NAME: G.Sindhu

CLASS: 4th year ECE

SUBJECT: IBM

REGISTER NO: 611419106061

{

"cells": [

{

"cell\_type": "markdown",

"id": "0736c0c4",

"metadata": {},

"source": [

"# 1 DOWNLOAD THE DATASET \n",

"# 2 LOAD THE DATASET"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "66f84b03",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<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>0.150</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>0.070</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>0.210</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>0.155</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>0.055</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395 \n",

"\n",

" Shell weight Rings \n",

"0 0.150 15 \n",

"1 0.070 7 \n",

"2 0.210 9 \n",

"3 0.155 10 \n",

"4 0.055 7 "

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"df =pd.read\_csv('/Users/ELCOT/Desktop/Assignment\_3/abalone.csv')\n",

"df.head()"

]

},

{

"cell\_type": "markdown",

"id": "1da4739b",

"metadata": {},

"source": [

"# 3. PERFORM BELOW VISUALIZATIONS\n"

]

},

{

"cell\_type": "markdown",

"id": "0af1d50c",

"metadata": {},

"source": [

"# Univariate Analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "c6b514b6",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Height', ylabel='Count'>"

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"import seaborn as sns\n",

"sns.histplot(df.Height,kde=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "fec5af35",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Rings', ylabel='Count'>"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"import seaborn as sns\n",

"sns.histplot(df.Rings,kde=True)"

]

},

{

"cell\_type": "markdown",

"id": "ad18b0dd",

"metadata": {},

"source": [

"## Bivariate analysis"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "654b1752",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\ELCOT\\anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. 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",

" warnings.warn(\n"

]

},

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='Length', ylabel='Height'>"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"sns.scatterplot(df.Length,df.Height)"

]

},

{

"cell\_type": "markdown",

"id": "7a69e7cb",

"metadata": {},

"source": [

"## MULTIVARIATE ANALYSIS"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "656aa7f8",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<seaborn.axisgrid.PairGrid at 0x144f2cd9e80>"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 1440x1440 with 72 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"import seaborn as sns\n",

"df= pd.read\_csv(\"/Users/ELCOT/Desktop/Assignment\_3/abalone.csv\")\n",

"sns.pairplot(df)"

]

},

{

"cell\_type": "markdown",

"id": "885dbdb7",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "4f69f362",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

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

" <td>4177</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",

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>1528</td>\n",

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

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

" <td>NaN</td>\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>NaN</td>\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>NaN</td>\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>NaN</td>\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>NaN</td>\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>NaN</td>\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>NaN</td>\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>"

],

"text/plain": [

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

"count 4177 4177.000000 4177.000000 4177.000000 4177.000000 \n",

"unique 3 NaN NaN NaN NaN \n",

"top M NaN NaN NaN NaN \n",

"freq 1528 NaN NaN NaN NaN \n",

"mean NaN 0.523992 0.407881 0.139516 0.828742 \n",

"std NaN 0.120093 0.099240 0.041827 0.490389 \n",

"min NaN 0.075000 0.055000 0.000000 0.002000 \n",

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

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

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

"max NaN 0.815000 0.650000 1.130000 2.825500 \n",

"\n",

" Shucked weight Viscera weight Shell weight Rings \n",

"count 4177.000000 4177.000000 4177.000000 4177.000000 \n",

"unique NaN NaN NaN NaN \n",

"top NaN NaN NaN NaN \n",

"freq NaN NaN NaN NaN \n",

"mean 0.359367 0.180594 0.238831 9.933684 \n",

"std 0.221963 0.109614 0.139203 3.224169 \n",

"min 0.001000 0.000500 0.001500 1.000000 \n",

"25% 0.186000 0.093500 0.130000 8.000000 \n",

"50% 0.336000 0.171000 0.234000 9.000000 \n",

"75% 0.502000 0.253000 0.329000 11.000000 \n",

"max 1.488000 0.760000 1.005000 29.000000 "

]

},

"execution\_count": 25,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df=pd.read\_csv(\"/Users/ELCOT/Desktop/Assignment\_3/abalone.csv\")\n",

"df.describe(include='all')"

]

},

{

"cell\_type": "markdown",

"id": "3b3a709a",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "05f3803a",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<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>0.150</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>0.070</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>0.210</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>0.155</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>0.055</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395 \n",

"\n",

" Shell weight Rings \n",

"0 0.150 15 \n",

"1 0.070 7 \n",

"2 0.210 9 \n",

"3 0.155 10 \n",

"4 0.055 7 "

]

},

"execution\_count": 27,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"from ast import increment\_lineno\n",

"import pandas as pd\n",

"import numpy as np\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"%matplotlib inline\n",

"sns.set(color\_codes=True)\n",

"df=pd.read\_csv(\"/Users/ELCOT/Desktop/Assignment\_3/abalone.csv\")\n",

"df.head()"

]

},

{

"cell\_type": "markdown",

"id": "118936fb",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"id": "ea189b97",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<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>3980</th>\n",

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

" <td>0.525</td>\n",

" <td>0.410</td>\n",

" <td>0.115</td>\n",

" <td>0.7745</td>\n",

" <td>0.4160</td>\n",

" <td>0.1630</td>\n",

" <td>0.1800</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>2419</th>\n",

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

" <td>0.465</td>\n",

" <td>0.360</td>\n",

" <td>0.120</td>\n",

" <td>0.4765</td>\n",

" <td>0.1920</td>\n",

" <td>0.1125</td>\n",

" <td>0.1600</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>3288</th>\n",

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

" <td>0.605</td>\n",

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

" <td>0.145</td>\n",

" <td>1.0185</td>\n",

" <td>0.4695</td>\n",

" <td>0.2250</td>\n",

" <td>0.2700</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>503</th>\n",

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

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

" <td>0.505</td>\n",

" <td>0.190</td>\n",

" <td>1.1290</td>\n",

" <td>0.4385</td>\n",

" <td>0.2560</td>\n",

" <td>0.3600</td>\n",

" <td>13</td>\n",

" </tr>\n",

" <tr>\n",

" <th>798</th>\n",

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

" <td>0.520</td>\n",

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

" <td>0.160</td>\n",

" <td>0.7450</td>\n",

" <td>0.2550</td>\n",

" <td>0.1570</td>\n",

" <td>0.2885</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>3881</th>\n",

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

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

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

" <td>0.130</td>\n",

" <td>1.0580</td>\n",

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

" <td>0.2645</td>\n",

" <td>0.3000</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>1234</th>\n",

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

" <td>0.375</td>\n",

" <td>0.290</td>\n",

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

" <td>0.2130</td>\n",

" <td>0.0960</td>\n",

" <td>0.0410</td>\n",

" <td>0.0610</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>2216</th>\n",

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

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

" <td>0.260</td>\n",

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

" <td>0.1900</td>\n",

" <td>0.0765</td>\n",

" <td>0.0385</td>\n",

" <td>0.0650</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>2200</th>\n",

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

" <td>0.650</td>\n",

" <td>0.515</td>\n",

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

" <td>1.4005</td>\n",

" <td>0.5195</td>\n",

" <td>0.3600</td>\n",

" <td>0.4400</td>\n",

" <td>13</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3169</th>\n",

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

" <td>0.560</td>\n",

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

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

" <td>0.9125</td>\n",

" <td>0.3595</td>\n",

" <td>0.2710</td>\n",

" <td>0.3500</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"3980 F 0.525 0.410 0.115 0.7745 0.4160 \n",

"2419 F 0.465 0.360 0.120 0.4765 0.1920 \n",

"3288 F 0.605 0.475 0.145 1.0185 0.4695 \n",

"503 F 0.600 0.505 0.190 1.1290 0.4385 \n",

"798 M 0.520 0.420 0.160 0.7450 0.2550 \n",

"3881 F 0.565 0.455 0.130 1.0580 0.4390 \n",

"1234 I 0.375 0.290 0.095 0.2130 0.0960 \n",

"2216 I 0.330 0.260 0.080 0.1900 0.0765 \n",

"2200 F 0.650 0.515 0.195 1.4005 0.5195 \n",

"3169 M 0.560 0.450 0.155 0.9125 0.3595 \n",

"\n",

" Viscera weight Shell weight Rings \n",

"3980 0.1630 0.1800 7 \n",

"2419 0.1125 0.1600 10 \n",

"3288 0.2250 0.2700 15 \n",

"503 0.2560 0.3600 13 \n",

"798 0.1570 0.2885 11 \n",

"3881 0.2645 0.3000 10 \n",

"1234 0.0410 0.0610 5 \n",

"2216 0.0385 0.0650 7 \n",

"2200 0.3600 0.4400 13 \n",

"3169 0.2710 0.3500 10 "

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"import pandas as pd\n",

"import matplotlib\n",

"from matplotlib import pyplot as pyplot\n",

"%matplotlib inline\n",

"matplotlib.rcParams['figure.figsize']=(11,6)\n",

"df=pd.read\_csv(\"/Users/ELCOT/Desktop/Assignment\_3/abalone.csv\")\n",

"df.sample(10)"

]

},

{

"cell\_type": "markdown",

"id": "5a071e9c",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"id": "ff47c41d",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<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>0.150</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>0.070</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>0.210</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>0.155</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>0.055</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

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

"0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010 \n",

"1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485 \n",

"2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415 \n",

"3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140 \n",

"4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395 \n",

"\n",

" Shell weight Rings \n",

"0 0.150 15 \n",

"1 0.070 7 \n",

"2 0.210 9 \n",

"3 0.155 10 \n",

"4 0.055 7 "

]

},

"execution\_count": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df=pd.read\_csv(\"/Users/ELCOT/Desktop/Assignment\_3/abalone.csv\")\n",

"df.columns\n",

"import pandas as pd\n",

"import numpy as np\n",

"headers=['Sex','Length','Diameter','Height','Whole weight','Shucked weight','Viscera weight','Shell weight','Rings']\n",

"import seaborn as sns\n",

"df.head()"

]

},

{

"cell\_type": "markdown",

"id": "8287163e",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"id": "306a9016",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[['M' 0.455 0.365 ... 0.2245 0.101 0.15]\n",

" ['M' 0.35 0.265 ... 0.0995 0.0485 0.07]\n",

" ['F' 0.53 0.42 ... 0.2565 0.1415 0.21]\n",

" ...\n",

" ['M' 0.6 0.475 ... 0.5255 0.2875 0.308]\n",

" ['F' 0.625 0.485 ... 0.531 0.261 0.296]\n",

" ['M' 0.71 0.555 ... 0.9455 0.3765 0.495]]\n",

"[15 7 9 ... 9 10 12]\n"

]

}

],

"source": [

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

"print(x)\n",

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

"print(y)"

]

},

{

"cell\_type": "markdown",

"id": "87ebdb31",

"metadata": {},

"source": [

"# 9. Scale the independent variables"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "84264fa5",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 792x432 with 2 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"import seaborn as sns\n",

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

"dff=df[['Length','Height']]\n",

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

"sns.set(rc={'figure.figsize':(40,40)})"

]

},

{

"cell\_type": "markdown",

"id": "79abb718",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "7f80c909",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Row count of x\_train table-3,341\n",

"Row count of y\_train table-3,341\n",

"Row count of x\_test table-836\n",

"Row count of y\_test table-836\n"

]

}

],

"source": [

"from scipy.sparse.construct import random\n",

"x=df.iloc[:, 1:2].values\n",

"y=df.iloc[:,2].values\n",

"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)\n",

"print('Row count of x\_train table'+'-'+str(f\"{len(x\_train):,}\"))\n",

"print('Row count of y\_train table'+'-'+str(f\"{len(y\_train):,}\"))\n",

"print('Row count of x\_test table'+'-'+str(f\"{len(x\_test):,}\"))\n",

"print('Row count of y\_test table'+'-'+str(f\"{len(y\_test):,}\"))"

]

},

{

"cell\_type": "markdown",

"id": "892953f5",

"metadata": {},

"source": [

"# 11. Build the Model"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "0449c806",

"metadata": {},

"outputs": [],

"source": [

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

"model=LinearRegression()"

]

},

{

"cell\_type": "markdown",

"id": "977d097e",

"metadata": {},

"source": [

"# 12. Train the Model"

]

},

{

"cell\_type": "code",

"execution\_count": 44,

"id": "1d6f578e",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"LinearRegression()"

]

},

"execution\_count": 44,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

]

},

{

"cell\_type": "markdown",

"id": "adc30ebd",

"metadata": {},

"source": [

"# 13. Test the Model"

]

},

{

"cell\_type": "code",

"execution\_count": 45,

"id": "ba5de756",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([0.42910815, 0.38837123, 0.48613984, 0.16024445, 0.50650831,\n",

" 0.510582 , 0.35985538, 0.37615015, 0.31097107, 0.48613984,\n",

" 0.36800276, 0.22134984, 0.31504476, 0.39244492, 0.16839184,\n",

" 0.45355031, 0.26616045, 0.44132923, 0.47799246, 0.35985538,\n",

" 0.29874999, 0.27838153, 0.35578169, 0.26208676, 0.43725554,\n",

" 0.43725554, 0.10728645, 0.49836092, 0.44540292, 0.41688707,\n",

" 0.29060261, 0.11136014, 0.457624 , 0.57983477, 0.33948692,\n",

" 0.4005923 , 0.36800276, 0.48206615, 0.33541323, 0.49836092,\n",

" 0.44132923, 0.38022384, 0.404666 , 0.41688707, 0.46984508,\n",

" 0.35985538, 0.47799246, 0.52687677, 0.41688707, 0.31911846,\n",

" 0.457624 , 0.30282369, 0.33541323, 0.46577138, 0.39244492,\n",

" 0.36392907, 0.1887603 , 0.25801307, 0.31504476, 0.28652892,\n",

" 0.48206615, 0.48613984, 0.457624 , 0.38022384, 0.31097107,\n",

" 0.510582 , 0.55131892, 0.50650831, 0.42096077, 0.53095046,\n",

" 0.40873969, 0.57576108, 0.49428723, 0.457624 , 0.46169769,\n",

" 0.38837123, 0.45355031, 0.40873969, 0.47799246, 0.38022384,\n",

" 0.48613984, 0.20912876, 0.31097107, 0.44947661, 0.43725554,\n",

" 0.36800276, 0.47799246, 0.50650831, 0.12358122, 0.28245522,\n",

" 0.45355031, 0.44132923, 0.35578169, 0.27023415, 0.41688707,\n",

" 0.21727615, 0.404666 , 0.28245522, 0.404666 , 0.56761369,\n",

" 0.48206615, 0.36392907, 0.53502415, 0.49021354, 0.15617076,\n",

" 0.50243461, 0.38837123, 0.27838153, 0.27023415, 0.48206615,\n",

" 0.56761369, 0.55946631, 0.38837123, 0.44947661, 0.45355031,\n",

" 0.56354 , 0.51465569, 0.46169769, 0.20912876, 0.41688707,\n",

" 0.27838153, 0.42503446, 0.28652892, 0.36800276, 0.44947661,\n",

" 0.48613984, 0.47391877, 0.53095046, 0.31911846, 0.49836092,\n",

" 0.48206615, 0.36800276, 0.39244492, 0.510582 , 0.53502415,\n",

" 0.48613984, 0.39651861, 0.4005923 , 0.28652892, 0.510582 ,\n",

" 0.35985538, 0.351708 , 0.26616045, 0.44540292, 0.46169769,\n",

" 0.44132923, 0.35578169, 0.38429753, 0.37207646, 0.2417183 ,\n",

" 0.38429753, 0.53502415, 0.2946763 , 0.49836092, 0.25801307,\n",

" 0.28245522, 0.46984508, 0.34356061, 0.46577138, 0.23764461,\n",

" 0.38837123, 0.25393938, 0.41281338, 0.46984508, 0.17653922,\n",

" 0.46984508, 0.27838153, 0.2417183 , 0.42503446, 0.43318184,\n",

" 0.44947661, 0.15209707, 0.35578169, 0.49021354, 0.49021354,\n",

" 0.49836092, 0.18468661, 0.3476343 , 0.28652892, 0.43318184,\n",

" 0.40873969, 0.47391877, 0.404666 , 0.42096077, 0.43318184,\n",

" 0.41688707, 0.457624 , 0.38429753, 0.24986568, 0.38837123,\n",

" 0.45355031, 0.4005923 , 0.38837123, 0.48206615, 0.43318184,\n",

" 0.35578169, 0.31504476, 0.40873969, 0.12358122, 0.53502415,\n",

" 0.36392907, 0.38837123, 0.56761369, 0.43318184, 0.47799246,\n",

" 0.47799246, 0.20912876, 0.52280308, 0.404666 , 0.3476343 ,\n",

" 0.1887603 , 0.44947661, 0.42910815, 0.46577138, 0.47391877,\n",

" 0.44132923, 0.46169769, 0.52280308, 0.1887603 , 0.54724523,\n",

" 0.404666 , 0.510582 , 0.37615015, 0.50243461, 0.50650831,\n",

" 0.42910815, 0.48613984, 0.40873969, 0.60835062, 0.49021354,\n",

" 0.42503446, 0.40873969, 0.2417183 , 0.53502415, 0.4005923 ,\n",

" 0.46169769, 0.40873969, 0.17653922, 0.40873969, 0.43318184,\n",

" 0.39244492, 0.46577138, 0.42910815, 0.55539262, 0.42096077,\n",

" 0.46577138, 0.48206615, 0.36800276, 0.48206615, 0.51465569,\n",

" 0.49836092, 0.37615015, 0.53502415, 0.56761369, 0.43318184,\n",

" 0.37615015, 0.28652892, 0.38429753, 0.47799246, 0.51465569,\n",

" 0.49021354, 0.36392907, 0.44540292, 0.24986568, 0.41281338,\n",

" 0.36800276, 0.37207646, 0.351708 , 0.31504476, 0.26616045,\n",

" 0.44132923, 0.32319215, 0.27430784, 0.31504476, 0.22542353,\n",

" 0.50243461, 0.47391877, 0.351708 , 0.49428723, 0.34356061,\n",

" 0.2417183 , 0.58390846, 0.26208676, 0.50243461, 0.51872938,\n",

" 0.45355031, 0.35985538, 0.48206615, 0.43725554, 0.46577138,\n",

" 0.37615015, 0.404666 , 0.22542353, 0.45355031, 0.28245522,\n",

" 0.37207646, 0.25801307, 0.20098138, 0.46169769, 0.50243461,\n",

" 0.351708 , 0.37207646, 0.54317154, 0.41688707, 0.35578169,\n",

" 0.32726584, 0.39651861, 0.44540292, 0.44540292, 0.45355031,\n",

" 0.50243461, 0.510582 , 0.35985538, 0.48206615, 0.44540292,\n",

" 0.43725554, 0.38837123, 0.27023415, 0.351708 , 0.41688707,\n",

" 0.52687677, 0.43725554, 0.33948692, 0.54724523, 0.44947661,\n",

" 0.41688707, 0.41281338, 0.42096077, 0.48613984, 0.49021354,\n",

" 0.39651861, 0.54317154, 0.15617076, 0.52687677, 0.45355031,\n",

" 0.53502415, 0.29874999, 0.37615015, 0.49836092, 0.42910815,\n",

" 0.37207646, 0.38837123, 0.50243461, 0.48613984, 0.43318184,\n",

" 0.351708 , 0.21727615, 0.39244492, 0.44132923, 0.29874999,\n",

" 0.20912876, 0.44947661, 0.52687677, 0.58390846, 0.47799246,\n",

" 0.38022384, 0.48206615, 0.49021354, 0.35578169, 0.39651861,\n",

" 0.37615015, 0.47799246, 0.37207646, 0.46577138, 0.49836092,\n",

" 0.20505507, 0.20505507, 0.404666 , 0.42503446, 0.38837123,\n",

" 0.457624 , 0.30282369, 0.48613984, 0.351708 , 0.52280308,\n",

" 0.49428723, 0.50243461, 0.33541323, 0.38837123, 0.11950753,\n",

" 0.38022384, 0.46169769, 0.46169769, 0.47799246, 0.20505507,\n",

" 0.51465569, 0.57168739, 0.36392907, 0.32319215, 0.22949722,\n",

" 0.54724523, 0.46577138, 0.42910815, 0.37207646, 0.52280308,\n",

" 0.30282369, 0.50243461, 0.37615015, 0.44540292, 0.38837123,\n",

" 0.34356061, 0.457624 , 0.28245522, 0.54317154, 0.33133953,\n",

" 0.55131892, 0.20912876, 0.50650831, 0.51465569, 0.38429753,\n",

" 0.38837123, 0.36800276, 0.53502415, 0.42910815, 0.50650831,\n",

" 0.44132923, 0.45355031, 0.47391877, 0.55539262, 0.54724523,\n",

" 0.44132923, 0.42096077, 0.44132923, 0.50243461, 0.21727615,\n",

" 0.44540292, 0.42910815, 0.50650831, 0.47799246, 0.43725554,\n",

" 0.41688707, 0.51872938, 0.39244492, 0.44540292, 0.33948692,\n",

" 0.36800276, 0.53502415, 0.42910815, 0.44540292, 0.49021354,\n",

" 0.31504476, 0.25393938, 0.44132923, 0.32319215, 0.28652892,\n",

" 0.28652892, 0.45355031, 0.18468661, 0.39244492, 0.50243461,\n",

" 0.10728645, 0.27430784, 0.49428723, 0.26616045, 0.20505507,\n",

" 0.31911846, 0.48206615, 0.42910815, 0.35578169, 0.47391877,\n",

" 0.31911846, 0.60020323, 0.39244492, 0.45355031, 0.36392907,\n",

" 0.55131892, 0.2417183 , 0.57576108, 0.33948692, 0.37207646,\n",

" 0.45355031, 0.41281338, 0.35578169, 0.49428723, 0.41281338,\n",

" 0.351708 , 0.33541323, 0.33948692, 0.27023415, 0.49836092,\n",

" 0.4005923 , 0.51465569, 0.34356061, 0.41688707, 0.50650831,\n",

" 0.51872938, 0.43725554, 0.39651861, 0.40873969, 0.50650831,\n",

" 0.11543383, 0.43318184, 0.43725554, 0.43725554, 0.25393938,\n",

" 0.30689738, 0.30689738, 0.22134984, 0.44947661, 0.49428723,\n",

" 0.29060261, 0.49836092, 0.30282369, 0.45355031, 0.49428723,\n",

" 0.49021354, 0.39244492, 0.48206615, 0.27430784, 0.20912876,\n",

" 0.38022384, 0.46169769, 0.51872938, 0.48613984, 0.35985538,\n",

" 0.37207646, 0.44540292, 0.54317154, 0.510582 , 0.44132923,\n",

" 0.42910815, 0.42096077, 0.50243461, 0.48206615, 0.24986568,\n",

" 0.49836092, 0.3476343 , 0.38022384, 0.39651861, 0.38837123,\n",

" 0.31911846, 0.33133953, 0.31097107, 0.40873969, 0.17246553,\n",

" 0.45355031, 0.49021354, 0.52687677, 0.41688707, 0.51465569,\n",

" 0.4005923 , 0.44132923, 0.31911846, 0.457624 , 0.27430784,\n",

" 0.44540292, 0.39244492, 0.35578169, 0.47391877, 0.47799246,\n",

" 0.43318184, 0.51465569, 0.33541323, 0.51872938, 0.49428723,\n",

" 0.3476343 , 0.36800276, 0.42910815, 0.46577138, 0.40873969,\n",

" 0.40873969, 0.47391877, 0.50243461, 0.43318184, 0.56761369,\n",

" 0.19283399, 0.33133953, 0.44947661, 0.44540292, 0.351708 ,\n",

" 0.43725554, 0.41688707, 0.39651861, 0.44947661, 0.41688707,\n",

" 0.31504476, 0.41688707, 0.44947661, 0.56761369, 0.12765491,\n",

" 0.50243461, 0.20505507, 0.1887603 , 0.22134984, 0.39244492,\n",

" 0.44540292, 0.56354 , 0.44132923, 0.4005923 , 0.48613984,\n",

" 0.19690768, 0.351708 , 0.49021354, 0.27023415, 0.47799246,\n",

" 0.52687677, 0.09506537, 0.3476343 , 0.23764461, 0.49021354,\n",

" 0.38022384, 0.53502415, 0.46984508, 0.47799246, 0.49428723,\n",

" 0.38837123, 0.46169769, 0.24986568, 0.42910815, 0.49428723,\n",

" 0.31911846, 0.52687677, 0.44947661, 0.4005923 , 0.42910815,\n",

" 0.54724523, 0.37615015, 0.31097107, 0.55131892, 0.44540292,\n",

" 0.39244492, 0.52687677, 0.30689738, 0.44132923, 0.46984508,\n",

" 0.43725554, 0.34356061, 0.47391877, 0.32726584, 0.42910815,\n",

" 0.31911846, 0.27838153, 0.29060261, 0.33541323, 0.2946763 ,\n",

" 0.45355031, 0.15617076, 0.56354 , 0.36392907, 0.42910815,\n",

" 0.34356061, 0.55539262, 0.49428723, 0.48206615, 0.46577138,\n",

" 0.39651861, 0.31911846, 0.42910815, 0.48613984, 0.37615015,\n",

" 0.29874999, 0.38429753, 0.33948692, 0.45355031, 0.28652892,\n",

" 0.47799246, 0.31097107, 0.32319215, 0.43725554, 0.49428723,\n",

" 0.35578169, 0.33541323, 0.46169769, 0.41281338, 0.38022384,\n",

" 0.38837123, 0.36392907, 0.24986568, 0.25393938, 0.41281338,\n",

" 0.44947661, 0.60835062, 0.51872938, 0.51465569, 0.16839184,\n",

" 0.33133953, 0.33133953, 0.2417183 , 0.42503446, 0.55131892,\n",

" 0.46577138, 0.43318184, 0.48206615, 0.351708 , 0.41688707,\n",

" 0.33541323, 0.42096077, 0.457624 , 0.41688707, 0.55539262,\n",

" 0.47391877, 0.457624 , 0.40873969, 0.51872938, 0.404666 ,\n",

" 0.50243461, 0.55131892, 0.42503446, 0.41281338, 0.33541323,\n",

" 0.27430784, 0.46169769, 0.48613984, 0.457624 , 0.38837123,\n",

" 0.45355031, 0.38837123, 0.351708 , 0.51465569, 0.48206615,\n",

" 0.23764461, 0.17246553, 0.20912876, 0.46577138, 0.32726584,\n",

" 0.41688707, 0.44540292, 0.40873969, 0.33133953, 0.33541323,\n",

" 0.44132923, 0.14802337, 0.39651861, 0.27430784, 0.46577138,\n",

" 0.20505507, 0.50243461, 0.39651861, 0.40873969, 0.48613984,\n",

" 0.42910815, 0.49836092, 0.46984508, 0.54317154, 0.404666 ,\n",

" 0.40873969, 0.40873969, 0.51465569, 0.30689738, 0.21320245,\n",

" 0.44132923, 0.42096077, 0.26616045, 0.33541323, 0.44132923,\n",

" 0.29874999, 0.47391877, 0.43725554, 0.39244492, 0.33948692,\n",

" 0.44540292, 0.43725554, 0.27023415, 0.53502415, 0.31504476,\n",

" 0.47799246, 0.38022384, 0.29874999, 0.44947661, 0.49021354,\n",

" 0.27430784, 0.22542353, 0.27430784, 0.20505507, 0.45355031,\n",

" 0.44947661, 0.53095046, 0.48206615, 0.47391877, 0.35985538,\n",

" 0.27838153, 0.32726584, 0.41281338, 0.49428723, 0.41281338,\n",

" 0.404666 , 0.29060261, 0.404666 , 0.30282369, 0.46169769,\n",

" 0.52280308, 0.42096077, 0.3476343 , 0.24986568, 0.3476343 ,\n",

" 0.44947661, 0.404666 , 0.33133953, 0.49836092, 0.42910815,\n",

" 0.41688707, 0.53502415, 0.53095046, 0.43725554, 0.44540292,\n",

" 0.44947661, 0.42910815, 0.47391877, 0.11543383, 0.40873969,\n",

" 0.36392907, 0.33133953, 0.33541323, 0.22134984, 0.24986568,\n",

" 0.33541323, 0.4005923 , 0.51872938, 0.31911846, 0.38429753,\n",

" 0.37207646, 0.35985538, 0.44540292, 0.41688707, 0.55946631,\n",

" 0.404666 , 0.46577138, 0.46577138, 0.42096077, 0.36392907,\n",

" 0.22542353, 0.42910815, 0.510582 , 0.25801307, 0.38837123,\n",

" 0.43725554, 0.48206615, 0.37615015, 0.25393938, 0.25393938,\n",

" 0.41281338, 0.47799246, 0.31504476, 0.34356061, 0.42096077,\n",

" 0.18468661])"

]

},

"execution\_count": 45,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

"pred"

]

},

{

"cell\_type": "code",

"execution\_count": 47,

"id": "77ce9928",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([1.77342665])"

]

},

"execution\_count": 47,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y\_p=model.predict([[2.2]])\n",

"y\_p"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.10.7"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}