{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "markdown",

"source": [

"Importing Libraries"

],

"metadata": {

"id": "Y0XclbRGmSjK"

}

},

{

"cell\_type": "code",

"execution\_count": 85,

"metadata": {

"id": "cmD\_AhEoD8MD"

},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns"

]

},

{

"cell\_type": "markdown",

"source": [

"Uploading Dataset"

],

"metadata": {

"id": "rmveiFtZmX6v"

}

},

{

"cell\_type": "code",

"source": [

"from google.colab import files\n",

"file = files.upload()"

],

"metadata": {

"colab": {

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

"height": 74

},

"id": "CZqQq-M-D8zR",

"outputId": "6175a72d-2e75-4c4f-e5f1-2cd75fa72462"

},

"execution\_count": 86,

"outputs": [

{

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

"data": {

"text/plain": [

"<IPython.core.display.HTML object>"

],

"text/html": [

"\n",

" <input type=\"file\" id=\"files-a38f6225-2572-438d-b371-68ebf9a78ad4\" name=\"files[]\" multiple disabled\n",

" style=\"border:none\" />\n",

" <output id=\"result-a38f6225-2572-438d-b371-68ebf9a78ad4\">\n",

" Upload widget is only available when the cell has been executed in the\n",

" current browser session. Please rerun this cell to enable.\n",

" </output>\n",

" <script>// Copyright 2017 Google LLC\n",

"//\n",

"// Licensed under the Apache License, Version 2.0 (the \"License\");\n",

"// you may not use this file except in compliance with the License.\n",

"// You may obtain a copy of the License at\n",

"//\n",

"// http://www.apache.org/licenses/LICENSE-2.0\n",

"//\n",

"// Unless required by applicable law or agreed to in writing, software\n",

"// distributed under the License is distributed on an \"AS IS\" BASIS,\n",

"// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n",

"// See the License for the specific language governing permissions and\n",

"// limitations under the License.\n",

"\n",

"/\*\*\n",

" \* @fileoverview Helpers for google.colab Python module.\n",

" \*/\n",

"(function(scope) {\n",

"function span(text, styleAttributes = {}) {\n",

" const element = document.createElement('span');\n",

" element.textContent = text;\n",

" for (const key of Object.keys(styleAttributes)) {\n",

" element.style[key] = styleAttributes[key];\n",

" }\n",

" return element;\n",

"}\n",

"\n",

"// Max number of bytes which will be uploaded at a time.\n",

"const MAX\_PAYLOAD\_SIZE = 100 \* 1024;\n",

"\n",

"function \_uploadFiles(inputId, outputId) {\n",

" const steps = uploadFilesStep(inputId, outputId);\n",

" const outputElement = document.getElementById(outputId);\n",

" // Cache steps on the outputElement to make it available for the next call\n",

" // to uploadFilesContinue from Python.\n",

" outputElement.steps = steps;\n",

"\n",

" return \_uploadFilesContinue(outputId);\n",

"}\n",

"\n",

"// This is roughly an async generator (not supported in the browser yet),\n",

"// where there are multiple asynchronous steps and the Python side is going\n",

"// to poll for completion of each step.\n",

"// This uses a Promise to block the python side on completion of each step,\n",

"// then passes the result of the previous step as the input to the next step.\n",

"function \_uploadFilesContinue(outputId) {\n",

" const outputElement = document.getElementById(outputId);\n",

" const steps = outputElement.steps;\n",

"\n",

" const next = steps.next(outputElement.lastPromiseValue);\n",

" return Promise.resolve(next.value.promise).then((value) => {\n",

" // Cache the last promise value to make it available to the next\n",

" // step of the generator.\n",

" outputElement.lastPromiseValue = value;\n",

" return next.value.response;\n",

" });\n",

"}\n",

"\n",

"/\*\*\n",

" \* Generator function which is called between each async step of the upload\n",

" \* process.\n",

" \* @param {string} inputId Element ID of the input file picker element.\n",

" \* @param {string} outputId Element ID of the output display.\n",

" \* @return {!Iterable<!Object>} Iterable of next steps.\n",

" \*/\n",

"function\* uploadFilesStep(inputId, outputId) {\n",

" const inputElement = document.getElementById(inputId);\n",

" inputElement.disabled = false;\n",

"\n",

" const outputElement = document.getElementById(outputId);\n",

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

"\n",

" const pickedPromise = new Promise((resolve) => {\n",

" inputElement.addEventListener('change', (e) => {\n",

" resolve(e.target.files);\n",

" });\n",

" });\n",

"\n",

" const cancel = document.createElement('button');\n",

" inputElement.parentElement.appendChild(cancel);\n",

" cancel.textContent = 'Cancel upload';\n",

" const cancelPromise = new Promise((resolve) => {\n",

" cancel.onclick = () => {\n",

" resolve(null);\n",

" };\n",

" });\n",

"\n",

" // Wait for the user to pick the files.\n",

" const files = yield {\n",

" promise: Promise.race([pickedPromise, cancelPromise]),\n",

" response: {\n",

" action: 'starting',\n",

" }\n",

" };\n",

"\n",

" cancel.remove();\n",

"\n",

" // Disable the input element since further picks are not allowed.\n",

" inputElement.disabled = true;\n",

"\n",

" if (!files) {\n",

" return {\n",

" response: {\n",

" action: 'complete',\n",

" }\n",

" };\n",

" }\n",

"\n",

" for (const file of files) {\n",

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

" li.append(span(file.name, {fontWeight: 'bold'}));\n",

" li.append(span(\n",

" `(${file.type || 'n/a'}) - ${file.size} bytes, ` +\n",

" `last modified: ${\n",

" file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() :\n",

" 'n/a'} - `));\n",

" const percent = span('0% done');\n",

" li.appendChild(percent);\n",

"\n",

" outputElement.appendChild(li);\n",

"\n",

" const fileDataPromise = new Promise((resolve) => {\n",

" const reader = new FileReader();\n",

" reader.onload = (e) => {\n",

" resolve(e.target.result);\n",

" };\n",

" reader.readAsArrayBuffer(file);\n",

" });\n",

" // Wait for the data to be ready.\n",

" let fileData = yield {\n",

" promise: fileDataPromise,\n",

" response: {\n",

" action: 'continue',\n",

" }\n",

" };\n",

"\n",

" // Use a chunked sending to avoid message size limits. See b/62115660.\n",

" let position = 0;\n",

" do {\n",

" const length = Math.min(fileData.byteLength - position, MAX\_PAYLOAD\_SIZE);\n",

" const chunk = new Uint8Array(fileData, position, length);\n",

" position += length;\n",

"\n",

" const base64 = btoa(String.fromCharCode.apply(null, chunk));\n",

" yield {\n",

" response: {\n",

" action: 'append',\n",

" file: file.name,\n",

" data: base64,\n",

" },\n",

" };\n",

"\n",

" let percentDone = fileData.byteLength === 0 ?\n",

" 100 :\n",

" Math.round((position / fileData.byteLength) \* 100);\n",

" percent.textContent = `${percentDone}% done`;\n",

"\n",

" } while (position < fileData.byteLength);\n",

" }\n",

"\n",

" // All done.\n",

" yield {\n",

" response: {\n",

" action: 'complete',\n",

" }\n",

" };\n",

"}\n",

"\n",

"scope.google = scope.google || {};\n",

"scope.google.colab = scope.google.colab || {};\n",

"scope.google.colab.\_files = {\n",

" \_uploadFiles,\n",

" \_uploadFilesContinue,\n",

"};\n",

"})(self);\n",

"</script> "

]

},

"metadata": {}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Saving Churn\_Modelling.csv to Churn\_Modelling (1).csv\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"Reading Dataset"

],

"metadata": {

"id": "jzecWHkumapk"

}

},

{

"cell\_type": "code",

"source": [

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

"df.shape"

],

"metadata": {

"colab": {

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

},

"id": "S9WnPY6ND82G",

"outputId": "d61c0cbb-1790-4033-d3ea-bed1896cbb9d"

},

"execution\_count": 87,

"outputs": [

{

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

"data": {

"text/plain": [

"(10000, 14)"

]

},

"metadata": {},

"execution\_count": 87

}

]

},

{

"cell\_type": "code",

"source": [

"df.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "eGsa6rmoD84w",

"outputId": "cbb4fca3-6257-4139-9be4-1106eb8d6f29"

},

"execution\_count": 88,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

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

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

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

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

"\n",

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

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"2 8 159660.80 3 1 0 \n",

"3 1 0.00 2 0 0 \n",

"4 2 125510.82 1 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 "

],

"text/html": [

"\n",

" <div id=\"df-ca961a2d-330b-4138-9ff3-7415bcf6c3db\">\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-ca961a2d-330b-4138-9ff3-7415bcf6c3db')\"\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-ca961a2d-330b-4138-9ff3-7415bcf6c3db 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-ca961a2d-330b-4138-9ff3-7415bcf6c3db');\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": 88

}

]

},

{

"cell\_type": "code",

"source": [

"df.tail()"

],

"metadata": {

"id": "qf394XrED871",

"colab": {

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

"height": 206

},

"outputId": "e335e2a8-e8ed-4591-ddcd-b6ca18d056d2"

},

"execution\_count": 89,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

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

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

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

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

"\n",

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

"9995 5 0.00 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",

"9995 96270.64 0 \n",

"9996 101699.77 0 \n",

"9997 42085.58 1 \n",

"9998 92888.52 1 \n",

"9999 38190.78 0 "

],

"text/html": [

"\n",

" <div id=\"df-8b674cfa-75fb-496a-a728-af00cde26447\">\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>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",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-8b674cfa-75fb-496a-a728-af00cde26447')\"\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-8b674cfa-75fb-496a-a728-af00cde26447 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-8b674cfa-75fb-496a-a728-af00cde26447');\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": 89

}

]

},

{

"cell\_type": "code",

"source": [

"df.columns"

],

"metadata": {

"colab": {

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

},

"id": "0n2kR19gkB4O",

"outputId": "8d7d3b00-4709-4cb7-8c74-e7bf5e2db4dc"

},

"execution\_count": 90,

"outputs": [

{

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

"data": {

"text/plain": [

"Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',\n",

" 'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',\n",

" 'IsActiveMember', 'EstimatedSalary', 'Exited'],\n",

" dtype='object')"

]

},

"metadata": {},

"execution\_count": 90

}

]

},

{

"cell\_type": "markdown",

"source": [

"Categorical Columns and Perform Encoding"

],

"metadata": {

"id": "IVO-vWqSnmxa"

}

},

{

"cell\_type": "code",

"source": [

"df['Gender'].replace({'Male': 1, 'Female': 0}, inplace=True)\n",

"df.head(5)"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "PN1RKLv-6TBk",

"outputId": "ee6d8cdb-a2e4-418f-e0d3-3d82be1d8bc6"

},

"execution\_count": 91,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

"1 2 15647311 Hill 608 Spain 0 41 \n",

"2 3 15619304 Onio 502 France 0 42 \n",

"3 4 15701354 Boni 699 France 0 39 \n",

"4 5 15737888 Mitchell 850 Spain 0 43 \n",

"\n",

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

"0 2 0.00 1 1 1 \n",

"1 1 83807.86 1 0 1 \n",

"2 8 159660.80 3 1 0 \n",

"3 1 0.00 2 0 0 \n",

"4 2 125510.82 1 1 1 \n",

"\n",

" EstimatedSalary Exited \n",

"0 101348.88 1 \n",

"1 112542.58 0 \n",

"2 113931.57 1 \n",

"3 93826.63 0 \n",

"4 79084.10 0 "

],

"text/html": [

"\n",

" <div id=\"df-42e69d6e-cc3d-4d0b-abb6-0e5e0c21936d\">\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>0</td>\n",

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

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

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

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

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

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

" <td>101348.88</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

" <td>15647311</td>\n",

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

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

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

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

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

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

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

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

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

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

" <td>113931.57</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

" <td>15701354</td>\n",

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

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

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

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

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

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

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

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

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

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

" <td>93826.63</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

" <td>15737888</td>\n",

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

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

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

" <td>0</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-42e69d6e-cc3d-4d0b-abb6-0e5e0c21936d')\"\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-42e69d6e-cc3d-4d0b-abb6-0e5e0c21936d 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-42e69d6e-cc3d-4d0b-abb6-0e5e0c21936d');\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": 91

}

]

},

{

"cell\_type": "markdown",

"source": [

"Dropping Unwanted Columns"

],

"metadata": {

"id": "XnzGiLrBoEN\_"

}

},

{

"cell\_type": "code",

"source": [

"df = df.drop(columns=['RowNumber', 'CustomerId', 'Surname', 'Geography'])\n",

"df.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "K3bhFpQHjZuu",

"outputId": "ef1636bc-2246-44df-a93d-cc009255147e"

},

"execution\_count": 92,

"outputs": [

{

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

"data": {

"text/plain": [

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

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

"1 608 0 41 1 83807.86 1 0 \n",

"2 502 0 42 8 159660.80 3 1 \n",

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

"4 850 0 43 2 125510.82 1 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 "

],

"text/html": [

"\n",

" <div id=\"df-7b6b30d5-c3ae-43eb-9314-48ca2703f91a\">\n",

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

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

" <td>101348.88</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

" <td>112542.58</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

" <td>113931.57</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

" <td>93826.63</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

" <td>0</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-7b6b30d5-c3ae-43eb-9314-48ca2703f91a')\"\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-7b6b30d5-c3ae-43eb-9314-48ca2703f91a 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-7b6b30d5-c3ae-43eb-9314-48ca2703f91a');\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": 92

}

]

},

{

"cell\_type": "markdown",

"source": [

"Handling Missing Values"

],

"metadata": {

"id": "vc\_lZC3JpqKO"

}

},

{

"cell\_type": "code",

"source": [

"df.duplicated().sum()"

],

"metadata": {

"colab": {

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

},

"id": "RaUSZOoIjZp4",

"outputId": "82e2780b-cd83-4312-a0a0-8ec0437c2735"

},

"execution\_count": 93,

"outputs": [

{

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

"data": {

"text/plain": [

"0"

]

},

"metadata": {},

"execution\_count": 93

}

]

},

{

"cell\_type": "code",

"source": [

"df.isna().sum()"

],

"metadata": {

"id": "bCpCIAEUD9Bb",

"colab": {

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

},

"outputId": "4c3d3573-aad6-48f1-bed3-eb4923e6c279"

},

"execution\_count": 94,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 0\n",

"Gender 0\n",

"Age 0\n",

"Tenure 0\n",

"Balance 0\n",

"NumOfProducts 0\n",

"HasCrCard 0\n",

"IsActiveMember 0\n",

"EstimatedSalary 0\n",

"Exited 0\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 94

}

]

},

{

"cell\_type": "code",

"source": [

"df.nunique()"

],

"metadata": {

"colab": {

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

},

"id": "jIig4dWhBrBD",

"outputId": "dcfcbefa-fa1a-423a-c329-56da7bb08360"

},

"execution\_count": 95,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 460\n",

"Gender 2\n",

"Age 70\n",

"Tenure 11\n",

"Balance 6382\n",

"NumOfProducts 4\n",

"HasCrCard 2\n",

"IsActiveMember 2\n",

"EstimatedSalary 9999\n",

"Exited 2\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 95

}

]

},

{

"cell\_type": "code",

"source": [

"df.info()"

],

"metadata": {

"colab": {

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

},

"id": "ImJ9xeBRmYNz",

"outputId": "d9d96d25-cab2-444a-eceb-9cb3f030c1da"

},

"execution\_count": 96,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

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

"Data columns (total 10 columns):\n",

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

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

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

" 1 Gender 10000 non-null int64 \n",

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

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

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

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

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

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

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

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

"dtypes: float64(2), int64(8)\n",

"memory usage: 781.4 KB\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

". Perform Descriptive Statistics"

],

"metadata": {

"id": "-YH8sSQroXxH"

}

},

{

"cell\_type": "code",

"source": [

"df.drop(columns=['Gender', 'HasCrCard', 'IsActiveMember', 'Exited']).describe()"

],

"metadata": {

"id": "DOaEvFw-D8-W",

"colab": {

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

"height": 300

},

"outputId": "2886ef5f-a132-4183-8757-3bad8c2fbfc3"

},

"execution\_count": 97,

"outputs": [

{

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

"data": {

"text/plain": [

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

"count 10000.000000 10000.000000 10000.000000 10000.000000 10000.000000 \n",

"mean 650.528800 38.921800 5.012800 76485.889288 1.530200 \n",

"std 96.653299 10.487806 2.892174 62397.405202 0.581654 \n",

"min 350.000000 18.000000 0.000000 0.000000 1.000000 \n",

"25% 584.000000 32.000000 3.000000 0.000000 1.000000 \n",

"50% 652.000000 37.000000 5.000000 97198.540000 1.000000 \n",

"75% 718.000000 44.000000 7.000000 127644.240000 2.000000 \n",

"max 850.000000 92.000000 10.000000 250898.090000 4.000000 \n",

"\n",

" EstimatedSalary \n",

"count 10000.000000 \n",

"mean 100090.239881 \n",

"std 57510.492818 \n",

"min 11.580000 \n",

"25% 51002.110000 \n",

"50% 100193.915000 \n",

"75% 149388.247500 \n",

"max 199992.480000 "

],

"text/html": [

"\n",

" <div id=\"df-d7d7ddef-b732-403f-a3dd-e814c1f1d37e\">\n",

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

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-d7d7ddef-b732-403f-a3dd-e814c1f1d37e')\"\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-d7d7ddef-b732-403f-a3dd-e814c1f1d37e 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-d7d7ddef-b732-403f-a3dd-e814c1f1d37e');\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": 97

}

]

},

{

"cell\_type": "markdown",

"source": [

"Find Outliers"

],

"metadata": {

"id": "bX58XsnRo6WI"

}

},

{

"cell\_type": "code",

"source": [

"qnt = df.drop(columns=['Gender', 'Tenure', 'HasCrCard', 'IsActiveMember', 'NumOfProducts', 'Exited']).quantile(q=[0.25, 0.75])\n",

"qnt"

],

"metadata": {

"colab": {

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

"height": 112

},

"id": "rL377fcDklH3",

"outputId": "f4c6db43-f8d7-4176-9411-79ef07ba132a"

},

"execution\_count": 98,

"outputs": [

{

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

"data": {

"text/plain": [

" CreditScore Age Balance EstimatedSalary\n",

"0.25 584.0 32.0 0.00 51002.1100\n",

"0.75 718.0 44.0 127644.24 149388.2475"

],

"text/html": [

"\n",

" <div id=\"df-0ac134a4-6a58-4593-b0fb-92e3fa1887e0\">\n",

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

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0.25</th>\n",

" <td>584.0</td>\n",

" <td>32.0</td>\n",

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

" <td>51002.1100</td>\n",

" </tr>\n",

" <tr>\n",

" <th>0.75</th>\n",

" <td>718.0</td>\n",

" <td>44.0</td>\n",

" <td>127644.24</td>\n",

" <td>149388.2475</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-0ac134a4-6a58-4593-b0fb-92e3fa1887e0')\"\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-0ac134a4-6a58-4593-b0fb-92e3fa1887e0 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-0ac134a4-6a58-4593-b0fb-92e3fa1887e0');\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": 98

}

]

},

{

"cell\_type": "code",

"source": [

"Q1 = qnt.iloc[0]\n",

"Q3 = qnt.iloc[1]\n",

"iqr = Q3 - Q1\n",

"iqr"

],

"metadata": {

"colab": {

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

},

"id": "gVHyi\_RWkhCx",

"outputId": "c7ed6a79-5a16-471a-8b6f-bfa41ea2ca14"

},

"execution\_count": 99,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 134.0000\n",

"Age 12.0000\n",

"Balance 127644.2400\n",

"EstimatedSalary 98386.1375\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 99

}

]

},

{

"cell\_type": "code",

"source": [

"upper = qnt.iloc[1] + 1.5\*iqr\n",

"upper"

],

"metadata": {

"colab": {

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

},

"id": "pkN4elqGkdAI",

"outputId": "4d1de429-5bc3-4c87-bde4-80475061908b"

},

"execution\_count": 100,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 919.00000\n",

"Age 62.00000\n",

"Balance 319110.60000\n",

"EstimatedSalary 296967.45375\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 100

}

]

},

{

"cell\_type": "code",

"source": [

"lower = qnt.iloc[0] - 1.5\*iqr\n",

"lower"

],

"metadata": {

"colab": {

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

},

"id": "SzEW2D2-kgUB",

"outputId": "789708ad-546b-4f3d-d0f6-5273ceabc2e7"

},

"execution\_count": 101,

"outputs": [

{

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

"data": {

"text/plain": [

"CreditScore 383.00000\n",

"Age 14.00000\n",

"Balance -191466.36000\n",

"EstimatedSalary -96577.09625\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 101

}

]

},

{

"cell\_type": "markdown",

"source": [

"Replace Outliers"

],

"metadata": {

"id": "nNFaVqmTpRIv"

}

},

{

"cell\_type": "code",

"source": [

"df['CreditScore'] = np.where(df['CreditScore']>756, 650.5288, df['CreditScore'])\n",

"df['Age'] = np.where(df['Age']>62, 38.9218, df['Age'])"

],

"metadata": {

"id": "M8jmy-qnouYz"

},

"execution\_count": 102,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"Performing Visualizations"

],

"metadata": {

"id": "H2J8BH-VorXu"

}

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "w4u7kTR1uA7T",

"colab": {

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

"height": 354

},

"outputId": "1b7037d5-46be-4a8e-c221-73e1d1deb2d4"

},

"execution\_count": 103,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff748021cd0>"

]

},

"metadata": {},

"execution\_count": 103

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "NX0wB1IiD9EE",

"colab": {

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

"height": 354

},

"outputId": "4382f629-9068-4d19-92af-d5c757344675"

},

"execution\_count": 104,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff747f9f250>"

]

},

"metadata": {},

"execution\_count": 104

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"colab": {

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

"height": 354

},

"id": "-RNZkb7zxN3d",

"outputId": "bca96d1e-69e2-431d-aa80-f49d95ca86d6"

},

"execution\_count": 105,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff747f07ed0>"

]

},

"metadata": {},

"execution\_count": 105

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"colab": {

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

"height": 354

},

"id": "m37y5PAim9Ff",

"outputId": "e1bdbf7b-a196-47ae-ba25-e6eb5c14add6"

},

"execution\_count": 106,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff747e72290>"

]

},

"metadata": {},

"execution\_count": 106

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "PTC82BGnuA-t",

"colab": {

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

"height": 354

},

"outputId": "d0d80120-1b35-44eb-86ae-ec557cb6a427"

},

"execution\_count": 107,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff748c6ef10>"

]

},

"metadata": {},

"execution\_count": 107

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.heatmap(df.corr(), annot=True)"

],

"metadata": {

"colab": {

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

"height": 359

},

"id": "wJFJAFzDqLRn",

"outputId": "81608abc-337c-4503-85cb-ef7784ec1bce"

},

"execution\_count": 108,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff747db12d0>"

]

},

"metadata": {},

"execution\_count": 108

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

" Split Data into Dependent and Independent Variables"

],

"metadata": {

"id": "bLREYVMspyoG"

}

},

{

"cell\_type": "code",

"source": [

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

"x.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "j3WNsWuDqj9m",

"outputId": "5ed90b36-53f3-4786-d28c-52a99c12c457"

},

"execution\_count": 109,

"outputs": [

{

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

"data": {

"text/plain": [

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

"0 619.0000 0 42.0 2 0.00 1 1 \n",

"1 608.0000 0 41.0 1 83807.86 1 0 \n",

"2 502.0000 0 42.0 8 159660.80 3 1 \n",

"3 699.0000 0 39.0 1 0.00 2 0 \n",

"4 650.5288 0 43.0 2 125510.82 1 1 \n",

"\n",

" IsActiveMember EstimatedSalary \n",

"0 1 101348.88 \n",

"1 1 112542.58 \n",

"2 0 113931.57 \n",

"3 0 93826.63 \n",

"4 1 79084.10 "

],

"text/html": [

"\n",

" <div id=\"df-20dd267c-5971-4a0c-87fd-bfdb7fdb8d33\">\n",

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

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>619.0000</td>\n",

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

" <td>42.0</td>\n",

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

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

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

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

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

" <td>101348.88</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>608.0000</td>\n",

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

" <td>41.0</td>\n",

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

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

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

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

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

" <td>112542.58</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>502.0000</td>\n",

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

" <td>42.0</td>\n",

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

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

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

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

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

" <td>113931.57</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>699.0000</td>\n",

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

" <td>39.0</td>\n",

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

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

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

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

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

" <td>93826.63</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>650.5288</td>\n",

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

" <td>43.0</td>\n",

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

" <td>125510.82</td>\n",

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

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

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

" <td>79084.10</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-20dd267c-5971-4a0c-87fd-bfdb7fdb8d33')\"\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-20dd267c-5971-4a0c-87fd-bfdb7fdb8d33 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-20dd267c-5971-4a0c-87fd-bfdb7fdb8d33');\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": 109

}

]

},

{

"cell\_type": "code",

"source": [

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

"y.head()"

],

"metadata": {

"colab": {

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

},

"id": "UYqcAustrCG4",

"outputId": "e038673e-40b8-4a30-f79a-bdb9f023883e"

},

"execution\_count": 110,

"outputs": [

{

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

"data": {

"text/plain": [

"0 1\n",

"1 0\n",

"2 1\n",

"3 0\n",

"4 0\n",

"Name: Exited, dtype: int64"

]

},

"metadata": {},

"execution\_count": 110

}

]

},

{

"cell\_type": "markdown",

"source": [

" Scale the independent variables"

],

"metadata": {

"id": "wyP8uicbrbt-"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.preprocessing import StandardScaler\n",

"ss = StandardScaler()\n",

"x = ss.fit\_transform(x)"

],

"metadata": {

"id": "aFJmLbaE1h8k"

},

"execution\_count": 120,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"x"

],

"metadata": {

"colab": {

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

},

"id": "pwGr9U1Sto-k",

"outputId": "aa25d15f-1e4e-4ea0-e0a0-90b683b6b3c3"

},

"execution\_count": 122,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[-0.13284832, -1.09598752, 0.48205148, ..., 0.64609167,\n",

" 0.97024255, 0.02188649],\n",

" [-0.28182929, -1.09598752, 0.36638802, ..., -1.54776799,\n",

" 0.97024255, 0.21653375],\n",

" [-1.71746409, -1.09598752, 0.48205148, ..., 0.64609167,\n",

" -1.03067011, 0.2406869 ],\n",

" ...,\n",

" [ 1.08608688, -1.09598752, -0.21192932, ..., -1.54776799,\n",

" 0.97024255, -1.00864308],\n",

" [ 0.29416906, 0.91241915, 0.48205148, ..., 0.64609167,\n",

" -1.03067011, -0.12523071],\n",

" [ 0.29416906, -1.09598752, -1.13723705, ..., 0.64609167,\n",

" -1.03067011, -1.07636976]])"

]

},

"metadata": {},

"execution\_count": 122

}

]

},

{

"cell\_type": "markdown",

"source": [

"Split the data into training and testing"

],

"metadata": {

"id": "iGJt040wrJMO"

}

},

{

"cell\_type": "code",

"source": [

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

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

],

"metadata": {

"id": "vUfNmpfvtfdm"

},

"execution\_count": 123,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"x\_train.shape"

],

"metadata": {

"id": "9s4qe5qJtfwa",

"colab": {

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

},

"outputId": "8d6805cb-f990-4b4d-98ac-e28b6fa1aeb5"

},

"execution\_count": 124,

"outputs": [

{

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

"data": {

"text/plain": [

"(8000, 9)"

]

},

"metadata": {},

"execution\_count": 124

}

]

},

{

"cell\_type": "code",

"source": [

"x\_test.shape"

],

"metadata": {

"colab": {

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

},

"id": "3\_3k6kKt1h\_0",

"outputId": "3f2eae31-9173-47a8-d4ed-c367bd693a28"

},

"execution\_count": 125,

"outputs": [

{

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

"data": {

"text/plain": [

"(2000, 9)"

]

},

"metadata": {},

"execution\_count": 125

}

]

},

{

"cell\_type": "code",

"source": [

"x\_train"

],

"metadata": {

"colab": {

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

},

"id": "VepARGox1h3F",

"outputId": "313cc229-f4d9-4405-965c-6abdeaee505d"

},

"execution\_count": 127,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[ 0.51725045, -1.09598752, -0.44325625, ..., 0.64609167,\n",

" -1.03067011, 1.10838187],\n",

" [-2.73324343, 0.91241915, 0.48205148, ..., 0.64609167,\n",

" 0.97024255, -0.74759209],\n",

" [-1.27052118, -1.09598752, -1.02157358, ..., 0.64609167,\n",

" -1.03067011, 1.48746417],\n",

" ...,\n",

" [ 1.47885489, 0.91241915, -0.32759278, ..., 0.64609167,\n",

" -1.03067011, 1.41441489],\n",

" [-0.52561634, -1.09598752, 0.01939762, ..., 0.64609167,\n",

" 0.97024255, 0.84614739],\n",

" [-0.07867343, -1.09598752, 1.17603229, ..., 0.64609167,\n",

" -1.03067011, 0.32630495]])"

]

},

"metadata": {},

"execution\_count": 127

}

]

},

{

"cell\_type": "code",

"source": [

"x\_test"

],

"metadata": {

"id": "Y9HiyQpWtxtX",

"outputId": "09f6b06a-b1d1-4b6c-fb09-a13614bd102a",

"colab": {

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

}

},

"execution\_count": 128,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[-0.43081026, -1.09598752, -0.32759278, ..., 0.64609167,\n",

" 0.97024255, 1.61304597],\n",

" [-1.43304588, -1.09598752, 0.25072455, ..., 0.64609167,\n",

" -1.03067011, 0.49753166],\n",

" [ 1.04545571, -1.09598752, 0.48205148, ..., 0.64609167,\n",

" 0.97024255, -0.4235611 ],\n",

" ...,\n",

" [-0.68814103, 0.91241915, -0.21192932, ..., 0.64609167,\n",

" -1.03067011, 0.72065149],\n",

" [ 0.28700714, 0.91241915, -0.44325625, ..., 0.64609167,\n",

" 0.97024255, -1.54438254],\n",

" [-0.75585965, 0.91241915, -0.90591012, ..., 0.64609167,\n",

" -1.03067011, 1.61474887]])"

]

},

"metadata": {},

"execution\_count": 128

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "18hXtuJ7txqM"

},

"execution\_count": null,

"outputs": []

}

]

}