{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"execution\_count": 24,

"metadata": {

"id": "xjPnsK9H4bOR"

},

"outputs": [],

"source": [

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"import numpy as np\n",

"sns.set\_style('darkgrid')\n",

"sns.set(font\_scale=1.3)"

]

},

{

"cell\_type": "code",

"source": [

"df=pd.read\_csv(\"/content/drive/MyDrive/IBM/Assignment - 2 /Churn\_Modelling.csv\")"

],

"metadata": {

"id": "TVSC1aBB4udm"

},

"execution\_count": 25,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"df.head()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 270

},

"id": "cSx6-Xbp5Lcr",

"outputId": "eeb9d064-ac76-48b9-c573-7709d0a0b8f3"

},

"execution\_count": 26,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"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 "

],

"text/html": [

"\n",

" <div id=\"df-7a2e5609-ac7a-4bcf-89ab-c8cac6a64b60\">\n",

" <div class=\"colab-df-container\">\n",

" <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>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-7a2e5609-ac7a-4bcf-89ab-c8cac6a64b60')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-7a2e5609-ac7a-4bcf-89ab-c8cac6a64b60 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-7a2e5609-ac7a-4bcf-89ab-c8cac6a64b60');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 26

}

]

},

{

"cell\_type": "code",

"source": [

"df.drop([\"RowNumber\",\"CustomerId\",\"Surname\"],axis=1,inplace=True)"

],

"metadata": {

"id": "2dtrUKrH4f\_V"

},

"execution\_count": 29,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"df.info()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "ZbBqA4Ngm7B0",

"outputId": "67f6b10c-8953-4843-c1b4-fdbb92901ada"

},

"execution\_count": 30,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

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

"Data columns (total 11 columns):\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

"dtypes: float64(2), int64(7), object(2)\n",

"memory usage: 859.5+ KB\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Univariate Analysis\n",

"plt.figure(figsize=(8,8))\n",

"sns.countplot(x='Tenure',data=df)\n",

"plt.xlabel('0:Customers with Bank, 1: exited from bank')\n",

"plt.ylabel('No.of.Customers')\n",

"plt.title(\"Bank Customers viz\")\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 526

},

"id": "GhD5wsQmhBxb",

"outputId": "63de0294-0e79-4bfb-a5dd-d150e0595b8e"

},

"execution\_count": 28,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 576x576 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Univariate Analysis\n",

"plt.figure(figsize=(8,8))\n",

"sns.kdeplot(x=df['Balance'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 539

},

"id": "244VxHucnITi",

"outputId": "175d02bf-fabc-46d0-9f3a-73f1fff053f8"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fa0c03906d0>"

]

},

"metadata": {},

"execution\_count": 9

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 576x576 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Bivariate Analysis \n",

"plt.scatter(df.Age,df.Balance)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 289

},

"id": "08uNa2XPjdUY",

"outputId": "afa4a00d-27e7-4576-a670-5f8279bd352d"

},

"execution\_count": 10,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fa0d35a7dd0>"

]

},

"metadata": {},

"execution\_count": 10

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Bivariate Analysis\n",

"df.corr()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 427

},

"id": "DF2AVP6Qn6CY",

"outputId": "6f1b5829-0282-4704-fd4b-8c80a22c9b5a"

},

"execution\_count": 54,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" CreditScore Gender Age Tenure Balance \\\n",

"CreditScore 1.000000 0.007888 -0.003965 0.000842 0.006268 \n",

"Gender 0.007888 1.000000 0.022812 0.003739 0.069408 \n",

"Age -0.003965 0.022812 1.000000 -0.009997 0.028308 \n",

"Tenure 0.000842 0.003739 -0.009997 1.000000 -0.012254 \n",

"Balance 0.006268 0.069408 0.028308 -0.012254 1.000000 \n",

"NumOfProducts 0.012238 0.003972 -0.030680 0.013444 -0.304180 \n",

"HasCrCard -0.005458 -0.008523 -0.011721 0.022583 -0.014858 \n",

"IsActiveMember 0.025651 0.006724 0.085472 -0.028362 -0.010084 \n",

"EstimatedSalary -0.001384 -0.001369 -0.007201 0.007784 0.012797 \n",

"Exited -0.027094 0.035943 0.285323 -0.014001 0.118533 \n",

"\n",

" NumOfProducts HasCrCard IsActiveMember EstimatedSalary \\\n",

"CreditScore 0.012238 -0.005458 0.025651 -0.001384 \n",

"Gender 0.003972 -0.008523 0.006724 -0.001369 \n",

"Age -0.030680 -0.011721 0.085472 -0.007201 \n",

"Tenure 0.013444 0.022583 -0.028362 0.007784 \n",

"Balance -0.304180 -0.014858 -0.010084 0.012797 \n",

"NumOfProducts 1.000000 0.003183 0.009612 0.014204 \n",

"HasCrCard 0.003183 1.000000 -0.011866 -0.009933 \n",

"IsActiveMember 0.009612 -0.011866 1.000000 -0.011421 \n",

"EstimatedSalary 0.014204 -0.009933 -0.011421 1.000000 \n",

"Exited -0.047820 -0.007138 -0.156128 0.012097 \n",

"\n",

" Exited \n",

"CreditScore -0.027094 \n",

"Gender 0.035943 \n",

"Age 0.285323 \n",

"Tenure -0.014001 \n",

"Balance 0.118533 \n",

"NumOfProducts -0.047820 \n",

"HasCrCard -0.007138 \n",

"IsActiveMember -0.156128 \n",

"EstimatedSalary 0.012097 \n",

"Exited 1.000000 "

],

"text/html": [

"\n",

" <div id=\"df-ad99fe11-af0f-4ecf-9a6f-439b4c1cf072\">\n",

" <div class=\"colab-df-container\">\n",

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

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

" <td>0.007888</td>\n",

" <td>-0.003965</td>\n",

" <td>0.000842</td>\n",

" <td>0.006268</td>\n",

" <td>0.012238</td>\n",

" <td>-0.005458</td>\n",

" <td>0.025651</td>\n",

" <td>-0.001384</td>\n",

" <td>-0.027094</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.007888</td>\n",

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

" <td>0.022812</td>\n",

" <td>0.003739</td>\n",

" <td>0.069408</td>\n",

" <td>0.003972</td>\n",

" <td>-0.008523</td>\n",

" <td>0.006724</td>\n",

" <td>-0.001369</td>\n",

" <td>0.035943</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>-0.003965</td>\n",

" <td>0.022812</td>\n",

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

" <td>-0.009997</td>\n",

" <td>0.028308</td>\n",

" <td>-0.030680</td>\n",

" <td>-0.011721</td>\n",

" <td>0.085472</td>\n",

" <td>-0.007201</td>\n",

" <td>0.285323</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.000842</td>\n",

" <td>0.003739</td>\n",

" <td>-0.009997</td>\n",

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

" <td>-0.012254</td>\n",

" <td>0.013444</td>\n",

" <td>0.022583</td>\n",

" <td>-0.028362</td>\n",

" <td>0.007784</td>\n",

" <td>-0.014001</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.006268</td>\n",

" <td>0.069408</td>\n",

" <td>0.028308</td>\n",

" <td>-0.012254</td>\n",

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

" <td>-0.304180</td>\n",

" <td>-0.014858</td>\n",

" <td>-0.010084</td>\n",

" <td>0.012797</td>\n",

" <td>0.118533</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.012238</td>\n",

" <td>0.003972</td>\n",

" <td>-0.030680</td>\n",

" <td>0.013444</td>\n",

" <td>-0.304180</td>\n",

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

" <td>0.003183</td>\n",

" <td>0.009612</td>\n",

" <td>0.014204</td>\n",

" <td>-0.047820</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>-0.005458</td>\n",

" <td>-0.008523</td>\n",

" <td>-0.011721</td>\n",

" <td>0.022583</td>\n",

" <td>-0.014858</td>\n",

" <td>0.003183</td>\n",

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

" <td>-0.011866</td>\n",

" <td>-0.009933</td>\n",

" <td>-0.007138</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.025651</td>\n",

" <td>0.006724</td>\n",

" <td>0.085472</td>\n",

" <td>-0.028362</td>\n",

" <td>-0.010084</td>\n",

" <td>0.009612</td>\n",

" <td>-0.011866</td>\n",

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

" <td>-0.011421</td>\n",

" <td>-0.156128</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>-0.001384</td>\n",

" <td>-0.001369</td>\n",

" <td>-0.007201</td>\n",

" <td>0.007784</td>\n",

" <td>0.012797</td>\n",

" <td>0.014204</td>\n",

" <td>-0.009933</td>\n",

" <td>-0.011421</td>\n",

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

" <td>0.012097</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>-0.027094</td>\n",

" <td>0.035943</td>\n",

" <td>0.285323</td>\n",

" <td>-0.014001</td>\n",

" <td>0.118533</td>\n",

" <td>-0.047820</td>\n",

" <td>-0.007138</td>\n",

" <td>-0.156128</td>\n",

" <td>0.012097</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ad99fe11-af0f-4ecf-9a6f-439b4c1cf072')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-ad99fe11-af0f-4ecf-9a6f-439b4c1cf072 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-ad99fe11-af0f-4ecf-9a6f-439b4c1cf072');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 54

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Bivariate Analysis\n",

"import statsmodels.api as sm\n",

"\n",

"#define response variable\n",

"y = df['CreditScore']\n",

"\n",

"#define explanatory variable\n",

"x = df[['EstimatedSalary']]\n",

"\n",

"#add constant to predictor variables\n",

"x = sm.add\_constant(x)\n",

"\n",

"#fit linear regression model\n",

"model = sm.OLS(y, x).fit()\n",

"\n",

"#view model summary\n",

"print(model.summary())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "JYaJzAX6n6I1",

"outputId": "36037946-db28-4c5e-a95c-9aebba238f6c"

},

"execution\_count": 36,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" OLS Regression Results \n",

"==============================================================================\n",

"Dep. Variable: CreditScore R-squared: 0.000\n",

"Model: OLS Adj. R-squared: -0.000\n",

"Method: Least Squares F-statistic: 0.01916\n",

"Date: Sat, 24 Sep 2022 Prob (F-statistic): 0.890\n",

"Time: 05:06:19 Log-Likelihood: -59900.\n",

"No. Observations: 10000 AIC: 1.198e+05\n",

"Df Residuals: 9998 BIC: 1.198e+05\n",

"Df Model: 1 \n",

"Covariance Type: nonrobust \n",

"===================================================================================\n",

" coef std err t P>|t| [0.025 0.975]\n",

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

"const 650.7617 1.940 335.407 0.000 646.958 654.565\n",

"EstimatedSalary -2.326e-06 1.68e-05 -0.138 0.890 -3.53e-05 3.06e-05\n",

"==============================================================================\n",

"Omnibus: 132.939 Durbin-Watson: 2.014\n",

"Prob(Omnibus): 0.000 Jarque-Bera (JB): 84.242\n",

"Skew: -0.072 Prob(JB): 5.10e-19\n",

"Kurtosis: 2.574 Cond. No. 2.32e+05\n",

"==============================================================================\n",

"\n",

"Notes:\n",

"[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.\n",

"[2] The condition number is large, 2.32e+05. This might indicate that there are\n",

"strong multicollinearity or other numerical problems.\n"

]

},

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:142: FutureWarning: In a future version of pandas all arguments of concat except for the argument 'objs' will be keyword-only\n",

" x = pd.concat(x[::order], 1)\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Multivariate Analysis\n",

"plt.figure(figsize=(4,4))\n",

"sns.pairplot(data=df[[\"Balance\",\"CreditScore\",\"EstimatedSalary\",\"NumOfProducts\",\"Tenure\",\"Exited\"]],hue=\"Exited\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 925

},

"id": "A4zmHGbtn6NI",

"outputId": "c26bd1f6-d0e2-4aa3-f4cd-c3e461858d9c"

},

"execution\_count": 35,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x7fa0b00a1b10>"

]

},

"metadata": {},

"execution\_count": 35

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 288x288 with 0 Axes>"

]

},

"metadata": {}

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 960.69x900 with 30 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Descriptive Statistics\n",

"df=pd.DataFrame(df)\n",

"print(df.sum())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3d5liF1vn6Rt",

"outputId": "20ab9ea4-9f16-4f9b-e0df-9b769b49a1b6"

},

"execution\_count": 40,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"CreditScore 6505288\n",

"Geography FranceSpainFranceFranceSpainSpainFranceGermany...\n",

"Gender FemaleFemaleFemaleFemaleFemaleMaleMaleFemaleMa...\n",

"Age 389218\n",

"Tenure 50128\n",

"Balance 764858892.88\n",

"NumOfProducts 15302\n",

"HasCrCard 7055\n",

"IsActiveMember 5151\n",

"EstimatedSalary 1000902398.81\n",

"Exited 2037\n",

"dtype: object\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Descriptive Statistics\n",

"print(\"----Sum Value-----\")\n",

"print(df.sum(1))\n",

"print(\"----------------------------------\")\n",

"print(\"-----Product Value-----\")\n",

"print(df.prod())\n",

"print(\"----------------------------------\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "w6XIplHdtmFV",

"outputId": "387a63c8-a1a6-4cc5-d9a5-c4655c96a4b1"

},

"execution\_count": 39,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"----Sum Value-----\n",

"0 102015.88\n",

"1 197002.44\n",

"2 274149.37\n",

"3 94567.63\n",

"4 205492.92\n",

" ... \n",

"9995 97088.64\n",

"9996 159633.38\n",

"9997 42840.58\n",

"9998 168784.83\n",

"9999 169159.57\n",

"Length: 10000, dtype: float64\n",

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

"-----Product Value-----\n",

"CreditScore 0.0\n",

"Age 0.0\n",

"Tenure 0.0\n",

"Balance 0.0\n",

"NumOfProducts 0.0\n",

"HasCrCard 0.0\n",

"IsActiveMember 0.0\n",

"EstimatedSalary inf\n",

"Exited 0.0\n",

"dtype: float64\n",

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

]

},

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.\n",

" This is separate from the ipykernel package so we can avoid doing imports until\n",

"/usr/local/lib/python3.7/dist-packages/numpy/core/\_methods.py:52: RuntimeWarning: overflow encountered in reduce\n",

" return umr\_prod(a, axis, dtype, out, keepdims, initial, where)\n",

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:6: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.\n",

" \n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Perform Descriptive Statistics\n",

"print(\"----------Mean Value-----------\")\n",

"print(df.mean())\n",

"print(\"-------------------------------\")\n",

"print(\"----------Median Value---------\")\n",

"print(df.median())\n",

"print(\"-------------------------------\")\n",

"print(\"----------Mode Value------------\")\n",

"print(df.mode())\n",

"print(\"-------------------------------\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "O0cpbrbQv\_Qe",

"outputId": "daabf582-1e3b-4c08-cbfb-c848b543e1c6"

},

"execution\_count": 38,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"----------Mean Value-----------\n",

"CreditScore 650.528800\n",

"Age 38.921800\n",

"Tenure 5.012800\n",

"Balance 76485.889288\n",

"NumOfProducts 1.530200\n",

"HasCrCard 0.705500\n",

"IsActiveMember 0.515100\n",

"EstimatedSalary 100090.239881\n",

"Exited 0.203700\n",

"dtype: float64\n",

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

"----------Median Value---------\n",

"CreditScore 652.000\n",

"Age 37.000\n",

"Tenure 5.000\n",

"Balance 97198.540\n",

"NumOfProducts 1.000\n",

"HasCrCard 1.000\n",

"IsActiveMember 1.000\n",

"EstimatedSalary 100193.915\n",

"Exited 0.000\n",

"dtype: float64\n",

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

"----------Mode Value------------\n",

" CreditScore Geography Gender Age Tenure Balance NumOfProducts \\\n",

"0 850 France Male 37 2 0.0 1 \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary Exited \n",

"0 1 1 24924.92 0 \n",

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

]

},

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.\n",

" This is separate from the ipykernel package so we can avoid doing imports until\n",

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:6: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.\n",

" \n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Handling with missing Values\n",

"df.isnull()#Checking values are null"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 488

},

"id": "fccEJs8AxJay",

"outputId": "0260e613-51f2-446e-d53a-39403d082ec9"

},

"execution\_count": 41,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" CreditScore Geography Gender Age Tenure Balance NumOfProducts \\\n",

"0 False False False False False False False \n",

"1 False False False False False False False \n",

"2 False False False False False False False \n",

"3 False False False False False False False \n",

"4 False False False False False False False \n",

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

"9995 False False False False False False False \n",

"9996 False False False False False False False \n",

"9997 False False False False False False False \n",

"9998 False False False False False False False \n",

"9999 False False False False False False False \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary Exited \n",

"0 False False False False \n",

"1 False False False False \n",

"2 False False False False \n",

"3 False False False False \n",

"4 False False False False \n",

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

"9995 False False False False \n",

"9996 False False False False \n",

"9997 False False False False \n",

"9998 False False False False \n",

"9999 False False False False \n",

"\n",

"[10000 rows x 11 columns]"

],

"text/html": [

"\n",

" <div id=\"df-26afd5b0-2cd1-46d0-8dea-6414cc00c0a9\">\n",

" <div class=\"colab-df-container\">\n",

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 11 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-26afd5b0-2cd1-46d0-8dea-6414cc00c0a9')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-26afd5b0-2cd1-46d0-8dea-6414cc00c0a9 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-26afd5b0-2cd1-46d0-8dea-6414cc00c0a9');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 41

}

]

},

{

"cell\_type": "code",

"source": [

"#Handling with missing Values\n",

"df.notnull()#Checking values are not null"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 488

},

"id": "\_ZS4SlF7yGlY",

"outputId": "af63c971-6da6-47fc-a63a-fece5f0d6393"

},

"execution\_count": 42,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" CreditScore Geography Gender Age Tenure Balance NumOfProducts \\\n",

"0 True True True True True True True \n",

"1 True True True True True True True \n",

"2 True True True True True True True \n",

"3 True True True True True True True \n",

"4 True True True True True True True \n",

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

"9995 True True True True True True True \n",

"9996 True True True True True True True \n",

"9997 True True True True True True True \n",

"9998 True True True True True True True \n",

"9999 True True True True True True True \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary Exited \n",

"0 True True True True \n",

"1 True True True True \n",

"2 True True True True \n",

"3 True True True True \n",

"4 True True True True \n",

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

"9995 True True True True \n",

"9996 True True True True \n",

"9997 True True True True \n",

"9998 True True True True \n",

"9999 True True True True \n",

"\n",

"[10000 rows x 11 columns]"

],

"text/html": [

"\n",

" <div id=\"df-64cc4df9-11a2-4fc8-838b-8f0f1ac6715e\">\n",

" <div class=\"colab-df-container\">\n",

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 11 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-64cc4df9-11a2-4fc8-838b-8f0f1ac6715e')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-64cc4df9-11a2-4fc8-838b-8f0f1ac6715e button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-64cc4df9-11a2-4fc8-838b-8f0f1ac6715e');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 42

}

]

},

{

"cell\_type": "code",

"source": [

"#Find outliers & replace the outliers\n",

"sns.boxplot(df['Balance'])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 363

},

"id": "VPhv3ENi06La",

"outputId": "28dcbbb1-e2b6-4bd6-d2dc-fcb7025bad4f"

},

"execution\_count": 43,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.\n",

" FutureWarning\n"

]

},

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fa0af6dcf90>"

]

},

"metadata": {},

"execution\_count": 43

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"#Find outliers & replace the outliers\n",

"print(np.where(df['Balance']>100000))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "kPtx-p3H0\_vd",

"outputId": "5c8a7572-5fae-4aab-ec01-dd9602bf9011"

},

"execution\_count": 44,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(array([ 2, 4, 5, ..., 9987, 9993, 9999]),)\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Find outliers & replace the outliers\n",

"from scipy import stats\n",

"import numpy as np\n",

" \n",

"z = np.abs(stats.zscore(df[\"EstimatedSalary\"]))\n",

"print(z)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "BDtanEN4z6qt",

"outputId": "0fbeae8b-ce0a-4eb4-9643-cdd41604e61f"

},

"execution\_count": 45,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"0 0.021886\n",

"1 0.216534\n",

"2 0.240687\n",

"3 0.108918\n",

"4 0.365276\n",

" ... \n",

"9995 0.066419\n",

"9996 0.027988\n",

"9997 1.008643\n",

"9998 0.125231\n",

"9999 1.076370\n",

"Name: EstimatedSalary, Length: 10000, dtype: float64\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Check for categorical columns & performs encoding\n",

"from sklearn.preprocessing import LabelEncoder\n",

"df['Gender'].unique()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "gNaP4k2uKL\_C",

"outputId": "8764f6f6-e827-48e9-ddd1-b90d6ff17508"

},

"execution\_count": 48,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array(['Female', 'Male'], dtype=object)"

]

},

"metadata": {},

"execution\_count": 48

}

]

},

{

"cell\_type": "code",

"source": [

"#Check for categorical columns & performs encoding\n",

"df['Gender'].value\_counts()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "fiVWeu-431Fq",

"outputId": "1d93e2f3-721b-463d-f79a-599e7d494368"

},

"execution\_count": 49,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Male 5457\n",

"Female 4543\n",

"Name: Gender, dtype: int64"

]

},

"metadata": {},

"execution\_count": 49

}

]

},

{

"cell\_type": "code",

"source": [

"#Check for categorical columns & performs encoding\n",

"encoding=LabelEncoder()\n",

"df[\"Gender\"]=encoding.fit\_transform(df.iloc[:,1].values)\n",

"df"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 488

},

"id": "V8Qb6O2c5imZ",

"outputId": "bbdcab03-2d43-41f0-d216-8d69b011030c"

},

"execution\_count": 57,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" CreditScore Geography Gender Age Tenure Balance NumOfProducts \\\n",

"0 619 France 0 42 2 0.00 1 \n",

"1 608 Spain 2 41 1 83807.86 1 \n",

"2 502 France 0 42 8 159660.80 3 \n",

"3 699 France 0 39 1 0.00 2 \n",

"4 850 Spain 2 43 2 125510.82 1 \n",

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

"9995 771 France 0 39 5 0.00 2 \n",

"9996 516 France 0 35 10 57369.61 1 \n",

"9997 709 France 0 36 7 0.00 1 \n",

"9998 772 Germany 1 42 3 75075.31 2 \n",

"9999 792 France 0 28 4 130142.79 1 \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary Exited \n",

"0 1 1 101348.88 1 \n",

"1 0 1 112542.58 0 \n",

"2 1 0 113931.57 1 \n",

"3 0 0 93826.63 0 \n",

"4 1 1 79084.10 0 \n",

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

"9995 1 0 96270.64 0 \n",

"9996 1 1 101699.77 0 \n",

"9997 0 1 42085.58 1 \n",

"9998 1 0 92888.52 1 \n",

"9999 1 0 38190.78 0 \n",

"\n",

"[10000 rows x 11 columns]"

],

"text/html": [

"\n",

" <div id=\"df-e6909c37-e349-4ad9-bc98-83ae9498bb3c\">\n",

" <div class=\"colab-df-container\">\n",

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

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

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

" <td>2</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>502</td>\n",

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

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

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

" <td>2</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",

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

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

" <td>0</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>96270.64</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>0</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>101699.77</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 11 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-e6909c37-e349-4ad9-bc98-83ae9498bb3c')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-e6909c37-e349-4ad9-bc98-83ae9498bb3c button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-e6909c37-e349-4ad9-bc98-83ae9498bb3c');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 57

}

]

},

{

"cell\_type": "code",

"source": [

"#Check for categorical columns & performs encoding\n"

],

"metadata": {

"id": "34X8xKGZ6cvx"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"#Split the data into Dependent & Independent Variables\n",

"print(\"----------Dependent Variables----------\")\n",

"X=df.iloc[:,1:4]\n",

"print(X)\n",

"print(\"---------------------------------------\")\n",

"print(\"---------Independent Variables---------\")\n",

"Y=df.iloc[:,4]\n",

"print(Y)\n",

"print(\"---------------------------------------\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "sJuCT-DYJMLt",

"outputId": "269bc4e3-5ca1-4fb2-9a96-457647655888"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"----------Dependent Variables-----------\n",

" Age Tenure Balance\n",

"0 42 2 0.00\n",

"1 41 1 83807.86\n",

"2 42 8 159660.80\n",

"3 39 1 0.00\n",

"4 43 2 125510.82\n",

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

"9995 39 5 0.00\n",

"9996 35 10 57369.61\n",

"9997 36 7 0.00\n",

"9998 42 3 75075.31\n",

"9999 28 4 130142.79\n",

"\n",

"[10000 rows x 3 columns]\n",

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

"---------Independent Variables---------\n",

"0 1\n",

"1 1\n",

"2 3\n",

"3 2\n",

"4 1\n",

" ..\n",

"9995 2\n",

"9996 1\n",

"9997 1\n",

"9998 2\n",

"9999 1\n",

"Name: NumOfProducts, Length: 10000, dtype: int64\n",

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

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Scale the independent Variables\n",

"from sklearn.preprocessing import StandardScaler\n",

"object= StandardScaler()\n",

"# standardization \n",

"scale = object.fit\_transform(df) \n",

"print(scale)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "dj\_IrRNLHFn3",

"outputId": "caa37ac7-8284-4fef-fd64-b980f1d58621"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[[-0.32622142 0.29351742 -1.04175968 ... 0.97024255 0.02188649\n",

" 1.97716468]\n",

" [-0.44003595 0.19816383 -1.38753759 ... 0.97024255 0.21653375\n",

" -0.50577476]\n",

" [-1.53679418 0.29351742 1.03290776 ... -1.03067011 0.2406869\n",

" 1.97716468]\n",

" ...\n",

" [ 0.60498839 -0.27860412 0.68712986 ... 0.97024255 -1.00864308\n",

" 1.97716468]\n",

" [ 1.25683526 0.29351742 -0.69598177 ... -1.03067011 -0.12523071\n",

" 1.97716468]\n",

" [ 1.46377078 -1.04143285 -0.35020386 ... -1.03067011 -1.07636976\n",

" -0.50577476]]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Split the data into training & testing\n",

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

],

"metadata": {

"id": "LnCceQdIGTkV"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"#Split the data into training & testing\n",

"x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=4,random\_state=4)\n",

"x\_train"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 424

},

"id": "p1jQH1SmGTwj",

"outputId": "afcbf65f-40e6-4a36-bb16-2a1ae67c9923"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" const EstimatedSalary\n",

"2558 1.0 137903.54\n",

"7642 1.0 121765.00\n",

"8912 1.0 109470.34\n",

"3319 1.0 2923.61\n",

"6852 1.0 7312.25\n",

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

"456 1.0 7666.73\n",

"6017 1.0 9085.00\n",

"709 1.0 147794.63\n",

"8366 1.0 102515.42\n",

"1146 1.0 54776.64\n",

"\n",

"[9996 rows x 2 columns]"

],

"text/html": [

"\n",

" <div id=\"df-ecfcf90e-4dee-4d01-9237-7681e63217b7\">\n",

" <div class=\"colab-df-container\">\n",

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>2558</th>\n",

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

" <td>137903.54</td>\n",

" </tr>\n",

" <tr>\n",

" <th>7642</th>\n",

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

" <td>121765.00</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8912</th>\n",

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

" <td>109470.34</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3319</th>\n",

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

" <td>2923.61</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6852</th>\n",

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

" <td>7312.25</td>\n",

" </tr>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" <tr>\n",

" <th>456</th>\n",

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

" <td>7666.73</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6017</th>\n",

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

" <td>9085.00</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>147794.63</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8366</th>\n",

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

" <td>102515.42</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1146</th>\n",

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

" <td>54776.64</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>9996 rows × 2 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ecfcf90e-4dee-4d01-9237-7681e63217b7')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-ecfcf90e-4dee-4d01-9237-7681e63217b7 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-ecfcf90e-4dee-4d01-9237-7681e63217b7');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 23

}

]

},

{

"cell\_type": "code",

"source": [

"#Split the data into training & testing\n",

"x\_test"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 175

},

"id": "2riy7kDDGv5q",

"outputId": "ca13cd66-bb1d-4a0b-9c14-5ec1de26b938"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" const EstimatedSalary\n",

"1603 1.0 23305.85\n",

"8713 1.0 41248.80\n",

"4561 1.0 143317.42\n",

"6600 1.0 174123.16"

],

"text/html": [

"\n",

" <div id=\"df-27c7c51c-8c5c-4537-8b83-f3df6679917c\">\n",

" <div class=\"colab-df-container\">\n",

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>1603</th>\n",

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

" <td>23305.85</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8713</th>\n",

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

" <td>41248.80</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4561</th>\n",

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

" <td>143317.42</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6600</th>\n",

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

" <td>174123.16</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-27c7c51c-8c5c-4537-8b83-f3df6679917c')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-27c7c51c-8c5c-4537-8b83-f3df6679917c button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-27c7c51c-8c5c-4537-8b83-f3df6679917c');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 24

}

]

},

{

"cell\_type": "code",

"source": [

"#Split the data into training & testing\n",

"y\_train"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "qFRH7IGvG0ng",

"outputId": "1348eedf-a3c7-4172-9a3c-5b1c4b6b1ea8"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"2558 727\n",

"7642 811\n",

"8912 623\n",

"3319 430\n",

"6852 600\n",

" ... \n",

"456 733\n",

"6017 487\n",

"709 686\n",

"8366 637\n",

"1146 614\n",

"Name: CreditScore, Length: 9996, dtype: int64"

]

},

"metadata": {},

"execution\_count": 25

}

]

},

{

"cell\_type": "code",

"source": [

"#Split the data into training & testing\n",

"y\_test"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "jrF5k\_-tG0rO",

"outputId": "ed396d7d-2a6e-4741-9105-53e7b978087e"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"1603 576\n",

"8713 786\n",

"4561 562\n",

"6600 505\n",

"Name: CreditScore, dtype: int64"

]

},

"metadata": {},

"execution\_count": 26

}

]

}

]

}