{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "markdown",

"source": [

"# \*\*1. Download the dataset: Dataset\*\*\n",

"# \*\*2. Load the dataset into the tool.\*\*"

],

"metadata": {

"id": "wWpPrzeEkW4w"

}

},

{

"cell\_type": "code",

"execution\_count": 155,

"metadata": {

"colab": {

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

"height": 424

},

"id": "CMUwmETPjfpf",

"outputId": "88fa62e2-18d9-4d6e-9114-c7d7abfba3d2"

},

"outputs": [

{

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

"data": {

"text/plain": [

" Sex Length Diameter Height Whole weight Shucked weight \\\n",

"0 M 0.455 0.365 0.095 0.5140 0.2245 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 \n",

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

"4172 F 0.565 0.450 0.165 0.8870 0.3700 \n",

"4173 M 0.590 0.440 0.135 0.9660 0.4390 \n",

"4174 M 0.600 0.475 0.205 1.1760 0.5255 \n",

"4175 F 0.625 0.485 0.150 1.0945 0.5310 \n",

"4176 M 0.710 0.555 0.195 1.9485 0.9455 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"0 0.1010 0.1500 15 \n",

"1 0.0485 0.0700 7 \n",

"2 0.1415 0.2100 9 \n",

"3 0.1140 0.1550 10 \n",

"4 0.0395 0.0550 7 \n",

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

"4172 0.2390 0.2490 11 \n",

"4173 0.2145 0.2605 10 \n",

"4174 0.2875 0.3080 9 \n",

"4175 0.2610 0.2960 10 \n",

"4176 0.3765 0.4950 12 \n",

"\n",

"[4177 rows x 9 columns]"

],

"text/html": [

"\n",

" <div id=\"df-73cc8892-2acb-4a50-839e-61df53246002\">\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>Sex</th>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

" <td>0.455</td>\n",

" <td>0.365</td>\n",

" <td>0.095</td>\n",

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.1500</td>\n",

" <td>15</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>0.350</td>\n",

" <td>0.265</td>\n",

" <td>0.090</td>\n",

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.0700</td>\n",

" <td>7</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>0.530</td>\n",

" <td>0.420</td>\n",

" <td>0.135</td>\n",

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.2100</td>\n",

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

" </tr>\n",

" <tr>\n",

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

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

" <td>0.440</td>\n",

" <td>0.365</td>\n",

" <td>0.125</td>\n",

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.1550</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>0.330</td>\n",

" <td>0.255</td>\n",

" <td>0.080</td>\n",

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.0550</td>\n",

" <td>7</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",

" </tr>\n",

" <tr>\n",

" <th>4172</th>\n",

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

" <td>0.565</td>\n",

" <td>0.450</td>\n",

" <td>0.165</td>\n",

" <td>0.8870</td>\n",

" <td>0.3700</td>\n",

" <td>0.2390</td>\n",

" <td>0.2490</td>\n",

" <td>11</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4173</th>\n",

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

" <td>0.590</td>\n",

" <td>0.440</td>\n",

" <td>0.135</td>\n",

" <td>0.9660</td>\n",

" <td>0.4390</td>\n",

" <td>0.2145</td>\n",

" <td>0.2605</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4174</th>\n",

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

" <td>0.600</td>\n",

" <td>0.475</td>\n",

" <td>0.205</td>\n",

" <td>1.1760</td>\n",

" <td>0.5255</td>\n",

" <td>0.2875</td>\n",

" <td>0.3080</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>4175</th>\n",

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

" <td>0.625</td>\n",

" <td>0.485</td>\n",

" <td>0.150</td>\n",

" <td>1.0945</td>\n",

" <td>0.5310</td>\n",

" <td>0.2610</td>\n",

" <td>0.2960</td>\n",

" <td>10</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4176</th>\n",

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

" <td>0.710</td>\n",

" <td>0.555</td>\n",

" <td>0.195</td>\n",

" <td>1.9485</td>\n",

" <td>0.9455</td>\n",

" <td>0.3765</td>\n",

" <td>0.4950</td>\n",

" <td>12</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>4177 rows × 9 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-73cc8892-2acb-4a50-839e-61df53246002')\"\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-73cc8892-2acb-4a50-839e-61df53246002 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-73cc8892-2acb-4a50-839e-61df53246002');\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": 155

}

],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"import matplotlib.pyplot as plt\n",

"import seaborn as sns\n",

"import plotly as py\n",

"from sklearn.tree import DecisionTreeRegressor\n",

"from sklearn.metrics import mean\_absolute\_error\n",

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

"from sklearn.model\_selection import cross\_val\_score\n",

"from sklearn.ensemble import RandomForestRegressor\n",

"#from sklearn.preprocessing.imputation import Imputer\n",

"from xgboost import XGBRegressor\n",

"from sklearn.ensemble import GradientBoostingRegressor, GradientBoostingClassifier\n",

"#from sklearn.ensemble.partial\_dependence import partial\_dependence, plot\_partial\_dependence\n",

"from sklearn.pipeline import make\_pipeline\n",

"from sklearn.metrics import confusion\_matrix\n",

"from sklearn.metrics import accuracy\_score\n",

"from sklearn import metrics\n",

"\n",

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

"data"

]

},

{

"cell\_type": "code",

"source": [

"data.describe()"

],

"metadata": {

"colab": {

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

"height": 300

},

"id": "EBeMk5u-mYCd",

"outputId": "047ef46a-df10-4da9-8272-02f9d3134f30"

},

"execution\_count": 156,

"outputs": [

{

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

"data": {

"text/plain": [

" Length Diameter Height Whole weight Shucked weight \\\n",

"count 4177.000000 4177.000000 4177.000000 4177.000000 4177.000000 \n",

"mean 0.523992 0.407881 0.139516 0.828742 0.359367 \n",

"std 0.120093 0.099240 0.041827 0.490389 0.221963 \n",

"min 0.075000 0.055000 0.000000 0.002000 0.001000 \n",

"25% 0.450000 0.350000 0.115000 0.441500 0.186000 \n",

"50% 0.545000 0.425000 0.140000 0.799500 0.336000 \n",

"75% 0.615000 0.480000 0.165000 1.153000 0.502000 \n",

"max 0.815000 0.650000 1.130000 2.825500 1.488000 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"count 4177.000000 4177.000000 4177.000000 \n",

"mean 0.180594 0.238831 9.933684 \n",

"std 0.109614 0.139203 3.224169 \n",

"min 0.000500 0.001500 1.000000 \n",

"25% 0.093500 0.130000 8.000000 \n",

"50% 0.171000 0.234000 9.000000 \n",

"75% 0.253000 0.329000 11.000000 \n",

"max 0.760000 1.005000 29.000000 "

],

"text/html": [

"\n",

" <div id=\"df-fe5421d0-0d2f-4c64-92a0-3ee4369f324a\">\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>Length</th>\n",

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" <td>4177.000000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.523992</td>\n",

" <td>0.407881</td>\n",

" <td>0.139516</td>\n",

" <td>0.828742</td>\n",

" <td>0.359367</td>\n",

" <td>0.180594</td>\n",

" <td>0.238831</td>\n",

" <td>9.933684</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.120093</td>\n",

" <td>0.099240</td>\n",

" <td>0.041827</td>\n",

" <td>0.490389</td>\n",

" <td>0.221963</td>\n",

" <td>0.109614</td>\n",

" <td>0.139203</td>\n",

" <td>3.224169</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.075000</td>\n",

" <td>0.055000</td>\n",

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

" <td>0.002000</td>\n",

" <td>0.001000</td>\n",

" <td>0.000500</td>\n",

" <td>0.001500</td>\n",

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

" </tr>\n",

" <tr>\n",

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

" <td>0.450000</td>\n",

" <td>0.350000</td>\n",

" <td>0.115000</td>\n",

" <td>0.441500</td>\n",

" <td>0.186000</td>\n",

" <td>0.093500</td>\n",

" <td>0.130000</td>\n",

" <td>8.000000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.545000</td>\n",

" <td>0.425000</td>\n",

" <td>0.140000</td>\n",

" <td>0.799500</td>\n",

" <td>0.336000</td>\n",

" <td>0.171000</td>\n",

" <td>0.234000</td>\n",

" <td>9.000000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.615000</td>\n",

" <td>0.480000</td>\n",

" <td>0.165000</td>\n",

" <td>1.153000</td>\n",

" <td>0.502000</td>\n",

" <td>0.253000</td>\n",

" <td>0.329000</td>\n",

" <td>11.000000</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.815000</td>\n",

" <td>0.650000</td>\n",

" <td>1.130000</td>\n",

" <td>2.825500</td>\n",

" <td>1.488000</td>\n",

" <td>0.760000</td>\n",

" <td>1.005000</td>\n",

" <td>29.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-fe5421d0-0d2f-4c64-92a0-3ee4369f324a')\"\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-fe5421d0-0d2f-4c64-92a0-3ee4369f324a 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-fe5421d0-0d2f-4c64-92a0-3ee4369f324a');\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": 156

}

]

},

{

"cell\_type": "markdown",

"source": [

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

" Univariate Analysis"

],

"metadata": {

"id": "333Cg9NnkKp4"

}

},

{

"cell\_type": "code",

"source": [

"rows = 2\n",

"cols = 2\n",

"i = 0\n",

"\n",

"plt.figure(figsize=(cols \* 5, rows \* 5))\n",

"\n",

"i += 1\n",

"plt.subplot(rows, cols, i)\n",

"plt.xticks(range(0, 31, 4))\n",

"plt.xlim(0, 30)\n",

"\_ = sns.distplot(data['Rings'], kde=False, bins=range(0, 31, 2))\n",

"\n",

"i += 1\n",

"plt.subplot(rows, cols, i)\n",

"\_ = sns.distplot(data['Rings'])\n",

"\n",

"i += 1\n",

"plt.subplot(rows, cols, i)\n",

"plt.xticks(range(0, 31, 4))\n",

"plt.xlim(0, 30)\n",

"\_ = sns.boxplot(data['Rings'])"

],

"metadata": {

"colab": {

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

"height": 695

},

"id": "HPgaVxJzkKTc",

"outputId": "482a9cef-e33e-4e61-c665-befb55d27d2e"

},

"execution\_count": 157,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).\n",

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

"/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": "display\_data",

"data": {

"text/plain": [

"<Figure size 720x720 with 3 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"Bi - Variate Analysis"

],

"metadata": {

"id": "Hi6d3\_9auZWF"

}

},

{

"cell\_type": "code",

"source": [

"plt.scatter(data.Rings, data.Height)\n"

],

"metadata": {

"colab": {

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

"height": 282

},

"id": "PSNpsD6xj2kI",

"outputId": "a3c5043a-719d-4527-e694-cce77bed23ba"

},

"execution\_count": 158,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.collections.PathCollection at 0x7fd586464910>"

]

},

"metadata": {},

"execution\_count": 158

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

"Multi-Variate Analysis\n"

],

"metadata": {

"id": "iKl7vsMem2Yz"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

"plt.show()"

],

"metadata": {

"colab": {

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

"height": 880

},

"id": "Q1CBs4OIt3D-",

"outputId": "814196b9-e797-4a46-9673-122f5f9ec71d"

},

"execution\_count": 159,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 1440x1080 with 64 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.figure(figsize=(10, 10))\n",

"corr = data.corr()\n",

"\_ = sns.heatmap(corr, annot=True)"

],

"metadata": {

"colab": {

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

"height": 665

},

"id": "Sg0KOlSqm198",

"outputId": "c5eb4f70-6168-460e-d2af-a26ca7209af4"

},

"execution\_count": 160,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 720x720 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"numerical\_features = data.select\_dtypes(include = [np.number]).columns\n",

"categorical\_features = data.select\_dtypes(include = [np.object]).columns\n",

"numerical\_features\n",

"categorical\_features"

],

"metadata": {

"colab": {

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

},

"id": "nsK-axT9yEDA",

"outputId": "3615e806-5b15-4da5-e612-53eca673076e"

},

"execution\_count": 161,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:2: DeprecationWarning: `np.object` is a deprecated alias for the builtin `object`. To silence this warning, use `object` by itself. Doing this will not modify any behavior and is safe. \n",

"Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations\n",

" \n"

]

},

{

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

"data": {

"text/plain": [

"Index(['Sex'], dtype='object')"

]

},

"metadata": {},

"execution\_count": 161

}

]

},

{

"cell\_type": "code",

"source": [

"plt.figure(figsize = (20,7))\n",

"sns.heatmap(data[numerical\_features].corr(),annot = True)"

],

"metadata": {

"colab": {

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

"height": 449

},

"id": "xKWyvStvyOwx",

"outputId": "f6802a94-c34b-4cea-bb59-424d4ab10678"

},

"execution\_count": 162,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fd572f5f550>"

]

},

"metadata": {},

"execution\_count": 162

},

{

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

"data": {

"text/plain": [

"<Figure size 1440x504 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "fPUbdicUnaW7"

}

},

{

"cell\_type": "code",

"source": [

"data.columns"

],

"metadata": {

"colab": {

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

},

"id": "wptJzNiivQks",

"outputId": "3ad91dbc-5a2e-46f5-dc80-634fa89a8d30"

},

"execution\_count": 163,

"outputs": [

{

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

"data": {

"text/plain": [

"Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',\n",

" 'Viscera weight', 'Shell weight', 'Rings'],\n",

" dtype='object')"

]

},

"metadata": {},

"execution\_count": 163

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].mean()"

],

"metadata": {

"colab": {

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

},

"id": "mQukFXMhm9QI",

"outputId": "2b563720-4e6e-4acb-cbbe-c1989b48c556"

},

"execution\_count": 164,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.523992\n",

"Diameter 0.407881\n",

"Height 0.139516\n",

"Whole weight 0.828742\n",

"Shucked weight 0.359367\n",

"Viscera weight 0.180594\n",

"Shell weight 0.238831\n",

"Rings 9.933684\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 164

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].median()"

],

"metadata": {

"colab": {

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

},

"id": "jDKs\_mVfve\_q",

"outputId": "a3231fa7-02a4-4312-8449-f71cb7fe20a0"

},

"execution\_count": 165,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.5450\n",

"Diameter 0.4250\n",

"Height 0.1400\n",

"Whole weight 0.7995\n",

"Shucked weight 0.3360\n",

"Viscera weight 0.1710\n",

"Shell weight 0.2340\n",

"Rings 9.0000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 165

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].mode()"

],

"metadata": {

"colab": {

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

"height": 112

},

"id": "5h6LSXO4vib8",

"outputId": "94cad28b-9ddd-4071-ecae-abf0bb0a3aeb"

},

"execution\_count": 166,

"outputs": [

{

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

"data": {

"text/plain": [

" Length Diameter Height Whole weight Shucked weight Viscera weight \\\n",

"0 0.550 0.45 0.15 0.2225 0.175 0.1715 \n",

"1 0.625 NaN NaN NaN NaN NaN \n",

"\n",

" Shell weight Rings \n",

"0 0.275 9.0 \n",

"1 NaN NaN "

],

"text/html": [

"\n",

" <div id=\"df-dfbf62ad-7b00-4ea2-a79d-11b0dd583c75\">\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>Length</th>\n",

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>0.550</td>\n",

" <td>0.45</td>\n",

" <td>0.15</td>\n",

" <td>0.2225</td>\n",

" <td>0.175</td>\n",

" <td>0.1715</td>\n",

" <td>0.275</td>\n",

" <td>9.0</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.625</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",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-dfbf62ad-7b00-4ea2-a79d-11b0dd583c75')\"\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-dfbf62ad-7b00-4ea2-a79d-11b0dd583c75 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-dfbf62ad-7b00-4ea2-a79d-11b0dd583c75');\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": 166

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].sum()"

],

"metadata": {

"colab": {

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

},

"id": "Iv18NMF5viQm",

"outputId": "545e8c58-deb6-4ea3-f55a-3bacb2564333"

},

"execution\_count": 167,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 2188.7150\n",

"Diameter 1703.7200\n",

"Height 582.7600\n",

"Whole weight 3461.6560\n",

"Shucked weight 1501.0780\n",

"Viscera weight 754.3395\n",

"Shell weight 997.5965\n",

"Rings 41493.0000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 167

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].quantile()"

],

"metadata": {

"colab": {

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

},

"id": "01w9cSOcvpV6",

"outputId": "e055b334-aa9e-42a6-bd29-2776d108f31d"

},

"execution\_count": 168,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.5450\n",

"Diameter 0.4250\n",

"Height 0.1400\n",

"Whole weight 0.7995\n",

"Shucked weight 0.3360\n",

"Viscera weight 0.1710\n",

"Shell weight 0.2340\n",

"Rings 9.0000\n",

"Name: 0.5, dtype: float64"

]

},

"metadata": {},

"execution\_count": 168

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].var()"

],

"metadata": {

"colab": {

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

},

"id": "zsCGu9OGvpSs",

"outputId": "690a6b10-76d0-4e80-dbd1-2e4b769e6837"

},

"execution\_count": 169,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.014422\n",

"Diameter 0.009849\n",

"Height 0.001750\n",

"Whole weight 0.240481\n",

"Shucked weight 0.049268\n",

"Viscera weight 0.012015\n",

"Shell weight 0.019377\n",

"Rings 10.395266\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 169

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].std()"

],

"metadata": {

"colab": {

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

},

"id": "L-b-Y\_NQvpQL",

"outputId": "fbd02515-728b-4ab6-bb53-d2c2a78de8bd"

},

"execution\_count": 170,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.120093\n",

"Diameter 0.099240\n",

"Height 0.041827\n",

"Whole weight 0.490389\n",

"Shucked weight 0.221963\n",

"Viscera weight 0.109614\n",

"Shell weight 0.139203\n",

"Rings 3.224169\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 170

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].skew()"

],

"metadata": {

"id": "3S0spnzJvpMM",

"colab": {

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

},

"outputId": "be471584-0599-46ad-a1f4-9a4a3f51e06c"

},

"execution\_count": 171,

"outputs": [

{

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

"data": {

"text/plain": [

"Length -0.639873\n",

"Diameter -0.609198\n",

"Height 3.128817\n",

"Whole weight 0.530959\n",

"Shucked weight 0.719098\n",

"Viscera weight 0.591852\n",

"Shell weight 0.620927\n",

"Rings 1.114102\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 171

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].min()"

],

"metadata": {

"colab": {

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

},

"id": "LQMURlSSv2CH",

"outputId": "8beafc7e-c1ec-4583-d0c5-b3f4ff9909e1"

},

"execution\_count": 172,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.0750\n",

"Diameter 0.0550\n",

"Height 0.0000\n",

"Whole weight 0.0020\n",

"Shucked weight 0.0010\n",

"Viscera weight 0.0005\n",

"Shell weight 0.0015\n",

"Rings 1.0000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 172

}

]

},

{

"cell\_type": "code",

"source": [

"data[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight','Viscera weight', 'Shell weight', 'Rings']].max()"

],

"metadata": {

"id": "LjDyQFPhv-AP",

"colab": {

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

},

"outputId": "fd955f18-222f-45b4-b9ed-512f25b781b8"

},

"execution\_count": 173,

"outputs": [

{

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

"data": {

"text/plain": [

"Length 0.8150\n",

"Diameter 0.6500\n",

"Height 1.1300\n",

"Whole weight 2.8255\n",

"Shucked weight 1.4880\n",

"Viscera weight 0.7600\n",

"Shell weight 1.0050\n",

"Rings 29.0000\n",

"dtype: float64"

]

},

"metadata": {},

"execution\_count": 173

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 5. Check for Missing values and deal with them."

],

"metadata": {

"id": "x5OaSEnQwIkM"

}

},

{

"cell\_type": "code",

"source": [

"data.isnull().sum()"

],

"metadata": {

"colab": {

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

},

"id": "VVjX8RzdwLjT",

"outputId": "8b80d796-e7ed-4729-b9fd-79049135efdb"

},

"execution\_count": 174,

"outputs": [

{

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

"data": {

"text/plain": [

"Sex 0\n",

"Length 0\n",

"Diameter 0\n",

"Height 0\n",

"Whole weight 0\n",

"Shucked weight 0\n",

"Viscera weight 0\n",

"Shell weight 0\n",

"Rings 0\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 174

}

]

},

{

"cell\_type": "markdown",

"source": [

"There is no missing values"

],

"metadata": {

"id": "80oDeOVPwP6\_"

}

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "00PnLQoxoZpf"

}

},

{

"cell\_type": "code",

"source": [

"data = pd.get\_dummies(data)\n",

"dummy\_data = data.copy()"

],

"metadata": {

"id": "WcG2W9hcyvYd"

},

"execution\_count": 175,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"data.boxplot( rot = 90, figsize=(20,5))"

],

"metadata": {

"id": "um00cijgy1a1",

"colab": {

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

"height": 406

},

"outputId": "c66001e2-f2b4-4da3-b916-c4edd79dded7"

},

"execution\_count": 176,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7fd572f4ba10>"

]

},

"metadata": {},

"execution\_count": 176

},

{

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

"data": {

"text/plain": [

"<Figure size 1440x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data['age'] = data['Rings']+1.5\n",

"\n"

],

"metadata": {

"id": "VzttI8iSOFcT"

},

"execution\_count": 177,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Viscera weight'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 265

},

"id": "khiJqNXwy5CV",

"outputId": "c01868bb-cd14-4db4-8097-09be7d06cfb8"

},

"execution\_count": 178,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Viscera weight']> 0.5) & (data['age'] < 20)].index, inplace=True)\n",

"data.drop(data[(data['Viscera weight']<0.5) & (data['age'] > 25)].index, inplace=True)"

],

"metadata": {

"id": "3HIcy3PyoZBP"

},

"execution\_count": 179,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Shell weight'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 265

},

"id": "kC\_vYvo7zONg",

"outputId": "15027c1e-1a4e-4178-916d-7defb23d854c"

},

"execution\_count": 180,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Shucked weight']>= 1) & (data['age'] < 20)].index, inplace=True)\n",

"data.drop(data[(data['Shucked weight']<1) & (data['age'] > 20)].index, inplace=True)"

],

"metadata": {

"id": "zAmtGcuS0EYZ"

},

"execution\_count": 181,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Whole weight'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 265

},

"id": "vMnLtf1E0IbD",

"outputId": "423a395d-b5b7-4b09-9b3e-713332e25f34"

},

"execution\_count": 182,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Whole weight']>= 2.5) & (data['age'] < 25)].index, inplace=True)\n",

"data.drop(data[(data['Whole weight']<2.5) & (data['age'] > 25)].index, inplace=True)"

],

"metadata": {

"id": "8QpxfAHVoniz"

},

"execution\_count": 183,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Diameter'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 265

},

"id": "ixE6W5k\_0ZF9",

"outputId": "de4829fd-5ebb-476a-9368-07ff6a9f9e64"

},

"execution\_count": 184,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Diameter']<0.1) & (data['age'] < 5)].index, inplace=True)\n",

"data.drop(data[(data['Diameter']<0.6) & (data['age'] > 25)].index, inplace=True)\n",

"data.drop(data[(data['Diameter']>=0.6) & (data['age']< 25)].index, inplace=True)"

],

"metadata": {

"id": "J5o1jtkz0ZAs"

},

"execution\_count": 185,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Height'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 266

},

"id": "JUT3vEE10gqH",

"outputId": "9d744973-ad12-477a-bea1-c87c5e555cde"

},

"execution\_count": 186,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Height']>0.4) & (data['age'] < 15)].index, inplace=True)\n",

"data.drop(data[(data['Height']<0.4) & (data['age'] > 25)].index, inplace=True)"

],

"metadata": {

"id": "bccyHeUv0jnu"

},

"execution\_count": 187,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"var = 'Length'\n",

"plt.scatter(x = data[var], y = data['age'],)\n",

"plt.grid(True)"

],

"metadata": {

"colab": {

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

"height": 266

},

"id": "9CeX1MS20mje",

"outputId": "a78494cc-cfd1-4ccd-fd74-92abd7d02c0e"

},

"execution\_count": 188,

"outputs": [

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"data.drop(data[(data['Length']<0.1) & (data['age'] < 5)].index, inplace=True)\n",

"data.drop(data[(data['Length']<0.8) & (data['age'] > 25)].index, inplace=True)\n",

"data.drop(data[(data['Length']>=0.8) & (data['age']< 25)].index, inplace=True)"

],

"metadata": {

"id": "Vi8b6Q7A0wdW"

},

"execution\_count": 189,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "tXnBoPn1pQ3-"

}

},

{

"cell\_type": "code",

"source": [

"from scipy import stats\n",

"z= np.abs(stats.zscore(data.select\_dtypes(include=[np.number])))\n",

"print(z)"

],

"metadata": {

"colab": {

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

},

"id": "uk5f18nrpoIu",

"outputId": "b9633a1c-388f-4d7e-cac1-209bb58cc119"

},

"execution\_count": 190,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" Length Diameter Height Whole weight Shucked weight \\\n",

"0 0.544555 0.399071 1.128902 0.620330 0.594508 \n",

"1 1.433400 1.424312 1.261866 1.247197 1.198628 \n",

"2 0.090335 0.164811 0.065193 0.266155 0.439854 \n",

"3 0.671532 0.399071 0.331120 0.615984 0.638005 \n",

"4 1.602703 1.526836 1.527793 1.291740 1.246958 \n",

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

"4172 0.386616 0.472383 0.732590 0.190143 0.108688 \n",

"4173 0.598246 0.369859 0.065193 0.361798 0.442162 \n",

"4174 0.682898 0.728693 1.796299 0.818097 0.860213 \n",

"4175 0.894528 0.831217 0.333699 0.641009 0.886794 \n",

"4176 1.614069 1.548885 1.530372 2.496623 2.890056 \n",

"\n",

" Viscera weight Shell weight Rings Sex\_F Sex\_I Sex\_M \\\n",

"0 0.711684 0.611842 1.908736 0.666846 0.704866 1.332557 \n",

"1 1.217824 1.221744 0.952190 0.666846 0.704866 1.332557 \n",

"2 0.321234 0.154415 0.236958 1.499596 0.704866 0.750437 \n",

"3 0.586355 0.573723 0.120657 0.666846 0.704866 1.332557 \n",

"4 1.304590 1.336101 0.952190 0.666846 1.418709 0.750437 \n",

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

"4172 0.618738 0.142913 0.478273 1.499596 0.704866 0.750437 \n",

"4173 0.382540 0.230586 0.120657 0.666846 0.704866 1.332557 \n",

"4174 1.086314 0.592716 0.236958 0.666846 0.704866 1.332557 \n",

"4175 0.830835 0.501230 0.120657 1.499596 0.704866 0.750437 \n",

"4176 1.944341 2.018363 0.835889 0.666846 0.704866 1.332557 \n",

"\n",

" age \n",

"0 1.908736 \n",

"1 0.952190 \n",

"2 0.236958 \n",

"3 0.120657 \n",

"4 0.952190 \n",

"... ... \n",

"4172 0.478273 \n",

"4173 0.120657 \n",

"4174 0.236958 \n",

"4175 0.120657 \n",

"4176 0.835889 \n",

"\n",

"[4022 rows x 12 columns]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"data\_o = data[(z < 3).all(axis=1)]"

],

"metadata": {

"id": "u63ubOsephFR"

},

"execution\_count": 191,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"low\_cardinality\_cols = [cname for cname in data\_o.columns if\n",

" data\_o[cname].nunique() < 10 and \n",

" data\_o[cname].dtype == \"object\"]\n",

"numeric\_cols = [cname for cname in data\_o.columns if\n",

" data\_o[cname].dtype in ['int64','float64']]\n",

"\n",

"my\_cols = low\_cardinality\_cols + numeric\_cols\n",

"data\_predictors = data\_o[my\_cols]"

],

"metadata": {

"id": "Il8cbzBso6dp"

},

"execution\_count": 192,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"print(\"Shape of Abalones with outliers: \"+ str(data.shape) , \n",

" \"Shape of Abalones without outliers: \" + str(data\_o.shape))"

],

"metadata": {

"colab": {

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

},

"id": "y3KApYAFp2Pp",

"outputId": "216597bb-8d9a-4276-cac0-4e98ae21ef88"

},

"execution\_count": 193,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Shape of Abalones with outliers: (4022, 12) Shape of Abalones without outliers: (3973, 12)\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"data\_encoded\_predictors = pd.get\_dummies(data\_predictors)"

],

"metadata": {

"id": "cpPcDBzJqOBe"

},

"execution\_count": 194,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "ys-Vxu\_WxARo"

}

},

{

"cell\_type": "code",

"source": [

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

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

"x.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "2lFbOeyCw\_3G",

"outputId": "08f1b44c-de71-4598-9e31-4bf29896f4d5"

},

"execution\_count": 195,

"outputs": [

{

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

"data": {

"text/plain": [

" Whole weight Shucked weight Viscera weight Shell weight Rings Sex\_F \\\n",

"0 0.5140 0.2245 0.1010 0.150 15 0 \n",

"1 0.2255 0.0995 0.0485 0.070 7 0 \n",

"2 0.6770 0.2565 0.1415 0.210 9 1 \n",

"3 0.5160 0.2155 0.1140 0.155 10 0 \n",

"4 0.2050 0.0895 0.0395 0.055 7 0 \n",

"\n",

" Sex\_I Sex\_M \n",

"0 0 1 \n",

"1 0 1 \n",

"2 0 0 \n",

"3 0 1 \n",

"4 1 0 "

],

"text/html": [

"\n",

" <div id=\"df-be8a9697-2382-4829-bf14-88df1a0881fc\">\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>Whole weight</th>\n",

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

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

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

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

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

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

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>0.5140</td>\n",

" <td>0.2245</td>\n",

" <td>0.1010</td>\n",

" <td>0.150</td>\n",

" <td>15</td>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>0.2255</td>\n",

" <td>0.0995</td>\n",

" <td>0.0485</td>\n",

" <td>0.070</td>\n",

" <td>7</td>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>0.6770</td>\n",

" <td>0.2565</td>\n",

" <td>0.1415</td>\n",

" <td>0.210</td>\n",

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

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>0.5160</td>\n",

" <td>0.2155</td>\n",

" <td>0.1140</td>\n",

" <td>0.155</td>\n",

" <td>10</td>\n",

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>0.2050</td>\n",

" <td>0.0895</td>\n",

" <td>0.0395</td>\n",

" <td>0.055</td>\n",

" <td>7</td>\n",

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

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

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-be8a9697-2382-4829-bf14-88df1a0881fc')\"\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-be8a9697-2382-4829-bf14-88df1a0881fc 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-be8a9697-2382-4829-bf14-88df1a0881fc');\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": 195

}

]

},

{

"cell\_type": "code",

"source": [

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

"y = data.iloc[:, 3:13].values"

],

"metadata": {

"id": "YSB8wxJ51Yd1"

},

"execution\_count": 196,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

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

"\n",

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

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

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

"print(y\_test.shape)"

],

"metadata": {

"colab": {

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

},

"id": "e7EnBVYo1m2b",

"outputId": "968af537-1ee3-48ff-8ac9-69cd238ab3ab"

},

"execution\_count": 197,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(3016, 9)\n",

"(3016, 9)\n",

"(1006, 9)\n",

"(1006, 9)\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 9. Scale the independent variables\n"

],

"metadata": {

"id": "qxHd4TtqxZQM"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.preprocessing import StandardScaler\n",

"\n",

"sc = StandardScaler()\n",

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

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

"\n",

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

"x\_train.head()"

],

"metadata": {

"colab": {

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

"height": 206

},

"id": "l6QqmVPO2GjW",

"outputId": "edcaa6c5-af98-46f6-b0fe-25be8cb758c4"

},

"execution\_count": 198,

"outputs": [

{

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

"data": {

"text/plain": [

" 0 1 2 3 4 5 6 \\\n",

"0 -0.769217 -0.763992 -0.861091 -0.660073 0.846424 -0.681232 1.429472 \n",

"1 -1.469024 -1.471684 -1.374250 -1.431125 -0.236960 -0.681232 -0.699559 \n",

"2 1.559468 1.118323 2.275953 1.210298 1.929808 -0.681232 -0.699559 \n",

"3 1.245047 1.254511 1.399711 1.248469 0.846424 -0.681232 -0.699559 \n",

"4 0.546332 0.213645 1.554627 0.408710 0.846424 1.467928 -0.699559 \n",

"\n",

" 7 8 \n",

"0 -0.740979 0.846424 \n",

"1 1.349566 -0.236960 \n",

"2 1.349566 1.929808 \n",

"3 1.349566 0.846424 \n",

"4 -0.740979 0.846424 "

],

"text/html": [

"\n",

" <div id=\"df-30f46d54-44d3-439d-baff-00c087578576\">\n",

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

" <div>\n",

"<style scoped>\n",

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

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

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

" <thead>\n",

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

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

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

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

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

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

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

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

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

" <th>7</th>\n",

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

" <td>-0.769217</td>\n",

" <td>-0.763992</td>\n",

" <td>-0.861091</td>\n",

" <td>-0.660073</td>\n",

" <td>0.846424</td>\n",

" <td>-0.681232</td>\n",

" <td>1.429472</td>\n",

" <td>-0.740979</td>\n",

" <td>0.846424</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>-1.469024</td>\n",

" <td>-1.471684</td>\n",

" <td>-1.374250</td>\n",

" <td>-1.431125</td>\n",

" <td>-0.236960</td>\n",

" <td>-0.681232</td>\n",

" <td>-0.699559</td>\n",

" <td>1.349566</td>\n",

" <td>-0.236960</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>1.559468</td>\n",

" <td>1.118323</td>\n",

" <td>2.275953</td>\n",

" <td>1.210298</td>\n",

" <td>1.929808</td>\n",

" <td>-0.681232</td>\n",

" <td>-0.699559</td>\n",

" <td>1.349566</td>\n",

" <td>1.929808</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>1.245047</td>\n",

" <td>1.254511</td>\n",

" <td>1.399711</td>\n",

" <td>1.248469</td>\n",

" <td>0.846424</td>\n",

" <td>-0.681232</td>\n",

" <td>-0.699559</td>\n",

" <td>1.349566</td>\n",

" <td>0.846424</td>\n",

" </tr>\n",

" <tr>\n",

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

" <td>0.546332</td>\n",

" <td>0.213645</td>\n",

" <td>1.554627</td>\n",

" <td>0.408710</td>\n",

" <td>0.846424</td>\n",

" <td>1.467928</td>\n",

" <td>-0.699559</td>\n",

" <td>-0.740979</td>\n",

" <td>0.846424</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-30f46d54-44d3-439d-baff-00c087578576')\"\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-30f46d54-44d3-439d-baff-00c087578576 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-30f46d54-44d3-439d-baff-00c087578576');\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": 198

}

]

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "SfhsqYhwrx1E"

}

},

{

"cell\_type": "code",

"source": [

"train, test = train\_test\_split(data, test\_size=0.25, random\_state=1)\n",

"print('Train data points :', len(train))\n",

"print('Test data points :', len(test))"

],

"metadata": {

"colab": {

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

},

"id": "nMTIbZI61CYw",

"outputId": "6c202fdf-d398-4614-9cc3-06beea5cae19"

},

"execution\_count": 199,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Train data points : 3016\n",

"Test data points : 1006\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

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

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

"\n",

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

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

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

"print(y\_test.shape)"

],

"metadata": {

"colab": {

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

},

"id": "fr3FTv022DgH",

"outputId": "f4722eaa-7045-4332-a399-c64455dc7401"

},

"execution\_count": 200,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(3016, 9)\n",

"(3016, 9)\n",

"(1006, 9)\n",

"(1006, 9)\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 11. Build the Model"

],

"metadata": {

"id": "Kr5kCUS32Sp7"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.ensemble import RandomForestRegressor\n",

"\n",

"# instantiate model\n",

"rf = RandomForestRegressor(n\_jobs=-1, #n\_jobs=-1 means that we are using all computer power to fit the model\n",

" random\_state=14)\n",

"\n",

"# fit the model\n",

"rf.fit(x\_train, y\_train)"

],

"metadata": {

"id": "aq3VIkoQ7F0p",

"colab": {

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

},

"outputId": "17fdda1e-ed14-4815-9e3b-092b922eff29"

},

"execution\_count": 201,

"outputs": [

{

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

"data": {

"text/plain": [

"RandomForestRegressor(n\_jobs=-1, random\_state=14)"

]

},

"metadata": {},

"execution\_count": 201

}

]

},

{

"cell\_type": "code",

"source": [

"from sklearn.linear\_model import LinearRegression\n",

"from sklearn.linear\_model import Lasso\n",

"models = {'linear\_regression':LinearRegression(),\n",

" \n",

" 'lasso':Lasso(random\_state=1),\n",

" \n",

" 'decision\_tree':DecisionTreeRegressor(random\_state=1),\n",

" \n",

" 'random\_forest':RandomForestRegressor(random\_state=1),\n",

" \n",

" 'xgboost':XGBRegressor(random\_state=1),\n",

" }"

],

"metadata": {

"id": "4FqWavN73Xde"

},

"execution\_count": 202,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"rf\_params = {'n\_estimators': 200, \n",

" 'min\_samples\_split': 2,\n",

" 'min\_samples\_leaf': 4, \n",

" 'max\_features': 'sqrt', \n",

" 'max\_depth': None, \n",

" 'bootstrap': True}\n",

"\n",

"model = RandomForestRegressor(random\_state=1, \*\*rf\_params)\n",

"\n",

"model.fit(x\_train, y\_train)"

],

"metadata": {

"colab": {

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

},

"id": "UvCy2aOt63wN",

"outputId": "965a8542-a7c3-459e-f906-f22830106230"

},

"execution\_count": 203,

"outputs": [

{

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

"data": {

"text/plain": [

"RandomForestRegressor(max\_features='sqrt', min\_samples\_leaf=4, n\_estimators=200,\n",

" random\_state=1)"

]

},

"metadata": {},

"execution\_count": 203

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 12. Train the Model"

],

"metadata": {

"id": "PcaNV\_5io6Hh"

}

},

{

"cell\_type": "code",

"source": [

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

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

"train\_X,val\_X,train\_y,val\_y = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)"

],

"metadata": {

"id": "YZmqxghjq476"

},

"execution\_count": 204,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"print(\"Shape of Training X :\",train\_X.shape)\n",

"print(\"Shape of Validation X :\",val\_X.shape)\n",

"print(\"Shape of Training y :\",train\_y.shape)\n",

"print(\"Shape of Validation y :\",val\_y.shape)"

],

"metadata": {

"colab": {

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

},

"id": "tWDMCLXaNT7M",

"outputId": "29450997-b4a8-422d-b533-71cd86ad7f13"

},

"execution\_count": 205,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Shape of Training X : (3217, 11)\n",

"Shape of Validation X : (805, 11)\n",

"Shape of Training y : (3217,)\n",

"Shape of Validation y : (805,)\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"lr = LinearRegression()\n",

"lr.fit(train\_X,train\_y)\n",

"print('Attempting to fit Linear Regressor')"

],

"metadata": {

"colab": {

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

},

"id": "BpsRuXLxNZyl",

"outputId": "fd7491c7-250b-4f02-fc98-28620bf480e6"

},

"execution\_count": 206,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Attempting to fit Linear Regressor\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"%%time\n",

"y\_pred\_val\_lr = lr.predict(val\_X)\n",

"print('MAE on Validation set :',metrics.mean\_absolute\_error(val\_y, y\_pred\_val\_lr))\n",

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

"print('MSE on Validation set :',metrics.mean\_squared\_error(val\_y, y\_pred\_val\_lr))\n",

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

"print('RMSE on Validation set :',np.sqrt(metrics.mean\_absolute\_error(val\_y, y\_pred\_val\_lr)))\n",

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

"print('R2 Score on Validation set :',metrics.r2\_score(val\_y, y\_pred\_val\_lr))\n",

"print(\"\\n\")"

],

"metadata": {

"colab": {

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

},

"id": "tY\_W5\_XnNdvp",

"outputId": "e6c1ed3d-2517-41ea-d866-22f2d0b31aaf"

},

"execution\_count": 207,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"MAE on Validation set : 1.1546319456101628e-15\n",

"\n",

"\n",

"MSE on Validation set : 3.1821472974893545e-30\n",

"\n",

"\n",

"RMSE on Validation set : 3.397987559733206e-08\n",

"\n",

"\n",

"R2 Score on Validation set : 1.0\n",

"\n",

"\n",

"CPU times: user 6.83 ms, sys: 4.96 ms, total: 11.8 ms\n",

"Wall time: 12.7 ms\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 13. Test the Model"

],

"metadata": {

"id": "XBw0Bly7slH0"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

"import numpy\n",

"from sklearn.metrics import r2\_score\n",

"numpy.random.seed(2)\n",

"x = numpy.random.normal(3, 1, 100)\n",

"y = numpy.random.normal(150, 40, 100) / x\n",

"train\_x = x[:80]\n",

"train\_y = y[:80]\n",

"\n",

"test\_x = x[80:]\n",

"test\_y = y[80:]\n",

"\n",

"mymodel = numpy.poly1d(numpy.polyfit(train\_x, train\_y, 4))\n",

"\n",

"r2 = r2\_score(test\_y, mymodel(test\_x))\n",

"\n",

"print(r2)"

],

"metadata": {

"colab": {

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

},

"id": "dk9bC5MQkM9B",

"outputId": "042d4e50-f7bf-486f-dac6-f509433cde86"

},

"execution\_count": 208,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"0.8086921460343566\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# 14. Measure the performance using Metrics."

],

"metadata": {

"id": "zvcnd5VNkfJK"

}

},

{

"cell\_type": "code",

"source": [

"from sklearn.metrics import mean\_absolute\_error, r2\_score, mean\_squared\_log\_error\n",

"\n",

"# create an evaluation function\n",

"def show\_score(model):\n",

" train\_preds= model.predict(x\_train)\n",

" test\_preds = model.predict(x\_test)\n",

" scores = {\"Training MAE\": mean\_absolute\_error(y\_train, train\_preds),\n",

" \"Test MAE\": mean\_absolute\_error(y\_test, test\_preds),\n",

" \"Training MSE\": mean\_squared\_log\_error(y\_train, train\_preds),\n",

" \"Test MSE\": mean\_squared\_log\_error(y\_test, test\_preds),\n",

" \"Training RMSE\": np.sqrt(mean\_squared\_log\_error(y\_train, train\_preds)),\n",

" \"Test RMSE\": np.sqrt(mean\_squared\_log\_error(y\_test, test\_preds)),\n",

" \"Training R2\": r2\_score(y\_train, train\_preds),\n",

" \"Test R2\": r2\_score(y\_test, test\_preds)}\n",

" return scores\n",

"# fit "

],

"metadata": {

"id": "AGRc6gy0k8di"

},

"execution\_count": 209,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"rf.fit(x\_train, y\_train)"

],

"metadata": {

"id": "XgtI7-2pC2Mc",

"colab": {

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

},

"outputId": "5b39dc1c-d2c8-432e-ad47-e0ce65d571b9"

},

"execution\_count": 210,

"outputs": [

{

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

"data": {

"text/plain": [

"RandomForestRegressor(n\_jobs=-1, random\_state=14)"

]

},

"metadata": {},

"execution\_count": 210

}

]

},

{

"cell\_type": "code",

"source": [

"show\_score(rf)"

],

"metadata": {

"colab": {

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

},

"id": "j5DfRZPNDFGG",

"outputId": "46beab84-3fc2-4623-fc79-c1601daa1957"

},

"execution\_count": 211,

"outputs": [

{

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

"data": {

"text/plain": [

"{'Training MAE': 0.002969515362511056,\n",

" 'Test MAE': 0.008333732052131658,\n",

" 'Training MSE': 3.377316030763978e-05,\n",

" 'Test MSE': 0.00022606804480543698,\n",

" 'Training RMSE': 0.0058114679993646855,\n",

" 'Test RMSE': 0.015035559344614919,\n",

" 'Training R2': 0.9977286616319563,\n",

" 'Test R2': 0.984112781560057}"

]

},

"metadata": {},

"execution\_count": 211

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "qLuaR6GnMpaD"

},

"execution\_count": 211,

"outputs": []

}

]

}