{

"cells": [

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {

"id": "yyWytQguV73l"

},

"outputs": [],

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zrySH7iAXrS1"

},

"source": [

"# \*\*2. LOADING DATASET\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {

"id": "Iq1PAZtVXDk9"

},

"outputs": [],

"source": [

"df=pd.read\_csv(r'/content/Churn\_Modelling.csv')"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"colab": {

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

"height": 488

},

"id": "9wusfu5DXY\_r",

"outputId": "9e3952fb-fbd8-43b1-fd62-8ce2c57d5df3"

},

"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",

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

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

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

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

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

"\n",

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

"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",

"9995 5 0.00 2 1 0 \n",

"9996 10 57369.61 1 1 1 \n",

"9997 7 0.00 1 0 1 \n",

"9998 3 75075.31 2 1 0 \n",

"9999 4 130142.79 1 1 0 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 \n",

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

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 \n",

"\n",

"[10000 rows x 14 columns]"

],

"text/html": [

"\n",

" <div id=\"df-47932970-6c1b-4c13-9207-2f8637b57517\">\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",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>4</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 × 14 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-47932970-6c1b-4c13-9207-2f8637b57517')\"\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-47932970-6c1b-4c13-9207-2f8637b57517 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-47932970-6c1b-4c13-9207-2f8637b57517');\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": 5

}

],

"source": [

"df"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "NouPJbj\_DUjv"

},

"source": [

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

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "phDE6UDDp7LH"

},

"source": [

"\*\*UNIVARIATE ANALYSIS\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {

"id": "GcNRMCFkqY1o"

},

"outputs": [],

"source": [

"df\_france=df.loc[df['Geography']=='France']\n",

"df\_spain=df.loc[df['Geography']=='Spain']\n",

"df\_germany=df.loc[df['Geography']=='Germany']"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {

"colab": {

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

"height": 279

},

"id": "ozQceRnLqjh1",

"outputId": "356a204c-a52a-455c-923d-7394d2794554"

},

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"plt.plot(df\_france['Balance'],np.zeros\_like(df\_france['Balance']),'o')\n",

"plt.plot(df\_spain['Balance'],np.zeros\_like(df\_spain['Balance']),'o')\n",

"plt.plot(df\_germany['Balance'],np.zeros\_like(df\_germany['Balance']),'o')\n",

"plt.xlabel('Age')\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "XdoDWXsdqrbx"

},

"source": [

"\n",

"\*\*BIVARIATE ANALYSIS\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 260

},

"id": "-UZ4QZCyqpAC",

"outputId": "c1603254-035f-4db0-92af-cc4730a7a2f4"

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning: The `size` parameter has been renamed to `height`; please update your code.\n",

" warnings.warn(msg, UserWarning)\n"

]

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 297.875x216 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.FacetGrid(df,hue=\"Geography\",size=3).map(plt.scatter,\"CreditScore\",\"EstimatedSalary\").add\_legend();\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "M27WzRTgrRRX"

},

"source": [

"\*\*MULTIVARIATE ANALYSIS\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"colab": {

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

"height": 1000

},

"id": "MtmBkgMgrJQ8",

"outputId": "8d999e44-c3e5-4b76-8280-fc795b186bc7"

},

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `height`; please update your code.\n",

" warnings.warn(msg, UserWarning)\n"

]

},

{

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

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x7f9028adefd0>"

]

},

"metadata": {},

"execution\_count": 7

},

{

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

"data": {

"text/plain": [

"<Figure size 2233.75x2160 with 110 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

],

"source": [

"sns.pairplot(df,hue=\"Balance\",size=3)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "ejSkD0E5sslK"

},

"source": [

"# \*\*Descriptive Statistics\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "GeFPbccRs3MD",

"outputId": "e0b3a6e0-f180-4578-9e34-33f79767ee55"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-127cd7ff-da55-46bc-99c7-4cc57d523042\">\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-127cd7ff-da55-46bc-99c7-4cc57d523042')\"\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-127cd7ff-da55-46bc-99c7-4cc57d523042 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-127cd7ff-da55-46bc-99c7-4cc57d523042');\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",

" "

],

"text/plain": [

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

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

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

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

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

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

"\n",

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

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"2 8 159660.80 3 1 0 \n",

"3 1 0.00 2 0 0 \n",

"4 2 125510.82 1 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 "

]

},

"execution\_count": 10,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.head()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "VvDpUaPdtJUz"

},

"source": [

"Mean for each column"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "jJw3OYEis9xG",

"outputId": "c04d0cdd-1051-46b5-88c9-ddb80d82e02a"

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:1: 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",

" \"\"\"Entry point for launching an IPython kernel.\n"

]

},

{

"data": {

"text/plain": [

"RowNumber 5.000500e+03\n",

"CustomerId 1.569094e+07\n",

"CreditScore 6.505288e+02\n",

"Age 3.892180e+01\n",

"Tenure 5.012800e+00\n",

"Balance 7.648589e+04\n",

"NumOfProducts 1.530200e+00\n",

"HasCrCard 7.055000e-01\n",

"IsActiveMember 5.151000e-01\n",

"EstimatedSalary 1.000902e+05\n",

"Exited 2.037000e-01\n",

"dtype: float64"

]

},

"execution\_count": 11,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.mean() "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "shjHQk0wtOKq"

},

"source": [

"Mean for each row"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "\_DOs9i67tCVG",

"outputId": "41c9e282-2b64-4430-a151-c3f47157fa47"

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:1: 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",

" \"\"\"Entry point for launching an IPython kernel.\n"

]

},

{

"data": {

"text/plain": [

"0 1.430602e+06\n",

"1 1.440392e+06\n",

"2 1.444860e+06\n",

"3 1.435993e+06\n",

"4 1.449399e+06\n",

" ... \n",

"9995 1.428483e+06\n",

"9996 1.430866e+06\n",

"9997 1.421579e+06\n",

"9998 1.441922e+06\n",

"9999 1.437044e+06\n",

"Length: 10000, dtype: float64"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.mean(axis=1) "

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "3xvgD6sItaux"

},

"source": [

"Median of each column"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "vkTq-\_yRtbJ8",

"outputId": "56b801b9-7b73-444b-8056-4dd187035fa0"

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:1: 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",

" \"\"\"Entry point for launching an IPython kernel.\n"

]

},

{

"data": {

"text/plain": [

"RowNumber 5.000500e+03\n",

"CustomerId 1.569074e+07\n",

"CreditScore 6.520000e+02\n",

"Age 3.700000e+01\n",

"Tenure 5.000000e+00\n",

"Balance 9.719854e+04\n",

"NumOfProducts 1.000000e+00\n",

"HasCrCard 1.000000e+00\n",

"IsActiveMember 1.000000e+00\n",

"EstimatedSalary 1.001939e+05\n",

"Exited 0.000000e+00\n",

"dtype: float64"

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.median()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zqI6L\_y0tr1L"

},

"source": [

"Mode of each row"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 424

},

"id": "a8CMnSL1thmT",

"outputId": "c2d7ea76-6904-4672-d1d0-8c8ca8fb6ff2"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-024688c1-3998-4e0a-bea6-b5900d84c87d\">\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>15565701</td>\n",

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 14 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-024688c1-3998-4e0a-bea6-b5900d84c87d')\"\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-024688c1-3998-4e0a-bea6-b5900d84c87d 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-024688c1-3998-4e0a-bea6-b5900d84c87d');\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",

" "

],

"text/plain": [

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

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

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

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

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

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

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

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

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

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

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

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

"\n",

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

"0 2.0 0.0 1.0 1.0 1.0 \n",

"1 NaN NaN NaN NaN NaN \n",

"2 NaN NaN NaN NaN NaN \n",

"3 NaN NaN NaN NaN NaN \n",

"4 NaN NaN NaN NaN NaN \n",

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

"9995 NaN NaN NaN NaN NaN \n",

"9996 NaN NaN NaN NaN NaN \n",

"9997 NaN NaN NaN NaN NaN \n",

"9998 NaN NaN NaN NaN NaN \n",

"9999 NaN NaN NaN NaN NaN \n",

"\n",

" EstimatedSalary Exited \n",

"0 24924.92 0.0 \n",

"1 NaN NaN \n",

"2 NaN NaN \n",

"3 NaN NaN \n",

"4 NaN NaN \n",

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

"9995 NaN NaN \n",

"9996 NaN NaN \n",

"9997 NaN NaN \n",

"9998 NaN NaN \n",

"9999 NaN NaN \n",

"\n",

"[10000 rows x 14 columns]"

]

},

"execution\_count": 14,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.mode()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "SgyuOhVwtydc"

},

"source": [

"# \*\*Measures of Spread\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "-jgi8tSRtx7r",

"outputId": "c57a4de6-6433-49aa-8674-927578888e6f"

},

"outputs": [

{

"data": {

"text/plain": [

"199980.90000000002"

]

},

"execution\_count": 15,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"max(df[\"EstimatedSalary\"]) - min(df[\"EstimatedSalary\"])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "evUOt1vwuKNg",

"outputId": "b880864b-998d-4b95-9f52-0ae946332fb6"

},

"outputs": [

{

"data": {

"text/plain": [

"[18.0, 32.0, 37.0, 44.0, 92.0]"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"five\_num = [df[\"Age\"].quantile(0), \n",

" df[\"Age\"].quantile(0.25),\n",

" df[\"Age\"].quantile(0.50),\n",

" df[\"Age\"].quantile(0.75),\n",

" df[\"Age\"].quantile(1)]\n",

"\n",

"five\_num"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "rqhWwAbMuOdt",

"outputId": "589f6bd3-23be-4cc4-efbf-054787fcf0a7"

},

"outputs": [

{

"data": {

"text/plain": [

"count 10000.000000\n",

"mean 76485.889288\n",

"std 62397.405202\n",

"min 0.000000\n",

"25% 0.000000\n",

"50% 97198.540000\n",

"75% 127644.240000\n",

"max 250898.090000\n",

"Name: Balance, dtype: float64"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[\"Balance\"].describe()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 483

},

"id": "gfFX2mAluc\_0",

"outputId": "f5598ee0-f501-494f-d07c-834d6e8eda0d"

},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 576x576 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"df.boxplot(column=\"Balance\",\n",

" return\_type='axes',\n",

" figsize=(8,8))\n",

"\n",

"plt.text(x=0.74, y=22.25, s=\"3rd Quartile\")\n",

"plt.text(x=0.8, y=18.75, s=\"Median\")\n",

"plt.text(x=0.75, y=15.5, s=\"1st Quartile\")\n",

"plt.text(x=0.9, y=10, s=\"Min\")\n",

"plt.text(x=0.9, y=33.5, s=\"Max\")\n",

"plt.text(x=0.7, y=19.5, s=\"IQR\", rotation=90, size=25);"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "WnBVuDFEuqEH",

"outputId": "2521fd3d-fcee-4f4e-cc99-73c4cc38e529"

},

"outputs": [

{

"data": {

"text/plain": [

"3893436175.990742"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[\"Balance\"].var()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "wWNY-ZnyuwGX",

"outputId": "e6eb44f0-aaeb-436b-b289-08d3279bf791"

},

"outputs": [

{

"data": {

"text/plain": [

"62397.405202385955"

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[\"Balance\"].std()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "L40GOQ5Nuz6v",

"outputId": "142a12f2-61b1-45a7-cd4b-85c078d8f3c2"

},

"outputs": [

{

"data": {

"text/plain": [

"69336.44285399996"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"abs\_median\_devs = abs(df[\"Balance\"] - df[\"Balance\"].median())\n",

"\n",

"abs\_median\_devs.median() \* 1.4826"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "WDajiYC9vA9e"

},

"source": [

"# \*\*Skewness and Kurtosis\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "4hX0w4hYu7Fh",

"outputId": "06a47bc4-aa13-4699-c219-0630aee1296d"

},

"outputs": [

{

"data": {

"text/plain": [

"-0.14110871094154384"

]

},

"execution\_count": 25,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[\"Balance\"].skew()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "exvqd0lwvM1T",

"outputId": "405ed16d-ce02-470c-c264-48f275a6cb36"

},

"outputs": [

{

"data": {

"text/plain": [

"-1.489411767941925"

]

},

"execution\_count": 27,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df[\"Balance\"].kurt() "

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "i8YyYW-SvyY7"

},

"outputs": [],

"source": [

"norm\_data = np.random.normal(size=100000)\n",

"skewed\_data = np.concatenate((np.random.normal(size=35000)+2, \n",

" np.random.exponential(size=65000)), \n",

" axis=0)\n",

"uniform\_data = np.random.uniform(0,2, size=100000)\n",

"peaked\_data = np.concatenate((np.random.exponential(size=50000),\n",

" np.random.exponential(size=50000)\*(-1)),\n",

" axis=0)\n",

"\n",

"data\_df = pd.DataFrame({\"norm\":norm\_data,\n",

" \"skewed\":skewed\_data,\n",

" \"uniform\":uniform\_data,\n",

" \"peaked\":peaked\_data})"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 592

},

"id": "GJ4fYlTNv5Tu",

"outputId": "3c5bbbb7-1a75-44a5-f07d-9ebe35fc9f65"

},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x720 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"data\_df.plot(kind=\"density\",\n",

" figsize=(10,10),\n",

" xlim=(-5,5));"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "c\_cfAOuYwD00",

"outputId": "a5dd00df-1a84-4f81-bf49-b420d1e2b8f8"

},

"outputs": [

{

"data": {

"text/plain": [

"norm -0.001353\n",

"skewed 0.998384\n",

"uniform 0.002243\n",

"peaked 0.031832\n",

"dtype: float64"

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data\_df.skew()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "\_B48hQMgwHFH",

"outputId": "e5d2133f-60dc-4775-8231-8f3ea04887f5"

},

"outputs": [

{

"data": {

"text/plain": [

"norm -0.012000\n",

"skewed 1.225030\n",

"uniform -1.198906\n",

"peaked 2.938847\n",

"dtype: float64"

]

},

"execution\_count": 31,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data\_df.kurt()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "CYrlomlfwV2V"

},

"source": [

"# \*\*Handle the Missing values\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 424

},

"id": "psz1SffFwUrA",

"outputId": "8984381a-55a9-4330-ca81-4991fc2ff35f"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-ae964bf9-667a-42ea-8cb5-e67900ba6dff\">\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",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>4</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 × 14 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ae964bf9-667a-42ea-8cb5-e67900ba6dff')\"\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-ae964bf9-667a-42ea-8cb5-e67900ba6dff 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-ae964bf9-667a-42ea-8cb5-e67900ba6dff');\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",

" "

],

"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",

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

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

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

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

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

"\n",

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

"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",

"9995 5 0.00 2 1 0 \n",

"9996 10 57369.61 1 1 1 \n",

"9997 7 0.00 1 0 1 \n",

"9998 3 75075.31 2 1 0 \n",

"9999 4 130142.79 1 1 0 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 \n",

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

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 \n",

"\n",

"[10000 rows x 14 columns]"

]

},

"execution\_count": 32,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 424

},

"id": "-NcYMcGBwdyx",

"outputId": "f3d95a5e-d152-4e13-9e41-74e43cbe94c2"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-6d479198-d76a-4ac5-bf57-a734f6cb218c\">\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>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",

" <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",

" <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",

" <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",

" <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",

" <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",

" <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",

" <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",

" <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",

" <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",

" <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",

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>10000 rows × 14 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-6d479198-d76a-4ac5-bf57-a734f6cb218c')\"\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-6d479198-d76a-4ac5-bf57-a734f6cb218c 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-6d479198-d76a-4ac5-bf57-a734f6cb218c');\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",

" "

],

"text/plain": [

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

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

"0 False False False False False \n",

"1 False False False False False \n",

"2 False False False False False \n",

"3 False False False False False \n",

"4 False False False False False \n",

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

"9995 False False False False False \n",

"9996 False False False False False \n",

"9997 False False False False False \n",

"9998 False False False False False \n",

"9999 False False False False False \n",

"\n",

" EstimatedSalary Exited \n",

"0 False False \n",

"1 False False \n",

"2 False False \n",

"3 False False \n",

"4 False False \n",

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

"9995 False False \n",

"9996 False False \n",

"9997 False False \n",

"9998 False False \n",

"9999 False False \n",

"\n",

"[10000 rows x 14 columns]"

]

},

"execution\_count": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.isnull()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 296

},

"id": "ZMRbxmTxwgDS",

"outputId": "101f28e2-ceed-4b63-9cbf-9bf456982b7e"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc711380310>"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.distplot(df['Balance'].dropna(),kde=False,color='darkred',bins=40)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 282

},

"id": "BJD4cSxdwuEn",

"outputId": "021acc96-67d6-47f1-bbc1-d0b84a126e8c"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc7112e7a10>"

]

},

"execution\_count": 36,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"df['Balance'].hist(bins=30,color='darkred',alpha=0.3)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 296

},

"id": "Lj3fd27lw6Pb",

"outputId": "e6717cf6-4c34-4cbd-ec85-47902a2ae7e9"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc7111baad0>"

]

},

"execution\_count": 37,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.countplot(x='NumOfProducts',data=df)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 17

},

"id": "KKINq472w-by",

"outputId": "595e082e-b596-4cd1-d7a7-ba47c0832b2f"

},

"outputs": [

{

"data": {

"text/html": [

" <script type=\"text/javascript\">\n",

" window.PlotlyConfig = {MathJaxConfig: 'local'};\n",

" if (window.MathJax) {MathJax.Hub.Config({SVG: {font: \"STIX-Web\"}});}\n",

" if (typeof require !== 'undefined') {\n",

" require.undef(\"plotly\");\n",

" requirejs.config({\n",

" paths: {\n",

" 'plotly': ['https://cdn.plot.ly/plotly-2.8.3.min']\n",

" }\n",

" });\n",

" require(['plotly'], function(Plotly) {\n",

" window.\_Plotly = Plotly;\n",

" });\n",

" }\n",

" </script>\n",

" "

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"import cufflinks as cf\n",

"cf.go\_offline()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 459

},

"id": "iS4M5\_QexUMf",

"outputId": "82f4da04-1f54-4e03-fa00-6e89cc2b4211"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc70ae99950>"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x504 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(12, 7))\n",

"sns.boxplot(x='Gender',y='Age',data=df,palette='winter')"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "mC\_S95pRxdr4"

},

"outputs": [],

"source": [

"def impute\_age(cols):\n",

" Age = cols[0]\n",

" Pclass = cols[1]\n",

" \n",

" if pd.isnull(Age):\n",

"\n",

" if Pclass == 1:\n",

" return 37\n",

"\n",

" elif Pclass == 2:\n",

" return 29\n",

"\n",

" else:\n",

" return 24\n",

"\n",

" else:\n",

" return Age"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "FjF7pTJ5xhVQ"

},

"outputs": [],

"source": [

"df.drop('Gender',axis=1,inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "KFuPY84pxnTL",

"outputId": "3db1ebe6-bfb8-4adb-eb00-12a5b849e6ba"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-1cd0033f-60e2-4766-8e87-56b4e2143500\">\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>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>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>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>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>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>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-1cd0033f-60e2-4766-8e87-56b4e2143500')\"\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-1cd0033f-60e2-4766-8e87-56b4e2143500 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-1cd0033f-60e2-4766-8e87-56b4e2143500');\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",

" "

],

"text/plain": [

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

"0 1 15634602 Hargrave 619 France 42 2 \n",

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

"2 3 15619304 Onio 502 France 42 8 \n",

"3 4 15701354 Boni 699 France 39 1 \n",

"4 5 15737888 Mitchell 850 Spain 43 2 \n",

"\n",

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

"0 0.00 1 1 1 101348.88 \n",

"1 83807.86 1 0 1 112542.58 \n",

"2 159660.80 3 1 0 113931.57 \n",

"3 0.00 2 0 0 93826.63 \n",

"4 125510.82 1 1 1 79084.10 \n",

"\n",

" Exited \n",

"0 1 \n",

"1 0 \n",

"2 1 \n",

"3 0 \n",

"4 0 "

]

},

"execution\_count": 42,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.head()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "vgpuveDJxq9Q"

},

"source": [

" Converting Categorical Features"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "V8hKGjjyxyj7",

"outputId": "15a03387-600f-429a-a903-45ad241ce23e"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

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

"Data columns (total 13 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 Age 10000 non-null int64 \n",

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

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

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

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

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

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

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

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

"memory usage: 1015.8+ KB\n"

]

}

],

"source": [

"df.info()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "Sl5taNaFx1lV",

"outputId": "a4bbb4a5-19b9-4cb5-c148-e9c79c36bca8"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-c2570e5f-ba7f-4635-99e3-544ef00702e2\">\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>Germany</th>\n",

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-c2570e5f-ba7f-4635-99e3-544ef00702e2')\"\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-c2570e5f-ba7f-4635-99e3-544ef00702e2 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-c2570e5f-ba7f-4635-99e3-544ef00702e2');\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",

" "

],

"text/plain": [

" Germany Spain\n",

"0 0 0\n",

"1 0 1\n",

"2 0 0\n",

"3 0 0\n",

"4 0 1"

]

},

"execution\_count": 44,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"pd.get\_dummies(df['Geography'],drop\_first=True).head()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "II0BV7MXx8c6",

"outputId": "96fd8371-da23-496b-a806-d136cd30d047"

},

"outputs": [

{

"data": {

"text/plain": [

"<bound method DataFrame.info of RowNumber CustomerId Surname CreditScore Geography Age Tenure \\\n",

"0 1 15634602 Hargrave 619 France 42 2 \n",

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

"2 3 15619304 Onio 502 France 42 8 \n",

"3 4 15701354 Boni 699 France 39 1 \n",

"4 5 15737888 Mitchell 850 Spain 43 2 \n",

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

"9995 9996 15606229 Obijiaku 771 France 39 5 \n",

"9996 9997 15569892 Johnstone 516 France 35 10 \n",

"9997 9998 15584532 Liu 709 France 36 7 \n",

"9998 9999 15682355 Sabbatini 772 Germany 42 3 \n",

"9999 10000 15628319 Walker 792 France 28 4 \n",

"\n",

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

"0 0.00 1 1 1 101348.88 \n",

"1 83807.86 1 0 1 112542.58 \n",

"2 159660.80 3 1 0 113931.57 \n",

"3 0.00 2 0 0 93826.63 \n",

"4 125510.82 1 1 1 79084.10 \n",

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

"9995 0.00 2 1 0 96270.64 \n",

"9996 57369.61 1 1 1 101699.77 \n",

"9997 0.00 1 0 1 42085.58 \n",

"9998 75075.31 2 1 0 92888.52 \n",

"9999 130142.79 1 1 0 38190.78 \n",

"\n",

" Exited \n",

"0 1 \n",

"1 0 \n",

"2 1 \n",

"3 0 \n",

"4 0 \n",

"... ... \n",

"9995 0 \n",

"9996 0 \n",

"9997 1 \n",

"9998 1 \n",

"9999 0 \n",

"\n",

"[10000 rows x 13 columns]>"

]

},

"execution\_count": 45,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.info"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "q2JGFP11x\_9x"

},

"outputs": [],

"source": [

"sex = pd.get\_dummies(df['Age'],drop\_first=True)\n",

"embark = pd.get\_dummies(df['Balance'],drop\_first=True)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "2-HTB2-0yE1Q"

},

"outputs": [],

"source": [

"df.drop(['Age','HasCrCard','Surname','CustomerId'],axis=1,inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "J4kJK1EgyGjz",

"outputId": "7d7e0ca6-c069-4036-9c04-348edf2b9c30"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-e5fc5bd3-4a66-40f3-8362-4ddde0fd0c78\">\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>CreditScore</th>\n",

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

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

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

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

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

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

" <td>0.00</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>608</td>\n",

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

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

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

" <td>1</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>502</td>\n",

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

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

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

" <td>3</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>699</td>\n",

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

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

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

" <td>2</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>850</td>\n",

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

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

" <td>125510.82</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-e5fc5bd3-4a66-40f3-8362-4ddde0fd0c78')\"\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-e5fc5bd3-4a66-40f3-8362-4ddde0fd0c78 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-e5fc5bd3-4a66-40f3-8362-4ddde0fd0c78');\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",

" "

],

"text/plain": [

" RowNumber CreditScore Geography Tenure Balance NumOfProducts \\\n",

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

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

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

"3 4 699 France 1 0.00 2 \n",

"4 5 850 Spain 2 125510.82 1 \n",

"\n",

" IsActiveMember EstimatedSalary Exited \n",

"0 1 101348.88 1 \n",

"1 1 112542.58 0 \n",

"2 0 113931.57 1 \n",

"3 0 93826.63 0 \n",

"4 1 79084.10 0 "

]

},

"execution\_count": 48,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.head()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "FHv4Q636ykqu"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "paJWgubSy1am"

},

"outputs": [],

"source": [

"dataset1= [11,10,12,14,12,15,14,13,15,102,12,14,17,19,107, 10,13,12,14,12,108,12,11,14,13,15,10,15,12,10,14,13,15,10]"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "feV1JTaHy4jG"

},

"source": [

"Using Z score"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "ro\_e3dI\_yJHI"

},

"outputs": [],

"source": [

"outliers=[]\n",

"def detect\_outliers(data):\n",

" \n",

" threshold=3\n",

" mean = np.mean(data)\n",

" std =np.std(data)\n",

" \n",

" \n",

" for i in data:\n",

" z\_score= (i - mean)/std \n",

" if np.abs(z\_score) > threshold:\n",

" outliers.append(y)\n",

" return outliers"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 269

},

"id": "fRbmatqjys10",

"outputId": "50041aaa-d774-4b57-c60b-25606c0391fe"

},

"outputs": [

{

"ename": "NameError",

"evalue": "ignored",

"output\_type": "error",

"traceback": [

"\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",

"\u001b[0;31mNameError\u001b[0m Traceback (most recent call last)",

"\u001b[0;32m<ipython-input-65-8d6d4d7620c1>\u001b[0m in \u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[0;32m----> 1\u001b[0;31m \u001b[0moutlier\_pt\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mdetect\_outliers\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mdataset1\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m",

"\u001b[0;32m<ipython-input-64-a36189340103>\u001b[0m in \u001b[0;36mdetect\_outliers\u001b[0;34m(data)\u001b[0m\n\u001b[1;32m 10\u001b[0m \u001b[0mz\_score\u001b[0m\u001b[0;34m=\u001b[0m \u001b[0;34m(\u001b[0m\u001b[0mi\u001b[0m \u001b[0;34m-\u001b[0m \u001b[0mmean\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m/\u001b[0m\u001b[0mstd\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 11\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0mnp\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mabs\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mz\_score\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;34m>\u001b[0m \u001b[0mthreshold\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m---> 12\u001b[0;31m \u001b[0moutliers\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mappend\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0my\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 13\u001b[0m \u001b[0;32mreturn\u001b[0m \u001b[0moutliers\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;31mNameError\u001b[0m: name 'y' is not defined"

]

}

],

"source": [

"outlier\_pt=detect\_outliers(dataset1)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "EWiwSJnQ0D4Q"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "q3fSmsJf0ogA"

},

"outputs": [],

"source": [

"df=pd.read\_csv('/content/Churn\_Modelling.csv')"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 424

},

"id": "KnFPuFlw0IzS",

"outputId": "409f85fb-38b4-454a-fdab-301343293574"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-be7e12b3-cabb-488c-a8a7-03cbce6f0d04\">\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",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

" <td>4</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 × 14 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-be7e12b3-cabb-488c-a8a7-03cbce6f0d04')\"\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-be7e12b3-cabb-488c-a8a7-03cbce6f0d04 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-be7e12b3-cabb-488c-a8a7-03cbce6f0d04');\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",

" "

],

"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",

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

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

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

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

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

"\n",

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

"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",

"9995 5 0.00 2 1 0 \n",

"9996 10 57369.61 1 1 1 \n",

"9997 7 0.00 1 0 1 \n",

"9998 3 75075.31 2 1 0 \n",

"9999 4 130142.79 1 1 0 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 \n",

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

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 \n",

"\n",

"[10000 rows x 14 columns]"

]

},

"execution\_count": 71,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "e20-p-5l0NBR"

},

"outputs": [],

"source": [

"df\_numeric = df[['RowNumber', 'CustomerId', 'CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','Exited']]"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "tlkQAyll0sqf",

"outputId": "8479482a-6440-4b8d-f7d0-1a26506eccd2"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-cae4e68b-4a6d-4516-aec1-cd4d054481a9\">\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>0</th>\n",

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

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

" <td>619</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>608</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>502</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>699</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>850</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-cae4e68b-4a6d-4516-aec1-cd4d054481a9')\"\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-cae4e68b-4a6d-4516-aec1-cd4d054481a9 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-cae4e68b-4a6d-4516-aec1-cd4d054481a9');\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",

" "

],

"text/plain": [

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

"0 1 15634602 619 42 2 0.00 1 \n",

"1 2 15647311 608 41 1 83807.86 1 \n",

"2 3 15619304 502 42 8 159660.80 3 \n",

"3 4 15701354 699 39 1 0.00 2 \n",

"4 5 15737888 850 43 2 125510.82 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 "

]

},

"execution\_count": 73,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_categorical = df[['Surname', 'Geography', 'Gender']]\n",

"\n",

"df\_numeric.head()\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 206

},

"id": "oZF-eG\_U0xxa",

"outputId": "7b52c360-609c-4818-8674-dba26b622a14"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-b4ed8f00-9ab3-4377-8042-efbf3fc46bfd\">\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>Surname</th>\n",

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-b4ed8f00-9ab3-4377-8042-efbf3fc46bfd')\"\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-b4ed8f00-9ab3-4377-8042-efbf3fc46bfd 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-b4ed8f00-9ab3-4377-8042-efbf3fc46bfd');\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",

" "

],

"text/plain": [

" Surname Geography Gender\n",

"0 Hargrave France Female\n",

"1 Hill Spain Female\n",

"2 Onio France Female\n",

"3 Boni France Female\n",

"4 Mitchell Spain Female"

]

},

"execution\_count": 74,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_categorical.head()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "MRtmsUa502aE",

"outputId": "a7bfd824-5685-474b-94d0-b3ea6ae1644d"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"['Hargrave' 'Hill' 'Onio' ... 'Kashiwagi' 'Aldridge' 'Burbidge']\n",

"['France' 'Spain' 'Germany']\n",

"['Female' 'Male']\n"

]

}

],

"source": [

"print(df['Surname'].unique())\n",

"print(df['Geography'].unique())\n",

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "sgmPVH7P09Kw"

},

"outputs": [],

"source": [

"from sklearn.preprocessing import LabelEncoder\n",

"marry\_encoder = LabelEncoder()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "GzpygOrS1A2M"

},

"outputs": [],

"source": [

"marry\_encoder.fit(df\_categorical['Gender'])\n",

"marry\_values = marry\_encoder.transform(df\_categorical['Gender'])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "Oo0H3Lm41GVW",

"outputId": "ffd19c55-3cd4-425f-d4c2-7b24f6179f2f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Before Encoding: ['Male', 'Female', 'Male', 'Male', 'Female', 'Male', 'Male', 'Female', 'Male', 'Female']\n",

"After Encoding: [1 0 1 1 0 1 1 0 1 0]\n",

"The inverse from the encoding result: ['Male' 'Female' 'Male' 'Male' 'Female' 'Male' 'Male' 'Female' 'Male'\n",

" 'Female']\n"

]

}

],

"source": [

"print(\"Before Encoding:\", list(df\_categorical['Gender'][-10:]))\n",

"print(\"After Encoding:\", marry\_values[-10:])\n",

"print(\"The inverse from the encoding result:\", marry\_encoder.inverse\_transform(marry\_values[-10:]))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "gnaTrJYq1MHS",

"outputId": "245a69e8-7ad9-4b51-b590-d53e9add19a8"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Before Encoding: ['France', 'Spain', 'France', 'France', 'Spain']\n",

"After Encoding: [0 2 0 0 2]\n",

"The inverse from the encoding result: ['France' 'Spain' 'France' 'France' 'Spain']\n"

]

}

],

"source": [

"residence\_encoder = LabelEncoder()\n",

"residence\_values = residence\_encoder.fit\_transform(df\_categorical['Geography'])\n",

"\n",

"print(\"Before Encoding:\", list(df\_categorical['Geography'][:5]))\n",

"print(\"After Encoding:\", residence\_values[:5])\n",

"print(\"The inverse from the encoding result:\", residence\_encoder.inverse\_transform(residence\_values[:5]))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "J9EUo0b41Xux",

"outputId": "0560b924-e957-47c6-ed89-0d31be85d364"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0 Female\n",

"1 Female\n",

"2 Female\n",

"3 Female\n",

"4 Female\n",

"Name: Gender, dtype: object\n",

"\n",

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

" [1. 0.]\n",

" [1. 0.]\n",

" [1. 0.]\n",

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

"\n",

"[['Female']\n",

" ['Female']\n",

" ['Female']\n",

" ['Female']\n",

" ['Female']]\n"

]

}

],

"source": [

"from sklearn.preprocessing import OneHotEncoder\n",

"gender\_encoder = OneHotEncoder()\n",

"\n",

"from sklearn.preprocessing import OneHotEncoder\n",

"import numpy as np\n",

"\n",

"gender\_encoder = OneHotEncoder()\n",

"gender\_reshaped = np.array(df\_categorical['Gender']).reshape(-1, 1)\n",

"gender\_values = gender\_encoder.fit\_transform(gender\_reshaped)\n",

"\n",

"print(df\_categorical['Gender'][:5])\n",

"print()\n",

"print(gender\_values.toarray()[:5])\n",

"print()\n",

"print(gender\_encoder.inverse\_transform(gender\_values)[:5])\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "Iuy9YBwX1cXU",

"outputId": "82f96708-a557-4a4e-80e6-455e7ba47b33"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0 Hargrave\n",

"1 Hill\n",

"2 Onio\n",

"3 Boni\n",

"4 Mitchell\n",

"Name: Surname, dtype: object\n",

"\n",

"[[0. 0. 0. ... 0. 0. 0.]\n",

" [0. 0. 0. ... 0. 0. 0.]\n",

" [0. 0. 0. ... 0. 0. 0.]\n",

" [0. 0. 0. ... 0. 0. 0.]\n",

" [0. 0. 0. ... 0. 0. 0.]]\n",

"\n",

"[['Hargrave']\n",

" ['Hill']\n",

" ['Onio']\n",

" ['Boni']\n",

" ['Mitchell']]\n"

]

}

],

"source": [

"smoke\_encoder = OneHotEncoder()\n",

"smoke\_reshaped = np.array(df\_categorical['Surname']).reshape(-1, 1)\n",

"smoke\_values = smoke\_encoder.fit\_transform(smoke\_reshaped)\n",

"\n",

"print(df\_categorical['Surname'][:5])\n",

"print()\n",

"print(smoke\_values.toarray()[:5])\n",

"print()\n",

"print(smoke\_encoder.inverse\_transform(smoke\_values)[:5])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "5RNYzNMP1f-0",

"outputId": "44a172b7-301d-4579-cd8c-c348f5cc9106"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0 France\n",

"1 Spain\n",

"2 France\n",

"3 France\n",

"4 Spain\n",

"Name: Geography, dtype: object\n",

"\n",

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

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

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

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

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

"\n",

"[['France']\n",

" ['Spain']\n",

" ['France']\n",

" ['France']\n",

" ['Spain']]\n"

]

}

],

"source": [

"work\_encoder = OneHotEncoder()\n",

"work\_reshaped = np.array(df\_categorical['Geography']).reshape(-1, 1)\n",

"work\_values = work\_encoder.fit\_transform(work\_reshaped)\n",

"\n",

"print(df\_categorical['Geography'][:5])\n",

"print()\n",

"print(work\_values.toarray()[:5])\n",

"print()\n",

"print(work\_encoder.inverse\_transform(work\_values)[:5])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 299

},

"id": "JrBJS-J81kgx",

"outputId": "2549f845-9532-4a8f-8497-0c478f047b2a"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-98b5d781-12d8-434f-bce5-cae5d0c12431\">\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>Surname\_Abbie</th>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 2934 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-98b5d781-12d8-434f-bce5-cae5d0c12431')\"\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-98b5d781-12d8-434f-bce5-cae5d0c12431 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-98b5d781-12d8-434f-bce5-cae5d0c12431');\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",

" "

],

"text/plain": [

" Surname\_Abbie Surname\_Abbott Surname\_Abdullah Surname\_Abdulov \\\n",

"0 0 0 0 0 \n",

"1 0 0 0 0 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 0 0 \n",

"\n",

" Surname\_Abel Surname\_Abernathy Surname\_Abramov Surname\_Abramova \\\n",

"0 0 0 0 0 \n",

"1 0 0 0 0 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 0 0 \n",

"\n",

" Surname\_Abramovich Surname\_Abramowitz ... Surname\_Zotova Surname\_Zox \\\n",

"0 0 0 ... 0 0 \n",

"1 0 0 ... 0 0 \n",

"2 0 0 ... 0 0 \n",

"3 0 0 ... 0 0 \n",

"4 0 0 ... 0 0 \n",

"\n",

" Surname\_Zubarev Surname\_Zubareva Surname\_Zuev Surname\_Zuyev \\\n",

"0 0 0 0 0 \n",

"1 0 0 0 0 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 0 0 \n",

"\n",

" Surname\_Zuyeva Geography\_Germany Geography\_Spain Gender\_Male \n",

"0 0 0 0 0 \n",

"1 0 0 1 0 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 1 0 \n",

"\n",

"[5 rows x 2934 columns]"

]

},

"execution\_count": 84,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_categorical\_encoded = pd.get\_dummies(df\_categorical, drop\_first=True)\n",

"df\_categorical\_encoded.head()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 299

},

"id": "ocOIhQB21piW",

"outputId": "88e90dd7-f416-48e8-dfd9-af24abe66cb3"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-275fc87d-e9f3-42f9-a6a2-91c9dd1fba63\">\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>...</th>\n",

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 2945 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-275fc87d-e9f3-42f9-a6a2-91c9dd1fba63')\"\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-275fc87d-e9f3-42f9-a6a2-91c9dd1fba63 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-275fc87d-e9f3-42f9-a6a2-91c9dd1fba63');\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",

" "

],

"text/plain": [

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

"0 1 15634602 619 42 2 0.00 1 \n",

"1 2 15647311 608 41 1 83807.86 1 \n",

"2 3 15619304 502 42 8 159660.80 3 \n",

"3 4 15701354 699 39 1 0.00 2 \n",

"4 5 15737888 850 43 2 125510.82 1 \n",

"\n",

" HasCrCard IsActiveMember EstimatedSalary ... Surname\_Zotova \\\n",

"0 1 1 101348.88 ... 0 \n",

"1 0 1 112542.58 ... 0 \n",

"2 1 0 113931.57 ... 0 \n",

"3 0 0 93826.63 ... 0 \n",

"4 1 1 79084.10 ... 0 \n",

"\n",

" Surname\_Zox Surname\_Zubarev Surname\_Zubareva Surname\_Zuev \\\n",

"0 0 0 0 0 \n",

"1 0 0 0 0 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 0 0 \n",

"\n",

" Surname\_Zuyev Surname\_Zuyeva Geography\_Germany Geography\_Spain \\\n",

"0 0 0 0 0 \n",

"1 0 0 0 1 \n",

"2 0 0 0 0 \n",

"3 0 0 0 0 \n",

"4 0 0 0 1 \n",

"\n",

" Gender\_Male \n",

"0 0 \n",

"1 0 \n",

"2 0 \n",

"3 0 \n",

"4 0 \n",

"\n",

"[5 rows x 2945 columns]"

]

},

"execution\_count": 85,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_new = pd.concat([df\_numeric, df\_categorical\_encoded], axis=1)\n",

"df\_new.head()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Sh4-Ej3j10zp"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "FItF8DAA1rmF",

"outputId": "a963c975-b9f9-414b-c509-f67963c4dc45"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"0.0\n",

"250898.09\n",

"76485.889288\n",

"RowNumber 10000\n",

"CustomerId 10000\n",

"Surname 10000\n",

"CreditScore 10000\n",

"Geography 10000\n",

"Gender 10000\n",

"Age 10000\n",

"Tenure 10000\n",

"Balance 10000\n",

"NumOfProducts 10000\n",

"HasCrCard 10000\n",

"IsActiveMember 10000\n",

"EstimatedSalary 10000\n",

"Exited 10000\n",

"dtype: int64\n"

]

}

],

"source": [

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

"\n",

"print(df[\"Balance\"].min())\n",

"print(df[\"Balance\"].max())\n",

"print(df[\"Balance\"].mean())\n",

"print(df.count(0))\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "nZGM3LMp2ATb",

"outputId": "8b5848d9-fa87-433b-c119-4b8a725c10e2"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"(10000, 14)\n",

"140000\n",

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

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

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

" ...\n",

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

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

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

]

}

],

"source": [

"print(df.shape)\n",

"print(df.size)\n",

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

"print(X)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "\_xs41fUz2JPd",

"outputId": "f7e103d3-9ba5-4b11-f106-a95c40605d94"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

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

"print(Y)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "I5mHfzSY2trB"

},

"source": [

"# \*\*Scale the independent variables\*\*"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 265

},

"id": "xhIs7Q-F2qEN",

"outputId": "4daab377-4a60-4fcd-9a72-ddedca1eb1cc"

},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 2 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

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

"\n",

"x = df[['Age', 'Tenure']].values\n",

"y = df['Gender'].values\n",

"\n",

"fig, ax = plt.subplots(ncols=2, figsize=(12, 4))\n",

"\n",

"ax[0].scatter(x[:,0], y)\n",

"ax[1].scatter(x[:,1], y)\n",

"\n",

"plt.show()\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 282

},

"id": "-uy5Smbn22ol",

"outputId": "467713d6-48f6-4cbe-b803-388702722ed8"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fc70a5be990>"

]

},

"execution\_count": 90,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"ax.scatter(x[:,0], y)\n",

"ax.scatter(x[:,1], y)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 334

},

"id": "AR2axrVM25zh",

"outputId": "c26f4498-05e8-44f5-d92c-818f0357cef2"

},

"outputs": [

{

"data": {

"text/plain": [

"(array([ 413., 1035., 1048., 1009., 989., 1012., 967., 1028., 1025.,\n",

" 1474.]),\n",

" array([ 0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.]),\n",

" <a list of 10 Patch objects>)"

]

},

"execution\_count": 91,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"ax.hist(x[:,0])\n",

"ax.hist(x[:,1])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 369

},

"id": "SPJwIR8D3EFF",

"outputId": "b4c8ce8f-f580-4606-eb3e-3011e1b02c5d"

},

"outputs": [

{

"data": {

"text/plain": [

"(array([ 413., 1035., 1048., 1009., 2001., 0., 1995., 0., 1025.,\n",

" 1474.]),\n",

" array([-1.73331549, -1.38753759, -1.04175968, -0.69598177, -0.35020386,\n",

" -0.00442596, 0.34135195, 0.68712986, 1.03290776, 1.37868567,\n",

" 1.72446358]),\n",

" <a list of 10 Patch objects>)"

]

},

"execution\_count": 93,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"from sklearn.preprocessing import StandardScaler\n",

"from sklearn.preprocessing import MinMaxScaler\n",

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"scaler = StandardScaler()\n",

"x\_std = scaler.fit\_transform(x)\n",

"\n",

"ax.hist(x\_std[:,0])\n",

"ax.hist(x\_std[:,1])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 282

},

"id": "9UP7SliG28sI",

"outputId": "1edeeceb-80f7-40f3-b677-18b19608098e"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fc70a3c5e10>"

]

},

"execution\_count": 94,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"scaler = StandardScaler()\n",

"x\_std = scaler.fit\_transform(x)\n",

"\n",

"ax.scatter(x\_std[:,0], y)\n",

"ax.scatter(x\_std[:,1], y)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 334

},

"id": "Vg6FAwGP3HeG",

"outputId": "97f9b603-a598-4abe-8429-59c27b0911ec"

},

"outputs": [

{

"data": {

"text/plain": [

"(array([ 413., 1035., 1048., 1009., 989., 1012., 967., 1028., 1025.,\n",

" 1474.]),\n",

" array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),\n",

" <a list of 10 Patch objects>)"

]

},

"execution\_count": 95,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"scaler = MinMaxScaler()\n",

"x\_minmax = scaler.fit\_transform(x)\n",

"\n",

"ax.hist(x\_minmax [:,0])\n",

"ax.hist(x\_minmax [:,1])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

"height": 282

},

"id": "QJsWcO2S3Mxa",

"outputId": "1451c6eb-cbeb-4a54-c86d-5aa03fa18004"

},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fc70a268f10>"

]

},

"execution\_count": 96,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 864x288 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig, ax = plt.subplots(figsize=(12, 4))\n",

"\n",

"scaler = MinMaxScaler()\n",

"x\_minmax = scaler.fit\_transform(x)\n",

"\n",

"ax.scatter(x\_minmax [:,0], y)\n",

"ax.scatter(x\_minmax [:,1], y)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "tg4v2TSl3QoP",

"outputId": "ba36015a-1788-4274-cbf8-c457e0ca08bc"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Mean Absolute Error: 55921.259463411116\n",

"Score -0.0031618755354316797\n"

]

}

],

"source": [

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

"from sklearn.pipeline import Pipeline\n",

"from sklearn.linear\_model import SGDRegressor\n",

"from sklearn.preprocessing import StandardScaler\n",

"from sklearn.preprocessing import MinMaxScaler\n",

"from sklearn.metrics import mean\_absolute\_error\n",

"import sklearn.metrics as metrics\n",

"\n",

"import pandas as pd\n",

"import numpy as np\n",

"import matplotlib.pyplot as plt\n",

"\n",

"# Import Data\n",

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

"x = df[['Age', 'Tenure']].values\n",

"y = df['Balance'].values\n",

"\n",

"# Split into a training and testing set\n",

"X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x, y)\n",

"\n",

"# Define the pipeline for scaling and model fitting\n",

"pipeline = Pipeline([\n",

" (\"MinMax Scaling\", MinMaxScaler()),\n",

" (\"SGD Regression\", SGDRegressor())\n",

"])\n",

"\n",

"# Scale the data and fit the model\n",

"pipeline.fit(X\_train, Y\_train)\n",

"\n",

"# Evaluate the model\n",

"Y\_pred = pipeline.predict(X\_test)\n",

"print('Mean Absolute Error: ', mean\_absolute\_error(Y\_pred, Y\_test))\n",

"print('Score', pipeline.score(X\_test, Y\_test))"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "1PrsJIdq3ezw"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "ouCWnFUF3Ydc"

},

"outputs": [],

"source": [

"dataset = pd.read\_csv('/content/Churn\_Modelling.csv')"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "1I0iTyYZ3m7G",

"outputId": "3fd47e6c-c990-4c53-b920-14d9721de86e"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"(10000, 13)\n",

" 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",

"5 6 15574012 Chu 645 Spain Male 44 \n",

"6 7 15592531 Bartlett 822 France Male 50 \n",

"7 8 15656148 Obinna 376 Germany Female 29 \n",

"8 9 15792365 He 501 France Male 44 \n",

"9 10 15592389 H? 684 France Male 27 \n",

"\n",

" Tenure Balance NumOfProducts IsActiveMember EstimatedSalary Exited \n",

"0 2 0.00 1 1 101348.88 1 \n",

"1 1 83807.86 1 1 112542.58 0 \n",

"2 8 159660.80 3 0 113931.57 1 \n",

"3 1 0.00 2 0 93826.63 0 \n",

"4 2 125510.82 1 1 79084.10 0 \n",

"5 8 113755.78 2 0 149756.71 1 \n",

"6 7 0.00 2 1 10062.80 0 \n",

"7 4 115046.74 4 0 119346.88 1 \n",

"8 4 142051.07 2 1 74940.50 0 \n",

"9 2 134603.88 1 1 71725.73 0 \n"

]

}

],

"source": [

"dataset.drop([\"HasCrCard\"],axis=1,inplace=True)\n",

"\n",

"print(dataset.shape)\n",

"print(dataset.head(10))\n",

"\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "\_EL1DWd53tKm",

"outputId": "775dd017-d4a1-4f77-8d24-3c98c9333030"

},

"outputs": [

{

"data": {

"text/plain": [

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

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

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

" ...,\n",

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

" [9999, 15682355, 'Sabbatini', ..., 2, 0, 92888.52],\n",

" [10000, 15628319, 'Walker', ..., 1, 0, 38190.78]], dtype=object)"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

"X"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "tl0SGtGy4Lkh",

"outputId": "4258f880-ea97-4434-806e-060f0c3cc6df"

},

"outputs": [

{

"data": {

"text/plain": [

"array([1, 0, 1, ..., 1, 1, 0])"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

"Y"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"colab": {

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

},

"id": "ZKMutOssZ4QM",

"outputId": "ffd33f70-9c49-4f0e-ba1e-0e7374905e62"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" x\_train.shape : (7500, 2)\n",

" y\_train.shape : (7500,)\n",

" x\_test.shape : (2500, 2)\n",

" y\_test.shape : (2500,)\n"

]

}

],

"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",

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

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

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

"print(' y\_test.shape : ',y\_test.shape)"

]

}

],

"metadata": {

"colab": {

"collapsed\_sections": [],

"provenance": []

},

"kernelspec": {

"display\_name": "Python 3",

"name": "python3"

},

"language\_info": {

"name": "python"

}

},

"nbformat": 4,

"nbformat\_minor": 0

}