

# Laboratório 3.2 - Notebook do aluno

## Visão geral

Este laboratório é uma continuação dos laboratórios guiados do Módulo 3.

## Apresentação do cenário de negócios

Você trabalha para um provedor de serviços médicos e deseja melhorar a detecção de anormalidades em pacientes ortopédicos.

Você tem a incumbência de resolver esse problema usando machine learning (ML). Você tem acesso a um conjunto de dados que contém seis componentes biomecânicos (features) e um alvo (target) de *normal* (normal) ou *abnormal* (anormal). Você pode usar esse conjunto de dados (datasets) para treinar um modelo de ML para prever se um paciente terá uma anomalia.

## Sobre esse conjunto de dados

Esse conjunto de dados (dataset) biomédicos foi criado pelo Dr. Henrique da Mota durante um período de residência médica no Group of Applied Research in Orthopaedics (GARO) do Centre Médico-Chirurgical de Réadaptation des Massues em Lyon, na França. Os dados foram organizados em duas tarefas de classificação diferentes, mas relacionadas.

A primeira tarefa consiste em classificar os pacientes como pertencentes a uma das três categorias a seguir:

- *Normal* (Normal) (100 pacientes)
- *Disk Hernia* (Hérnia de disco) (60 pacientes)
- *Spondylolisthesis* (Espondilolistese) (150 pacientes)

Para a segunda tarefa, as categorias *Disk Hernia* (Hérnia de disco) e *Spondylolisthesis* (Espondilolistese) foram mescladas em uma única categoria, rotulada como *abnormal* (anormal). Portanto, a segunda tarefa consiste em classificar os pacientes como pertencentes a uma das categorias: *Normal* (Normal) (100 pacientes) ou *Abnormal* (Anormal) (210 pacientes).

## Informações de atributo

Cada paciente é representado no conjunto de dados por seis atributos biomecânicos derivados da forma e da orientação da pelve e da coluna lombar (nesta ordem):

- Incidência pélvica

- Inclinação pélvica
- Ângulo da lordose lombar
- Inclinação sacral
- Raio pélvico
- Grau de espondilolistese

A convenção a seguir é usada para os rótulos de classe (labels):

- DH (hérnia de disco)
- Espondilolistese (SL)
- Normal (NO)
- Anormal (AB)

Para obter mais informações sobre esse conjunto de dados, consulte a [página da Web Conjunto de dados de coluna vertebral](#).

## Atribuições do conjunto de dados (dataset)

Esse conjunto de dados foi obtido de: Dua, D. e Graff, C. (2019). repositório UCI Machine Learning (<http://archive.ics.uci.edu/ml>). Irvine, CA: University of California, School of Information and Computer Science.

## Configuração do laboratório

Como essa solução é dividida em vários laboratórios neste módulo, você executará as seguintes células para poder carregar os dados:

### Importação de dados

```
In [1]: import warnings, requests, zipfile, io
warnings.simplefilter('ignore')
import pandas as pd
from scipy.io import arff
```

```
In [2]: f_zip = 'http://archive.ics.uci.edu/ml/machine-learning-databases/00212/vertebra
r = requests.get(f_zip, stream=True)
Vertebral_zip = zipfile.ZipFile(io.BytesIO(r.content))
Vertebral_zip.extractall()
```

```
In [3]: data = arff.loadarff('column_2C_weka.arff')
df = pd.DataFrame(data[0])
```

## Etapa 1: Exploração dos dados

Você começará a examinar os dados no conjunto de dados (dataset).

Para aproveitar ao máximo este laboratório, leia atentamente as instruções e o código antes de executar as células. Reserve um tempo para experimentar!

Primeiro, você usará a função **shape** para examinar o número de linhas e colunas

```
In [4]: df.shape
```

```
Out[4]: (310, 7)
```

Agora, você obterá uma lista das colunas.

```
In [5]: df.columns
```

```
Out[5]: Index(['pelvic_incidence', 'pelvic_tilt', 'lumbar_lordosis_angle',
   'sacral_slope', 'pelvic_radius', 'degree_spondylolisthesis', 'class'],
   dtype='object')
```

É possível ver os seis componentes biomecânicos, e a coluna alvo (target) é chamada *class* (classe).

Quais tipos de coluna você tem?

```
In [6]: df.dtypes
```

```
Out[6]: pelvic_incidence      float64
pelvic_tilt                  float64
lumbar_lordosis_angle        float64
sacral_slope                 float64
pelvic_radius                float64
degree_spondylolisthesis    float64
class                         object
dtype: object
```

Você tem seis tipos float (ponto flutuante) para os componentes (features) biomecânicos, mas o alvo (target) é uma class (classe).

Para examinar as estatísticas da primeira coluna, você pode usar a função **describe**.

```
In [7]: df['pelvic_incidence'].describe()
```

```
Out[7]: count    310.000000
mean     60.496653
std      17.236520
min     26.147921
25%     46.430294
50%     58.691038
75%     72.877696
max     129.834041
Name: pelvic_incidence, dtype: float64
```

**Tarefa de desafio:** Tente atualizar o código na célula anterior para visualizar as estatísticas de outros componentes (features). Quais componentes têm anomalias (outliers) que talvez você queira examinar?

Como esse conjunto de dados (dataset) tem apenas seis componentes (features), você pode exibir as estatísticas de cada componente (feature) executando **describe** em todo o DataFrame.

In [8]: `df.describe()`

Out[8]:

	<b>pelvic_incidence</b>	<b>pelvic_tilt</b>	<b>lumbar_lordosis_angle</b>	<b>sacral_slope</b>	<b>pelvic_radius</b>	<b>degree_s</b>
<b>count</b>	310.000000	310.000000	310.000000	310.000000	310.000000	310.000000
<b>mean</b>	60.496653	17.542822	51.930930	42.953831	117.920655	
<b>std</b>	17.236520	10.008330	18.554064	13.423102	13.317377	
<b>min</b>	26.147921	-6.554948	14.000000	13.366931	70.082575	
<b>25%</b>	46.430294	10.667069	37.000000	33.347122	110.709196	
<b>50%</b>	58.691038	16.357689	49.562398	42.404912	118.268178	
<b>75%</b>	72.877696	22.120395	63.000000	52.695888	125.467674	
<b>max</b>	129.834041	49.431864	125.742385	121.429566	163.071041	

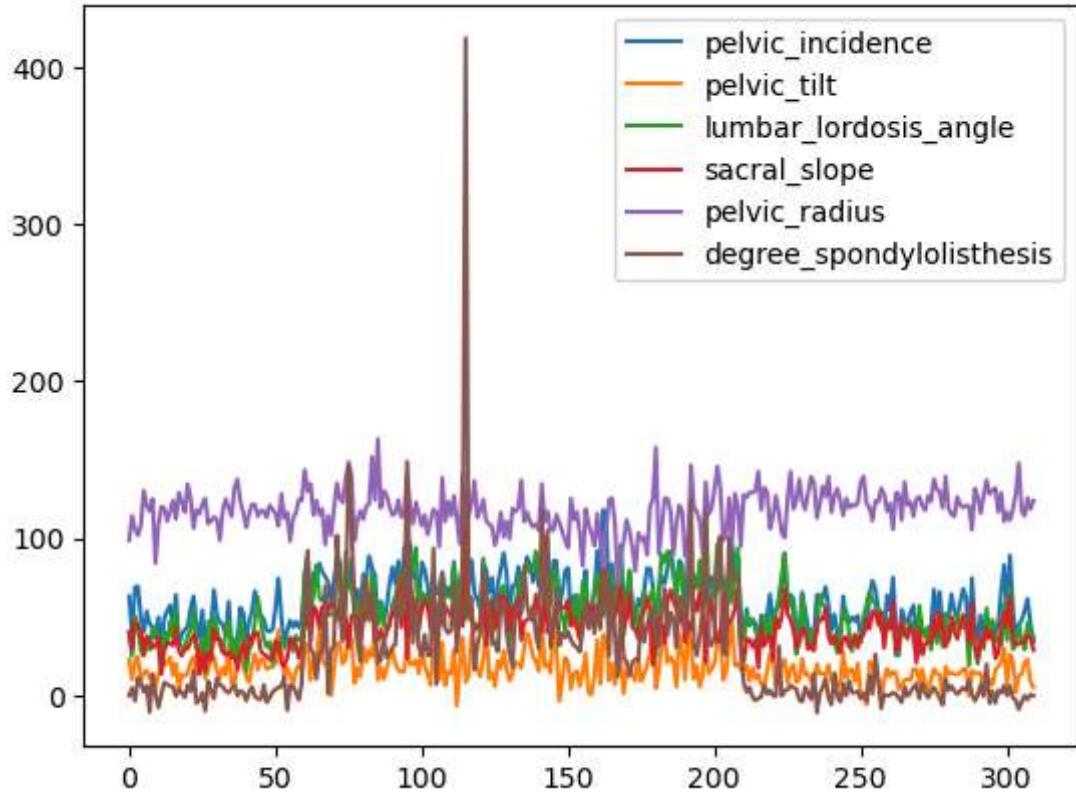
**Pergunta:** Há algum componente (feature) que não se apresente bem distribuído? Há algum componente (feature) com anomalias (outliers) que você deseja examinar? Parece que pode haver alguma correlação entre os componentes (features)?

Nem sempre é fácil fazer observações quando você olha apenas para números, portanto, agora você plotará esses valores.

In [9]: `import matplotlib.pyplot as plt  
%matplotlib inline  
df.plot()`

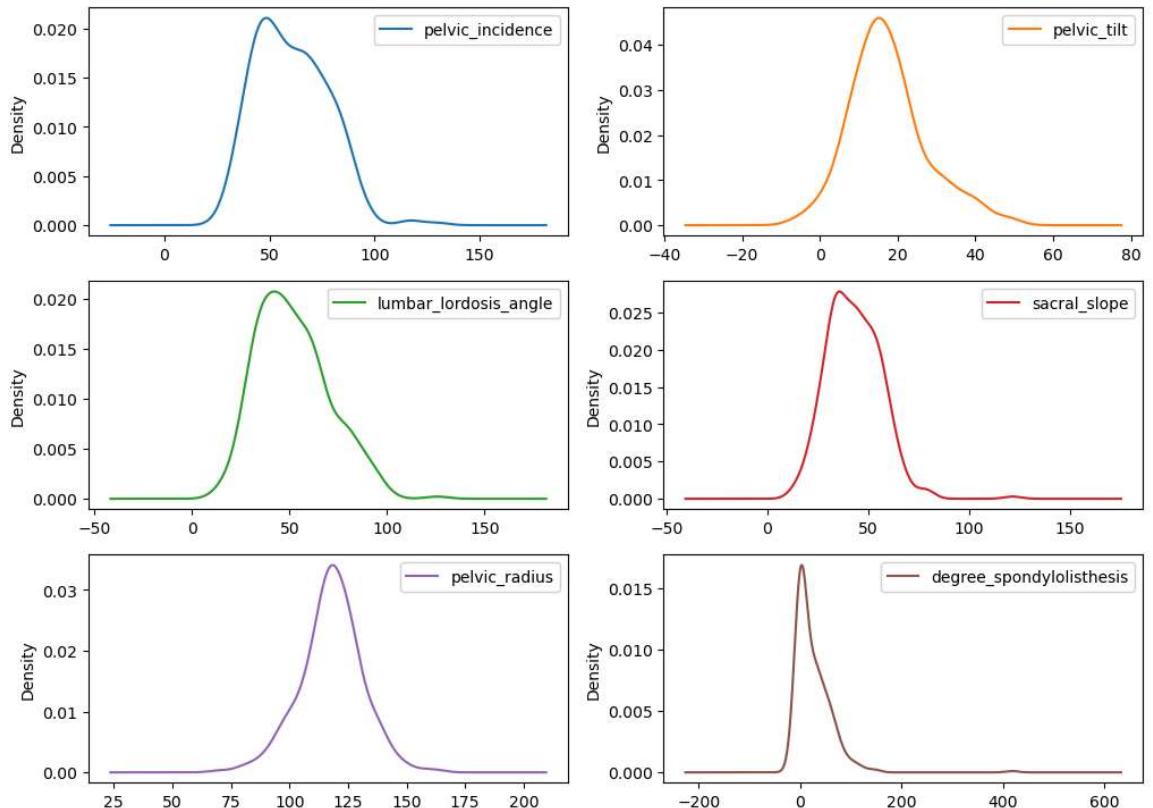
Matplotlib is building the font cache; this may take a moment.

Out[9]: `<Axes: >`



Agora, você plotará a distribuição dos valores para cada componente usando um gráfico *density or kernel density estimate (KDE)* (densidade ou estimativa de densidade por kernel).

```
In [10]: df.plot(kind='density', subplots=True, layout=(4,2), figsize=(12,12), sharex=False)
plt.show()
```



Algumas das visualizações se destacam?

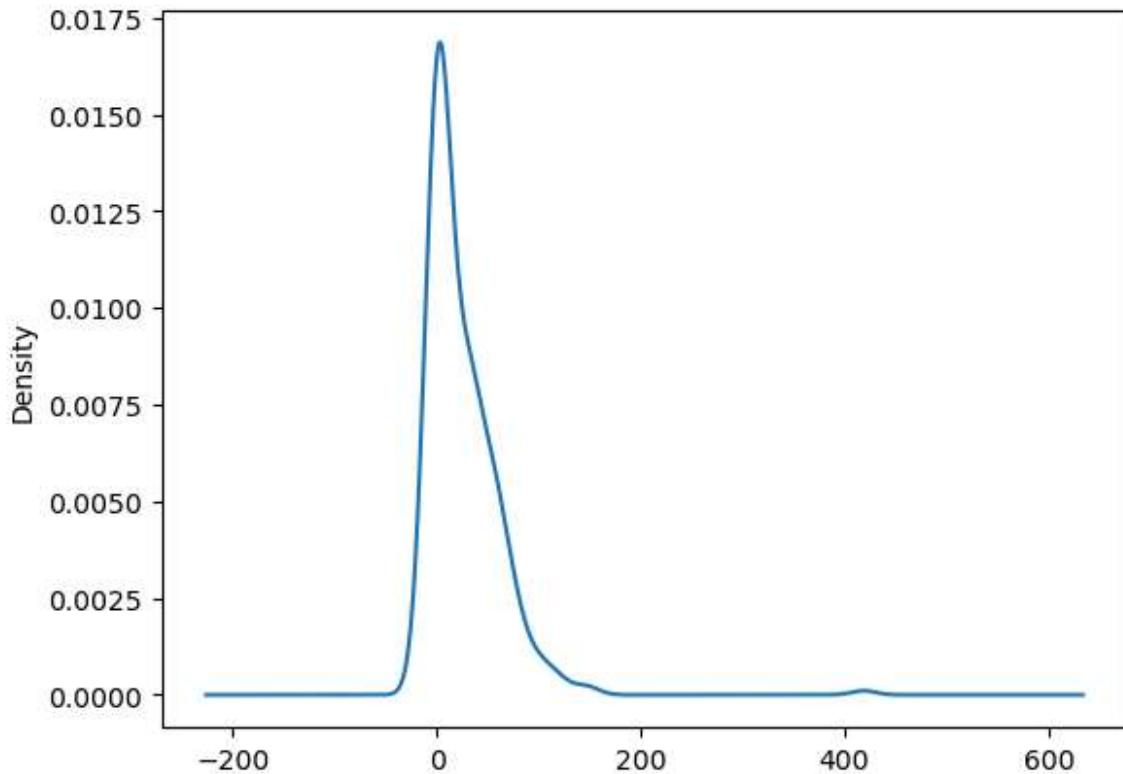
## Investigar degree\_spondylolisthesis

Agora você investigará **degree\_spondylolisthesis**:

Comece com o *density plot* (gráfico de densidade), que, se você lembrar, mostra a *distribution of the values* (distribuição dos valores).

```
In [11]: df['degree_spondylolisthesis'].plot.density()
```

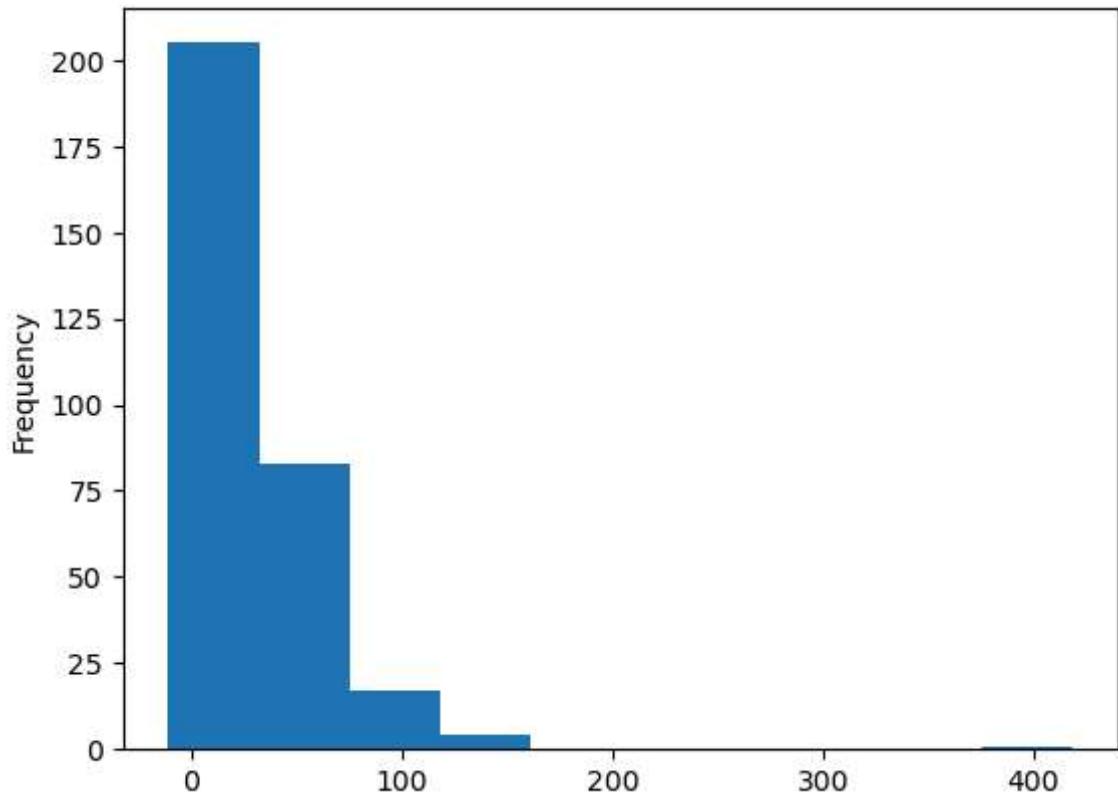
```
Out[11]: <Axes: ylabel='Density'>
```



Um gráfico de densidade ameniza a curva. Parece que pode haver um aumento em torno de **400**. Visualize os dados com um *histogram* (histograma).

```
In [12]: df['degree_spondylolisthesis'].plot.hist()
```

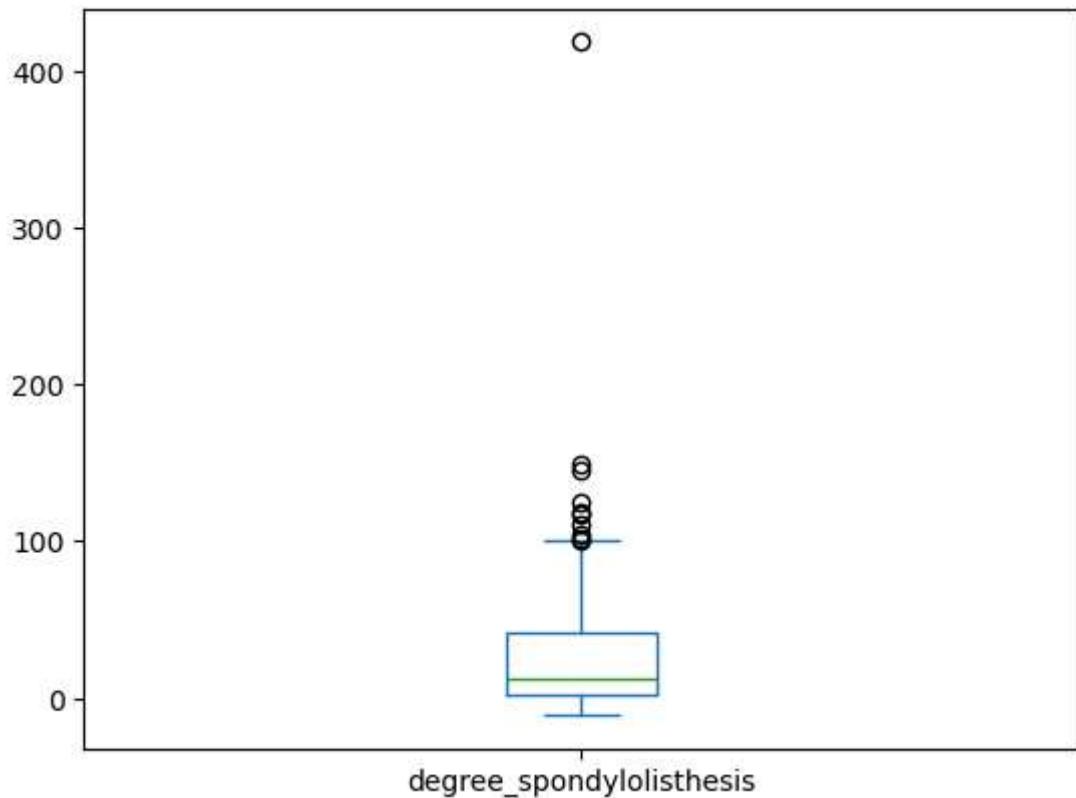
```
Out[12]: <Axes: ylabel='Frequency'>
```



Ao usar um *box plot* (gráfico diagrama de caixa), você pode ver se há anomalias (outliers).

```
In [13]: df['degree_spondylolisthesis'].plot.box()
```

```
Out[13]: <Axes: >
```



Você pode ver um pequeno aumento em torno de **400**. Às vezes, exceções como esta podem levar ao descarte de modelos de treinamento. A única maneira de descobrir seria

testar o modelo com e sem as anomalias (outliers) e comparar as pontuações (scores) dos modelos. No entanto, esta é uma tarefa para um laboratório posterior.

Você pode ver o que já parece um cluster no gráfico diagrama de caixa (box plot) acima, o qual parece ter um valor máximo já definido. Há uma correlação entre esses pontos de dados (data points) e o alvo (target)?

Antes de procurar uma correlação, você examinará mais o alvo (target).

## Analisando o alvo (target)

Primeiro, que tipo de distribuição você tem?

```
In [14]: df['class'].value_counts()
```

```
Out[14]: b'Abnormal'    210  
b'Normal'      100  
Name: class, dtype: int64
```

Parece que você tem cerca de 1/3 *Normal* e 2/3 *Abormal* (Anormal). Esse resultado deve ser aceitável, mas se você pudesse obter mais dados, você gostaria de fazer mais tentativas e de equilibrar melhor os números.

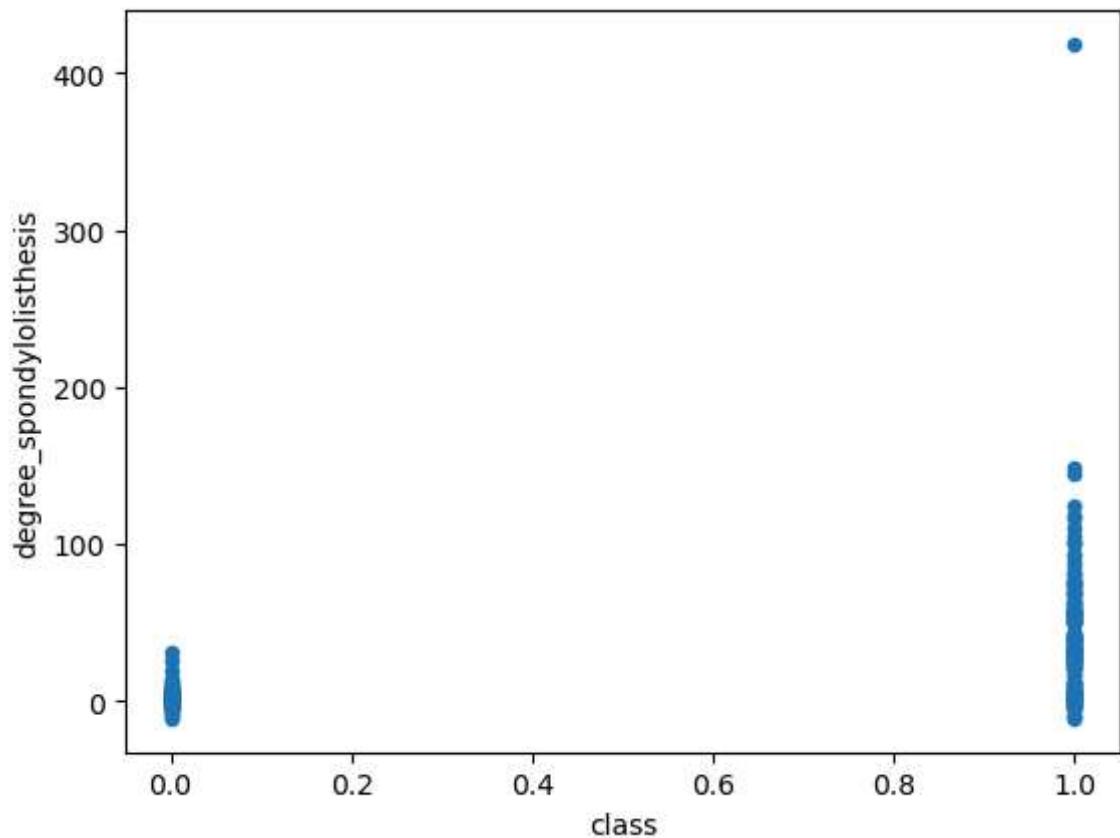
Os valores do tipo *class* (classe) não funcionarão para o seu modelo de ML, portanto, você irá converter essa coluna em um valor numérico. Você pode usar o *mapper* (mapeador) para essa tarefa.

```
In [15]: class_mapper = {b'Abnormal':1,b'Normal':0}  
df['class']=df['class'].replace(class_mapper)
```

Agora, é possível plotar *degree\_spondylolisthesis* para o alvo (target).

```
In [16]: df.plot.scatter(y='degree_spondylolisthesis',x='class')
```

```
Out[16]: <Axes: xlabel='class', ylabel='degree_spondylolisthesis'>
```



O que você vê?

Embora pareça haver uma conexão entre os valores altos e as anormalidades, também há muitos valores que estão no mesmo intervalo. Portanto, pode haver uma correlação, mas vale a pena analisar melhor os dados.

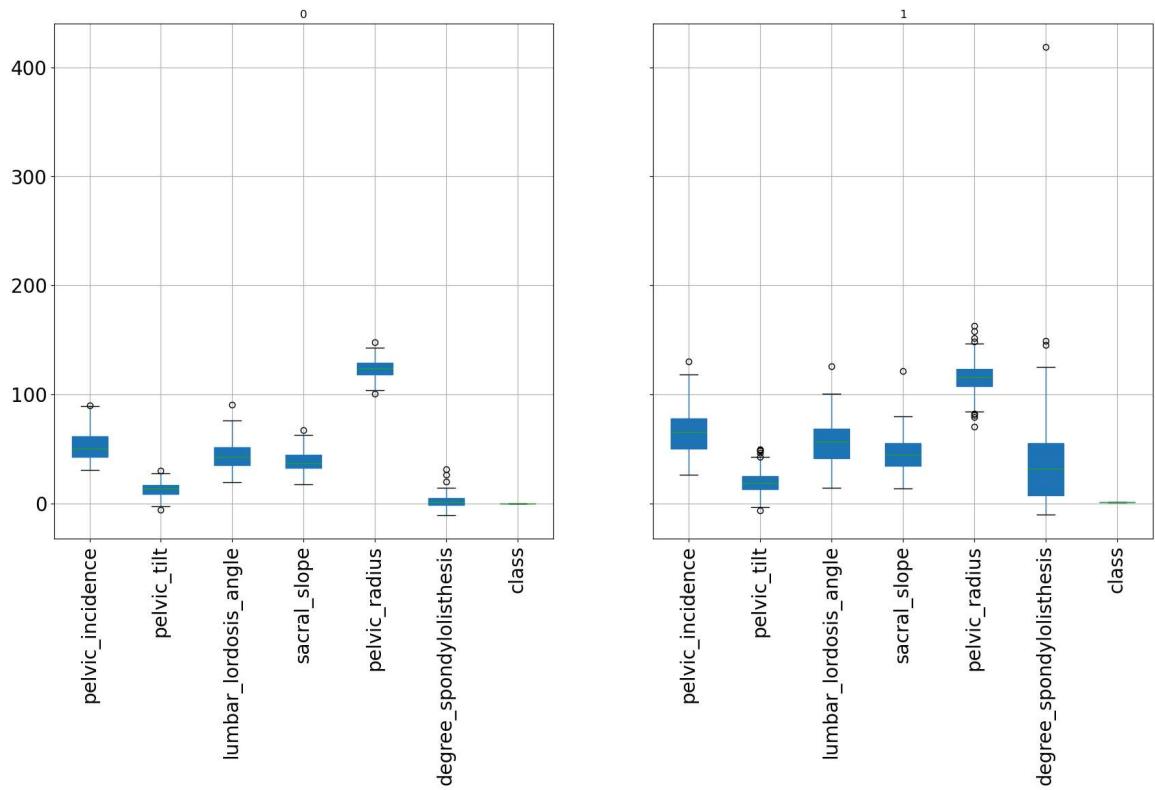
**Tarefa de desafio:** Usando as células anteriores, determine como os valores de outros componentes (features) correspondem em relação ao alvo (target).

## Visualizando múltiplas variáveis

Como demonstram as etapas anteriores, as visualizações podem ser muito importantes. Às vezes, você poderá querer analisar múltiplos pontos de dados (data points). Você pode fazer isso usando *groupby*.

A colocação dos componentes (features) dos dois valores *Abnormal* (Anormal) e *Normal* (normal) lado a lado pode ajudá-lo a observar outras diferenças.

```
In [17]: df.groupby('class').boxplot(fontsize=20, rot=90, figsize=(20,10), patch_artist=True)
Out[17]: 0      Axes(0.1,0.15;0.363636x0.75)
          1      Axes(0.536364,0.15;0.363636x0.75)
          dtype: object
```



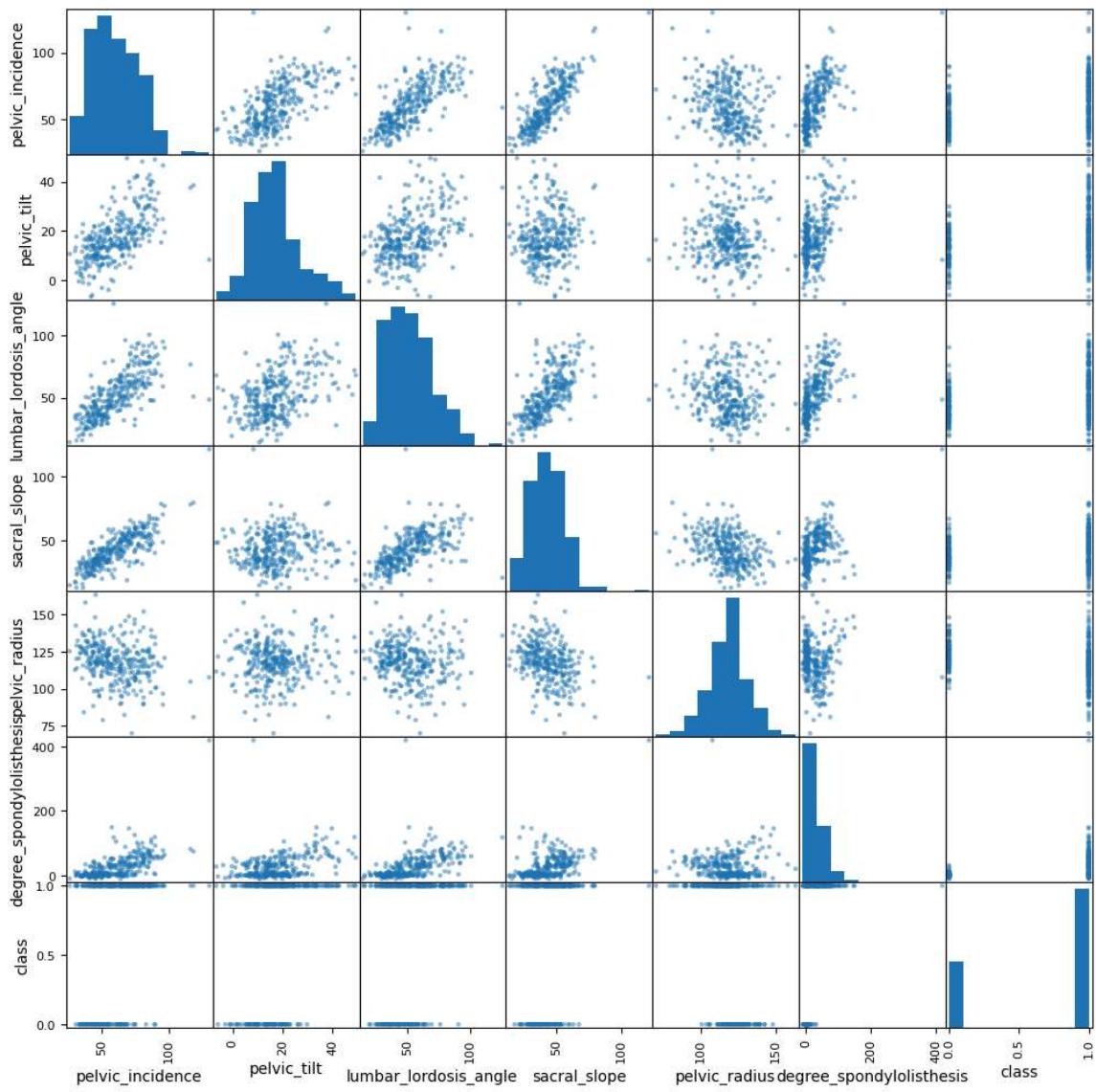
Usando a função **corr**, você pode criar uma matriz de correlação para todo o conjunto de dados (dataset).

```
In [18]: corr_matrix = df.corr()
corr_matrix["class"].sort_values(ascending=False)
```

```
Out[18]: class           1.000000
degree_spondylolisthesis 0.443687
pelvic_incidence         0.353336
pelvic_tilt               0.326063
lumbar_lordosis_angle    0.312484
sacral_slope              0.210602
pelvic_radius             -0.309857
Name: class, dtype: float64
```

Você também pode plotar esses dados.

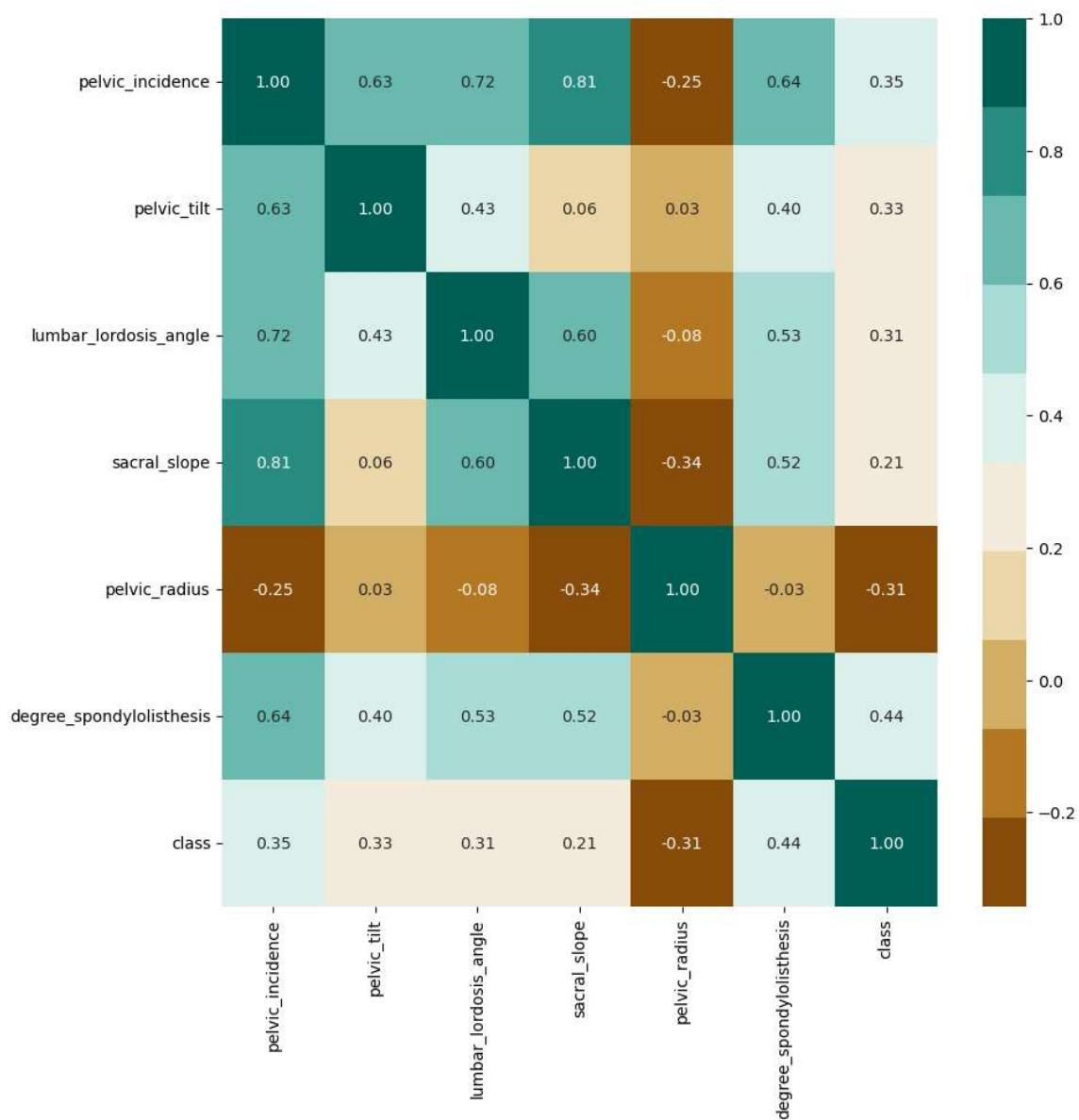
```
In [19]: pd.plotting.scatter_matrix(df, figsize=(12,12))
plt.show()
```



Ao usar **seaborn**, você pode visualizar a correlação como um *heatmap* (mapa de calor).

In [20]:

```
import seaborn as sns
# Plot figsize
fig, ax = plt.subplots(figsize=(10, 10))
# Generate Color Map
# colormap = sns.diverging_palette(220, 10, as_cmap=True)
colormap = sns.color_palette("BrBG", 10)
# Generate Heat Map, allow annotations and place floats in map
sns.heatmap(corr_matrix, cmap=colormap, annot=True, fmt=".2f")
#ax.set_yticklabels(column_names);
plt.show()
```



**Tarefa de desafio:** Busque outros dados disponíveis no repositório UCI Machine Learning. Usando o código anterior como referência, explore!

## Parabéns!

Você concluiu este laboratório e agora pode encerrá-lo seguindo as instruções do guia do laboratório.

In [ ]:

In [ ]:

In [ ]: