{

"cells": [

{

"cell\_type": "markdown",

"id": "51f3786c",

"metadata": {},

"source": [

"###### 1.Download the Dataset"

]

},

{

"cell\_type": "markdown",

"id": "5641a064",

"metadata": {},

"source": [

"###### 2.Load the Dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"id": "acadd095",

"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": [

"import pandas as pd\n",

"df = pd.read\_csv('Churn\_Modelling.csv')\n",

"df.info()"

]

},

{

"cell\_type": "markdown",

"id": "541c4b90",

"metadata": {},

"source": [

"###### 3.Perform Below Visualizations"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "2b15550c",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:ylabel='Frequency'>"

]

},

"execution\_count": 2,

"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": [

"# 3.1.Univariate Analysis\n",

"import matplotlib.pyplot as plt\n",

"\n",

"df[df['EstimatedSalary'] < 182012.4]['EstimatedSalary'].plot.hist()"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "91acfde6",

"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>13</th>\n",

" <td>14</td>\n",

" <td>15691483</td>\n",

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

" <td>549</td>\n",

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

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

" <td>25</td>\n",

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

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

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

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

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

" <td>190857.79</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>19</td>\n",

" <td>15661507</td>\n",

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

" <td>587</td>\n",

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

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

" <td>45</td>\n",

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

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

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

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

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

" <td>158684.81</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>21</td>\n",

" <td>15577657</td>\n",

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

" <td>732</td>\n",

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

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

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

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

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

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

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

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

" <td>170886.17</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>24</th>\n",

" <td>25</td>\n",

" <td>15625047</td>\n",

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

" <td>846</td>\n",

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

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

" <td>38</td>\n",

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

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

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

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

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

" <td>187616.16</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>26</th>\n",

" <td>27</td>\n",

" <td>15736816</td>\n",

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

" <td>756</td>\n",

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

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

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

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

" <td>136815.64</td>\n",

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

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

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

" <td>170041.95</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",

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

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

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

" </tr>\n",

" <tr>\n",

" <th>9978</th>\n",

" <td>9979</td>\n",

" <td>15703563</td>\n",

" <td>P'eng</td>\n",

" <td>774</td>\n",

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

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

" <td>40</td>\n",

" <td>9</td>\n",

" <td>93017.47</td>\n",

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

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

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

" <td>191608.97</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>9983</th>\n",

" <td>9984</td>\n",

" <td>15656710</td>\n",

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

" <td>613</td>\n",

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

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

" <td>40</td>\n",

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

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

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

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

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

" <td>151325.24</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>9989</th>\n",

" <td>9990</td>\n",

" <td>15605622</td>\n",

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

" <td>841</td>\n",

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

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

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

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

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

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

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

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

" <td>179436.60</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>9992</th>\n",

" <td>9993</td>\n",

" <td>15657105</td>\n",

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

" <td>726</td>\n",

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

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

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

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

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

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

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

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

" <td>195192.40</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>9994</th>\n",

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

" <td>15719294</td>\n",

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

" <td>800</td>\n",

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

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

" <td>29</td>\n",

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

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

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

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

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

" <td>167773.55</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>2455 rows × 14 columns</p>\n",

"</div>"

],

"text/plain": [

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

"13 14 15691483 Chin 549 France Female 25 \n",

"18 19 15661507 Muldrow 587 Spain Male 45 \n",

"20 21 15577657 McDonald 732 France Male 41 \n",

"24 25 15625047 Yen 846 France Female 38 \n",

"26 27 15736816 Young 756 Germany Male 36 \n",

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

"9978 9979 15703563 P'eng 774 France Male 40 \n",

"9983 9984 15656710 Cocci 613 France Male 40 \n",

"9989 9990 15605622 McMillan 841 Spain Male 28 \n",

"9992 9993 15657105 Chukwualuka 726 Spain Male 36 \n",

"9994 9995 15719294 Wood 800 France Female 29 \n",

"\n",

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

"13 5 0.00 2 0 0 \n",

"18 6 0.00 1 0 0 \n",

"20 8 0.00 2 1 1 \n",

"24 5 0.00 1 1 1 \n",

"26 2 136815.64 1 1 1 \n",

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

"9978 9 93017.47 2 1 0 \n",

"9983 4 0.00 1 0 0 \n",

"9989 4 0.00 2 1 1 \n",

"9992 2 0.00 1 1 0 \n",

"9994 2 0.00 2 0 0 \n",

"\n",

" EstimatedSalary Exited \n",

"13 190857.79 0 \n",

"18 158684.81 0 \n",

"20 170886.17 0 \n",

"24 187616.16 0 \n",

"26 170041.95 0 \n",

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

"9978 191608.97 0 \n",

"9983 151325.24 0 \n",

"9989 179436.60 0 \n",

"9992 195192.40 0 \n",

"9994 167773.55 0 \n",

"\n",

"[2455 rows x 14 columns]"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[df['EstimatedSalary'] > 150000]"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "f3ff8340",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"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": [

"# 3.2.Bi-Variate Analysis\n",

"df[df['EstimatedSalary'] < 10000].sample(100).plot.scatter(x='EstimatedSalary', y='CreditScore')"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "251737d0",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x1080 with 64 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"# 3.3.Multi-Variate Analysis\n",

"pd.plotting.scatter\_matrix(df.loc[:, \"CreditScore\":\"EstimatedSalary\"], diagonal=\"kde\",figsize=(15,15))\n",

"plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "343c9f40",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1440x1080 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"ax = df[[\"CreditScore\",\"Age\"]].plot(figsize=(20,15))\n",

"ax.legend(loc='center left', bbox\_to\_anchor=(1, 0.5));"

]

},

{

"cell\_type": "markdown",

"id": "7d181744",

"metadata": {},

"source": [

"###### 4.Perform descriptive statistics on the Dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "baf13835",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"100090.2398809998\n"

]

}

],

"source": [

"import numpy as np\n",

"print(df['EstimatedSalary'].mean())"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "d9d04e68",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"1000902398.8100001\n"

]

}

],

"source": [

"import numpy as np\n",

"print(df['EstimatedSalary'].sum())"

]

},

{

"cell\_type": "markdown",

"id": "af6d2b4d",

"metadata": {},

"source": [

"###### 5.Handle the Missing Values"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "01050542",

"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": 10,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"missing\_values\_count = df.isnull().sum()\n",

"missing\_values\_count[0:15]"

]

},

{

"cell\_type": "markdown",

"id": "21a91247",

"metadata": {},

"source": [

"###### 6.Find the Outliers and replace the outliers"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "e9ad82b0",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"5.0\n"

]

}

],

"source": [

"median = float(df['Tenure'].median())\n",

"df[\"Tenure\"] = np.where(df[\"Tenure\"] > median, median, df['Tenure'])\n",

"print(median)"

]

},

{

"cell\_type": "markdown",

"id": "b612bbc9",

"metadata": {},

"source": [

"###### 7.Check for categorical columns and perform encoding"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "0f05f470",

"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>2.0</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>1.0</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>5.0</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>1.0</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>2.0</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 0.00 1 1 1 \n",

"1 1.0 83807.86 1 0 1 \n",

"2 5.0 159660.80 3 1 0 \n",

"3 1.0 0.00 2 0 0 \n",

"4 2.0 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": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "46c258a8",

"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>9995</th>\n",

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

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

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

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

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

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

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

" <td>5.0</td>\n",

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

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

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

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

" <td>96270.64</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>5.0</td>\n",

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

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

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

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

" <td>101699.77</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>5.0</td>\n",

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

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

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

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

" <td>42085.58</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

" <td>92888.52</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>4.0</td>\n",

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

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

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

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

" <td>38190.78</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"9995 9996 15606229 Obijiaku 771 France Male 39 \n",

"9996 9997 15569892 Johnstone 516 France Male 35 \n",

"9997 9998 15584532 Liu 709 France Female 36 \n",

"9998 9999 15682355 Sabbatini 772 Germany Male 42 \n",

"9999 10000 15628319 Walker 792 France Female 28 \n",

"\n",

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

"9995 5.0 0.00 2 1 0 \n",

"9996 5.0 57369.61 1 1 1 \n",

"9997 5.0 0.00 1 0 1 \n",

"9998 3.0 75075.31 2 1 0 \n",

"9999 4.0 130142.79 1 1 0 \n",

"\n",

" EstimatedSalary Exited \n",

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 "

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.tail()"

]

},

{

"cell\_type": "markdown",

"id": "8c1be16d",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"id": "dc105ee2",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[1 0 1 ... 1 1 0]\n"

]

}

],

"source": [

"# 8.1.Dependent Variables\n",

"Y = df.iloc[:, -1].values\n",

"print(Y)"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "8175c2b5",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[1 15634602 'Hargrave' ... 1 1 101348.88]\n",

" [2 15647311 'Hill' ... 0 1 112542.58]\n",

" [3 15619304 'Onio' ... 1 0 113931.57]\n",

" ...\n",

" [9998 15584532 'Liu' ... 0 1 42085.58]\n",

" [9999 15682355 'Sabbatini' ... 1 0 92888.52]\n",

" [10000 15628319 'Walker' ... 1 0 38190.78]]\n"

]

}

],

"source": [

"# 8.2.Independent Variables\n",

"X = df.iloc[:, :-1].values\n",

"print(X)"

]

},

{

"cell\_type": "markdown",

"id": "2f117d50",

"metadata": {},

"source": [

"###### 9.Scale the independent variables"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "1c76d74a",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

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

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

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

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

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

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

"9995 9996 15606229 Obijiaku 771 France Male \n",

"9996 9997 15569892 Johnstone 516 France Male \n",

"9997 9998 15584532 Liu 709 France Female \n",

"9998 9999 15682355 Sabbatini 772 Germany Male \n",

"9999 10000 15628319 Walker 792 France Female \n",

"\n",

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

"0 0.324324 0.4 0.00 1 1 1 \n",

"1 0.310811 0.2 83807.86 1 0 1 \n",

"2 0.324324 1.0 159660.80 3 1 0 \n",

"3 0.283784 0.2 0.00 2 0 0 \n",

"4 0.337838 0.4 125510.82 1 1 1 \n",

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

"9995 0.283784 1.0 0.00 2 1 0 \n",

"9996 0.229730 1.0 57369.61 1 1 1 \n",

"9997 0.243243 1.0 0.00 1 0 1 \n",

"9998 0.324324 0.6 75075.31 2 1 0 \n",

"9999 0.135135 0.8 130142.79 1 1 0 \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 \n",

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

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 \n",

"\n",

"[10000 rows x 14 columns]\n"

]

}

],

"source": [

"from sklearn.preprocessing import MinMaxScaler\n",

"min\_max\_scaler = MinMaxScaler()\n",

"df[[\"Age\", \"Tenure\"]] = min\_max\_scaler.fit\_transform(df[[\"Age\", \"Tenure\"]])\n",

"print(df)"

]

},

{

"cell\_type": "markdown",

"id": "a89307b0",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"id": "8055063a",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

"3555 3556 15629094 Fomin 528 France Female \n",

"4078 4079 15690440 Stiles 656 Spain Male \n",

"8445 8446 15678333 Parry-Okeden 683 France Female \n",

"5939 5940 15709861 He 766 Germany Male \n",

"5583 5584 15620579 Dunn 695 Spain Female \n",

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

"805 806 15756026 Hooper 790 Spain Female \n",

"4169 4170 15716728 Basedow 513 Spain Female \n",

"588 589 15614782 Hao 526 France Male \n",

"6968 6969 15721793 Chiu 510 Germany Female \n",

"2547 2548 15634772 Mario 682 Spain Female \n",

"\n",

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

"3555 0.243243 0.2 156948.41 1 1 1 \n",

"4078 0.391892 0.2 0.00 2 1 1 \n",

"8445 0.108108 1.0 0.00 2 1 0 \n",

"5939 0.162162 0.8 127786.28 2 1 1 \n",

"5583 0.175676 1.0 0.00 2 0 1 \n",

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

"805 0.378378 1.0 0.00 1 0 0 \n",

"4169 0.324324 1.0 0.00 2 0 1 \n",

"588 0.243243 0.2 0.00 1 1 0 \n",

"6968 0.432432 1.0 123936.54 1 1 1 \n",

"2547 0.554054 0.0 122661.39 1 0 1 \n",

"\n",

" EstimatedSalary Exited \n",

"3555 149912.28 1 \n",

"4078 197961.93 0 \n",

"8445 86619.77 0 \n",

"5939 28879.30 0 \n",

"5583 131644.41 0 \n",

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

"805 14679.81 1 \n",

"4169 73151.25 0 \n",

"588 160696.72 0 \n",

"6968 23768.01 0 \n",

"2547 84803.76 0 \n",

"\n",

"[8000 rows x 14 columns]\n"

]

}

],

"source": [

"training\_data = df.sample(frac=0.8, random\_state=25)\n",

"testing\_data = df.drop(training\_data.index)\n",

"print(training\_data)"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "80fe2100",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

"8 9 15792365 He 501 France Male \n",

"14 15 15600882 Scott 635 Spain Female \n",

"23 24 15725737 Mosman 669 France Male \n",

"25 26 15738191 Maclean 577 France Male \n",

"26 27 15736816 Young 756 Germany Male \n",

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

"9954 9955 15739850 Trentino 645 France Male \n",

"9959 9960 15677783 Graham 764 Spain Male \n",

"9986 9987 15581736 Bartlett 673 Germany Male \n",

"9988 9989 15589329 Pirozzi 775 France Male \n",

"9995 9996 15606229 Obijiaku 771 France Male \n",

"\n",

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

"8 0.351351 0.8 142051.07 2 0 1 \n",

"14 0.229730 1.0 0.00 2 1 1 \n",

"23 0.378378 0.6 0.00 2 0 1 \n",

"25 0.094595 0.6 0.00 2 0 1 \n",

"26 0.243243 0.4 136815.64 1 1 1 \n",

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

"9954 0.364865 1.0 155417.61 1 0 1 \n",

"9959 0.270270 0.8 113607.47 1 1 0 \n",

"9986 0.391892 0.2 183579.54 2 0 1 \n",

"9988 0.162162 0.8 0.00 2 1 0 \n",

"9995 0.283784 1.0 0.00 2 1 0 \n",

"\n",

" EstimatedSalary Exited \n",

"8 74940.50 0 \n",

"14 65951.65 0 \n",

"23 8487.75 0 \n",

"25 124508.29 0 \n",

"26 170041.95 0 \n",

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

"9954 3449.22 0 \n",

"9959 91094.46 0 \n",

"9986 34047.54 0 \n",

"9988 49337.84 0 \n",

"9995 96270.64 0 \n",

"\n",

"[2000 rows x 14 columns]\n"

]

}

],

"source": [

"print(testing\_data)"

]

}

],

"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": "3.9.7"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}