{

"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": 11,

"metadata": {

"id": "05e\_ePvU5yge"

},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns"

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*2. Load the dataset.\*\*\n"

],

"metadata": {

"id": "zhaJ4bI8BBRU"

}

},

{

"cell\_type": "code",

"source": [

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

"data.head()\n",

"\n"

],

"metadata": {

"colab": {

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

"height": 270

},

"id": "AMfBKspK6Tsg",

"outputId": "4f84f8ff-bf93-4efe-f993-eaf89aa487fd"

},

"execution\_count": 3,

"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-efaee10b-0c8e-4c64-9074-0474121377da\">\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-efaee10b-0c8e-4c64-9074-0474121377da')\"\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-efaee10b-0c8e-4c64-9074-0474121377da 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-efaee10b-0c8e-4c64-9074-0474121377da');\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": 3

}

]

},

{

"cell\_type": "markdown",

"source": [],

"metadata": {

"id": "DDmyAyvwAjzZ"

}

},

{

"cell\_type": "markdown",

"source": [

"# \*\*3. Perform Below Visualizations.\*\*\n",

"\n",

"# Univariate Analysis\n"

],

"metadata": {

"id": "ADMV0QwnBwHH"

}

},

{

"cell\_type": "code",

"source": [

"data.boxplot(column=['EstimatedSalary'], grid=False, color='blue')"

],

"metadata": {

"colab": {

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

"height": 283

},

"id": "V76Zh3sg8O0M",

"outputId": "a94dc43a-9864-4aa6-8a25-3efaa004a3d3"

},

"execution\_count": null,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fe0c7bbd950>"

]

},

"metadata": {},

"execution\_count": 13

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.hist(column='EstimatedSalary', grid=False, edgecolor='black')"

],

"metadata": {

"colab": {

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

"height": 318

},

"id": "dXy5mwlZ87wT",

"outputId": "6fd185b1-070c-4dea-85c6-2c9249cc3288"

},

"execution\_count": 12,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f4f02695090>]],\n",

" dtype=object)"

]

},

"metadata": {},

"execution\_count": 12

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.kdeplot(data['EstimatedSalary'])"

],

"metadata": {

"colab": {

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

"height": 308

},

"id": "tU4BNr\_J\_b32",

"outputId": "2d936935-7c54-4f35-cf28-6ebe9af9759a"

},

"execution\_count": 13,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4f0261cf90>"

]

},

"metadata": {},

"execution\_count": 13

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.hist(column='Balance', grid=False, edgecolor='black')"

],

"metadata": {

"colab": {

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

"height": 318

},

"id": "W6AsFEsD-q5Q",

"outputId": "4926f4f8-be9a-43e5-bd43-bb332c4bbfdb"

},

"execution\_count": 14,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f4f025ab890>]],\n",

" dtype=object)"

]

},

"metadata": {},

"execution\_count": 14

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"import seaborn as sns\n",

"\n",

"sns.kdeplot(data['Balance'])\n"

],

"metadata": {

"colab": {

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

"height": 308

},

"id": "3SCQihrU-1xv",

"outputId": "0f1dbd38-0db0-4809-857b-e8641a4573b2"

},

"execution\_count": 15,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4f02f08250>"

]

},

"metadata": {},

"execution\_count": 15

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"# Bi - Variate Analysis"

],

"metadata": {

"id": "TJQnw4lPBzCL"

}

},

{

"cell\_type": "code",

"source": [

"plt.scatter(data.CustomerId, data.Balance\t)\n",

"\n"

],

"metadata": {

"colab": {

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

"height": 296

},

"id": "GZDFY4tM\_OKs",

"outputId": "30fb7a6a-744b-4b2c-bd09-080af09d36fa"

},

"execution\_count": 16,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7f4f02ec8e50>"

]

},

"metadata": {},

"execution\_count": 16

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.corr()"

],

"metadata": {

"colab": {

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

"height": 458

},

"id": "2aYd8GN0ETMr",

"outputId": "a85e690d-cf39-4af5-c2ea-86698a1cfb3a"

},

"execution\_count": 17,

"outputs": [

{

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

"data": {

"text/plain": [

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

"RowNumber 1.000000 0.004202 0.005840 0.000783 -0.006495 \n",

"CustomerId 0.004202 1.000000 0.005308 0.009497 -0.014883 \n",

"CreditScore 0.005840 0.005308 1.000000 -0.003965 0.000842 \n",

"Age 0.000783 0.009497 -0.003965 1.000000 -0.009997 \n",

"Tenure -0.006495 -0.014883 0.000842 -0.009997 1.000000 \n",

"Balance -0.009067 -0.012419 0.006268 0.028308 -0.012254 \n",

"NumOfProducts 0.007246 0.016972 0.012238 -0.030680 0.013444 \n",

"HasCrCard 0.000599 -0.014025 -0.005458 -0.011721 0.022583 \n",

"IsActiveMember 0.012044 0.001665 0.025651 0.085472 -0.028362 \n",

"EstimatedSalary -0.005988 0.015271 -0.001384 -0.007201 0.007784 \n",

"Exited -0.016571 -0.006248 -0.027094 0.285323 -0.014001 \n",

"\n",

" Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"RowNumber -0.009067 0.007246 0.000599 0.012044 \n",

"CustomerId -0.012419 0.016972 -0.014025 0.001665 \n",

"CreditScore 0.006268 0.012238 -0.005458 0.025651 \n",

"Age 0.028308 -0.030680 -0.011721 0.085472 \n",

"Tenure -0.012254 0.013444 0.022583 -0.028362 \n",

"Balance 1.000000 -0.304180 -0.014858 -0.010084 \n",

"NumOfProducts -0.304180 1.000000 0.003183 0.009612 \n",

"HasCrCard -0.014858 0.003183 1.000000 -0.011866 \n",

"IsActiveMember -0.010084 0.009612 -0.011866 1.000000 \n",

"EstimatedSalary 0.012797 0.014204 -0.009933 -0.011421 \n",

"Exited 0.118533 -0.047820 -0.007138 -0.156128 \n",

"\n",

" EstimatedSalary Exited \n",

"RowNumber -0.005988 -0.016571 \n",

"CustomerId 0.015271 -0.006248 \n",

"CreditScore -0.001384 -0.027094 \n",

"Age -0.007201 0.285323 \n",

"Tenure 0.007784 -0.014001 \n",

"Balance 0.012797 0.118533 \n",

"NumOfProducts 0.014204 -0.047820 \n",

"HasCrCard -0.009933 -0.007138 \n",

"IsActiveMember -0.011421 -0.156128 \n",

"EstimatedSalary 1.000000 0.012097 \n",

"Exited 0.012097 1.000000 "

],

"text/html": [

"\n",

" <div id=\"df-56cf040e-8c9c-4a96-93d9-0db3320af29b\">\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>CreditScore</th>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

" <td>0.004202</td>\n",

" <td>0.005840</td>\n",

" <td>0.000783</td>\n",

" <td>-0.006495</td>\n",

" <td>-0.009067</td>\n",

" <td>0.007246</td>\n",

" <td>0.000599</td>\n",

" <td>0.012044</td>\n",

" <td>-0.005988</td>\n",

" <td>-0.016571</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.004202</td>\n",

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

" <td>0.005308</td>\n",

" <td>0.009497</td>\n",

" <td>-0.014883</td>\n",

" <td>-0.012419</td>\n",

" <td>0.016972</td>\n",

" <td>-0.014025</td>\n",

" <td>0.001665</td>\n",

" <td>0.015271</td>\n",

" <td>-0.006248</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.005840</td>\n",

" <td>0.005308</td>\n",

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

" <td>0.000783</td>\n",

" <td>0.009497</td>\n",

" <td>-0.003965</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>-0.006495</td>\n",

" <td>-0.014883</td>\n",

" <td>0.000842</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>-0.009067</td>\n",

" <td>-0.012419</td>\n",

" <td>0.006268</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>0.007246</td>\n",

" <td>0.016972</td>\n",

" <td>0.012238</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>0.000599</td>\n",

" <td>-0.014025</td>\n",

" <td>-0.005458</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>0.012044</td>\n",

" <td>0.001665</td>\n",

" <td>0.025651</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>-0.005988</td>\n",

" <td>0.015271</td>\n",

" <td>-0.001384</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>-0.016571</td>\n",

" <td>-0.006248</td>\n",

" <td>-0.027094</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-56cf040e-8c9c-4a96-93d9-0db3320af29b')\"\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-56cf040e-8c9c-4a96-93d9-0db3320af29b 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-56cf040e-8c9c-4a96-93d9-0db3320af29b');\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": 17

}

]

},

{

"cell\_type": "markdown",

"source": [

"# Multi - Variate Analysis"

],

"metadata": {

"id": "GFqHsZBwQu5H"

}

},

{

"cell\_type": "code",

"source": [

"\n",

"pd.plotting.scatter\_matrix(data.loc[:, \"CustomerId\":\"Balance\"], diagonal=\"kde\",figsize=(20,15))\n",

"plt.show()"

],

"metadata": {

"colab": {

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

"height": 846

},

"id": "lW6Co9PaQxcn",

"outputId": "0f074739-f57a-4605-fe6f-a0d50f8639b0"

},

"execution\_count": 19,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 1440x1080 with 25 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*4. Perform descriptive statistics on the dataset.\*\*"

],

"metadata": {

"id": "tL5nMRY\_HZ6Y"

}

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].mean()"

],

"metadata": {

"colab": {

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

},

"id": "Z5tiJf0J5KeA",

"outputId": "0ff9b4fc-3392-4abb-d47c-e7aef998cd46"

},

"execution\_count": 51,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 140.000000\n",

"Balance 76485.889288\n",

"EstimatedSalary 100090.239881\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 51

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].median()"

],

"metadata": {

"colab": {

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

},

"id": "U2zllhVG6kzU",

"outputId": "3ccd977a-a991-4250-f7c3-a77452163131"

},

"execution\_count": 52,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 140.000\n",

"Balance 97198.540\n",

"EstimatedSalary 100193.915\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 52

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].mode()"

],

"metadata": {

"colab": {

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

"height": 81

},

"id": "hKK0HdMU6nUT",

"outputId": "867d6401-1172-4a50-fd54-063834d23f8b"

},

"execution\_count": 53,

"outputs": [

{

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

"data": {

"text/plain": [

" CreditScore Balance EstimatedSalary\n",

"0 140 0.0 24924.92"

],

"text/html": [

"\n",

" <div id=\"df-393fb06d-0438-4e19-9bac-1a5b05bdf5fc\">\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>Balance</th>\n",

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>140</td>\n",

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-393fb06d-0438-4e19-9bac-1a5b05bdf5fc')\"\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-393fb06d-0438-4e19-9bac-1a5b05bdf5fc 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-393fb06d-0438-4e19-9bac-1a5b05bdf5fc');\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": 53

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].quantile()"

],

"metadata": {

"colab": {

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

},

"id": "gEooUGci6ri8",

"outputId": "ae639230-1265-4c77-d3fa-4ecb820b3f7c"

},

"execution\_count": 54,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 140.000\n",

"Balance 97198.540\n",

"EstimatedSalary 100193.915\n",

"Name: 0.5, dtype: float64"

]

},

"metadata": {},

"execution\_count": 54

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].std()"

],

"metadata": {

"colab": {

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

},

"id": "38Ad0sF46u\_j",

"outputId": "816f73c7-6857-4ba7-9b11-fe765fed4d09"

},

"execution\_count": 57,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 0.000000\n",

"Balance 62397.405202\n",

"EstimatedSalary 57510.492818\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 57

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].min()"

],

"metadata": {

"colab": {

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

},

"id": "eQtGs2dS7HF3",

"outputId": "7b110d18-0e4a-4757-e713-33cbae505d77"

},

"execution\_count": 58,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 140.00\n",

"Balance 0.00\n",

"EstimatedSalary 11.58\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 58

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].max()"

],

"metadata": {

"colab": {

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

},

"id": "KBafVzDD7L5c",

"outputId": "d0e8d70c-0de4-4440-abae-e9a261e47b54"

},

"execution\_count": 59,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 140.00\n",

"Balance 250898.09\n",

"EstimatedSalary 199992.48\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 59

}

]

},

{

"cell\_type": "code",

"source": [

"data[['CreditScore', 'Balance', 'EstimatedSalary']].skew()"

],

"metadata": {

"colab": {

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

},

"id": "1JX19ha57O2P",

"outputId": "d685d5cf-f2c1-41a8-d0c3-7cac5284975d"

},

"execution\_count": 60,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 0.000000\n",

"Balance -0.141109\n",

"EstimatedSalary 0.002085\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 60

}

]

},

{

"cell\_type": "code",

"source": [

"data.info()"

],

"metadata": {

"colab": {

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

},

"id": "YFZNNsU4HSdb",

"outputId": "d600a41d-b2e7-4f1f-c47d-bbb06bee9344"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"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"

]

}

]

},

{

"cell\_type": "code",

"source": [

"data.shape"

],

"metadata": {

"colab": {

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

},

"id": "SJ3aUQgRjR5a",

"outputId": "105d085b-fa2a-438e-df02-eca4e7a3d43f"

},

"execution\_count": 21,

"outputs": [

{

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

"data": {

"text/plain": [

"(10000, 14)"

]

},

"metadata": {},

"execution\_count": 21

}

]

},

{

"cell\_type": "code",

"source": [

"data.describe()\n"

],

"metadata": {

"colab": {

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

"height": 364

},

"id": "m4VaxsccLYWE",

"outputId": "9e39054d-25b8-49f5-9f36-27a9d4b689cc"

},

"execution\_count": null,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

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

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

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

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

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

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

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

"\n",

" Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"count 10000.000000 10000.000000 10000.00000 10000.000000 \n",

"mean 76485.889288 1.530200 0.70550 0.515100 \n",

"std 62397.405202 0.581654 0.45584 0.499797 \n",

"min 0.000000 1.000000 0.00000 0.000000 \n",

"25% 0.000000 1.000000 0.00000 0.000000 \n",

"50% 97198.540000 1.000000 1.00000 1.000000 \n",

"75% 127644.240000 2.000000 1.00000 1.000000 \n",

"max 250898.090000 4.000000 1.00000 1.000000 \n",

"\n",

" EstimatedSalary Exited \n",

"count 10000.000000 10000.000000 \n",

"mean 100090.239881 0.203700 \n",

"std 57510.492818 0.402769 \n",

"min 11.580000 0.000000 \n",

"25% 51002.110000 0.000000 \n",

"50% 100193.915000 0.000000 \n",

"75% 149388.247500 0.000000 \n",

"max 199992.480000 1.000000 "

],

"text/html": [

"\n",

" <div id=\"df-8997b36a-e0b3-4ff9-827c-7d7d0a8253c2\">\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>CreditScore</th>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-8997b36a-e0b3-4ff9-827c-7d7d0a8253c2')\"\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-8997b36a-e0b3-4ff9-827c-7d7d0a8253c2 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-8997b36a-e0b3-4ff9-827c-7d7d0a8253c2');\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": 32

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 5. Handle the Missing \*\*values\*\*.\n",

"There is no missing values\n"

],

"metadata": {

"id": "nwtjtbfvO1X2"

}

},

{

"cell\_type": "code",

"source": [

"data.isnull().sum()"

],

"metadata": {

"colab": {

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

},

"id": "VfuaWTaTM-L4",

"outputId": "7a323e6d-7a11-4acf-ce7b-27d8d8593783"

},

"execution\_count": 22,

"outputs": [

{

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

"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"

]

},

"metadata": {},

"execution\_count": 22

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*6. Find the outliers and replace the outliers\*\*"

],

"metadata": {

"id": "8CBpnl\_6POcw"

}

},

{

"cell\_type": "code",

"source": [

"data.describe()"

],

"metadata": {

"colab": {

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

"height": 364

},

"id": "H80lV2UV1Bbj",

"outputId": "cacdaa06-6052-40c6-87e9-b1f69306f49f"

},

"execution\_count": 28,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

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

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

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

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

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

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

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

"\n",

" Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"count 10000.000000 10000.000000 10000.00000 10000.000000 \n",

"mean 76485.889288 1.530200 0.70550 0.515100 \n",

"std 62397.405202 0.581654 0.45584 0.499797 \n",

"min 0.000000 1.000000 0.00000 0.000000 \n",

"25% 0.000000 1.000000 0.00000 0.000000 \n",

"50% 97198.540000 1.000000 1.00000 1.000000 \n",

"75% 127644.240000 2.000000 1.00000 1.000000 \n",

"max 250898.090000 4.000000 1.00000 1.000000 \n",

"\n",

" EstimatedSalary Exited \n",

"count 10000.000000 10000.000000 \n",

"mean 100090.239881 0.203700 \n",

"std 57510.492818 0.402769 \n",

"min 11.580000 0.000000 \n",

"25% 51002.110000 0.000000 \n",

"50% 100193.915000 0.000000 \n",

"75% 149388.247500 0.000000 \n",

"max 199992.480000 1.000000 "

],

"text/html": [

"\n",

" <div id=\"df-36750fab-1b57-46c1-9135-21073942e6e4\">\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>CreditScore</th>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-36750fab-1b57-46c1-9135-21073942e6e4')\"\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-36750fab-1b57-46c1-9135-21073942e6e4 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-36750fab-1b57-46c1-9135-21073942e6e4');\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": 28

}

]

},

{

"cell\_type": "code",

"source": [

"\n",

"numeric\_col = ['RowNumber\tCustomerId','CreditScore','\tAge',\t'Tenure',\t'Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','Exited']\n",

"categorical\_col = ['Surname', 'Geography', 'Gender']\n"

],

"metadata": {

"id": "DsbO\_zcKk\_vg"

},

"execution\_count": 26,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"print(data['CreditScore'].skew())\n",

"data['CreditScore'].describe()"

],

"metadata": {

"colab": {

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

},

"id": "FCZ1pFWh1R6i",

"outputId": "80f4cddf-90ea-461e-86a4-024f3365ae79"

},

"execution\_count": 29,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"-0.07160660820092675\n"

]

},

{

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

"data": {

"text/plain": [

"count 10000.000000\n",

"mean 650.528800\n",

"std 96.653299\n",

"min 350.000000\n",

"25% 584.000000\n",

"50% 652.000000\n",

"75% 718.000000\n",

"max 850.000000\n",

"Name: CreditScore, dtype: float64"

]

},

"metadata": {},

"execution\_count": 29

}

]

},

{

"cell\_type": "code",

"source": [

"Q1 = data.quantile(0.25)\n",

"Q3 = data.quantile(0.75)\n",

"IQR = Q3 - Q1\n",

"print(IQR)"

],

"metadata": {

"colab": {

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

},

"id": "8rAEKxoy1jUZ",

"outputId": "55745a79-1e3e-4c9e-c95e-eca23668fdfa"

},

"execution\_count": 31,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"RowNumber 4999.5000\n",

"CustomerId 124705.5000\n",

"CreditScore 134.0000\n",

"Age 12.0000\n",

"Tenure 4.0000\n",

"Balance 127644.2400\n",

"NumOfProducts 1.0000\n",

"HasCrCard 1.0000\n",

"IsActiveMember 1.0000\n",

"EstimatedSalary 98386.1375\n",

"Exited 0.0000\n",

"dtype: float64\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"plt.boxplot(data[\"CreditScore\"])\n",

"plt.show()"

],

"metadata": {

"colab": {

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

"height": 265

},

"id": "knVHwF9g18hl",

"outputId": "61b32e0c-b638-4337-ac10-d38164cd58a0"

},

"execution\_count": 36,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"print(data['CreditScore'].quantile(0.50)) \n",

"print(data['CreditScore'].quantile(0.95)) \n",

"data['CreditScore'] = np.where(data['CreditScore'] > 325, 140, data['CreditScore'])\n",

"data.describe()"

],

"metadata": {

"colab": {

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

"height": 401

},

"id": "wyF7uGu52WL8",

"outputId": "e0aeefb1-c5e4-4de4-9af7-d821cabe8a9c"

},

"execution\_count": 37,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"652.0\n",

"812.0\n"

]

},

{

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

"data": {

"text/plain": [

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

"count 10000.00000 1.000000e+04 10000.0 10000.000000 10000.000000 \n",

"mean 5000.50000 1.569094e+07 140.0 38.921800 5.012800 \n",

"std 2886.89568 7.193619e+04 0.0 10.487806 2.892174 \n",

"min 1.00000 1.556570e+07 140.0 18.000000 0.000000 \n",

"25% 2500.75000 1.562853e+07 140.0 32.000000 3.000000 \n",

"50% 5000.50000 1.569074e+07 140.0 37.000000 5.000000 \n",

"75% 7500.25000 1.575323e+07 140.0 44.000000 7.000000 \n",

"max 10000.00000 1.581569e+07 140.0 92.000000 10.000000 \n",

"\n",

" Balance NumOfProducts HasCrCard IsActiveMember \\\n",

"count 10000.000000 10000.000000 10000.00000 10000.000000 \n",

"mean 76485.889288 1.530200 0.70550 0.515100 \n",

"std 62397.405202 0.581654 0.45584 0.499797 \n",

"min 0.000000 1.000000 0.00000 0.000000 \n",

"25% 0.000000 1.000000 0.00000 0.000000 \n",

"50% 97198.540000 1.000000 1.00000 1.000000 \n",

"75% 127644.240000 2.000000 1.00000 1.000000 \n",

"max 250898.090000 4.000000 1.00000 1.000000 \n",

"\n",

" EstimatedSalary Exited \n",

"count 10000.000000 10000.000000 \n",

"mean 100090.239881 0.203700 \n",

"std 57510.492818 0.402769 \n",

"min 11.580000 0.000000 \n",

"25% 51002.110000 0.000000 \n",

"50% 100193.915000 0.000000 \n",

"75% 149388.247500 0.000000 \n",

"max 199992.480000 1.000000 "

],

"text/html": [

"\n",

" <div id=\"df-59bf9477-41e3-40cc-b187-baa63f9c3066\">\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>CreditScore</th>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

" <td>10000.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" <td>140.0</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-59bf9477-41e3-40cc-b187-baa63f9c3066')\"\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-59bf9477-41e3-40cc-b187-baa63f9c3066 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-59bf9477-41e3-40cc-b187-baa63f9c3066');\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": 37

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*7. Check for Categorical columns and perform encoding\*\*"

],

"metadata": {

"id": "FcWJ01AAlAMI"

}

},

{

"cell\_type": "code",

"source": [

"X = data.iloc[:, 3:13].values\n",

"y = data.iloc[:, 13].values\n"

],

"metadata": {

"id": "M\_Ko7hqFuh\_8"

},

"execution\_count": 38,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

"from sklearn.compose import ColumnTransformer\n",

"\n",

"labelencoder\_X\_1 = LabelEncoder()\n",

"X[:, 1] = labelencoder\_X\_1.fit\_transform(X[:, 1])\n",

"labelencoder\_X\_2 = LabelEncoder()\n",

"X[:, 2] = labelencoder\_X\_2.fit\_transform(X[:, 2])\n",

"\n",

"# remove categorical\_features, it works 100% perfectly\n",

"onehotencoder = OneHotEncoder()\n",

"X = onehotencoder.fit\_transform(X).toarray()\n",

"X = X[:, 1:]"

],

"metadata": {

"id": "1mhcxkCglCFe"

},

"execution\_count": 39,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "N1HNQU\_Nk6\_6"

}

},

{

"cell\_type": "code",

"source": [

"X= data.iloc[:,3:-1]\n",

"y=data.iloc[:,-1]\n",

"X.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "8bZt5hCizRwG",

"outputId": "70021e7f-9ddf-41ac-b891-d5374dffaa8a"

},

"execution\_count": 40,

"outputs": [

{

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

"data": {

"text/plain": [

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

"0 140 France Female 42 2 0.00 1 \n",

"1 140 Spain Female 41 1 83807.86 1 \n",

"2 140 France Female 42 8 159660.80 3 \n",

"3 140 France Female 39 1 0.00 2 \n",

"4 140 Spain Female 43 2 125510.82 1 \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary \n",

"0 1 1 101348.88 \n",

"1 0 1 112542.58 \n",

"2 1 0 113931.57 \n",

"3 0 0 93826.63 \n",

"4 1 1 79084.10 "

],

"text/html": [

"\n",

" <div id=\"df-1a4e1921-ea6d-42c1-a259-c36fe83c9b5e\">\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",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>140</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",

" </tr>\n",

" <tr>\n",

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

" <td>140</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",

" </tr>\n",

" <tr>\n",

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

" <td>140</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",

" </tr>\n",

" <tr>\n",

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

" <td>140</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",

" </tr>\n",

" <tr>\n",

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

" <td>140</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",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-1a4e1921-ea6d-42c1-a259-c36fe83c9b5e')\"\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-1a4e1921-ea6d-42c1-a259-c36fe83c9b5e 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-1a4e1921-ea6d-42c1-a259-c36fe83c9b5e');\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": 40

}

]

},

{

"cell\_type": "code",

"source": [

"X = data.iloc[:, 3:13].values\n",

"y = data.iloc[:, 13].values"

],

"metadata": {

"id": "LBUGFhUHNOcU"

},

"execution\_count": 41,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"print(X)"

],

"metadata": {

"colab": {

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

},

"id": "28gvAhniohBZ",

"outputId": "025b9023-fb9f-437e-f330-85911746f761"

},

"execution\_count": 42,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[[140 'France' 'Female' ... 1 1 101348.88]\n",

" [140 'Spain' 'Female' ... 0 1 112542.58]\n",

" [140 'France' 'Female' ... 1 0 113931.57]\n",

" ...\n",

" [140 'France' 'Female' ... 0 1 42085.58]\n",

" [140 'Germany' 'Male' ... 1 0 92888.52]\n",

" [140 'France' 'Female' ... 1 0 38190.78]]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"print(y)"

],

"metadata": {

"colab": {

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

},

"id": "ducksijS3xA0",

"outputId": "4e38330e-186c-422b-cbea-88e21732738f"

},

"execution\_count": 43,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "code",

"source": [

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

"x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.25, random\_state = 0)\n",

"\n",

"print(x\_train.shape)\n",

"print(y\_train.shape)\n",

"print(x\_test.shape)\n",

"print(y\_test.shape)"

],

"metadata": {

"colab": {

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

},

"id": "1Yg-yCn33kpS",

"outputId": "c863e3ed-32ef-4ab9-bb9f-b96a277241ed"

},

"execution\_count": 49,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(7500, 2)\n",

"(7500,)\n",

"(2500, 2)\n",

"(2500,)\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*9. Scale the independant variables\*\*"

],

"metadata": {

"id": "wf2WrqKe33bS"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.preprocessing import StandardScaler\n",

"\n",

"sc = StandardScaler()\n",

"x\_train = sc.fit\_transform(x\_train)\n",

"x\_test = sc.fit\_transform(x\_test)\n",

"\n",

"x\_train = pd.DataFrame(x\_train)\n",

"x\_train.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "JFGeHxBf3plD",

"outputId": "62599ea4-c3a3-48af-d2ad-d24e48591664"

},

"execution\_count": 50,

"outputs": [

{

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

"data": {

"text/plain": [

" 0 1\n",

"0 0.0 -1.343330\n",

"1 0.0 1.558330\n",

"2 0.0 -0.655156\n",

"3 0.0 1.200594\n",

"4 0.0 0.778798"

],

"text/html": [

"\n",

" <div id=\"df-ba9db867-9c96-45c8-9434-4f307159a20f\">\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>0</th>\n",

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

" <td>-1.343330</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>1.558330</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>-0.655156</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>1.200594</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>0.778798</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ba9db867-9c96-45c8-9434-4f307159a20f')\"\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-ba9db867-9c96-45c8-9434-4f307159a20f 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-ba9db867-9c96-45c8-9434-4f307159a20f');\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": 50

}

]

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "inM-EIsZ7xml"

}

},

{

"cell\_type": "code",

"source": [

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

"x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.25, random\_state = 0)\n",

"\n",

"print(x\_train.shape)\n",

"print(y\_train.shape)\n",

"print(x\_test.shape)\n",

"print(y\_test.shape)"

],

"metadata": {

"colab": {

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

},

"id": "03KONW3O71Xx",

"outputId": "fcc8dcf3-4d32-400d-f8ad-c89fcfdaf096"

},

"execution\_count": 63,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(7500, 2)\n",

"(7500,)\n",

"(2500, 2)\n",

"(2500,)\n"

]

}

]

}

]

}