n",

" 72,\n",

" 71,\n",

" 69,\n",

" 67,\n",

" 64,\n",

" 81,\n",

" 73,\n",

" 63,\n",

" 67,\n",

" 74,\n",

" 83,\n",

" 69,\n",

" 71,\n",

" 78,\n",

" 63,\n",

" 70,\n",

" 69,\n",

" 72,\n",

" 70,\n",

" 63,\n",

" 74,\n",

" 80,\n",

" 69,\n",

" 72,\n",

" 67,\n",

" 76,\n",

" 71,\n",

" 67,\n",

" 71,\n",

" 78,\n",

" 63,\n",

" 63,\n",

" 68,\n",

" 64,\n",

" 70,\n",

" 78,\n",

" 69,\n",

" 68,\n",

" 64,\n",

" 64,\n",

" 77,\n",

" 77]"

]

},

"execution\_count": 56,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"outlier\_list=list(dataset[dataset[\"Age\"]>u\_b][\"Age\"])\n",

"outlier\_list"

]

},

{

"cell\_type": "code",

"execution\_count": 57,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"{66: 62.0,\n",

" 75: 62.0,\n",

" 65: 62.0,\n",

" 73: 62.0,\n",

" 72: 62.0,\n",

" 67: 62.0,\n",

" 79: 62.0,\n",

" 80: 62.0,\n",

" 68: 62.0,\n",

" 70: 62.0,\n",

" 63: 62.0,\n",

" 64: 62.0,\n",

" 82: 62.0,\n",

" 69: 62.0,\n",

" 74: 62.0,\n",

" 71: 62.0,\n",

" 76: 62.0,\n",

" 77: 62.0,\n",

" 88: 62.0,\n",

" 85: 62.0,\n",

" 84: 62.0,\n",

" 78: 62.0,\n",

" 81: 62.0,\n",

" 92: 62.0,\n",

" 83: 62.0}"

]

},

"execution\_count": 57,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"outlier\_dict={}.fromkeys(outlier\_list,u\_b)\n",

"outlier\_dict"

]

},

{

"cell\_type": "code",

"execution\_count": 58,

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\ProgramData\\Anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: 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.\n",

" warnings.warn(\n"

]

},

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Age'>"

]

},

"execution\_count": 58,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"dataset[\"Age\"]=dataset[\"Age\"].replace(outlier\_dict)\n",

"sns.boxplot(dataset[\"Age\"])"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"7. Check for Categorical columns and perform encoding.\n",

"8. Split the data into dependent and independent variables.\n",

"9. Scale the independent variables"

]

},

{

"cell\_type": "code",

"execution\_count": 59,

"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": 59,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset.isnull().any()"

]

},

{

"cell\_type": "code",

"execution\_count": 62,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([15634602, 15647311, 15619304, ..., 15584532, 15682355, 15628319],\n",

" dtype=int64)"

]

},

"execution\_count": 62,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataset[\"CustomerId\"].unique()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"from sklearn.compose import ColumnTransformer\n",

"from sklearn.pipeline import Pipeline\n",

"from sklearn.preprocessing import OneHotEncoder\n",

"dataset.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 67,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[1, 15634602, 'Hargrave', 619],\n",

" [2, 15647311, 'Hill', 608],\n",

" [3, 15619304, 'Onio', 502],\n",

" ...,\n",

" [9998, 15584532, 'Liu', 709],\n",

" [9999, 15682355, 'Sabbatini', 772],\n",

" [10000, 15628319, 'Walker', 792]], dtype=object)"

]

},

"execution\_count": 67,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x=dataset.iloc[:,0:4].values\n",

"x"

]

},

{

"cell\_type": "code",

"execution\_count": 68,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"numpy.ndarray"

]

},

"execution\_count": 68,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"type(x)"

]

},

{

"cell\_type": "code",

"execution\_count": 69,

"metadata": {},

"outputs": [],

"source": [

"y=dataset.iloc[:,4:5].values"

]

},

{

"cell\_type": "code",

"execution\_count": 70,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(10000, 4)"

]

},

"execution\_count": 70,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 71,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(10000, 1)"

]

},

"execution\_count": 71,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 72,

"metadata": {},

"outputs": [],

"source": [

"ct=ColumnTransformer([(\"oh\",OneHotEncoder(),[3])],remainder=\"Passthrough\")"

]

},

{

"cell\_type": "code",

"execution\_count": 73,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[1, 15634602, 'Hargrave', 619],\n",

" [2, 15647311, 'Hill', 608],\n",

" [3, 15619304, 'Onio', 502],\n",

" ...,\n",

" [9998, 15584532, 'Liu', 709],\n",

" [9999, 15682355, 'Sabbatini', 772],\n",

" [10000, 15628319, 'Walker', 792]], dtype=object)"

]

},

"execution\_count": 73,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"10. Split the data into training and testing"

]

},

{

"cell\_type": "code",

"execution\_count": 75,

"metadata": {},

"outputs": [],

"source": [

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

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

]

},

{

"cell\_type": "code",

"execution\_count": 76,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(8000, 4)"

]

},

"execution\_count": 76,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 77,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(2000, 4)"

]

},

"execution\_count": 77,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_test.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 78,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(8000, 1)"

]

},

"execution\_count": 78,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_train.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 79,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(2000, 1)"

]

},

"execution\_count": 79,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_test.shape"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3",

"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": "3.8.5"

}

},

"nbformat": 4,

"nbformat\_minor": 4

}