```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "# **[YBI Foundation](https://www.ybifoundation.org/)**\n",
        "\n",
        "**[Join Telegram to Get Updates of all Future FREE Bootcamps and Courses](https://telegram.me/ybif_ybifoundation)**"
      ],
      "metadata": {
        "id": "yCOD0EFTy74k"
      }
    },
    {
```

```
"cell_type": "markdown",
"source": [
 "# **Cancer Prediction**\n",
 "\n",
 "Dataset Information:\n",
 "\n",
 "Target Variable (y): \n",
 "- Diagnosis (M = malignant, B = benign)\n",
 "\n",
 "Ten features (X) are computed for each cell nucleus:\n",
 "\n",
 "1. radius (mean of distances from center to points on the perimeter)\n",
 "2. texture (standard deviation of gray-scale values)\n",
 "3. perimeter\n",
 "4. area\n",
 "5. smoothness (local variation in radius lengths)\n",
 "6.  compactness (perimeter^2 / area - 1.0)\n",
 "7. concavity (severity of concave portions of the contour)\n",
 "8. concave points (number of concave portions of the contour)\n",
 "9. symmetry\n",
 "10. fractal dimension (coastline approximation - 1)\n",
 "\n",
 "For each characteristic three measures are given:\n",
 "\n",
 "a. Mean\n",
 "\n",
 "b. Standard error\n",
 "\n",
 "c. Largest/ Worst"
```

```
   ],
   "metadata": {
    "id": "JaWtri-Iy74k"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
    "**[Watch Video Tutorial](https://www.youtube.com/c/YBIFoundation?sub_confirmation=1)**"
   ],
   "metadata": {
    "id": "9X8h9s4Yy74k"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "# Step 1 : import library\n",
    "import pandas as pd"
   ],
   "metadata": {
    "id": "1JlfKw5KzLsq"
   },
   "execution_count": 1,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "# Step 2 : import data\n",
    "cancer = pd.read_csv('https://github.com/YBIFoundation/Dataset/raw/main/Cancer.csv')"
   ],
   "metadata": {
    "id": "t3-VjbbQzLsq"
   },
   "execution_count": 2,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "cancer.head()"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 317
    },
    "outputId": "406b439c-4e6d-47d6-f5fe-1e210858bbd1",
    "id": "Rfcei9g4zLsr"
   },
   "execution_count": 3,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "        id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \\\n",
"0   842302       M      17.99       10.38        122.80    1001.0   \n",
"1   842517       M      20.57       17.77        132.90    1326.0   \n",
"2  84300903       M      19.69       21.25        130.00    1203.0   \n",
"3  84348301       M      11.42       20.38         77.58     386.1   \n",
"4  84358402       M      20.29       14.34        135.10    1297.0   \n",
"\n",
"   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \\\n",
"0         0.11840           0.27760          0.3001              0.14710   \n",
"1         0.08474           0.07864          0.0869              0.07017   \n",
"2         0.10960           0.15990          0.1974              0.12790   \n",
"3         0.14250           0.28390          0.2414              0.10520   \n",
"4         0.10030           0.13280          0.1980              0.10430   \n",
"\n",
"   ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \\\n",
"0  ...          17.33           184.60      2019.0            0.1622   \n",
"1  ...          23.41           158.80      1956.0            0.1238   \n",
"2  ...          25.53           152.50      1709.0            0.1444   \n",
"3  ...          26.50            98.87       567.7            0.2098   \n",
"4  ...          16.67           152.20      1575.0            0.1374   \n",
"\n",
"   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \\\n",
"0             0.6656           0.7119                0.2654          0.4601   \n",
"1             0.1866           0.2416                0.1860          0.2750   \n",
"2             0.4245           0.4504                0.2430          0.3613   \n",
"3             0.8663           0.6869                0.2575          0.6638   \n",
"4             0.2050           0.4000                0.1625          0.2364   \n",
"\n",
"   fractal_dimension_worst  Unnamed: 32  \n",
"0                  0.11890          NaN  \n",
   "1          0.08902      NaN  \n",
   "2          0.08758      NaN  \n",
   "3          0.17300      NaN  \n",
   "4          0.07678      NaN  \n",
   "\n",
   "[5 rows x 33 columns]"
  ],
  "text/html": [
   "\n",
   "  <div id=\"df-86b5fb7b-99e0-4309-9ed7-e8c91009e45d\">\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>id</th>\n",
"      <th>diagnosis</th>\n",
"      <th>radius_mean</th>\n",
"      <th>texture_mean</th>\n",
"      <th>perimeter_mean</th>\n",
"      <th>area_mean</th>\n",
"      <th>smoothness_mean</th>\n",
"      <th>compactness_mean</th>\n",
"      <th>concavity_mean</th>\n",
"      <th>concave points_mean</th>\n",
"      <th>...</th>\n",
"      <th>texture_worst</th>\n",
"      <th>perimeter_worst</th>\n",
"      <th>area_worst</th>\n",
"      <th>smoothness_worst</th>\n",
"      <th>compactness_worst</th>\n",
"      <th>concavity_worst</th>\n",
"      <th>concave points_worst</th>\n",
"      <th>symmetry_worst</th>\n",
"      <th>fractal_dimension_worst</th>\n",
"      <th>Unnamed: 32</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>842302</td>\n",
"      <td>M</td>\n",
"      <td>17.99</td>\n",
"     <td>10.38</td>\n",
"     <td>122.80</td>\n",
"     <td>1001.0</td>\n",
"     <td>0.11840</td>\n",
"     <td>0.27760</td>\n",
"     <td>0.3001</td>\n",
"     <td>0.14710</td>\n",
"     <td>...</td>\n",
"     <td>17.33</td>\n",
"     <td>184.60</td>\n",
"     <td>2019.0</td>\n",
"     <td>0.1622</td>\n",
"     <td>0.6656</td>\n",
"     <td>0.7119</td>\n",
"     <td>0.2654</td>\n",
"     <td>0.4601</td>\n",
"     <td>0.11890</td>\n",
"     <td>NaN</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>842517</td>\n",
"     <td>M</td>\n",
"     <td>20.57</td>\n",
"     <td>17.77</td>\n",
"     <td>132.90</td>\n",
"     <td>1326.0</td>\n",
"     <td>0.08474</td>\n",
"     <td>0.07864</td>\n",
"      <td>0.0869</td>\n",
"      <td>0.07017</td>\n",
"      <td>...</td>\n",
"      <td>23.41</td>\n",
"      <td>158.80</td>\n",
"      <td>1956.0</td>\n",
"      <td>0.1238</td>\n",
"      <td>0.1866</td>\n",
"      <td>0.2416</td>\n",
"      <td>0.1860</td>\n",
"      <td>0.2750</td>\n",
"      <td>0.08902</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>84300903</td>\n",
"      <td>M</td>\n",
"      <td>19.69</td>\n",
"      <td>21.25</td>\n",
"      <td>130.00</td>\n",
"      <td>1203.0</td>\n",
"      <td>0.10960</td>\n",
"      <td>0.15990</td>\n",
"      <td>0.1974</td>\n",
"      <td>0.12790</td>\n",
"      <td>...</td>\n",
"      <td>25.53</td>\n",
"      <td>152.50</td>\n",
"      <td>1709.0</td>\n",
"      <td>0.1444</td>\n",
"      <td>0.4245</td>\n",
"      <td>0.4504</td>\n",
"      <td>0.2430</td>\n",
"      <td>0.3613</td>\n",
"      <td>0.08758</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>84348301</td>\n",
"      <td>M</td>\n",
"      <td>11.42</td>\n",
"      <td>20.38</td>\n",
"      <td>77.58</td>\n",
"      <td>386.1</td>\n",
"      <td>0.14250</td>\n",
"      <td>0.28390</td>\n",
"      <td>0.2414</td>\n",
"      <td>0.10520</td>\n",
"      <td>...</td>\n",
"      <td>26.50</td>\n",
"      <td>98.87</td>\n",
"      <td>567.7</td>\n",
"      <td>0.2098</td>\n",
"      <td>0.8663</td>\n",
"      <td>0.6869</td>\n",
"      <td>0.2575</td>\n",
"      <td>0.6638</td>\n",
"      <td>0.17300</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>84358402</td>\n",
"      <td>M</td>\n",
"      <td>20.29</td>\n",
"      <td>14.34</td>\n",
"      <td>135.10</td>\n",
"      <td>1297.0</td>\n",
"      <td>0.10030</td>\n",
"      <td>0.13280</td>\n",
"      <td>0.1980</td>\n",
"      <td>0.10430</td>\n",
"      <td>...</td>\n",
"      <td>16.67</td>\n",
"      <td>152.20</td>\n",
"      <td>1575.0</td>\n",
"      <td>0.1374</td>\n",
"      <td>0.2050</td>\n",
"      <td>0.4000</td>\n",
"      <td>0.1625</td>\n",
"      <td>0.2364</td>\n",
"      <td>0.07678</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
        "</table>\n",
        "<p>5 rows × 33 columns</p>\n",
        "</div>\n",
        "      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-86b5fb7b-99e0-4309-
9ed7-e8c91009e45d')\"\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-86b5fb7b-99e0-4309-9ed7-e8c91009e45d 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-86b5fb7b-99e0-4309-9ed7-e8c91009e45d');\n",
        "            const dataTable =\n",
        "              await google.colab.kernel.invokeFunction('convertToInteractive',\n",
        "                                                        [key], {});\n",
        "            if (!dataTable) return;\n",
        "\n",
        "            const docLinkHtml = 'Like what you see? Visit the ' +\n",
        "              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
        "              + ' to learn more about interactive tables.';\n",
        "            element.innerHTML = '';\n",
        "            dataTable['output_type'] = 'display_data';\n",
        "            await google.colab.output.renderOutput(dataTable, element);\n",
        "            const docLink = document.createElement('div');\n",
        "            docLink.innerHTML = docLinkHtml;\n",
        "            element.appendChild(docLink);\n",
        "          }\n",
        "        </script>\n",
        "      </div>\n",
        "    </div>\n",
        "  "
       ]
      },
   "metadata": {},
   "execution_count": 3
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "cancer.info()"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "66b316cf-47b4-4a80-e6b4-76fb7b892f03",
  "id": "GfvX6UjkzLsr"
 },
 "execution_count": 4,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
    "RangeIndex: 569 entries, 0 to 568\n",
    "Data columns (total 33 columns):\n",
    " #   Column                   Non-Null Count  Dtype  \n",
    "---  ------                   --------------  -----  \n",
    " 0   id                       569 non-null    int64  \n",
" 1   diagnosis                569 non-null    object \n",
" 2   radius_mean              569 non-null    float64\n",
" 3   texture_mean             569 non-null    float64\n",
" 4   perimeter_mean           569 non-null    float64\n",
" 5   area_mean                569 non-null    float64\n",
" 6   smoothness_mean          569 non-null    float64\n",
" 7   compactness_mean         569 non-null    float64\n",
" 8   concavity_mean           569 non-null    float64\n",
" 9   concave points_mean      569 non-null    float64\n",
" 10  symmetry_mean            569 non-null    float64\n",
" 11  fractal_dimension_mean   569 non-null    float64\n",
" 12  radius_se                569 non-null    float64\n",
" 13  texture_se               569 non-null    float64\n",
" 14  perimeter_se             569 non-null    float64\n",
" 15  area_se                  569 non-null    float64\n",
" 16  smoothness_se            569 non-null    float64\n",
" 17  compactness_se           569 non-null    float64\n",
" 18  concavity_se             569 non-null    float64\n",
" 19  concave points_se        569 non-null    float64\n",
" 20  symmetry_se              569 non-null    float64\n",
" 21  fractal_dimension_se     569 non-null    float64\n",
" 22  radius_worst             569 non-null    float64\n",
" 23  texture_worst            569 non-null    float64\n",
" 24  perimeter_worst          569 non-null    float64\n",
" 25  area_worst               569 non-null    float64\n",
" 26  smoothness_worst         569 non-null    float64\n",
" 27  compactness_worst        569 non-null    float64\n",
" 28  concavity_worst          569 non-null    float64\n",
" 29  concave points_worst     569 non-null    float64\n",
    " 30  symmetry_worst           569 non-null    float64\n",
    " 31  fractal_dimension_worst  569 non-null    float64\n",
    " 32  Unnamed: 32              0 non-null      float64\n",
    "dtypes: float64(31), int64(1), object(1)\n",
    "memory usage: 146.8+ KB\n"
   ]
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "cancer.describe()"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 411
  },
  "outputId": "21864e24-85a5-44de-a422-7be35b597c51",
  "id": "wKeWsipAzLsr"
 },
 "execution_count": 5,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "               id  radius_mean  texture_mean  perimeter_mean    area_mean  \\\n",
"count  5.690000e+02   569.000000   569.000000    569.000000   569.000000   \n",
"mean   3.037183e+07    14.127292    19.289649     91.969033   654.889104   \n",
"std    1.250206e+08     3.524049     4.301036     24.298981   351.914129   \n",
"min    8.670000e+03     6.981000     9.710000     43.790000   143.500000   \n",
"25%    8.692180e+05    11.700000    16.170000     75.170000   420.300000   \n",
"50%    9.060240e+05    13.370000    18.840000     86.240000   551.100000   \n",
"75%    8.813129e+06    15.780000    21.800000    104.100000   782.700000   \n",
"max    9.113205e+08    28.110000    39.280000    188.500000  2501.000000   \n",
"\n",
"       smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \\\n",
"count       569.000000        569.000000      569.000000           569.000000   \n",
"mean          0.096360          0.104341        0.088799             0.048919   \n",
"std           0.014064          0.052813        0.079720             0.038803   \n",
"min           0.052630          0.019380        0.000000             0.000000   \n",
"25%           0.086370          0.064920        0.029560             0.020310   \n",
"50%           0.095870          0.092630        0.061540             0.033500   \n",
"75%           0.105300          0.130400        0.130700             0.074000   \n",
"max           0.163400          0.345400        0.426800             0.201200   \n",
"\n",
"       symmetry_mean  ...  texture_worst  perimeter_worst   area_worst  \\\n",
"count     569.000000  ...     569.000000       569.000000   569.000000   \n",
"mean        0.181162  ...      25.677223       107.261213   880.583128   \n",
"std         0.027414  ...       6.146258        33.602542   569.356993   \n",
"min         0.106000  ...      12.020000        50.410000   185.200000   \n",
"25%         0.161900  ...      21.080000        84.110000   515.300000   \n",
"50%         0.179200  ...      25.410000        97.660000   686.500000   \n",
"75%         0.195700  ...      29.720000       125.400000  1084.000000   \n",
"max         0.304000  ...      49.540000       251.200000  4254.000000   \n",
"\n",
"       smoothness_worst  compactness_worst  concavity_worst  \\\n",
"count       569.000000         569.000000       569.000000   \n",
"mean          0.132369           0.254265         0.272188   \n",
"std           0.022832           0.157336         0.208624   \n",
"min           0.071170           0.027290         0.000000   \n",
"25%           0.116600           0.147200         0.114500   \n",
"50%           0.131300           0.211900         0.226700   \n",
"75%           0.146000           0.339100         0.382900   \n",
"max           0.222600           1.058000         1.252000   \n",
"\n",
"       concave points_worst  symmetry_worst  fractal_dimension_worst  \\\n",
"count            569.000000      569.000000               569.000000   \n",
"mean               0.114606        0.290076                 0.083946   \n",
"std                0.065732        0.061867                 0.018061   \n",
"min                0.000000        0.156500                 0.055040   \n",
"25%                0.064930        0.250400                 0.071460   \n",
"50%                0.099930        0.282200                 0.080040   \n",
"75%                0.161400        0.317900                 0.092080   \n",
"max                0.291000        0.663800                 0.207500   \n",
"\n",
"       Unnamed: 32  \n",
"count          0.0  \n",
"mean           NaN  \n",
"std            NaN  \n",
"min            NaN  \n",
"25%            NaN  \n",
"50%            NaN  \n",
"75%            NaN  \n",
"max            NaN  \n",
 "\n",
 "[8 rows x 32 columns]"
],
"text/html": [
 "\n",
 "  <div id=\"df-fb737bca-f188-4205-b4aa-040719d6a725\">\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>id</th>\n",
 "      <th>radius_mean</th>\n",
 "      <th>texture_mean</th>\n",
 "      <th>perimeter_mean</th>\n",
"      <th>area_mean</th>\n",
"      <th>smoothness_mean</th>\n",
"      <th>compactness_mean</th>\n",
"      <th>concavity_mean</th>\n",
"      <th>concave points_mean</th>\n",
"      <th>symmetry_mean</th>\n",
"      <th>...</th>\n",
"      <th>texture_worst</th>\n",
"      <th>perimeter_worst</th>\n",
"      <th>area_worst</th>\n",
"      <th>smoothness_worst</th>\n",
"      <th>compactness_worst</th>\n",
"      <th>concavity_worst</th>\n",
"      <th>concave points_worst</th>\n",
"      <th>symmetry_worst</th>\n",
"      <th>fractal_dimension_worst</th>\n",
"      <th>Unnamed: 32</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>5.690000e+02</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>...</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>569.000000</td>\n",
"      <td>0.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>mean</th>\n",
"      <td>3.037183e+07</td>\n",
"      <td>14.127292</td>\n",
"      <td>19.289649</td>\n",
"      <td>91.969033</td>\n",
"      <td>654.889104</td>\n",
"      <td>0.096360</td>\n",
"      <td>0.104341</td>\n",
"      <td>0.088799</td>\n",
"      <td>0.048919</td>\n",
"      <td>0.181162</td>\n",
"      <td>...</td>\n",
"      <td>25.677223</td>\n",
"      <td>107.261213</td>\n",
"      <td>880.583128</td>\n",
"      <td>0.132369</td>\n",
"      <td>0.254265</td>\n",
"      <td>0.272188</td>\n",
"      <td>0.114606</td>\n",
"      <td>0.290076</td>\n",
"      <td>0.083946</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>std</th>\n",
"      <td>1.250206e+08</td>\n",
"      <td>3.524049</td>\n",
"      <td>4.301036</td>\n",
"      <td>24.298981</td>\n",
"      <td>351.914129</td>\n",
"      <td>0.014064</td>\n",
"      <td>0.052813</td>\n",
"      <td>0.079720</td>\n",
"      <td>0.038803</td>\n",
"      <td>0.027414</td>\n",
"      <td>...</td>\n",
"      <td>6.146258</td>\n",
"      <td>33.602542</td>\n",
"      <td>569.356993</td>\n",
"      <td>0.022832</td>\n",
"      <td>0.157336</td>\n",
"      <td>0.208624</td>\n",
"      <td>0.065732</td>\n",
"      <td>0.061867</td>\n",
"      <td>0.018061</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>min</th>\n",
"      <td>8.670000e+03</td>\n",
"      <td>6.981000</td>\n",
"      <td>9.710000</td>\n",
"      <td>43.790000</td>\n",
"      <td>143.500000</td>\n",
"      <td>0.052630</td>\n",
"      <td>0.019380</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.106000</td>\n",
"      <td>...</td>\n",
"      <td>12.020000</td>\n",
"      <td>50.410000</td>\n",
"      <td>185.200000</td>\n",
"      <td>0.071170</td>\n",
"      <td>0.027290</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.156500</td>\n",
"      <td>0.055040</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25%</th>\n",
"      <td>8.692180e+05</td>\n",
"      <td>11.700000</td>\n",
"      <td>16.170000</td>\n",
"      <td>75.170000</td>\n",
"      <td>420.300000</td>\n",
"      <td>0.086370</td>\n",
"      <td>0.064920</td>\n",
"      <td>0.029560</td>\n",
"      <td>0.020310</td>\n",
"      <td>0.161900</td>\n",
"      <td>...</td>\n",
"      <td>21.080000</td>\n",
"      <td>84.110000</td>\n",
"      <td>515.300000</td>\n",
"      <td>0.116600</td>\n",
"      <td>0.147200</td>\n",
"      <td>0.114500</td>\n",
"      <td>0.064930</td>\n",
"      <td>0.250400</td>\n",
"      <td>0.071460</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>9.060240e+05</td>\n",
"      <td>13.370000</td>\n",
"      <td>18.840000</td>\n",
"      <td>86.240000</td>\n",
"      <td>551.100000</td>\n",
"      <td>0.095870</td>\n",
"      <td>0.092630</td>\n",
"      <td>0.061540</td>\n",
"      <td>0.033500</td>\n",
"      <td>0.179200</td>\n",
"      <td>...</td>\n",
"      <td>25.410000</td>\n",
"      <td>97.660000</td>\n",
"      <td>686.500000</td>\n",
"      <td>0.131300</td>\n",
"      <td>0.211900</td>\n",
"      <td>0.226700</td>\n",
"      <td>0.099930</td>\n",
"      <td>0.282200</td>\n",
"      <td>0.080040</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>75%</th>\n",
"      <td>8.813129e+06</td>\n",
"      <td>15.780000</td>\n",
"      <td>21.800000</td>\n",
"      <td>104.100000</td>\n",
"      <td>782.700000</td>\n",
"      <td>0.105300</td>\n",
"      <td>0.130400</td>\n",
"      <td>0.130700</td>\n",
"      <td>0.074000</td>\n",
"      <td>0.195700</td>\n",
"      <td>...</td>\n",
"      <td>29.720000</td>\n",
"      <td>125.400000</td>\n",
"      <td>1084.000000</td>\n",
"      <td>0.146000</td>\n",
"      <td>0.339100</td>\n",
"      <td>0.382900</td>\n",
"      <td>0.161400</td>\n",
"      <td>0.317900</td>\n",
"      <td>0.092080</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>max</th>\n",
"      <td>9.113205e+08</td>\n",
"      <td>28.110000</td>\n",
"      <td>39.280000</td>\n",
"      <td>188.500000</td>\n",
"      <td>2501.000000</td>\n",
"      <td>0.163400</td>\n",
"      <td>0.345400</td>\n",
"      <td>0.426800</td>\n",
"      <td>0.201200</td>\n",
"      <td>0.304000</td>\n",
"      <td>...</td>\n",
"      <td>49.540000</td>\n",
"      <td>251.200000</td>\n",
"      <td>4254.000000</td>\n",
"      <td>0.222600</td>\n",
"      <td>1.058000</td>\n",
"      <td>1.252000</td>\n",
"      <td>0.291000</td>\n",
"      <td>0.663800</td>\n",
"      <td>0.207500</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>8 rows × 32 columns</p>\n",
"</div>\n",
"      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-fb737bca-f188-4205-b4aa-040719d6a725')\"\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-fb737bca-f188-4205-b4aa-040719d6a725 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-fb737bca-f188-4205-b4aa-040719d6a725');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"            '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
              "          const docLink = document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "metadata": {},
          "execution_count": 5
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# Step 3 : define target (y) and features (X)"
      ],
      "metadata": {
        "id": "UfNtsBgjzLsr"
      },
      "execution_count": 6,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
  "cancer.columns"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "b9f955ae-963a-4787-97fb-bb0a983458b4",
  "id": "P-ykMVLizLss"
},
"execution_count": 7,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',\n",
        "       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',\n",
        "       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',\n",
        "       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',\n",
        "       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',\n",
        "       'fractal_dimension_se', 'radius_worst', 'texture_worst',\n",
        "       'perimeter_worst', 'area_worst', 'smoothness_worst',\n",
        "       'compactness_worst', 'concavity_worst', 'concave points_worst',\n",
        "       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],\n",
        "      dtype='object')"
      ]
    },
    "metadata": {},
    "execution_count": 7
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "y = cancer['diagnosis']"
  ],
  "metadata": {
    "id": "34E6jhSFzLss"
  },
  "execution_count": 8,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "X = cancer.drop(['id','diagnosis','Unnamed: 32'],axis=1)"
  ],
  "metadata": {
    "id": "7lWJdtbWzLss"
  },
  "execution_count": 9,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "# Step 4 : train test split\n",
    "from sklearn.model_selection import train_test_split\n",
    "X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.7, random_state=2529)"
   ],
   "metadata": {
    "id": "3b1vrx-tzLst"
   },
   "execution_count": 10,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "# check shape of train and test sample\n",
    "X_train.shape, X_test.shape, y_train.shape, y_test.shape"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "6cdd2cc2-cd0a-4b44-aeab-44621d11f203",
    "id": "Z6eHvZi4zLst"
   },
   "execution_count": 11,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "((398, 30), (171, 30), (398,), (171,))"
```

```
            ]
          },
          "metadata": {},
          "execution_count": 11
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "# Step 5 : select model\n",
        "from sklearn.linear_model import LogisticRegression\n",
        "model = LogisticRegression(max_iter=5000)"
      ],
      "metadata": {
        "id": "Spqy8XsGzLst"
      },
      "execution_count": 12,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "# Step 6 : train or fit model\n",
        "model.fit(X_train,y_train)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
    "height": 75
   },
   "outputId": "85b9bf1a-0cc4-4b35-c73d-ea3ed03f226b",
   "id": "8bAOoepNzLst"
  },
  "execution_count": 13,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "LogisticRegression(max_iter=5000)"
     ],
     "text/html": [
      "<style>#sk-container-id-1 {color: black;background-color: white;}#sk-container-id-1 pre{padding: 0;}#sk-container-id-1 div.sk-toggleable {background-color: white;}#sk-container-id-1 label.sk-toggleable__label {cursor: pointer;display: block;width: 100%;margin-bottom: 0;padding: 0.3em;box-sizing: border-box;text-align: center;}#sk-container-id-1 label.sk-toggleable__label-arrow:before {content: \"▸\";float: left;margin-right: 0.25em;color: #696969;}#sk-container-id-1 label.sk-toggleable__label-arrow:hover:before {color: black;}#sk-container-id-1 div.sk-estimator:hover label.sk-toggleable__label-arrow:before {color: black;}#sk-container-id-1 div.sk-toggleable__content {max-height: 0;max-width: 0;overflow: hidden;text-align: left;background-color: #f0f8ff;}#sk-container-id-1 div.sk-toggleable__content pre {margin: 0.2em;color: black;border-radius: 0.25em;background-color: #f0f8ff;}#sk-container-id-1 input.sk-toggleable__control:checked~div.sk-toggleable__content {max-height: 200px;max-width: 100%;overflow: auto;}#sk-container-id-1 input.sk-toggleable__control:checked~label.sk-toggleable__label-arrow:before {content: \"▾\";}#sk-container-id-1 div.sk-estimator input.sk-toggleable__control:checked~label.sk-toggleable__label {background-color: #d4ebff;}#sk-container-id-1 div.sk-label input.sk-toggleable__control:checked~label.sk-toggleable__label {background-color: #d4ebff;}#sk-container-id-1 input.sk-hidden--visually {border: 0;clip: rect(1px 1px 1px 1px);clip: rect(1px, 1px, 1px, 1px);height: 1px;margin: -1px;overflow: hidden;padding: 0;position: absolute;width: 1px;}#sk-container-id-1 div.sk-estimator {font-family: monospace;background-color: #f0f8ff;border: 1px dotted black;border-radius: 0.25em;box-sizing: border-box;margin-bottom: 0.5em;}#sk-container-id-1 div.sk-estimator:hover {background-color: #d4ebff;}#sk-container-id-1 div.sk-parallel-item::after {content: \"\";width: 100%;border-bottom: 1px solid gray;flex-grow: 1;}#sk-container-id-1 div.sk-label:hover label.sk-toggleable__label {background-color: #d4ebff;}#sk-container-id-1 div.sk-serial::before {content: \"\";position: absolute;border-left: 1px solid gray;box-sizing: border-box;top: 0;bottom: 0;left: 50%;z-index: 0;}#sk-container-id-1 div.sk-serial
```

```
{display: flex;flex-direction: column;align-items: center;background-color: white;padding-right: 0.2em;padding-left: 0.2em;position: relative;}#sk-container-id-1 div.sk-item {position: relative;z-index: 1;}#sk-container-id-1 div.sk-parallel {display: flex;align-items: stretch;justify-content: center;background-color: white;position: relative;}#sk-container-id-1 div.sk-item::before, #sk-container-id-1 div.sk-parallel-item::before {content: \"\";position: absolute;border-left: 1px solid gray;box-sizing: border-box;top: 0;bottom: 0;left: 50%;z-index: -1;}#sk-container-id-1 div.sk-parallel-item {display: flex;flex-direction: column;z-index: 1;position: relative;background-color: white;}#sk-container-id-1 div.sk-parallel-item:first-child::after {align-self: flex-end;width: 50%;}#sk-container-id-1 div.sk-parallel-item:last-child::after {align-self: flex-start;width: 50%;}#sk-container-id-1 div.sk-parallel-item:only-child::after {width: 0;}#sk-container-id-1 div.sk-dashed-wrapped {border: 1px dashed gray;margin: 0 0.4em 0.5em 0.4em;box-sizing: border-box;padding-bottom: 0.4em;background-color: white;}#sk-container-id-1 div.sk-label label {font-family: monospace;font-weight: bold;display: inline-block;line-height: 1.2em;}#sk-container-id-1 div.sk-label-container {text-align: center;}#sk-container-id-1 div.sk-container {/* jupyter's `normalize.less` sets `[hidden] { display: none; }` but bootstrap.min.css set `[hidden] { display: none !important; }` so we also need the `!important` here to be able to override the default hidden behavior on the sphinx rendered scikit-learn.org. See: https://github.com/scikit-learn/scikit-learn/issues/21755 */display: inline-block !important;position: relative;}#sk-container-id-1 div.sk-text-repr-fallback {display: none;}</style><div id=\"sk-container-id-1\" class=\"sk-top-container\"><div class=\"sk-text-repr-fallback\"><pre>LogisticRegression(max_iter=5000)</pre><b>In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. <br />On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.</b></div><div class=\"sk-container\" hidden><div class=\"sk-item\"><div class=\"sk-estimator sk-toggleable\"><input class=\"sk-toggleable__control sk-hidden--visually\" id=\"sk-estimator-id-1\" type=\"checkbox\" checked><label for=\"sk-estimator-id-1\" class=\"sk-toggleable__label sk-toggleable__label-arrow\">LogisticRegression</label><div class=\"sk-toggleable__content\"><pre>LogisticRegression(max_iter=5000)</pre></div></div></div></div></div>"
       ]
      },
      "metadata": {},
      "execution_count": 13
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "model.intercept_"
    ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "4cbbcd85-39e8-4b4e-bf8c-31b1865c4d6d",
    "id": "_Cvr09y5zLst"
   },
   "execution_count": 14,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([-30.20269391])"
      ]
     },
     "metadata": {},
     "execution_count": 14
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "model.coef_"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "99b3e9fa-efe2-4633-d360-7ee8ea62fdad",
    "id": "-riH-UYezLsu"
   },
   "execution_count": 15,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([[-0.8644508 , -0.1823121 ,  0.26510852, -0.02688942,  0.13284582,\n",
       "         0.19445151,  0.40918278,  0.20206338,  0.17199488,  0.03798515,\n",
       "         0.0192444 , -1.13284188, -0.13597054,  0.11911954,  0.02266663,\n",
       "        -0.03006638,  0.04691738,  0.02805721,  0.03329433, -0.00980702,\n",
       "        -0.27140621,  0.44034405,  0.16566196,  0.01286379,  0.2719812 ,\n",
       "         0.59704539,  1.06177846,  0.40903862,  0.51193487,  0.08436947]])"
      ]
     },
     "metadata": {},
     "execution_count": 15
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "# Step 7 : predict model\n",
    "y_pred = model.predict(X_test)"
   ],
   "metadata": {
    "id": "Rt7vdl_gzLsu"
   },
   "execution_count": 16,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "y_pred"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "d6d07116-de5e-42a7-8855-f466ad6ad1bd",
    "id": "0TV_ji8KzLsu"
   },
   "execution_count": 17,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array(['B', 'M', 'M', 'B', 'M', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B',\n",
       "       'M', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'M',\n",
       "       'B', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'B',\n",
       "       'M', 'M', 'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B',\n",
       "       'B', 'B', 'B', 'B', 'B', 'M', 'M', 'M', 'B', 'M', 'B', 'M', 'M',\n",
       "      'M', 'M', 'B', 'M', 'M', 'B', 'M', 'B', 'M', 'B', 'M', 'B', 'B',\n",
       "      'M', 'M', 'M', 'B', 'B', 'M', 'M', 'M', 'B', 'B', 'B', 'B', 'M',\n",
       "      'B', 'B', 'B', 'M', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'B',\n",
       "      'B', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B',\n",
       "      'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M',\n",
       "      'M', 'B', 'M', 'M', 'M', 'B', 'B', 'M', 'B', 'M', 'B', 'M', 'B',\n",
       "      'M', 'B', 'M', 'B', 'B', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'B',\n",
       "      'B', 'M', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'B', 'M', 'B', 'B',\n",
       "      'B', 'B'], dtype=object)"
      ]
     },
     "metadata": {},
     "execution_count": 17
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "# Step 8 : model accuracy\n",
    "from sklearn.metrics import confusion_matrix, accuracy_score, classification_report"
   ],
   "metadata": {
    "id": "gjS8ioy-zLsu"
   },
   "execution_count": 18,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "confusion_matrix(y_test,y_pred)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "6c745193-0994-459a-91c5-b9a321568583",
    "id": "Owk-bowQzLsu"
   },
   "execution_count": 19,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([[97,  5],\n",
       "       [ 2, 67]])"
      ]
     },
     "metadata": {},
     "execution_count": 19
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "accuracy_score(y_test,y_pred)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "d71dd810-7e90-4307-9dba-62cadabe5725",
    "id": "C4-CPO1WzLsu"
   },
   "execution_count": 20,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "0.9590643274853801"
      ]
     },
     "metadata": {},
     "execution_count": 20
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "print(classification_report(y_test,y_pred))"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "9ddc6cbc-ae52-4325-8a11-cb02418f01b0",
    "id": "B4ilAmm5zLsv"
   },
   "execution_count": 21,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "              precision    recall  f1-score   support\n",
      "\n",
      "           B       0.98      0.95      0.97       102\n",
      "           M       0.93      0.97      0.95        69\n",
      "\n",
      "    accuracy                           0.96       171\n",
      "   macro avg       0.96      0.96      0.96       171\n",
      "weighted avg       0.96      0.96      0.96       171\n",
      "\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# **Don't Forget to Star and Watch on GitHub to Receive Updates**\n",
    "**Action 1: ⭐Star Repository as it make easy for you to find it again. You can see all the repositories and topics you have starred by going to your stars page.**\n",
    "\n",
    "**Action 2: 👁 Watch Repository and get notified of all future updates and activities in this repository.**\n",
    "\n",
    "**[Click Here to Visit Fundamental Repository on GitHub](https://github.com/YBIFoundation/Fundamentals)**"
   ],
   "metadata": {
    "id": "REsvrZJnzLsv"
   }
  },
  {
   "cell_type": "markdown",
   "source": [],
   "metadata": {
    "id": "SyBIrkCHzLsv"
   }
  },
  {
   "cell_type": "markdown",
   "source": [

"![image.png](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAksAAABHCAYAAAAX4ANQAAAZ
yUlEQVR4nO3de1RTd9ro8W8uJAQJclMrlnqpF0aPqIww9UTrK9YeWl1F22o91Nb3VeesXuwZj71Mzxo92sv
0rfaMdY6XmTWtrtdarco7IrPUoa1iW8vYpX3xwlsFdcTqwIwKCEagCSQ5f4RQkVxJQhJ4PmuxVkl29t70eXz2s
3/7t3cUTT+YbQghhBBCCKeUod4BIYQQQohwJs2SEEIIIYQb0iwJIYQQQrihDvUOiJ6pefjoUO+CcEJ38Wyod0
EIISKOQiZ4C39IU9QzSBMlhBCuSbMkfCYNUs8mjZMkvAsKvDkmzJLwiBazKvDkmzJLwiBaz3CbdiJXoOqSfCH6GoTdIsCY+k
sPVu0jQJf0kNEcHUHTVKmiXhkj8FTg6w4amrMZV4Cl9JgyRCIVi1Spol4ZQvhU4OpJHN21hLnIU3pEkS4SDQ9
UqaJeGUp4InB86eSeIu/BGIRklyTEDgmu5A5ZM0S6ITd0kqhax3cJUDEn/hTFcObJJLoitClWvSLIlO5EApQPJA
eM+bA5jkjQiG7ppGIE/wFl6RQtf76C6elfknwiMZiRahdGeOBbNeyXfDCSGECDhplER30108G7S5gnHDN7S8k2ZJCCF
EmjJHqaoF6Ga2hooLGxEXNLCzabTI0KFwqFAk1UFH369KFv375dWofENrJJDghfSc6IUAlE7vkrM1Sa2sr169
```

fRx2lpl+/fkRFqVEqZRArXFitVlpaWrl5s47q6kb69++PWu1dKkhsewZ/c+Da9etoNRqSk5LQRmtRKBRB3mNxt
79equT+YUO7bXtWqxWzyUxdfT23GxsZIDnTa0VS7gVKULZ27fp1YmJiSExIAJsNG2C1yllE+LB36QMG3ENdX
R3Xrl9nUEqKV5+U2PYU/uWAPlZP375x9hywIaMEIdK9//YUaLVaBt5zD/X19ZIzvVyk5F6gBHxIoKGhAbVKR
WJCAra2g6kIPzbsxSoxMRG1SklDQ4PHz0hse5au5ACAVqOhb984yYFeyJEz8fHxqFQq6r3ImfqGBskZ4bdOu
Vdf363bD/jI0u3GRpKSEuWsIULYbDb69o2ntq7O47VgiW3XmE1mzC0tWCytbWdjNkCBUqlApVKjiYpCo9W
EbP98yQEAvV4vOeBG53gTNrEOFJvNRkJ8PDW1tWg9LNvY2EhyUpLkTDfobbkXHx/fbdsNeLPU0tJClDoq0Ks
VQaSJiqKlpcXjchJb35hNZpp/aMZqtTp514bVasNqNdPSYl9OF60LWTHzNgcANJrIL7jB4C7e4RTrQInyoW5Izg
SX5F7wBbxZstlsqFQqv9ZReqqMz4u/JG3kcHJn5QRoz4QrSpXKq7O+QMS2t2hqasJkMnm9vNVqpbGpkVZL
KzExMUHcM+e8zQFAJuY64Uu8Qx3rQFEqlV7XjUDmjLl6DwCalHkBW2ckk9zrHmH3BO+vjx1ny7adJCUmUn
6+iKSkRCZPygr1bgnhNV8bpTs5PhfJhcyVmtqbFB89htHYSNqIYRge+GmH97/+5j+ouHAJvb4P2VMmkZyUEK
I99U1X492TYx0sVvMNzH/PB0CdPA2lpl+I9yi0JPe6T1g1S45GKfXeQby+fCn/7/dbKNxfRMa4scTE6EK9eyIA
mpqaKfnmOKWnyig/fxGAtJHDyRg/FsMDWREfZ7PJ3OVGycFkMqFWqSN+qPxufz70JSazmaGDUzn9XTkpA
/szdHAqAOUXLnHmu3KGDr63vamaN/vREO+xZ/7Gu6fGOlgco0qO/44e8mII9ya0JPe6V9g8IOfuRikmRsfsW
TnU1NbxSX4BTU3Nod5F4afy8xdZ9ev32LmngMamZnJn5ZA7K4fGpmZ27ilg1a/fa2+gIlXzD4HJ00CtJ5wYbzc
ybHAqWRnpgH2kqf09420AJj8wkZSBAzq8F84CEaeeGOtgsJpv0Fr7RfvvrbVfYDXfCOEehZbkXvcKi2ap9FRZp
0YJaJ+z9PWx47y64k22bNtJybHj1NTW+baBa4W8OCGT9Dt+XiysdbHcMvZeAzByOj+f404W87R+w5zX2Hb
WbH/PXMmh/CIuNfm2yz3N18eOs2bdRnQ6Hb9cvpQ3V7zK7Fk5zJ6Vw5srXuWXy5ei0+lYs24jXx877mItlex
6OpNHtlb8+JK5hHcm3PXa5XzmTVjArst+7nRbXH9z0rvFzSazi8ncbSz/4ND6ZczMfpgZT61gd4XZ5aKOh7D1J
Olj0ii/cIntu/eh0USRNvL+9vfSRt6PRhPF9t37qLhwifQxaT6t+1zRDjZvufPnMOd83cGaE3y8ZQeFFZ4XBQ/xtj
Xx/Ykv2FNwkE8KDvP5WdfNX0+MdaDYLI1YjGdprf0C0+VNnd43Xd5Ea+0XWIxnsVkau74hSy3HP3iN2dMcN
XwtR41+7LifagqXkb60kBoX73uqNeaqixza/S4vzFnBATf9pOSe98LiMtxnxV+SlJjYoVFymD0rh8mTsti3v4j/O
HWm/UD60nOLyRg/1qftzF9fzEvj2n6J0btf2FzOoc2/w9g/h6ypeownt7LqTVhYsIhxLj5iX7+ZqqK3eObnbzHo
wFs8ZDzDrs35TJmQw7DhPu1uj+EYNTRMyiJv7hynl9rSRg7n9eVL+XDbTrZs2wngZK7aUH6SqaGqtJyaRaNIB
vjuWwoA0zflVC0axSDAeOEM5SnZjB/iep+8iaevzB7uzriwewVrbjzJh0Vr0Xz+Js8u/wMjCpaS4WIU3NzS0qO
GyCc/8FNMZjMVFy7x1JyZ6GP7tL+nj+3DU3Nmsn33PkaNGMbku+YzeaXP/eTM+i/2vEBFHw+L+8tdvG+d+5
a//EPHxBkGBl4v5dNTpRzvN50sF1Nselqs/dVafxzT5c0eGyCL8Tssxu/af1eo+hB9/6uo9GN82l75tudZcnAimz
4qZrzeTNWXf+ZvTYC+moMrV/DVA+/x7sykrvwpQeG+1pzlo+U7MU+C7xpgphfrktzzLCxGltJGDqe2ro4Nv9/
i9DJMclIiSxbmsfn9d3njV6+SlJjIZ8Vf+rwdbZwefXzbj6fc0GTy8pFiVk+1N1WmK2c4dNn9qYZ9/UmkzX2a+U1
FnK4EUnP58MjHLOyljdKVq1V8kl9A6r2DWLIwr71RKj1Vxqpfv8erv3qT0lNlAMTE6Pifzy8m9d5BfJJfwJWrVZ
3Wl5aZCyUVnGs7GSo/WUzyjGlknfiWc23hOXemCO2MTNyNTXgTT19ZLK1u3jzL5zuuMGduDoM1GgbOeIy
ZzUWUurnq6HZ9EcqbSduDBg7o2soVKuJiY9t+dAT7vk3X8bnF+Su3iU4dyYjYKGKHDSdVY+LatdtdWFfvpIwZ
Cl16fKUNhba/j5+p5WxpJcOemMuU1LYanruAhwbY3zu3vwxjmIXHfb6MZsnut3lh+n0BWJdwCItmafasHBY
vzONGTR1r1m2kcH+Ry2U/K/6S2ro6Msb5NqrknJlLhWt5Zlom6YYFrD5wmfbpcndcgqkpXEb26hLgY57x4bI
MACd/S7rj0p5jnXuLeD3HQPqEbJZsq7Bv01LNoXcWkW3IJH2CgeVF3lz/C3/79hdhs9l4ffnSDq/v3GNvhmpq
6yg80DHery9fis1mY5+TPNCmZ/IohZyqAHuRq+Gh/76Yn6UUc/oiQCWXSmH6mCFANQdX/vj/NHvZx5Q3uY
jnncPwhhxeLKz+caOVTuLlhNvH/9dd4fvmLAYPavtdcx8jxpu5cMX1JeWe+DUyw9omdG/fve+uy2Y72L57Hw
BDB98buA02V/PV/r3t2/i3/Sf4vm2axrmiHWzedZgDe3exefeJjpc8rDV8lb+DLUV/xdUVdNfxaeBWE/SNdzSG
SST0hdv17i7FdU+sS0+V8S/PLXP64zhpCQdKTT90I99AofL+bi2FKgbdyDe6cIdcEg8+YqDqdytZXViB0eJ4vYzfT
FjENuDo6py2y2LOawrUsndpJumrf8sHz2aTvr6srfa/wG82ryTbYD8GmC4W8s6z2fZLfc+u5eDVtk01VbL3nU
UYJmSSOf9tCi66vzQWyHyR3PNOWFyGA/sll4xxY9mZX8C+/UUYJmWRnJTYYZkP2+YsGSZl8fD0qT5vY9uiTL
YBYGB10XpmVv+OF1dXM39HMdtHQfknK5kH3H0PTnLueopZRvbqoWw/+Qv3l20sZi4V7mBXzFw2jQG+67
zI0Qt6Nv3pCK8Uv8Uj//t3HJ61nqxv1rL85GT2fLWVNC9Oif/9jwX8+94Cp+89+fgcnnxijueVdBNPz1hx9rgMl5
/Rp5NlMHPwQjUMqeB4STYP/etQtAYz75ys4OX7Kjl61sD0dPuI4LjF7/Hn1UloKWPDY4vYcOQRNjmJZ/lWxz
B8CVnxp9mY/2OzeqhcY4d/OsJL+19j9lv2eD3q94i8Bvwc+Y6kHHDQx/Zh8TNzqfz+bxiNtym/cAmAtBHD0Gg

1DBucirarDzC8fZ49W87b/3vQA7yQk8hf9h/hPxnJzLwM7m2p5qtPv+LA4VgWz/qJfbnG28RMncsLA1VQc6J
tRc2c+8z+uccfvp9A3FytDJNHU2WMH8vihXntl7odFi/M83laQ7ApY4agG/kGP1zehLXZ/QREpW4I0cNf6/KjB
JJnrmV361peX7sAw/pMXt60noWjx/Lyya0wYRGXVhexKTcJMDqvKY5rXSdg3K5izuiBk8XACS71K6S4JAXqj/
D6/CNkfVTMmVSoyl/GI7/KZ9xHudT8YRmrq+ey58hW0qhg24oFwH/r0t8SriIp95wJi5Elh5gYHXlz56DTRXca
VbizUVqyMK9L65+/vpiSI8WUHHmLmclQfmwPVTm5PD5aDyo9aTOmMcWP/d+2KJP0iQZmbzCz9INlZLmo
+VMeNjBIoyF5wkSyqKSqBpKHTyStegv/Z8XHHPXi8tCTT8zhycc7HwzD7SA5e1YONpuNNe9v7PB63rw5pN4
7iKTERGbf9eDRd9dtxGazdXrdLonRGSkcP1FBzZkTHDRMZLxew/jMHKpKy6m6VMHR0QbGDwDQYirfw6qlC
5g38zk+qAacjjhXcPSPlTz6wvNMSdWg1WeycOaPX9L40CPTGKTRMMgwrT1ezih9OiKawcO8Sk/ri5QcuJtW
oyFtxDAyM9LRx/ZBH9uHzIx0xo1J6zCPyWd97ifnqVwWPJXLgqlD4GYll26pGDYxk8E6Faq4VP5rWjJcq+aS4z
OxKYwd2PHsxHjmM47cSGTarEzucVMhfYm3p5N333LHP5MnZbH4jhq6eGFe2D7LThkzBG3qP3tcTpv6z34+
c0nDsNwV7DlSyIdzf2Dj08+x66qz5TzUlBnZZHWYDmtg+oP2WlJTUsjB+hJWP2afRP7IOyVQVk0NFXyVX82ju
bmkxQPxo5g5w+B2bwOZL5J73gmbkSWHmBgdGePTOXn6x2E5R6ME4E9Y7XOK7njBbAa95scTfAsuL7F4w
z7BW4s+3pcz42puNQETFrDngIGjn+/gD0/nsOF/bGXPwlFuP+k4IDpGF8LxIHlf6iDy5j3Olm072fD7LSx+Nq8t
xmM7nU00NTWz5aOdXP1bFYsX5nFf6iCn60ybkI12SzkFqYWkTf7YPqF3RDppJRXsGl6CNvNthgHG4rU8tUH
Puh0fMyW+lr1LczjsdI1mfqiHuJgfv+UqeUASXHO2bFu8nFCp1FitLjqgvokM5BTf/wPoB5ivcOGUhhFLEp0v3
7Y+TyIhB1wxmc3tE1VNZnPXR5QcHHOWHL//AKCi09xVpRI1LvpmsD+lvvU2N4zwEzeP/XId7xhiNXCt4RYQB
9RyswFih7ier+VNrAPpzgNUuB+svLnLza874e6kSSHrhZW8XDKPo6W1zE/t+Lb3NcWFlOfZc2DRXfMpyzjU1L
H+mCzuz6Tc1hofSe55J6xGlhzSRg6nqamZK1erOowoGSZltd9Z1RWmW0aM9W0/TTAsPRftgXwKzhoxmWs
5vmcHrm5a1yenAEZuuBn0sTdjfhT8+KFMmbuCV+brKT9Z6fK20Ts5RhfC+SDpOJsoPVXGmvc3Op3EX37+Im
ve30jpqTLPZxtjJjKnaSsbPoCfjRpqf21IOjNSSjj0eTVzMu1NpslYgyk+iX56MF0u5nDJj6voGM9R/OwxDbv+73q
OXjVjMldz9JtKn/9OTZSb783TjOfBOVCwq4jvm818//mfOHBfHlPd9MNu13eHSMgBZwoPHKKm9iY1tTcpPH
go8BtIGMp9sWbO/eUE3zdbsNy6yl/Ka9DeOxR3U19jxkxn2kAz//npYc65eQyN6/gkMThFy+0r57hw28Ttv1
7kakssgwfHdmFdwTN5UlZEHKysTZ6fAeLNMq7Vcmjzxxy9XIux3kjVN3/iwNkUxo9IAmIZNBpu1Nursbua4kl
yuoFx1VvYsLcSkwWor2DXgTJgCONyNRTkF1Jeb8ZUe4Jd20+4XVcg80VyzzthN7IE9mYJYMPvt1BTW9fp0pvj
8QGLfbwct2tZNrscvyzcypmXnmfd3Fd5/elsfjPAwNJljzGFb51+VjvhEX4+6jmWP/hnfv5RCS8F+BJrzecryX2tC
CMakifksu7tnLZboD2LhAPk5LY5aB/+207WrNvIfamD2ifpl54u48rVKpISE/nl8qXt8XdJk05WDuwqyiWr/Q7h
oaQZqqnKzyEr3d6wJs9YzEt//AXzJm5l3Au/4CED7c1wp3i+ks+7a1awar6BGpKY8tpWpgz17W/UaDVuvjhXQ
8aS9Sx5ewVLZq5DM/oxVr6bxwgX89OUSqVPt/NGQg7cqerv16ipu4nhgZ+i1Wgo/uoYVX+/1vU74ZxK4MGZ
BkyffsOBnedBqSJuYAYzpw9B6/ZzOn7y8DRu/PFTjuw/QcITzi/HuYv3gAkZjDGW8u1nh0EVw+CJk0h38bQSX
2Pd29w9X0md9E8AHR5Q6WlOk3sa+kWfYdW//JZL9aAdMJY5q9fz7GiAoUzJm8aGFQtIP/kWxe+4rikepc7l/
T8YWbVyAZlvmdEPmcbjr/wS0PPQS2s5/b9WMm/aepKnPs8rTxjgGzd77LbWeE9yz3uKph/MAZ0KX3n5Mk
OHuHnAjZdKT5XxWfGXZIwb22kyt2O0aUb2VPLmRdZBIlzdGbfm4aM7va+7eDYgsW1qaubrY8cpPV1GRdsI
06iRw8kYN5bJk3rG1500Nvl/SaBPTJ9uL2LO4usqF/xlvN3I9t372pujqr9fY/Ezc/2/FNfNAhHvrsY6ULXWH5W
XL3PPQ52/lubOHPF3P5vPr8JiPIs66Z/QpMxrn5tkNd/AXL2H1tovUOlHoxv5Rpe3EYkk91zvQzDqVliOLAFO5
7Q4OEaZSk+VSbMUYWJidDw8fWqX7maMBBqthlZLq1/f2aTVanv82Z4+tg/ZD07iROkZALIfnBRxjRL4H+/eE
Gt/aQbOQzGkf6cJ3EpNP6KHvIg1ZR420/UQ7V3oSO51r7Btljzp6h1xQgSb45u8u1LEtFptr/km8LQRw0gbMS
zUu+G3rsa7N8XaH56exq3U9AO/7oSLXJJ73SfgzZJCocBisdjvKBERwWKxeHwWEkhsfRETE4NapfZ6XoFSqU
QXrQvZmZ63OQBgs9m8Xra38CXeoY51oFitVq/rhuRM8EjudY+AN0tqtRqz2YxOF9lzT3oTs9mMWu05FSS2v
tFoNWi0Gswm+y3yFktr29NybYACpVKBSqVGExUV8uLlbQ6A/Sw2Ojo6yHsUeVzHm7CKdaD4UjckZ4JLci/4
Ar41nU5Hw60GOaBGkIYG7+Ilse0aRyELZ97mAlDRaJQDnxuREO9AqK+vJ8aLnInR6SRnuonkXvAE/DlLfePia
G21UFfn+juvRPioq6uj1WKhb1ycx2Ultj2TLzkA9gdINjTcCvJeiXBWX19Pq8VCnBc5ExcXJzkjAsaX3AukoFyGS0
hI4ObNmzRVVdE3Lg6dTtftQ2bCtdbWVpqbm2m4ZS9eCQkJXg+nS2x7hq7mAEBiQgJ1N29iNpvQ6/VotVqZj
9ILWK1WzGZz+8EqMSEBlUpFi4fPqVQqyRnhF2e5p1R27zO1g3KUi9HpUKtUGI1GGm7dorauDpuzb0sVIaF
QKFCr1URrtej1ejQ+3LItse0Z/MmB6Oho+vfrx61bt6ipraW1tVVyIEQqL/vzMEbf3JkziQkJqKOivJ7gHR0dTf/k

```
ZG4ZjZIzPUTIci8xEbVa3TOaJYVCgVarRaVSYbFYsFmtyD+L8KEAFEolKpXK51EhiW3P4E8OKJVK1Go18fHxWK
1WrFarHPh6AYVCgVKhQKlSoVKpfBoZUiqVRGk09pyxWLDabJIzwmsKhQKlUtnhp7sF9fqJWq2WSzQ9RPPw
0R2egCqx7fmcPQXXIVQFS0Qux+gAUjdEBJJqJ7zm7uApeo7m4aNdxjoQX3Uiej6pFSJUgpV7Af9uONEzeEo4
OWj2PN4UGYm7cMZd7kjOiO4Q7GOWNEvCJV86dCmIkcfXMzCJsXDF21ySHBKB1J15J82ScCvQQ5pSLLtfIGI
ocROedCXPJK+Et7paxwKVY9IsCY9k/kHvJQcz4Ytg1wrJx54rGLkTyHyRZkl4TZqm3kMOSsIfUitEKAWjfkmzJLp
EimHPJE2SCCSpE6I7BbN+SbMkAkKKYmSS5kh0J6kTItC6q4ZJsyS6lRTL7iONkAh3Ug+EL0JZ06RZEkIIERGkue
pZIumETpolIYQQQgg35OtOhBBCCCHckGZJCCGEEMINaZaEEEIIIdyQZkkIIYQQwg1ploQQQggh3JBmSQghh
BDCDWmWhBBCCCHckGZJCCGEEMINaZaEEEIIIdyQZkkIIYQQwg1ploQQQggh3JBmSQghhBDCDWmWhBB
CCCHc+P8uze3szLlmngAAAABJRU5ErkJggg==)"
    ],
    "metadata": {
     "id": "1l-TzjzyzLsv"
    }
   },
   {
    "cell_type": "markdown",
    "source": [
     "# **Don't Forget to Upvote NoteBook on Kaggle and Receive Updates** \n",
     "**[Click Here to Visit Kaggle](https://www.kaggle.com/ybifoundation)**"
    ],
    "metadata": {
     "id": "-anszVM5zLsv"
    }
   },
   {
    "cell_type": "markdown",
    "source": [

"![image.png](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAA+8AAABgCAYAAABsdEO1AAAg
AElEQVR4nO3de1zUdb7H8RczDAMDjaKCaJApEF5QU9k2dFcts2xParu1aaXWsWzbblu5rh2z03q2PNv20O5
Zmtaxdlctt7K2Jc0yUylDyUteFi/RKKKgyITDDMPMnD9mQFBALoOM+X4+Hjweyvzm+/3/3+fj+/8vOf7/X2+YT6f
z4eIiIiIiIhCxDWw9ARERERERERBqm8C4iIiIiIiS4hTeRURERERERERKcwruIiIiIhIiFN4FxERERERERQlxCu8iIiIiI
iIU7hXURERERERCTEKbyLiIiIiIhDiFdxEREREREZEQp/AuIiIiIiEuIU3kVERERERERCnMK7iIiIiISIhTeBcRERER
EREJcQrvIiIiIiIiEuvK0HICIiIiJyPjmWlNrWQwg5HWx5bT0EkZAX5vP5fG09CBERERGRHyuF9aZTmBc5ncK7iIiIi
```

IiEgrUGhvOYV4kZMU3kVEREREgkihPfgU4kVUsE5EREREJGgU3FuHrquIZt5FREREIKisQFTs8in07UTOTOFd
xERERGRFmpM+FTwbJwzXUtdRzlfKbyLiIiIiLSAwmbraOi66prK+UjPvIuIiIiItIIOtjyFzBbQtROpTeFdREREQCTIF
DyDo77rqAJ2cj5SeBcREREaSaFyNanD0JE/BTeRURERESCSGFTRFqDwruIiIiIiIhIiFN4FxERERERQlxCu8iIiIiIt
Jo2mlapG2Et/UAREREREETkzByOco6VHKfUXsYJhwOnq4LKysqgtd++vZXjx+2NOjY8PJxIcwTRFgtWawyx7dsR
E20J2lhE5HQK7yIiIiIiIezQ4SIKCg/jdLro2CGWjh3a0y2pK5GRZozh4YQFqZ9P12Zz5dDMRh3r8XhxupyUlTkoK
bXz/YECoiLNdOkcT5eE+CCNKLj27t3HF1+sY3NuLnn/zqPg0CHKysrwer1tPbTzhsFgICYmhq5dupB6SSoDBwz
g5z//GcnJPdp6aOcEhXcREREREkRBUVHyM/fk2zBERdEvsSqeOHdp6SNWMRgPRFgvRFgud4zsBUHyshIMHC
7EdPMTFFyUSH9exjUfp987yf7B06TJycja19VDOe16vF7vdjt1uZ9fu3XzwwYcAZGQMYty4m7jxhl+18QhDW
6uFd5/PR6XHQ6XHg9fjxevz4fP5uHzPU/SK7MKFpvYkmmMZfEEyfaMvpF24ltmIiIiIiADsztvH8VI7yd0vCqnQ
3pBOHWLp1CGWo8dK2LMvn2PHS0lL6U5YWLDWBjTNxx+v5Nlnn2fX7t1t0r80Xk7OJnJyNrHwtUU8+OADX
HPN1W09pJAU5gtyxQmv14vL7cbtrvv5mz7//p/TvneBIZIH4q/groShGAyqoSciIiIi54ZjSamnfa8l+7y7KirYvvP
fRFsspKX2CNqS+MZoyrL5xtidt4+yEw769EwlMtLcoraaep0fffQx/vb3JS3qU9rOLTeP58kn/9TWwwg5QQ3vT
lcFFW539e/DjUbCw40YDQYMBgNhYWHYPeVsdxRgcx7je9cxVtp3sMtZCMCFpvY8EH8lt3T+abCGJCIiIiLSaoI
Z3l0VFWRvzOXiiy7k4osSWzq0Jgt2eAfI//4gh4uK6denZ4sCfGOv8+HDR7jv/ge0RP5HICNjEC++8DydO4dmD
YW2EJTw7vV6KXe68ASKPUSYTESYwhs9i77Bvoc5h1ax8cR3APSKTGBZ6m9ob9JSehEREREJXcEM75u2bKdj
bPs2Ce7QOuEd/AG++FgJA/v3afYS+sZc58OHj3D77ZO1TP5HpGdaGm+8sUgBPqDF4d3j8eBwuvD5fBgNBiL
NERiNRgB8pcV49u/Ac2g/vpIj+Jwn/J1GRhMWG4+xS3eM3XsT1s5f5GJ9aR5/PPAhu1yF9DQnsCz1LmIjolt4ii
IiIiIirSNY4X133j58QM/Utqu63VrhHVp+fo25zr++abxm3H+EMjIG8fYyPQIBLSxY5/V6q4O7KTycqMBSGO+h/
bjeeQH3+hWNasc0ZAzmG+9nSJdUlsfczQ27X2GXs5Cb8ubz9iWagRcRERGRH6+i4mMcL7VzWcalbT2UVpO
W2oOvcr7hSNHRVqlC/+ijjym4/0jl5Gzi0Ucf0zPwQIuqw5XXEdwrPl1K2dRRjQ7uAO71KyibOoqKT5diNUaxP
O1uekYmsMtVyE15r7ZkiCIiIiIiIiIW1/vo3k7hed1eJ0bSGlRze++/5A0Nv9+OOVKk73I/e3vy/h449XtvUw2lyzw
7vTVYHH68VoMFQHd9d78yhfNIuwDp2hfTzUfObdHEVYxy5g7QiBZfW1eD045z+K6715JwO8OYGdzkL+dvir
5g5TREREERCRkHTpchDki4pzZDq4lOnaIxRwRwaHCI0Ft99lnnw9qexKadJ+bGd69Xm91VflIcwTgn3F3LZ2LIT
6RyN/8L1H3PI3hwhT/G8JNhA8aQdSDzxNx3WTCrPUvlXEtmVM9A//HxOsAeP7Ip3gDxfBERERERH4sCgoPc
2HXzm09jLPmwgsTOHQ4eOH9neX/UIG688Su3bt5Z/k/2noYDcrJyWH4FVfRs1c6Cxe9jsfjCWr7zQrvrkBwjz
CZMBqNeA/tx/naf4PPh6/oIO7VSzHEdcV01S0Q0x5DYirhmb/A53RQuelTfCUN/4F1vvbfeA/tZ0i7VC6LvpiD
7uP8vWhjc4YqIiIiIhKSHI5ynE7XeTHrXqVTh1jKnS5OnHAEpb2lS5cFpR05N4Ty/bbb7Tz3/Ivk5+fjcrl45pnn2L
Zte1D7aHLBOp/Ph9tdCUCEyf925zvPgzfwqYLbReWOLzF8lUX4gOGYht1AWEw7jPEX4fpoEd49W87cideD6
50XiLp/LlO7jGTcngW8cOQzbon/abO3lxARERERCSXHSo7TsUNsWw+jeTwu7HYXGM1YrU3bv71jh1iOHS8l
OrplRan37t13VovU9Rkxmit+dgWP4jExn8iwHE1Xm7XBRtXs
mG/Dpe6jucsSkXgKuYzR9lYxvxOB/+ZQzWzS9z65SFDfQXfDk5m9i7dx/JyW23K0NbanJ4rwxM/YcbjRgMBn
yuctzrVtQO1fYS3Ovex5iYQsR/TMZnP4Y7dw2VtLomIp9LjwRTeogL5I
iIiIiIIhodReRscO7dt6GE1WtO5lHvzdM2woAYbPZuOiXxPXhPfHtrNSfLQELmzZOL74Yl3LGmisvvew6KV7uCKx
KvWOZtxvH8e1dyWz77yXxXUF3rPkir9kMe/GZMwAt03gvsKVzBl8LwvqPPoa7p39CIOtdb1mZ8MTK9mwqI
6XJj7Cszcmgz2b2R9ls2BEJn1irTBwGFewkMU/u4NZv+xDO2xkT32GpcE7vTp98cW6kAzvVquV3z1wHzbbA
QoLC3nood/Rt296UPtofngPD+zlfry4jqN8+Arzca99j8ge/fA67HgL9mLo3A2vqxc5Y3qy3voO4wp/Xmoy1U
4XRUK7yIiIiLyo3HC4aBbUte2HkYT2PlsxhVMXmLHbDEDrma1EhNjIdd92sMWj2Zyb2+I2zuwO/vrGQwyOBT
wuir47QClWkpLjMCdfzYzXZmMbOYPPzsJITvcIU3+ZjBk7e9Z8ib3nMAYmXM3YmbDgiYbf6Srci81e8zt2bCW
N7HbGwzzumkDc5pdZDHDJMK7/ZSZW9uI8C+F9c24ut98+qZV7aZ6MjAzWfPZJq7Xf5GfevR5/4ThjoJK8r+x
43QdWuvEWF+AtLoCy4xgv6knEiPGEXdD4pUHeo4dq9VXVt4iIiIjIuc7pqiAysmlLzutie2cK/ae8jS24tbHq4OJI
USLj/vI+GxdMaHYrkeZInK6KFo8m7995LW7jTAY/N8Ef3DlA1sOjuGzkKEaOHMzIP2djB8zJ13Dn5KqjezNp9
mI+XLuJXd9sYNWyl5g6OrG6rfvmZbEqxuzFjAjcMzAaQv4MCuLVctnM67qTd0eYlFWF
quy3ufZ204dnRWzEbB/y7LJ9/LiFn8aNzfiaQTbulGMHFXz6yYeeTfwYt8JPLvsM7Zs38Tavz7O2FM3Crv/XiZm
DmTUxHsY+8vZLJ/YB/9kfiJXZmUx7/4z998SZ+O+h6omT2N7fT4ADFXh3dmYYhNhYDCCsWnd+X4oqdVXVd

8t4S6z4yYKS4zp5DcddhzewPfc5Tgc7pOvmaKwWEy132+yYqnr71lXOQ5njfdiwnRBFKaaH5E4bOzZXoi7U3
dSLu5U67XT2va6cfxQebINVzkOd3itsdd+jxt3aTnVIzCecp7UbLccIk/tq44VEVXn7y7H4QBLu6jar7fkfJrCYafm
bal1bV3lOJyc3m593z9FnT8T1W0Uk79rP6WmJHr1TKi7nQauQbPUdX9qDdh/L2qfV+DeV73nlJ8Vd5kdd61/
0E3+e3lqW6feo8BYTBYrpqrL4ypkzxYbjpgGromIiIicUWVlJcYWriq1vTOFkX9Yg4s1jLwbVr3ya5Lq2JU5OKxc/
9z7jLMAOSua3YrRaKCysnGP0jak4NChFrdxJmPS/MHalbuC335w8hl32/x5vJhUTB8LgRnsRKYue5P7Mvwx1
uWClIyruS/jcgZ2Gcut8w+QmJJMSjK4Eh9hYNV/8qzDmTI3hXjHFTyY/QPW3yaThJVrfglL3wUmDueKS5LBkU
vW/506ulxsRb8mJW4gE9/Pol3fOHDt4LNXW3DC3e7gr288EvjAAqyZE3jKdcoKi6RkUi5JBvsR4mMhpVvVOn
wzcZckk7K1Bf03wtm4782Vk5PD76c9QmFhIdOmTeX22yZhrGub9GZqVsE64OQz7pUNfGrm84Lbia+yAiorC
LugPeH9h+IrL6t9nLsC7+F8vAX7oPJkQvNVOGv15QtCeOffS/j9K6VMemoal8cCjvW88uBbcNtc7h5igm1vcs/
L22kXG0UEbo6XVJJy4wymj0oACln751lkX/44M69LOK3polV/YdqKo8RVB9wujJ72IEPjAcrJf+cZnsiyYWpnxe
Sw47D8hIf/NJleFupu+8gnzJ25kcwnHmdEQqD990oZ9fBcxveu6z3bWTJ1PmtjrLQ3uTleUg4Jw5g+YzwpNT6
BK139NL9basPUfzIv3/8TTADFG1j49Ed8D5wo9Yfu9hag/0SenpAeuC4w/bW76EXjz+fNsiHMnDuBFMPp59
MUe/7xJK9W1Tr0lnO8tCPja12XQnqNe4rpI6v+8nCzaf7DvLAlgYkN9VfyCXOnLWenIZ37X7iXQTXCcmn2fJ54
PZfjFivt3XaKvElMnDmDEdXPZ53pGjRPwdv/w4xVxaRcX/fPGca9rHjsJQrGPsXDIwLna1vB47M2kjnjKUb3CFy
TLy/j6SeuJS5wL5aUBO4pAAP4zdPjSfFuZvHU5bSv+pna9ia/e+FrBv3n8/4/D9+t4PHZ2xkVuIaOzfP5r5e3Q2
wUEQ47xyOH8PDsCfRq+aSBiIjIeaklpZhPBnc/1+oZrRzgzY2a1T1bysrKznxQi9xBUtV/yw99ecpr2Sx4LPvkbyf
PZlKGFbCz4c9juXW+lfuWvcnUDCuDb/8vxs6/t/pQc+UO5oy9iRe5h+VvPsRAayKj7nkIbnibzfmjSeoWR8qITH
g3m6kDkgFw7VjDnNPG9zZzPprA4Nt6k9Q3Gew7WPzYvcxuxDP4KTfmsf/GGt/Y+zbdR85g8MOBlQaeA7z/
8EQe3DqAZ9+Yy9hu9TS0aBL9WcyWmf5l80t7jOKRM3ffIq1/35unZrV5gGeeeY5BAwdy6aX9g9ZHqz5A7jt
WSMUnS8DrwZCUiqFrD0xXjiPMW3tNj8/nxXe8CNd7r+D9bsfJyvWtwDRwPJOSp7N4aS6D7h5A/jvL2XThG
GYPqTmjnM7dT/tDatF7jzFtXS5Fo65tXCGO+Gv5wxN1HLtjOXOzyhk1/XluSDWBt5DVf5rF3IV9TwboRikna/5
bZFQF4joMnfQUEwcCbhtZs2czd3Eaz909INCHnS1f2UhJ7UHBlo1sdf3EH1jjh3H/08OAQlbPnEX25Q/VHRyb
ej5l63nhrZ/w3KS0Rp9hXVImPMXTgdVZRR/OYtq6dHrF1z5m5+pVFIy8ga4AJZ/zSSM2NijN2cjOhB70s2/n6x
w3g4YERl7yOQtfz6XLr//E0yM7AeVsfflR5r7+CRkzr6JdU65BUxxcwaLVVvpdXEy9a1oM6WQMiOKJnFwcI4Z
hAQq++oaCTpcxqIHaHd2ureOemvtyaepbLN+2j/G9e1CwazduYNOu3TAkndJduymKTadXAkAh2f/Ipf11jzPr
+gTw7iNr9qtk55TTa0hUHT2KiIhIazk1uFdp/QB/PnFBIJZEmjs1eOTYzBT/0vEDX/LSfP8M/YtZ3zIlIxNrQjKZNY
61f/sxL24DeJnFX09g4Ig4zIm9GcszzPl6L2O7JZOUNpokejMw2V9b4NsvXj6tz6S7FvP6hN5Uz6GYzZgdB5jy1
03MyLSyZ0kqI2fUd2ouXDUjl8P/k5SZGEgxuz/mwQ8OAAd4MPtexnZLbvD85exo8oLX02bBwyPqPdZXepTK
7H9SuW09GMLxlRZTmf1PXCvfqvXlXr8Cw0U9MaZeCqaT7YVFRNbqKzjbxFm5/LYxdM35B1nrV7BkbTRjJp4
att2UltpxlO5j05ZS4nr2aHwFTa8DR6nd/1V2chVBwbZtlKZexbWpgUhnSGDEtQNwb9vGnqYMP2EIo7qs54X
Xt+M+07GmJEZdOwDH5tyTfRSuZ+13CWTeNoahFn9gbY7Gnk/KkCF0W/cqb25pXj+nceWStbKQy8eOoWv
Nn96EHvRzrGfttsD4Vn9O/sVJ1PchoV8hOZ/bSLl8ImMyovgyZ2P1NXXv2M7WyCGMGVn1F3UU/e6ZyxtVw
Z0g3tMa41n92idw/USubfjfB1J+OhBL3ka2lgLY2JpTTNeMy2io5I37ROnJn83qZxCsJPdJoOg7Gw4K2bnNzoih
QzDt2k0Bbvbk2bD0TQ+025H2nSF/zRKyNu6j1N2DUTOfYrKCu4iISLM1Z12pbUndwb2KP8CfjWfg21ZMTEw
r9/AW3wZWaMd1GUhSrdeGc9/suTw7Zy5Tb4T4qmXwrlI2NKGHzfbAXTSbiQdsL2XzrQe4uA8Tf3k5qVbAsY
PPXjj1naOZdUcmcUawvfsws7PtYE5m3F/eZ2yKfwXAkd3197vng3R6ptf4GjvLf57R/vNw1VwqX2RvZmnC1t
H69715qqrNd+vWDbPZ3CrV5psc3g2BAO31+ovHhUU2Yu1MmIGwcBPewzbcX/2Lyi/eq/5yr30X9xfv4fuhh
DCLlTDDyY8Iq4rbVfVlCNYe7wnXMn7oCZa//i+KMscx6rTZyt38/YknefyJV1lR2I5+l3RsfNtHPuHxqdO5Z+p07
vnzJxQFvl1aYod2VmpdrXDA29TnfToy4v4J9PpqEUs2N6LewCl9FKzbwJ6Ey+iXkEZGRhRfrttQ/wxvAxp9PnFX
cfdtyax9/XV2/tBwmw7bdnbaGt6JoGjVClZbrmJM5qnz2ukMGxrN6jVf4/ZuZ+1aGDNiQMOz37b1rC5MIDM
jgZSMgVi2bWBTqf+l4yXFYO0YCOrbWT5tOtOmTWfatCXVwbzJ99RbTP7G3ZTWU3exdPWbLOFafjOqEc8U
9P4JQy372JBjB9tG1hYnMOJnSQ2+JX/Vs/6fy6nTuWfx9urvx/VOJy5vOzsLd7OlMJ3eN/Yhs9T/+x3fRjF0QN
WqCROD7v0T06+ysGXps/z+3t8ybe7nFKmOpIiISLOEh4fjacaz30njF7BrXx77G/jatSB0Z949Hi/hQdhBqmuXL
kEYTcPe/HqH/xe9x/Dn+3tXf/+KvzzCfeNHM/aXl9MHWLAz8Dx8lz7MCMweDe6f5J+Nd9irMwGAtWsfBgO

QyJ2BZ+opPODf3i1/IZt2A8ZkBt+RQhxg3/IxL542sj7EdwCwY9v9AQtuncicbDvE9qZPHFC0mRWnPSN/ZjuO
BIreJfauLpo3JSOZUHpC8mzc9+aqqja/a+d27pj8n0F93h2aE96N/rd4qsJ7TBP2pvT5wOfzz6THtAdzFMYe6YR
17AqG04di6NilVl9VfQdDytUjSaET145KryPgpXP300/x9NNP8fI9aWTPX8qmxn7clDCGp1+bxxuvzeONGsvn
OyUmwN59FNQ4tPRoMVgstfp31/z72+Oue3bdMoRJd/Yke/ESss8wrtp9+GdoObqKv0ybzqubKyFvPTmN3R
aihsaej3+4k7n/kl288rdcSutt0c7WpS8xd+Xe+jutOetex8sXjRhGypZVrP7oc1ZbLqVf94bPoeCrbyjgKCvmTGf
a65txs4/PN/r/worr2gVT4T6+dwGkMWbmo8wafRFFpY7qe9KUawDA3k+YO385OXXtruj6mr8vtdGrTycO5
XzNnmNQ+v12dh6sb8VCGkOHWNmak8vOr76h4OIhZJwh86dc/7j/5/K1ebxx94CTL/RIZ5BlL3n/2s3W1HR6
WdLp3beQLe9tYCs96d+n6kA37h8iSLnmLqbPeZ4FcybT67slvJpV1wmJiIjImUSaI3A6Q2lO8+xwupxEmutfvd
tYqZekBmE0DbPNeoale12AlcEPvc/+ndvZtTOPRYG91e3ZC3n8HeDNFWywA5beTFr2Pn/9axbzfuEP5kXZb9
d+Xr3baOatzWLV2veZ1BvAxbfrqpbFH+C1r3cAZvr0TgSK2PTuwjpG9jnbvyMwrg2syprLTT1PbuBuL9zLkQb
OK2X0dnZtr/n1GYtug8UrNvs/aIgbzqyvsli1agNTL6tzY/iTSlyBmflErlzV+tXmz8Z9D1VNTsPhgU8PKiv963DC2
p9hfW8NYdZYjH2HYBoxHtPIWzBddTPhmf9BxLBfEdbu9IXphi4X1+orPMifXEA4poaadJdTcPAQbhy4nS3rK
W7IcPqVfsKipbspdbtx29bz93/a6Dp0GCkAJJDSM4r8NSvYWuwGVzFbP1hDfkwaKXWEMstlE7m/ZzF76gjeJ
8r8S6OLdvyLxf+0kTLqKn8fgRna8dP+yKyZjzJr1l2MjrWRnWM/vZEWn09NUfS7fSIZJbZanzrWZuXy389jwR3
1Ly2pf9Y9IHYYV/W3seS97aSMGNngEvKqDzJSfvEg/zvzUWbN/CMPD7ey86uN/g8YLh3GiJjtLJ6/ngIH4D7E6
s+2Y+rbt/r8mnYNgNTxPPfaDEbE1/GaM4qLf3opcaXb2LJtG3l2OFG0m/zi+lcidB18GV3z3ueVdcX0+ull1cv56
1Nr2XxpzXbT6N+7nKz1ufQaOAALJvoNTGdrTi5FffrUqK2wj+WPTeepD2y4vYDXX6zSEoR/fEVERM5H0RYLZ
Seaswby3FZW5iDa0vLKdwMHDDjzQS22hkfu/C8WZB/wPyNuNmM2Ax47ez5+hltuXYgNIH8ht059i29LwBz
Xm8GZyViNLmzZC5k+5e1aLdq3rcFmTSYl0Qq4sK1+ht8+UaOS/azA0nmAom/5+J26xpXNIw8/w2cHXGCOI+
WSZJJiwXVgB98WurD2vYNn/3oH9a7LNJsxW2p+Bbade2cK0/9vB3YPmOOSSbkYNq3wb4tXr3cXsTS7CDATl
5zM4J80fxvBxjg79715cnJyGH7FVfTslc7CRa/j8QT32ZUmr1epDu8eD16vF4M5CtPPxlC5/oMG3uXD5/ViuD
AF05DRUHUSnkoIM0C0lbBoKz6fB1/gyR/TkDGERbfD6/VS6Wmt8F6fXJ6687f+X5qsXH7zQ1x+pmR0JrHDu
Huag4WvPMvvVgEGEylD72Xmr07+kep2yx+4+8gzvPDIA7gBU7s0Jk+9oZ7ntqPoNWkiI3bN59SCkl8uns6Xg
MnciUHXTuPhX/jTf/66jRQkDGdQj6ql3ulkZlj54PP1FI1sZEG+JpxPLZYB3DBpADkvN3NrB8fXfLCykMtvntFAK
DcxaPgQLNtKuWqoFRpaUZC3ntXFCYwa3ANL4N72GnwZcWvWk1N4FSMS0hg/cwKOp5Yy44G3AIjrPYaZd
9QoRNfUa9CQdumMqvHBxc5XvmZ54g2M6t/AJ51JlzG00ycsKU7ihowzfCJKYNn8qqrfDaixcwCk9E2HnGIy+
vrbMaWn04/tkN63xiqCNMZMGcbcV2Yz5SP/d+J6j+cPV5y5bxERETldO2sMx+0/kNC5Sf8LCw0Zj7B/X/Pqip
eU2rFaW/7c8s9//rMWt9Eo+R8w+9YPmA30GTGaFPby/uodpx+3ehbXDZpFUubVDEyoYM+7a/i2rvbKvuS6
S6fQZ8TVtNuzkg2nVYffy5Fj0CcOir5dydL6xrXtZSYPfRn6DmdsSgRFm+tqq6aF3HppXbP4tX02ayz9ZyUyeEQ
SttXZ/g8nptY44A+j6P6Hmu/IZs6tg5nTLZMrutr4LPsArems3fcmOhvV5sN8zdh/rdzlwu2uJMJkItIcgffQfsqmj
qq/SrwxnPDB12G+8QEM7TuB1+ffRi7AZzDCCTvlr0zHs+Mr8PmImZOFoUt3nK4KKtxuTKZwosyh9LRFK/K6c
btNmM6T05VzkKsctylKe7yLiMh571jS6Ut4O9jyGvVeh6Oc3G07GPLTQcEeVrN8ujabK4dmnvnAFlr/1Sb6p/
ciJrrxs+/1Xedf3zSenJxNwRxeq/nzqjzGJYM9+8/0v7XuED1l8Qam/iTOP7vv2svSKaN4ZN3ZHWcoy8gYxNvLlr
T1MOpkt9u5974HWLduPQDR0dG89eb/tf1WcWaTCbe70h+qw40Yu3Qn8s7/wTn/0brf4KmkctOneA/bMH
TqirFbL8I6J0FYGF7bv/EetuE9tB+vbTd4Kom860kMXbrj8XiocLur+zxvGBTcJcSZo5q3FZ6IiIhUs1iiiIw0U3z0GJ
06dljgPWUAAAamSURBVGjr4ZwVxcdKiIw0Nym4N2TcuJvOmfDucrhwOaDUUV+dgwEkx1vB48JVcoCsuXcq
uJ9i3Lib2noI9aqqNm+zHaCwsLBVqs03a+YdqJ4RNxoMRFv8W0W53puHa8mc+t9kMEK4CUNyPwwXJvvD+
56teA/th4py8Hoxj5+K+Xr/kvUTjnI8Xm/1DL+IiIiISChpycw7wKHDRRw5Ukz/vr3OfHArOxsz71u27SQ+riNdE
uoqAlS/hq7ztddex67dDeyLJj8KPdPS+Ne/PmzrYbSpZi96jTRHYDQY8Hi9lAeqZJqv/y2Rdz3pD+l18Xqgwok3L
9e/Vdzad/F+vwucJ4AwIu96sjq4lztdeLxejAaDgruIiIiI/Ch16RyHq6KC4qPH2noore7osRJcFRVNDu5n8uCDD
wS1PQlNus8tCO8AUZFmwsLCcFdWVgf4iCvHETMni/Aho/EFtoU77ctdgc/p8H9VugkfMpqYOVlEXOnfTbDc
6cJdWUlYWBhRkVo/LiIiIiI/Xt27JbF3//c0aznsOWTPvnwuvigx6O1ec83V3HLz+KC3K6HjlpvHc801V7f1MNpc
s5fNV/F4PDicLnw+X/UsefVm9JVuvC4HVDjxud0Q2K8dg4EwkwkiIjGYLRBuqm7L6arA4/USFhaGJdIc9I3tRU
RERESCpaXL5qvsztuHD+iZ2iMIo2qe1lw239Lza8x1PpeK10njhXKRurOtxbWijUYj0VGR1UvoT5Q7cboq8Hq9

```
/ufbo9thiO2MMT4RY8JF/q/4RAyxnTFEt4NwE16vF6erghPlzuql8tFRkQruIiIiInJeSEvtwQmHg+++b91tttpC/
vcHKTvhIC2le6v28+ILz9MzLa1V+5Czq2daGi++8HxbDyNkBGWjJ0OgaF1EoCJ8hdtNmaMcR7mTCrcbj8dDz
Ql+n89XXUneUe6kzFFeXVU+wmQi2hKFwaA9qERERETk/JHe6xK++/7gjyrA539/kMNFxfTpmUpYWFir9tW5
czxvvLGIjIzQ2HpPWiYjYxBvvLGIzp2DWyPhXBbUhBxpjiDGEoXJ5N+BrjKwDP5EuZMfTjiwl53AXnaCH044qm
foKz3+veFNpnBiLFEqTiciIiIi5yVzRASZlw3gaMlxdgWWmZ/Ldufto/hYCf369CTyLNWx6tw5nreXLdEz8Oe4W
24ez9vLlii4nyLo09sGg4Eos5kLoi1ERZoxmcIxGgy1PmkLCwvDaDBgMoUTFRk41mzWbLuIiIiInNfMEREM6p9
OGLAx55tzsgr90WMlfJXzDT5gYP8+Zy241/Tkk3/ilXkvaRn9OaZnWhqvzHuJJ5/8U1sPJSSFt1bDYWFhmMLD
MYW3WhciIiIiIj9Kaak9KCo+xr7vbBwsOMyFXTvTqWOHth5Wg4qPlXDwYCGuigq6d0siPq5jm47nmmuu5pp
rruad5f9g6dJlKmYXwjIyBjFu3E3ceMOv2nooIa3F1eZFRERERM5Xwao235BDh4soKDyM0+miY4dY2lsvICba
QmSkGWN4OMF6krwp1eY9Hi9Ol5OyMgclpXaOHishMtJM187xQd/HHYJznffu3ccXX6xjc24uef/Oo+DQIcr
KyvyFtuWsMBgMxMTE0LVLF1IvSWXggAH8/Oc/Izm57XZZOJcovIuIiIiINNPZCO9VHI5yjpUcp9RexgmHw18/
qrIyaO23b2/l+HF7o44NDw8n0hxBtMWC1RpDbPt2xERbgjaWU53N6ywSqrSmXUREREТkHGCxRGGxRJF4Y
VuPRETagirEiYiIiIiIIQ4hXcRERERERGREKfwLiIiIiIIhLiFN5FREREIKoruJq0ny6niJ+Cu8iIiIiIs1UX8VzBc7gqO8
6qtK8nI8U3kVEREREWoECfMvo+onUpn3eRUREREERa6ExBUzPFTdPQ9dS1lPOVwruIiIiISBA0ZqZYwbN+un4
iDVN4FxEREREJEi31bl0K73I+U3gXEREREQkiBfjgU2gXUcE6EREREZGgUtAMLl1PET/NvIuIiIiItCLNxDePQrtIb
QrvIiIiIiJngUL8mSmwi9RP4V1EREREREQkxOmZdxEREREREZEQp/AuIiIiIiIiEuIU3kVEREREREERCnMK7iIiIiIiISI
hTeBcREREREREJcQrvIiIiIiIiIiFO4V1EREREREQkxCm8i4iIiIiIiIQ4hXcRERERERGREKfwLiIiIiIiIhLiFN5FREREREER
EQpzCu4iIiIiIiEiIU3gXEREREREERCXH/D8Tn4pYJHtSKAAAAAElFTkSuQmCC)"
     ],
     "metadata": {
       "id": "v5iWFvexzLsw"
     }
   }
 ]
}
```