{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

},

"accelerator": "GPU"

},

"cells": [

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"import seaborn as sns\n"

],

"metadata": {

"id": "LjdlrQO9\_GNY"

},

"execution\_count": 2,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"from matplotlib import pyplot as pt"

],

"metadata": {

"id": "NGsmnRsq\_0dr"

},

"execution\_count": 3,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"%matplotlib inline"

],

"metadata": {

"id": "z9rAl9WJ\_GTE"

},

"execution\_count": 4,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"df = pd.read\_csv('/content/data.csv')\n",

"df"

],

"metadata": {

"colab": {

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

"height": 505

},

"id": "Uq4pmXfmAI2f",

"outputId": "68e470d3-0e62-4b88-a11c-cae5a206e00f"

},

"execution\_count": 5,

"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",

"564 926424 M 21.56 22.39 142.00 1479.0 \n",

"565 926682 M 20.13 28.25 131.20 1261.0 \n",

"566 926954 M 16.60 28.08 108.30 858.1 \n",

"567 927241 M 20.60 29.33 140.10 1265.0 \n",

"568 92751 B 7.76 24.54 47.92 181.0 \n",

"\n",

" smoothness\_mean compactness\_mean concavity\_mean concave points\_mean \\\n",

"0 0.11840 0.27760 0.30010 0.14710 \n",

"1 0.08474 0.07864 0.08690 0.07017 \n",

"2 0.10960 0.15990 0.19740 0.12790 \n",

"3 0.14250 0.28390 0.24140 0.10520 \n",

"4 0.10030 0.13280 0.19800 0.10430 \n",

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

"564 0.11100 0.11590 0.24390 0.13890 \n",

"565 0.09780 0.10340 0.14400 0.09791 \n",

"566 0.08455 0.10230 0.09251 0.05302 \n",

"567 0.11780 0.27700 0.35140 0.15200 \n",

"568 0.05263 0.04362 0.00000 0.00000 \n",

"\n",

" ... texture\_worst perimeter\_worst area\_worst smoothness\_worst \\\n",

"0 ... 17.33 184.60 2019.0 0.16220 \n",

"1 ... 23.41 158.80 1956.0 0.12380 \n",

"2 ... 25.53 152.50 1709.0 0.14440 \n",

"3 ... 26.50 98.87 567.7 0.20980 \n",

"4 ... 16.67 152.20 1575.0 0.13740 \n",

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

"564 ... 26.40 166.10 2027.0 0.14100 \n",

"565 ... 38.25 155.00 1731.0 0.11660 \n",

"566 ... 34.12 126.70 1124.0 0.11390 \n",

"567 ... 39.42 184.60 1821.0 0.16500 \n",

"568 ... 30.37 59.16 268.6 0.08996 \n",

"\n",

" compactness\_worst concavity\_worst concave points\_worst symmetry\_worst \\\n",

"0 0.66560 0.7119 0.2654 0.4601 \n",

"1 0.18660 0.2416 0.1860 0.2750 \n",

"2 0.42450 0.4504 0.2430 0.3613 \n",

"3 0.86630 0.6869 0.2575 0.6638 \n",

"4 0.20500 0.4000 0.1625 0.2364 \n",

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

"564 0.21130 0.4107 0.2216 0.2060 \n",

"565 0.19220 0.3215 0.1628 0.2572 \n",

"566 0.30940 0.3403 0.1418 0.2218 \n",

"567 0.86810 0.9387 0.2650 0.4087 \n",

"568 0.06444 0.0000 0.0000 0.2871 \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",

"564 0.07115 NaN \n",

"565 0.06637 NaN \n",

"566 0.07820 NaN \n",

"567 0.12400 NaN \n",

"568 0.07039 NaN \n",

"\n",

"[569 rows x 33 columns]"

],

"text/html": [

"\n",

" <div id=\"df-e9f6ce44-c133-4472-a33e-9291843a40dc\">\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>0.30010</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>0.16220</td>\n",

" <td>0.66560</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>0.08690</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>0.12380</td>\n",

" <td>0.18660</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>0.19740</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>0.14440</td>\n",

" <td>0.42450</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>0.24140</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>0.20980</td>\n",

" <td>0.86630</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>0.19800</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>0.13740</td>\n",

" <td>0.20500</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",

" <tr>\n",

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

" </tr>\n",

" <tr>\n",

" <th>564</th>\n",

" <td>926424</td>\n",

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

" <td>21.56</td>\n",

" <td>22.39</td>\n",

" <td>142.00</td>\n",

" <td>1479.0</td>\n",

" <td>0.11100</td>\n",

" <td>0.11590</td>\n",

" <td>0.24390</td>\n",

" <td>0.13890</td>\n",

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

" <td>26.40</td>\n",

" <td>166.10</td>\n",

" <td>2027.0</td>\n",

" <td>0.14100</td>\n",

" <td>0.21130</td>\n",

" <td>0.4107</td>\n",

" <td>0.2216</td>\n",

" <td>0.2060</td>\n",

" <td>0.07115</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>565</th>\n",

" <td>926682</td>\n",

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

" <td>20.13</td>\n",

" <td>28.25</td>\n",

" <td>131.20</td>\n",

" <td>1261.0</td>\n",

" <td>0.09780</td>\n",

" <td>0.10340</td>\n",

" <td>0.14400</td>\n",

" <td>0.09791</td>\n",

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

" <td>38.25</td>\n",

" <td>155.00</td>\n",

" <td>1731.0</td>\n",

" <td>0.11660</td>\n",

" <td>0.19220</td>\n",

" <td>0.3215</td>\n",

" <td>0.1628</td>\n",

" <td>0.2572</td>\n",

" <td>0.06637</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>566</th>\n",

" <td>926954</td>\n",

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

" <td>16.60</td>\n",

" <td>28.08</td>\n",

" <td>108.30</td>\n",

" <td>858.1</td>\n",

" <td>0.08455</td>\n",

" <td>0.10230</td>\n",

" <td>0.09251</td>\n",

" <td>0.05302</td>\n",

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

" <td>34.12</td>\n",

" <td>126.70</td>\n",

" <td>1124.0</td>\n",

" <td>0.11390</td>\n",

" <td>0.30940</td>\n",

" <td>0.3403</td>\n",

" <td>0.1418</td>\n",

" <td>0.2218</td>\n",

" <td>0.07820</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>567</th>\n",

" <td>927241</td>\n",

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

" <td>20.60</td>\n",

" <td>29.33</td>\n",

" <td>140.10</td>\n",

" <td>1265.0</td>\n",

" <td>0.11780</td>\n",

" <td>0.27700</td>\n",

" <td>0.35140</td>\n",

" <td>0.15200</td>\n",

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

" <td>39.42</td>\n",

" <td>184.60</td>\n",

" <td>1821.0</td>\n",

" <td>0.16500</td>\n",

" <td>0.86810</td>\n",

" <td>0.9387</td>\n",

" <td>0.2650</td>\n",

" <td>0.4087</td>\n",

" <td>0.12400</td>\n",

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

" </tr>\n",

" <tr>\n",

" <th>568</th>\n",

" <td>92751</td>\n",

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

" <td>7.76</td>\n",

" <td>24.54</td>\n",

" <td>47.92</td>\n",

" <td>181.0</td>\n",

" <td>0.05263</td>\n",

" <td>0.04362</td>\n",

" <td>0.00000</td>\n",

" <td>0.00000</td>\n",

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

" <td>30.37</td>\n",

" <td>59.16</td>\n",

" <td>268.6</td>\n",

" <td>0.08996</td>\n",

" <td>0.06444</td>\n",

" <td>0.0000</td>\n",

" <td>0.0000</td>\n",

" <td>0.2871</td>\n",

" <td>0.07039</td>\n",

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

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>569 rows × 33 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-e9f6ce44-c133-4472-a33e-9291843a40dc')\"\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-e9f6ce44-c133-4472-a33e-9291843a40dc 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-e9f6ce44-c133-4472-a33e-9291843a40dc');\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": [

"sns.displot(df.radius\_mean)"

],

"metadata": {

"colab": {

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

"height": 386

},

"id": "X28Z\_dAAAlrt",

"outputId": "fe396875-9002-466c-dee4-a16a2a53ad26"

},

"execution\_count": 7,

"outputs": [

{

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

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x7f05de4a50d0>"

]

},

"metadata": {},

"execution\_count": 7

},

{

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

"data": {

"text/plain": [

"<Figure size 360x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"df.plot.line()"

],

"metadata": {

"colab": {

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

"height": 574

},

"id": "kVAKxfc2BJnB",

"outputId": "fce48fdb-d008-4dfa-a5ec-50094b5381a3"

},

"execution\_count": 8,

"outputs": [

{

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

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x7f05de043c50>"

]

},

"metadata": {},

"execution\_count": 8

},

{

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

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.lmplot(\"id\",\"area\_mean\",df,hue=\"diagnosis\",fit\_reg=False);"

],

"metadata": {

"colab": {

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

"height": 424

},

"id": "k5OyEJ0NBXtT",

"outputId": "099d5510-0031-4fed-ef5b-635d75951427"

},

"execution\_count": 17,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

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

" FutureWarning\n"

]

},

{

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

"data": {

"text/plain": [

"<Figure size 415.5x360 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"source": [

"df.describe()"

],

"metadata": {

"colab": {

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

"height": 411

},

"id": "nd\_NeFs6DArL",

"outputId": "aa34b184-c0ce-478e-c99b-01d91e6fba27"

},

"execution\_count": 18,

"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-8eac2449-a509-4fbc-a97e-6108d3514473\">\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-8eac2449-a509-4fbc-a97e-6108d3514473')\"\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-8eac2449-a509-4fbc-a97e-6108d3514473 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-8eac2449-a509-4fbc-a97e-6108d3514473');\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": 18

}

]

},

{

"cell\_type": "code",

"source": [

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

"pd.isnull(df[\"diagnosis\"])"

],

"metadata": {

"colab": {

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

},

"id": "IMrw5bCzDIZf",

"outputId": "e76e30b5-749f-49a5-b781-88b1e38bb2ec"

},

"execution\_count": 21,

"outputs": [

{

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

"data": {

"text/plain": [

"0 False\n",

"1 False\n",

"2 False\n",

"3 False\n",

"4 False\n",

" ... \n",

"564 False\n",

"565 False\n",

"566 False\n",

"567 False\n",

"568 False\n",

"Name: diagnosis, Length: 569, dtype: bool"

]

},

"metadata": {},

"execution\_count": 21

}

]

},

{

"cell\_type": "code",

"source": [

"df[\"texture\_mean\"] = np.where(df[\"texture\_mean\"]>10,np.median,df[\"texture\_mean\"])\n",

"df[\"texture\_mean\"]"

],

"metadata": {

"colab": {

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

},

"id": "H2eI9YY0Dtto",

"outputId": "338b4907-47d8-482e-94fa-198c93d4c8e8"

},

"execution\_count": 28,

"outputs": [

{

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

"data": {

"text/plain": [

"0 <function median at 0x7f05fb954b00>\n",

"1 <function median at 0x7f05fb954b00>\n",

"2 <function median at 0x7f05fb954b00>\n",

"3 <function median at 0x7f05fb954b00>\n",

"4 <function median at 0x7f05fb954b00>\n",

" ... \n",

"564 <function median at 0x7f05fb954b00>\n",

"565 <function median at 0x7f05fb954b00>\n",

"566 <function median at 0x7f05fb954b00>\n",

"567 <function median at 0x7f05fb954b00>\n",

"568 <function median at 0x7f05fb954b00>\n",

"Name: texture\_mean, Length: 569, dtype: object"

]

},

"metadata": {},

"execution\_count": 28

}

]

},

{

"cell\_type": "code",

"source": [

"pd.get\_dummies(df, columns=[\"diagnosis\",\"id\"],prefix=[\"id\",\"diagnosis\"])\n",

"df.head()"

],

"metadata": {

"colab": {

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

"height": 490

},

"id": "fbKwxIHEENkU",

"outputId": "96d7c71c-29b1-4785-876f-6b169db7d0b1"

},

"execution\_count": 29,

"outputs": [

{

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

"data": {

"text/plain": [

" id diagnosis radius\_mean \\\n",

"0 842302 M <function median at 0x7f05fb954b00> \n",

"1 842517 M <function median at 0x7f05fb954b00> \n",

"2 84300903 M <function median at 0x7f05fb954b00> \n",

"3 84348301 M <function median at 0x7f05fb954b00> \n",

"4 84358402 M <function median at 0x7f05fb954b00> \n",

"\n",

" texture\_mean perimeter\_mean area\_mean \\\n",

"0 <function median at 0x7f05fb954b00> 122.80 1001.0 \n",

"1 <function median at 0x7f05fb954b00> 132.90 1326.0 \n",

"2 <function median at 0x7f05fb954b00> 130.00 1203.0 \n",

"3 <function median at 0x7f05fb954b00> 77.58 386.1 \n",

"4 <function median at 0x7f05fb954b00> 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-0f8775f7-ffdc-4e50-b1ea-3857a96e0f23\">\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>&lt;function median at 0x7f05fb954b00&gt;</td>\n",

" <td>&lt;function median at 0x7f05fb954b00&gt;</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>&lt;function median at 0x7f05fb954b00&gt;</td>\n",

" <td>&lt;function median at 0x7f05fb954b00&gt;</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>&lt;function median at 0x7f05fb954b00&gt;</td>\n",

" <td>&lt;function median at 0x7f05fb954b00&gt;</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>&lt;function median at 0x7f05fb954b00&gt;</td>\n",

" <td>&lt;function median at 0x7f05fb954b00&gt;</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>&lt;function median at 0x7f05fb954b00&gt;</td>\n",

" <td>&lt;function median at 0x7f05fb954b00&gt;</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-0f8775f7-ffdc-4e50-b1ea-3857a96e0f23')\"\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-0f8775f7-ffdc-4e50-b1ea-3857a96e0f23 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-0f8775f7-ffdc-4e50-b1ea-3857a96e0f23');\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": 29

}

]

},

{

"cell\_type": "code",

"source": [

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

"print(x)"

],

"metadata": {

"colab": {

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

},

"id": "\_z0jr5OgFNzX",

"outputId": "24a321ac-1d7d-47f8-b3c5-53dbfe5aab13"

},

"execution\_count": 31,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[[842302 'M' <function median at 0x7f05fb954b00> ... 0.2654 0.4601 0.1189]\n",

" [842517 'M' <function median at 0x7f05fb954b00> ... 0.186 0.275 0.08902]\n",

" [84300903 'M' <function median at 0x7f05fb954b00> ... 0.243 0.3613\n",

" 0.08758]\n",

" ...\n",

" [926954 'M' <function median at 0x7f05fb954b00> ... 0.1418 0.2218 0.0782]\n",

" [927241 'M' <function median at 0x7f05fb954b00> ... 0.265 0.4087 0.124]\n",

" [92751 'B' 7.76 ... 0.0 0.2871 0.07039]]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"y = df.iloc[:, -3].values\n",

"print(y)"

],

"metadata": {

"colab": {

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

},

"id": "s3n3w0QoGGAB",

"outputId": "c2c56a33-fba3-444d-9b34-bab1aa8c6b8b"

},

"execution\_count": 36,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[0.4601 0.275 0.3613 0.6638 0.2364 0.3985 0.3063 0.3196 0.4378 0.4366\n",

" 0.2948 0.3792 0.3176 0.2809 0.3596 0.4218 0.3029 0.3706 0.2768 0.2977\n",

" 0.3184 0.245 0.4667 0.2822 0.3613 0.4066 0.4264 0.2341 0.4027 0.2756\n",

" 0.3444 0.4761 0.353 0.3672 0.427 0.4863 0.3591 0.1987 0.1565 0.2807\n",

" 0.2994 0.2964 0.467 0.3739 0.3693 0.3799 0.3105 0.39 0.2747 0.2871\n",

" 0.2433 0.2346 0.2785 0.3021 0.2675 0.3306 0.3537 0.3698 0.2439 0.322\n",

" 0.3557 0.2972 0.2844 0.3282 0.3383 0.3321 0.2878 0.24 0.4228 0.2383\n",

" 0.2551 0.2254 0.3313 0.2589 0.2618 0.265 0.271 0.3751 0.544 0.2779\n",

" 0.2762 0.3527 0.2355 0.2311 0.3379 0.3695 0.302 0.3956 0.2972 0.3151\n",

" 0.2522 0.2556 0.2027 0.2678 0.2834 0.3689 0.2227 0.1934 0.2772 0.2718\n",

" 0.2651 0.2932 0.2694 0.2622 0.2826 0.3147 0.2806 0.2983 0.4055 0.2829\n",

" 0.2533 0.2226 0.2398 0.2383 0.2926 0.2438 0.1652 0.3585 0.3274 0.4882\n",

" 0.3016 0.2894 0.3222 0.2889 0.2048 0.2364 0.347 0.2841 0.259 0.3305\n",

" 0.3469 0.2837 0.348 0.2723 0.3109 0.2829 0.1712 0.2676 0.3414 0.2102\n",

" 0.3105 0.2792 0.2584 0.3549 0.23 0.2727 0.5774 0.2852 0.2691 0.235\n",

" 0.3196 0.3322 0.3108 0.2859 0.3849 0.3113 0.2463 0.2527 0.2514 0.2738\n",

" 0.3168 0.2443 0.3643 0.2268 0.3589 0.2646 0.209 0.281 0.216 0.2404\n",

" 0.2827 0.2884 0.3216 0.1902 0.271 0.2592 0.2614 0.3054 0.2295 0.1783\n",

" 0.2856 0.4098 0.3437 0.1811 0.3175 0.2933 0.3206 0.2572 0.32 0.2482\n",

" 0.5166 0.2179 0.1909 0.3215 0.3 0.3024 0.308 0.2369 0.3193 0.4753\n",

" 0.2668 0.2928 0.3198 0.4432 0.3014 0.3415 0.2989 0.3275 0.4128 0.232\n",

" 0.2909 0.2535 0.1648 0.1603 0.4724 0.363 0.3397 0.2868 0.307 0.2713\n",

" 0.238 0.3065 0.3055 0.3993 0.2506 0.3062 0.2615 0.2954 0.2826 0.3407\n",

" 0.3109 0.2849 0.2911 0.2437 0.2757 0.2226 0.3103 0.2238 0.189 0.2853\n",

" 0.253 0.2901 0.3308 0.2663 0.292 0.2883 0.2767 0.2639 0.3409 0.2664\n",

" 0.3126 0.274 0.2749 0.3138 0.3379 0.3068 0.2818 0.3258 0.3277 0.3512\n",

" 0.3151 0.2452 0.3067 0.2683 0.3216 0.2868 0.294 0.2446 0.3604 0.2605\n",

" 0.2458 0.2733 0.2833 0.2841 0.2504 0.222 0.2758 0.2567 0.2335 0.2955\n",

" 0.3258 0.3101 0.359 0.277 0.1999 0.2505 0.2465 0.2309 0.2955 0.3267\n",

" 0.2272 0.2962 0.338 0.3101 0.2369 0.2823 0.2143 0.1978 0.2636 0.2227\n",

" 0.2968 0.2685 0.3294 0.2213 0.2208 0.3244 0.2651 0.2991 0.2267 0.2107\n",

" 0.3487 0.253 0.2564 0.339 0.3142 0.1917 0.2293 0.2812 0.3135 0.1901\n",

" 0.2608 0.3055 0.2382 0.5558 0.2661 0.2688 0.21 0.2171 0.3103 0.2736\n",

" 0.3019 0.3596 0.3292 0.2815 0.2554 0.2623 0.2432 0.3679 0.2894 0.2593\n",

" 0.3053 0.2982 0.3301 0.4045 0.2765 0.2434 0.288 0.3109 0.2851 0.3124\n",

" 0.2731 0.4245 0.369 0.2654 0.2576 0.2121 0.3113 0.2362 0.2434 0.2454\n",

" 0.2233 0.2637 0.2744 0.2394 0.2741 0.2609 0.3271 0.3218 0.251 0.2741\n",

" 0.4824 0.2487 0.273 0.2689 0.3323 0.3153 0.2597 0.2694 0.3387 0.4154\n",

" 0.3343 0.3202 0.2191 0.2819 0.2736 0.2477 0.2677 0.2542 0.2157 0.2576\n",

" 0.2937 0.2445 0.3187 0.3828 0.3049 0.2523 0.2666 0.1988 0.259 0.2779\n",

" 0.3245 0.246 0.3207 0.3297 0.2298 0.2251 0.2778 0.2488 0.306 0.3455\n",

" 0.2973 0.2998 0.2376 0.3163 0.3233 0.3075 0.2872 0.3003 0.2775 0.2976\n",

" 0.3035 0.2827 0.2787 0.2744 0.3196 0.2349 0.302 0.2965 0.2383 0.2165\n",

" 0.2866 0.2556 0.3032 0.3007 0.2525 0.3179 0.3313 0.2725 0.2513 0.2136\n",

" 0.254 0.25 0.1859 0.2699 0.2842 0.2599 0.2972 0.36 0.2471 0.2268\n",

" 0.2305 0.2293 0.2447 0.2606 0.327 0.2196 0.2884 0.2444 0.2306 0.2321\n",

" 0.301 0.2641 0.2345 0.321 0.2235 0.2845 0.2563 0.3057 0.2301 0.266\n",

" 0.3174 0.2404 0.2688 0.2409 0.2581 0.3006 0.2534 0.2679 0.2941 0.3318\n",

" 0.2406 0.2356 0.3227 0.2518 0.2557 0.3231 0.2455 0.3266 0.2804 0.4677\n",

" 0.31 0.1783 0.3251 0.228 0.2687 0.2369 0.323 0.3035 0.251 0.248\n",

" 0.2177 0.3651 0.3155 0.2908 0.3038 0.2848 0.2709 0.278 0.23 0.3013\n",

" 0.2604 0.2334 0.3585 0.2626 0.2282 0.306 0.3074 0.2549 0.2582 0.3071\n",

" 0.3681 0.3157 0.2557 0.2849 0.2502 0.2983 0.3518 0.311 0.216 0.2781\n",

" 0.2478 0.3206 0.2806 0.322 0.2289 0.3077 0.2698 0.2803 0.3058 0.279\n",

" 0.2329 0.3187 0.2722 0.2473 0.2249 0.2642 0.2681 0.2691 0.2552 0.3059\n",

" 0.2458 0.3169 0.2407 0.2435 0.2372 0.2226 0.2262 0.2475 0.2258 0.2112\n",

" 0.225 0.1566 0.4089 0.2929 0.206 0.2572 0.2218 0.4087 0.2871]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"from sklearn.preprocessing import MinMaxScaler\n",

"scaler = MinMaxScaler()\n",

"df[[\"id\"]] = scaler.fit\_transform(df[[\"id\"]])\n",

"print(df)"

],

"metadata": {

"colab": {

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

},

"id": "nys21AsgGqaC",

"outputId": "7de3fb32-6268-47cc-b944-bcf1195995cb"

},

"execution\_count": 37,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" id diagnosis radius\_mean \\\n",

"0 0.000915 M <function median at 0x7f05fb954b00> \n",

"1 0.000915 M <function median at 0x7f05fb954b00> \n",

"2 0.092495 M <function median at 0x7f05fb954b00> \n",

"3 0.092547 M <function median at 0x7f05fb954b00> \n",

"4 0.092559 M <function median at 0x7f05fb954b00> \n",

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

"564 0.001007 M <function median at 0x7f05fb954b00> \n",

"565 0.001007 M <function median at 0x7f05fb954b00> \n",

"566 0.001008 M <function median at 0x7f05fb954b00> \n",

"567 0.001008 M <function median at 0x7f05fb954b00> \n",

"568 0.000092 B 7.76 \n",

"\n",

" texture\_mean perimeter\_mean area\_mean \\\n",

"0 <function median at 0x7f05fb954b00> 122.80 1001.0 \n",

"1 <function median at 0x7f05fb954b00> 132.90 1326.0 \n",

"2 <function median at 0x7f05fb954b00> 130.00 1203.0 \n",

"3 <function median at 0x7f05fb954b00> 77.58 386.1 \n",

"4 <function median at 0x7f05fb954b00> 135.10 1297.0 \n",

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

"564 <function median at 0x7f05fb954b00> 142.00 1479.0 \n",

"565 <function median at 0x7f05fb954b00> 131.20 1261.0 \n",

"566 <function median at 0x7f05fb954b00> 108.30 858.1 \n",

"567 <function median at 0x7f05fb954b00> 140.10 1265.0 \n",

"568 <function median at 0x7f05fb954b00> 47.92 181.0 \n",

"\n",

" smoothness\_mean compactness\_mean concavity\_mean concave points\_mean \\\n",

"0 0.11840 0.27760 0.30010 0.14710 \n",

"1 0.08474 0.07864 0.08690 0.07017 \n",

"2 0.10960 0.15990 0.19740 0.12790 \n",

"3 0.14250 0.28390 0.24140 0.10520 \n",

"4 0.10030 0.13280 0.19800 0.10430 \n",

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

"564 0.11100 0.11590 0.24390 0.13890 \n",

"565 0.09780 0.10340 0.14400 0.09791 \n",

"566 0.08455 0.10230 0.09251 0.05302 \n",

"567 0.11780 0.27700 0.35140 0.15200 \n",

"568 0.05263 0.04362 0.00000 0.00000 \n",

"\n",

" ... texture\_worst perimeter\_worst area\_worst smoothness\_worst \\\n",

"0 ... 17.33 184.60 2019.0 0.16220 \n",

"1 ... 23.41 158.80 1956.0 0.12380 \n",

"2 ... 25.53 152.50 1709.0 0.14440 \n",

"3 ... 26.50 98.87 567.7 0.20980 \n",

"4 ... 16.67 152.20 1575.0 0.13740 \n",

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

"564 ... 26.40 166.10 2027.0 0.14100 \n",

"565 ... 38.25 155.00 1731.0 0.11660 \n",

"566 ... 34.12 126.70 1124.0 0.11390 \n",

"567 ... 39.42 184.60 1821.0 0.16500 \n",

"568 ... 30.37 59.16 268.6 0.08996 \n",

"\n",

" compactness\_worst concavity\_worst concave points\_worst symmetry\_worst \\\n",

"0 0.66560 0.7119 0.2654 0.4601 \n",

"1 0.18660 0.2416 0.1860 0.2750 \n",

"2 0.42450 0.4504 0.2430 0.3613 \n",

"3 0.86630 0.6869 0.2575 0.6638 \n",

"4 0.20500 0.4000 0.1625 0.2364 \n",

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

"564 0.21130 0.4107 0.2216 0.2060 \n",

"565 0.19220 0.3215 0.1628 0.2572 \n",

"566 0.30940 0.3403 0.1418 0.2218 \n",

"567 0.86810 0.9387 0.2650 0.4087 \n",

"568 0.06444 0.0000 0.0000 0.2871 \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",

"564 0.07115 NaN \n",

"565 0.06637 NaN \n",

"566 0.07820 NaN \n",

"567 0.12400 NaN \n",

"568 0.07039 NaN \n",

"\n",

"[569 rows x 33 columns]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

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

"train\_size = 0.8\n",

"x = df.drop(columns = ['diagnosis']).copy()\n",

"y = df['diagnosis']\n",

"x\_train, x\_rem, y\_train, y\_rem = train\_test\_split(x,y,train\_size=0.8)\n",

"test\_size = 0.5\n",

"x\_valid, x\_test, y\_valid, y\_test = train\_test\_split(x\_rem,y\_rem,test\_size=0.5)\n",

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

"print(x\_valid.shape),print(y\_valid.shape)\n",

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

],

"metadata": {

"colab": {

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

},

"id": "5j8M1bW-HzvZ",

"outputId": "3e632be5-51b2-495d-ea2c-f3b612e08be5"

},

"execution\_count": 42,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"(455, 32)\n",

"(455,)\n",

"(57, 32)\n",

"(57,)\n",

"(57, 32)\n",

"(57,)\n"

]

},

{

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

"data": {

"text/plain": [

"(None, None)"

]

},

"metadata": {},

"execution\_count": 42

}

]

}

]

}