MARCELO MIKY MINE

MACHINE LEARNING ENGINEERING

APRENDIZADO SUPERVISIONADO

AULA 01

SUMÁRIO

O QUE VEM POR AÍ?

Nesta aula você irá aprender sobre aprendizado de máquina supervisionado, suas características, algoritmos e exemplos e verá que há duas tarefas para este tipo de aprendizado e o que difere uma da outra. Usamos aprendizado de máquina nos casos em que não é prático para os humanos detectarem padrões em dados complexos ou em grande volume. Com o big data inundando nossas vidas, o aprendizado de máquina é uma necessidade cada vez maior.

Há diversos campos que utilizam IA, como processamento de linguagem natural, visão computacional, sistema de recomendação e detecção de fraudes, entre outros.

HANDS ON

Nesta aula introdutória de Aprendizado Supervisionado iremos ver algumas ferramentas para manipular os dados, prepará-los para o algoritmo de aprendizado de máquina e termos um modelo final de IA.

A linguagem de programação utilizada para este curso será o Python, que é de alto nível, interpretada, de fácil aprendizado, muito popular e de comunidade engajada. Caso seja seu primeiro contato com a linguagem, não se preocupe! É uma linguagem de fácil sintaxe, com baixa curva de aprendizado e ótima documentação. Há diversos cursos na internet e aqui recomendamos o da comunidade Grupy-Sanca.

Os códigos serão feitos no Jupyter Notebook, um ambiente computacional que permite anotações em formato Markdown. Utilizaremos também algumas bibliotecas baseadas em Python, como o Pandas, para manipulação e análise de dados, o NumPy, para operações matemáticas, e o Scikit-Learn, para aprendizado de máquina supervisionado e não-supervisionado, com algoritmos implementados, transformações no conjunto de dados (dataset), visualizações e avaliações.

```
In [1]: import pandas as pd
        import numpy as np

In [2]: df = pd.DataFrame({
      "A": 3,
      "B": pd.Timestamp("20240101"),
      "C": pd.Series(2, index=list(range(4)), dtype="float32"),
      "D": np.array([5] * 4, dtype="int32"),
      "E": pd.Categorical(["train", "test", "test", "train"]),
      "F": "foo"})

In [3]: df
Out[3]:
           A          B    C    D      E      F
     0     3 2024-01-01  2.0  5  train  foo
     1     3 2024-01-01  2.0  5  test   foo
     2     3 2024-01-01  2.0  5  test   foo
     3     3 2024-01-01  2.0  5  train  foo
```

Código-fonte 1 – Código Python com a biblioteca Pandas e Numpy executado no Jupyter Notebook
Fonte: Elaborado pelo autor (2024)

SAIBA MAIS

O Aprendizado de Máquina (AM), ou Machine Learning (em inglês), é uma subárea da Inteligência Artificial (IA) que se concentra no desenvolvimento de algoritmos e modelos que permitem aos computadores aprender e melhorar a partir de experiências passadas.

Uma das definições do que é o Aprendizado de Máquina foi feita por Arthur Samuel em 1959: "Campo de estudo que permite aos computadores a capacidade de aprender sem serem explicitamente programados". Então, em vez de serem explicitamente programados para realizar uma tarefa específica ("dado isso, faça isso"), esses algoritmos são capazes de aprender e se adaptar, tornando-os incrivelmente poderosos em lidar com problemas complexos e dados variáveis.

Há diversos campos que utilizam IA, como o de processamento de linguagem natural para compreender contextos da linguagem humana, como a tradução de idiomas, pesquisa na internet e filtragem de spam na sua caixa de e-mail. O aprendizado de máquina também é usado para visão computacional: os algoritmos analisam imagens ou vídeos digitais para dar algum sentido para esses dados. Como exemplo, temos o auxílio em áreas como a medicina para o diagnóstico de pacientes com base em seus exames. Ao analisar dados visuais, também podemos utilizar a IA nos sistemas de navegação, como carros autônomos ou drones.

Figura 1 - Detecção de objetos em uma rua com visão computacional
Fonte: Security Informed (2021)

O(a) profissional que trabalha com modelos de IA chama Cientista de Dados, mas também é responsável por organizar, preparar, analisar, gerar visualizações, mensurar a qualidade dos modelos e preparar o modelo para subir para produção (deploy). É uma profissão relativamente nova e que exige conhecimento em diversas áreas como banco de dados, ciência da computação, matemática, estatística, métodos científicos e algoritmos de machine learning.

São muitas áreas que envolvem a formação completa de um cientista de dados e isso só cresce com o tempo. Mas vá com calma: estruture cada etapa do seu roadmap e faça com dedicação. Evite pular etapas. Uma frase famosa para os estudos do cientista de dados é: "não é um sprint, mas uma maratona". E cada esforço valerá a pena, pode ter certeza.

Aprendizado de Máquina

Dois dos principais tipos de **aprendizados** são o aprendizado **supervisionado** e **não supervisionado**. O aprendizado de máquina supervisionado é um tipo de aprendizado de máquina em que um algoritmo **aprende** a partir de um conjunto de dados **rotulados**. O conjunto de dados rotulados consiste em exemplos de entrada e seus resultados correspondentes. O algoritmo usa esses exemplos para aprender a relacionar as entradas para as saídas. O aprendizado de máquina não-supervisionado utiliza dados sem rótulos (ou classe).

Este aprendizado é chamado de **supervisionado** pois houve a atuação de um **supervisor** especialista ou algum sistema que **rotulou** cada exemplo do conjunto de dados. Alguns exemplos:

- Detecção de Spam em e-mails.
- Classificação de tipos de vinhos.
- Prever o preço de uma casa.
- Identificar o texto escrito a mão e converter para texto digital.
- Determinar se um tumor é benigno com base em uma imagem.
- Detectar uma atividade fraudulenta em uma transação de cartão de crédito.

Tabela, DataFrame e Nomes

Estes dados rotulados são estruturados em uma tabela. Vocês verão que na biblioteca Pandas chamaremos esta tabela de DataFrame. Vamos entender como se chama o nome de cada campo de um DataFrame (figura 2).

	comprim_sepala	largura_sepala	comprim_petala	largura_petala	classe
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figura 2 - Exemplo de tabela do conjunto de dados Íris
Fonte: Elaborado pelo autor (2024)

Este é uma parte do DataFrame, somente as 5 primeiras linhas de um famoso conjunto de dados da flor Íris. Este conjunto de dados pode ser baixado no repositório do UCI ou pela biblioteca Scikit-Learn (código-fonte 2).

A tabela possui 150 linhas, também chamadas de exemplos. As colunas "comprim_sepala", "largura_sepala", "comprim_petala" e "largura_petala" são os nomes dos atributos (ou features), ou seja, as características de cada exemplo. Estes atributos mostram as dimensões das flores, o comprimento e a largura da pétala e sépala, em centímetros. É importante mencionar que o nome das colunas foi inserido pelo usuário; no curso iremos ver de que forma alterar o nome de qualquer coluna pela biblioteca Pandas.

Na coluna "classe" temos o rótulo de cada exemplo, neste conjunto de dados há 3: Íris-setosa, Íris-virginica e Íris-versicolor; ou seja, há 3 tipos diferentes da flor Íris. Neste conjunto de dados há 50 exemplos para cada classe, ou seja, 50 linhas com valores distintos nos atributos para cada classe.

A coluna com os números de 0 a 4 é chamada de índice (index em inglês), um número inteiro que, por padrão, inicia no valor 0 na biblioteca Pandas. Esta coluna pode ou não existir ao carregar a tabela (tabela esta normalmente em formato csv - Comma-separated Values ou xlsx do Excel). Caso não exista, o Pandas atribui valores iniciando em 0. Este índice é customizável.

Lembre-se que todos estes conteúdos ficam sempre disponíveis quando precisar! Recorra sempre ao material e as videoaulas para fixar o conteúdo.

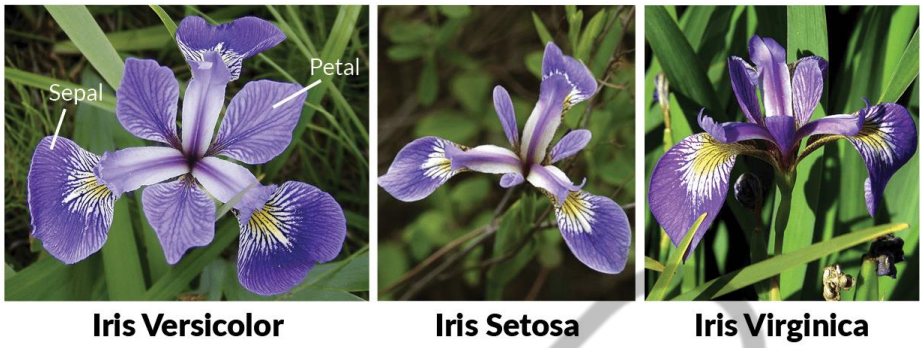


Figura 3 - Os 3 tipos da flor Íris do conjunto de dados
Fonte: LAC INPE [s.d.]

O aprendizado de máquina supervisionado, matematicamente, tem como objetivo aprender uma função que mapeia uma relação de uma ou mais entradas x (atributos) para uma saída y (classe)

$$\mathbf{f: x \rightarrow y}$$

O aprendizado acontece com o algoritmo de aprendizado observando cada exemplo e a respectiva classe como sendo a "resposta correta", ou mais correta possível. O "resposta correta" está entre aspas pois é uma estimativa que pode ser correta ou não, dependendo da quantidade de exemplos, do algoritmo escolhido para o aprendizado e outros.

```
In [1]: from sklearn import datasets

In [2]: iris = datasets.load_iris()
```

Código-fonte 2 – Código com a biblioteca Scikit-Learn carregando o dataset Íris executado no Jupyter Notebook
Fonte: Elaborado pelo autor (2024)

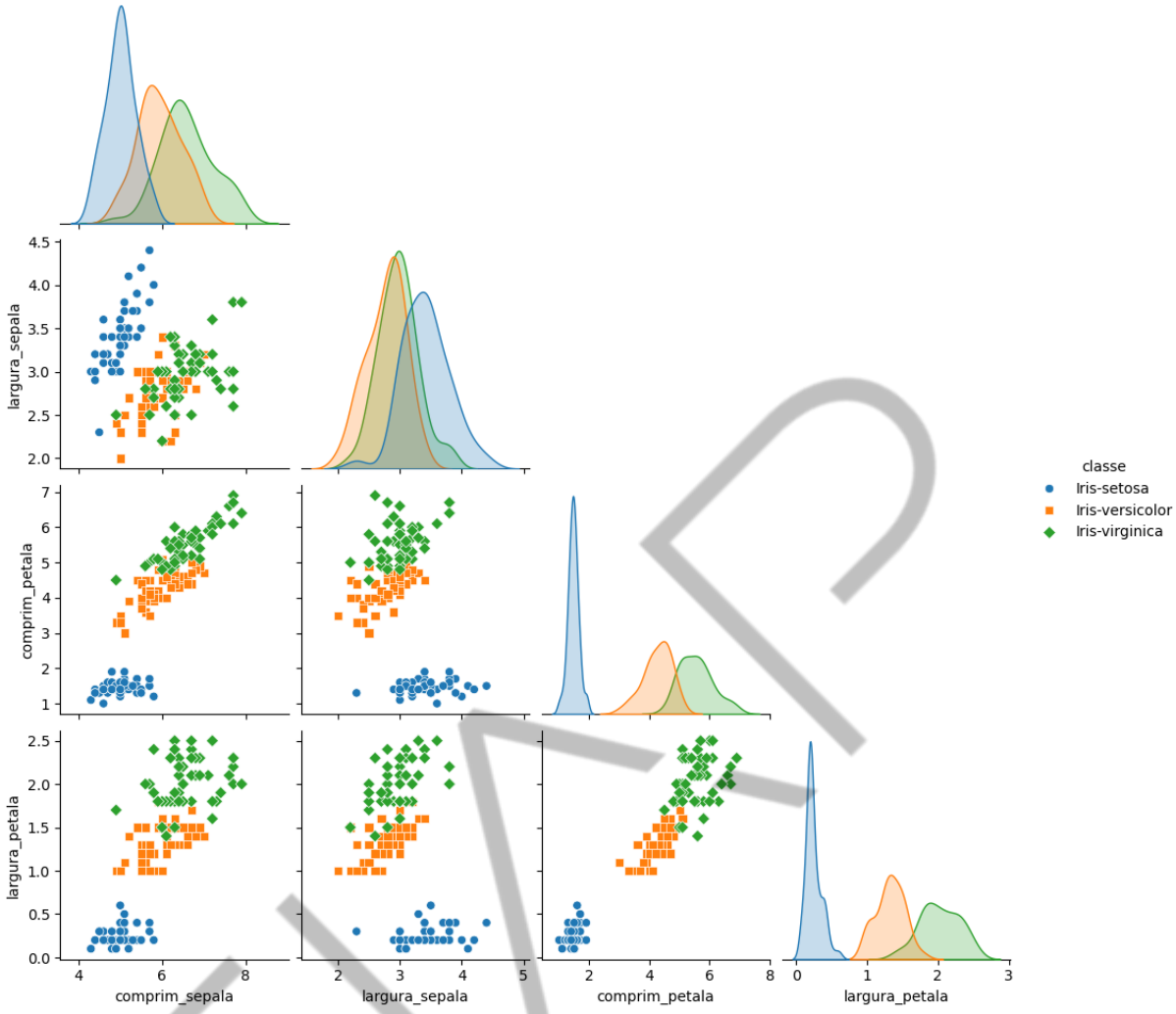


Figura 4 - Distribuição dos dados do conjunto de dados Íris feito com a biblioteca Seaborn
Fonte: Elaborado pelo autor (2024)

Etapas do Aprendizado de Máquina

Este aprendizado se dá aplicando o algoritmo escolhido em um conjunto de dados de **treino**. E avaliamos o desempenho deste algoritmo com o conjunto de **teste**. Mas como isso funciona na prática? A estrutura tradicional de um fluxo de aprendizado de máquina supervisionado é este:

1) O conjunto de dados é dividido, de forma randômica, em treino e teste. Normalmente escolhe-se uma porcentagem maior para treino e menor para teste, como por exemplo 70% e 30% respectivamente.

2) Após a divisão, aplica-se o algoritmo de aprendizado de máquina no conjunto de treino. Isso fará com que ele aprenda a relação entre os atributos e a respectiva classe para cada exemplo.

3) Uma vez com o algoritmo treinado, é mostrado para ele apenas os atributos do conjunto de teste para obter as respostas das classes que o algoritmo deduz ser para cada exemplo, baseado em seu aprendizado. A coluna com as classes do conjunto de teste são omitidas neste momento para o algoritmo.

4) Comparamos as respostas obtidas das classes pelo algoritmo com as verdadeiras e utilizamos alguma métrica para avaliar o desempenho.

5) Salvamos o resultado da métrica, voltamos à etapa inicial e realizamos ajustes para tentar melhorar o desempenho do algoritmo.

6) Com os ajustes feitos, temos um **modelo** de aprendizado de máquina.

Em cada exemplo do conjunto de dados de treino, o algoritmo sabe qual é a resposta correta. E o algoritmo usa seu conhecimento para tentar generalizar para novos exemplos que o algoritmo não viu antes e tentar obter a resposta certa.

Infelizmente, não podemos usar os dados que usamos para construir o modelo para avaliá-lo, ou seja, usar o mesmo conjunto de treino como teste, pois o algoritmo irá lembrar de todo o conjunto de treinamento e, portanto, sempre acertará a classe correta para qualquer exemplo do conjunto de treinamento, não possibilitando termos uma métrica de avaliação confiável.

Esta “lembrança” não nos indica se o nosso modelo generaliza bem (em outras palavras, se o aprendizado também terá um bom desempenho com novos dados). Estes conceitos de generalização e outros relacionados serão abordados em mais detalhes nas próximas aulas.

Avaliação do Modelo

Com os dados rotulados de teste e as predições que o modelo manda como saída, podemos medir o quão bom o modelo foi com relação à performance. Há diversas métricas de avaliação: acurácia, precisão, recall, F1-Score, Mean Squared Error, Root Mean Squared Error, Mean Absolute Error e Mean Average Precision, entre outros.

Tipos de Tarefas

Vimos que a coluna **classe** do conjunto de dados Íris possui 3 tipos de classe: setosa, virginica e versicolor. Pelo fato de a classe ser do tipo discreto (valores inteiros ou categorias), temos para este dataset a tarefa do tipo **Classificação**. Quando a classe possui valores contínuos (valores no intervalo dos números reais), temos uma **Regressão**. Estes são os dois tipos de tarefas em aprendizado supervisionado.

Um exemplo de conjunto de dados para a tarefa de Regressão é o dataset com preços de casas, o Real Estate Valuation disponível também no UCI.

No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
1	2012.916667	32.0	84.87882	10	24.98298	121.54024	37.9
2	2012.916667	19.5	306.59470	9	24.98034	121.53951	42.2
3	2013.583333	13.3	561.98450	5	24.98746	121.54391	47.3
4	2013.500000	13.3	561.98450	5	24.98746	121.54391	54.8
5	2012.833333	5.0	390.56840	5	24.97937	121.54245	43.1

Figura 5 - Exemplo de tabela do conjunto de dados Real Estate Valuation
Fonte: Elaborado pelo autor (2024)

Olhando a documentação deste dataset, vemos que são dados de casas na cidade de New Taipei, em Taiwan. As features são da segunda coluna, "X1 transaction date" até a penúltima "X6 longitude" e a coluna de rótulos é a "Y house price of unit area". Há uma coluna com um índice, a coluna "No". A primeira feature é um campo de data, a feature "X4" são dados com números inteiros e as demais features são valores contínuos (números decimais).

A classe são valores contínuos também, com o valor da casa por unidade de área. Este dataset possui 414 exemplos (linhas) e na figura 4 se apresentam apenas as 5 primeiras linhas.

Exemplo Prático e Didático de Regressão

Para exemplificar melhor a tarefa de regressão, vamos pensar em um exemplo simples para efeitos didáticos. Suponha que eu tenho um dataset com apenas uma feature e uma coluna com a classe e é fornecida inicialmente apenas uma linha de exemplo:

x	y
2	5

Apenas com este exemplo, você é capaz de dizer qual é a função que relaciona (mapeia) a feature com a classe? Podemos deduzir algumas, como:

- $x + 3 = y$
- $x^2 + 1 = y$
- $x^3 - 3 = y$

Mas, mesmo assim, caso decidamos por escolher uma destas funções, pode ser que não acertaremos para novos valores. Por exemplo: suponha que escolhamos a primeira função ($x + 3 = y$) e, no momento do teste do nosso algoritmo, precisemos calcular o valor de y para $x = 3$:

x	**y**
2	5
3	?

Utilizando a função $x + 3 = y$, com o valor de $x = 3$, somando 3 obtemos $y = 6$. Mas, no momento da avaliação, observamos que o valor correto é 7.

x	**y**
2	5
3	7

Observando este segundo exemplo é capaz que você perceba qual é a função correta. O que estamos fazendo aqui é similar à etapa de **treinamento** em aprendizado de máquina. Com mais alguns exemplos fica mais fácil saber qual é a função que relaciona a feature com a classe e **poder prever para novos exemplos**:

x	**y**
2	5
3	7
4	9
6	13

7	?

Com mais estes exemplos, podemos deduzir que a função que está mapeando a feature x com a classe y é do tipo: 2x + 1 = y. E, assim, conseguimos prever que a classe y para a feature valendo 7 é de: 7.2 + 1 = 15.

Este é um exemplo simples que conseguimos fazer mentalmente. Ao observar o conjunto de dados dos preços das casas, percebemos que essa acaba se tornando uma tarefa mais complexa.

Biblioteca Pandas

Para conseguirmos manipular os dados antes de aplicar o **machine learning**, vamos aprender alguns comandos básicos da biblioteca Pandas. É altamente recomendado que você veja e sempre recorra ao site da documentação oficial para aplicar suas implementações. Chegou a sua vez de aprender e praticar alguns comandos! Vamos lá?

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('iris.data')

In [3]: df.head()
```

Código-fonte 3 – Código Python com a biblioteca Pandas, executado no Jupyter Notebook, para carregar arquivos e visualizar
Fonte: Elaborado pelo autor (2024)

No código-fonte 3 vemos que a primeira entrada (In [1]) na célula do Jupyter é a importação da biblioteca. O "`as pd`" significa que sempre que precisarmos utilizar alguma função do Pandas, iremos usar antes "`pd`" como na segunda entrada (In [2]). A segunda entrada carrega o arquivo 'iris.data' do computador (supondo que o notebook do Jupyter, arquivo de extensão ipynb, esteja na mesma pasta que o arquivo a ser carregado) com a função `read_csv`; este DataFrame foi atribuído na variável de nome `df`. Na terceira entrada estamos utilizando a função `head()` para visualizar apenas as 5 primeiras linhas. A saída será semelhante à figura 2. Para visualizar as 5 últimas linhas, basta utilizar a função `tail()` (código-fonte 4).

```
In [4]: df.tail()

Out[4]:
```

	comprim_sepala	largura_sepala	comprim_petala	largura_petala	classe
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Código-fonte 4 – Código exibindo as últimas linhas do dataset Íris
Fonte: Elaborado pelo autor (2024)

Uma observação importante é que este dataset possui 150 linhas e os índices vão de 0 até 149.

Podemos observar uma linha do dataset. Para isso, usamos o comando "iloc[]" e dentro das chaves o número do índice da linha que queremos (código-fonte 5).

```
In [5]: df.iloc[0]

Out[5]:
comprim_sepala            5.1
largura_sepala            3.5
comprim_petala            1.4
largura_petala            0.2
classe            Iris-setosa
Name: 0, dtype: object
```

Código-fonte 5 – Código exibindo uma linha do dataset Íris
Fonte: Elaborado pelo autor (2024)

E também é possível observar os valores de uma coluna específica: para isso utilizamos as chaves e o nome da coluna (código-fonte 6) desejada.

```
In [6]: df['comprim_petala']

Out[6]:
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: comprim_petala, Length: 150, dtype: float64
```

Código-fonte 6 – Código exibindo uma coluna do dataset Íris
Fonte: Elaborado pelo autor (2024)

É possível combinar estes dois últimos comandos. Por exemplo: obter o valor da primeira linha da coluna "comprim_petala" ou as 5 primeiras linhas desta mesma coluna (código-fonte 7).

```
In [7]: df['comprim_petala'].iloc[0]

Out[7]: 1.4

In [8]: df['comprim_petala'].head()

Out[8]:
0    5.1
1    4.9
2    4.7
3    4.6
4    5.0
Name: comprim_sepala, dtype: float64
```

Código-fonte 7 – Código exibindo uma coluna do dataset Íris
Fonte: Elaborado pelo autor (2024)

Com a função `info()` é possível visualizar informações do DataFrame com intervalo dos índices, nome das colunas, quantidade de valores não-nulos nas colunas e tipo de dado em Dtype (código-fonte 8).

```
In [9]: df.info()

Out[9]:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   comprim_sepala  150 non-null    float64
 1   largura_sepala  150 non-null    float64
 2   comprim_petala  150 non-null    float64
 3   largura_petala  150 non-null    float64
 4   classe          150 non-null    object
```

Código-fonte 8 – Código exibindo as informações das colunas e os tipos de dados do dataset Íris
Fonte: Elaborado pelo autor (2024)

Uma forma alternativa - e mais simples - para ver a quantidade de linhas e colunas do DataFrame é com `shape` (código-fonte 9).

```
In [10]: df.shape

Out[10]: (150, 5)
```

Código-fonte 9 – Código exibindo a quantidade de linhas e colunas do dataset Íris
Fonte: Elaborado pelo autor (2024)

Para visualizar apenas algumas colunas, colocamos o nome da coluna separado por vírgula entre dois colchetes (código-fonte 10).

```
In [11]: df[['largura_sepala', 'largura_petala', 'comprim_petala']]

Out[11]:
```

	largura_sepala	largura_petala	comprim_petala
0	3.5	0.2	1.4
1	3.0	0.2	1.4
2	3.2	0.2	1.3
3	3.1	0.2	1.5
4	3.6	0.2	1.4
...	...	...	...
145	3.0	2.3	5.2
146	2.5	1.9	5.0
147	3.0	2.0	5.2
148	3.4	2.3	5.4
149	3.0	1.8	5.1

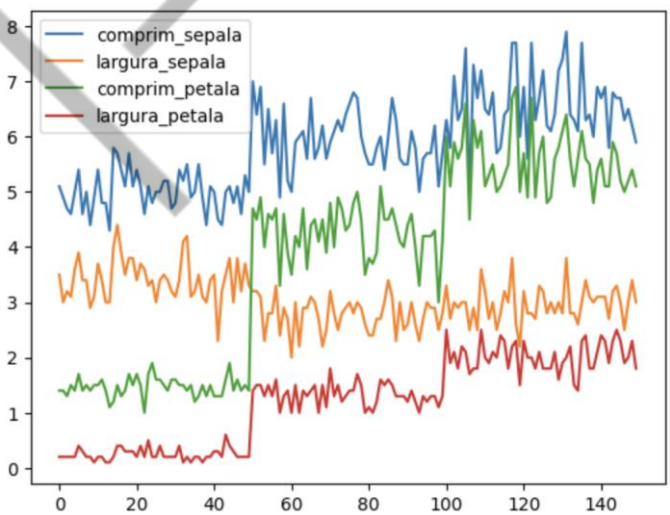
150 rows × 3 columns

Código-fonte 10 – Código exibindo as informações de colunas específicas do dataset Íris
Fonte: Elaborado pelo autor (2024)

Uma visualização gráfica de uma coluna pode ser feita com a função `plot()`, seja para o DataFrame todo (In[12]) como uma coluna específica (In[13]. No eixo vertical estão os valores de cada feature e no horizontal o índice (código-fonte 11).

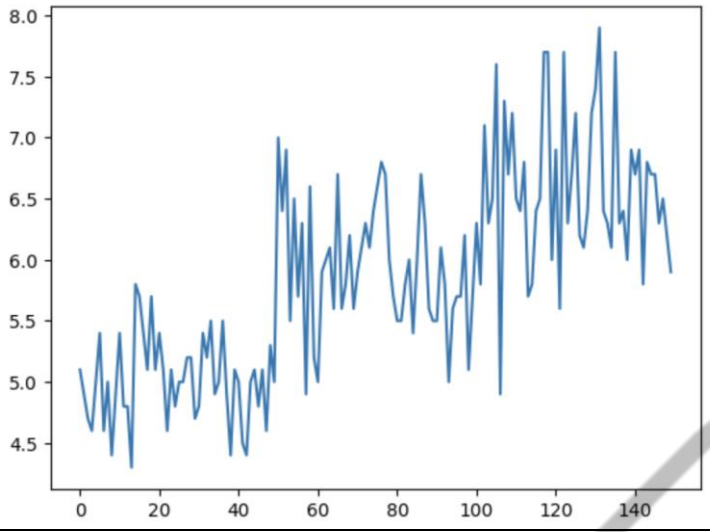
```
In [12]: df.plot()

Out[12]:
```




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

Out[13]:
```

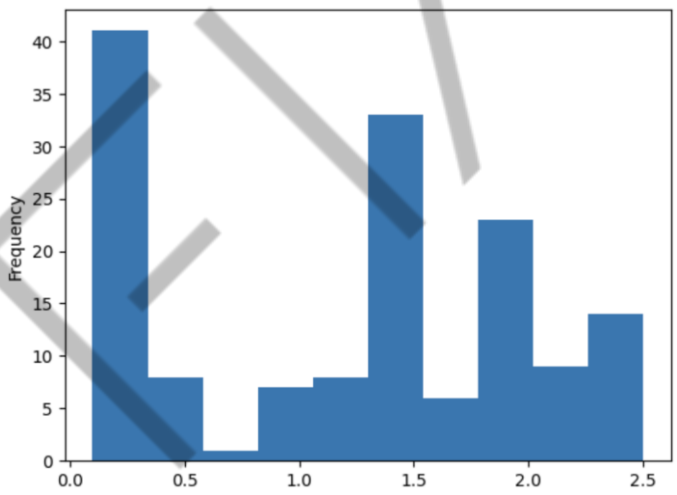


Código-fonte 11 – Código exibindo de forma gráfica as features do dataset Íris
Fonte: Elaborado pelo autor (2024)

Outra forma de visualização é com o histograma, um gráfico de barras que mostra a distribuição de frequências (quantidade de ocorrências) dada uma categorização (código-fonte 12). Neste caso foi feito de 0.5 em 0.5 cm.

```
In [14]: df['largura_petala'].plot(kind='hist')

Out[14]:
```



Código-fonte 12 – Código para exibir o histograma de uma coluna do dataset Íris
Fonte: Elaborado pelo autor (2024)

Outra função importante é obter os valores únicos (sem repetições) de uma coluna (código-fonte 13). O retorno será uma lista (array) com os valores.

```
In [15]: df['largura_petala'].unique()

Out[15]: array([0.2, 0.4, 0.3, 0.1, 0.5, 0.6, 1.4, 1.5,
1.3, 1.6, 1. , 1.1, 1.8, 1.2, 1.7, 2.5, 1.9, 2.1, 2.2, 2. , 2.4,
2.3])
```

Código-fonte 13 – Código para localizar a linha mediante condições do dataset Íris
Fonte: Elaborado pelo autor (2024)

Caso precisemos buscar no DataFrame a linha que possui um valor específico em alguma coluna, há duas formas (código-fonte 14). Por exemplo: qual/quais linha(s) possui(em), na coluna 'largura_petala', o valor 1.1?

```
In [16]: df[df['largura_petala'] == 1.1]

Out[16]:
```

	comprim_sepala	largura_sepala	comprim_petala	largura_petala	classe
69	5.6	2.5	3.9	1.1	Iris-versicolor
80	5.5	2.4	3.8	1.1	Iris-versicolor
98	5.1	2.5	3.0	1.1	Iris-versicolor

```
In [17]: df.query('largura_petala == 1.1')

Out[17]:
```

	comprim_sepala	largura_sepala	comprim_petala	largura_petala	classe
69	5.6	2.5	3.9	1.1	Iris-versicolor
80	5.5	2.4	3.8	1.1	Iris-versicolor
98	5.1	2.5	3.0	1.1	Iris-versicolor

Código-fonte 14 – Código para localizar as linhas mediante condições do dataset Íris (1)
Fonte: Elaborado pelo autor (2024)

Nesta busca pelas linhas, é possível aplicar outros operadores matemáticos como maior (>), maior igual (>=), menor (<), menor igual (<=) e diferente (!=). Coloque cada condição entre parênteses e o operador entre estas condições: o operador lógico E (&) ou OU (|).

```
In [18]:
df[(df['largura_petala']>1.1)&(df['largura_petala']<1.4)]
```

Out[18]:

	comprim_sepala	largura_sepala	comprim_petala	largura_petala	classe
53	5.5	2.3	4.0	1.3	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor
58	6.6	2.9	4.6	1.3	Iris-versicolor
64	5.6	2.9	3.6	1.3	Iris-versicolor
71	6.1	2.8	4.0	1.3	Iris-versicolor
73	6.1	2.8	4.7	1.2	Iris-versicolor
74	6.4	2.9	4.3	1.3	Iris-versicolor
82	5.8	2.7	3.9	1.2	Iris-versicolor
87	6.3	2.3	4.4	1.3	Iris-versicolor
88	5.6	3.0	4.1	1.3	Iris-versicolor
89	5.5	2.5	4.0	1.3	Iris-versicolor
90	5.5	2.6	4.4	1.2	Iris-versicolor
92	5.8	2.6	4.0	1.2	Iris-versicolor
94	5.6	2.7	4.2	1.3	Iris-versicolor
95	5.7	3.0	4.2	1.2	Iris-versicolor
96	5.7	2.9	4.2	1.3	Iris-versicolor
97	6.2	2.9	4.3	1.3	Iris-versicolor
99	5.7	2.8	4.1	1.3	Iris-versicolor

Código-fonte 15 – Código para localizar as linhas mediante condições do dataset Íris (2)
Fonte: Elaborado pelo autor (2024)

Outro comando interessante para obter um resumo das features com algumas métricas estatísticas é o `describe()`. Com ele obtemos a quantidade de exemplos (`count`), média (`mean`), desvio padrão (`std`), valor mínimo (`min`), os valores dos quartis (`25%`, `50%` e `75%`) e valor máximo (`max`).

```
In [19]: df.describe()
```

Out[19]:

	comprim_sepala	largura_sepala	comprim_petala	largura_petala
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Código-fonte 16 – Código para obter algumas métricas estatísticas das features do dataset Íris
Fonte: Elaborado pelo autor (2024)

MERCADO, CASES E TENDÊNCIAS

O estado do Texas, nos Estados Unidos, está testando um novo modelo de IA para corrigir provas realizadas por estudantes do sistema regular de ensino. A ideia é substituir a maior parte dos examinadores humanos que trabalham avaliando a performance dos alunos da região. O governo local estima, com o uso do modelo de **Processamento de Linguagem Natural**, uma economia entre US$15 milhões e US$20 milhões por ano. Leia mais aqui.

A tecnologia está cada vez mais presente em alguns golpes, algo que deve aumentar em 2024 segundo um levantamento feito pela Kaspersky. Diversas entidades e governos já sinalizaram que estão preocupados com alguns rumos adotados com o uso da inteligência artificial. Saiba mais sobre o assunto aqui.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula aprendemos os conceitos iniciais de Aprendizado Supervisionado e suas características. Vimos também a estrutura de um DataFrame, os nomes dos campos e alguns comandos para manipulá-lo. Por fim, entendemos que há uma ordem específica nas etapas do fluxo do aprendizado de máquina.

REFERÊNCIAS

ANDREW, N. G. **Machine learning yearning book**. 2024. Disponível em: <https://info.deeplearning.ai/machine-learning-yearning-book>. Acesso em: 24 abr. 2024.

GRUPY SANCA. **Curso Introdutório de Python**. 2024. Disponível em: <http://curso.grupysanca.com.br/pt/latest/>. Acesso em: 04 jul. 2024.

GRUS, J. **Data science from scratch**: first principles with python. 2. ed. Sebastopol: O'Reilly Media, 2019.

LAC INPE. **Data Science Example - Iris dataset**. [s.d.] Disponível em: <www.lac.inpe.br/~rafael.santos/Docs/CAP394/WholeStory-Iris.html>. Acesso em: 04 jul. 2024.

SCIPY.ORG. **Numpy and Scipy documentation**. 2024. Disponível em: <https://docs.scipy.org/doc/>. Acesso em: 24 abr. 2024.

Pandas documentation — pandas 2.2.2 documentation. Disponível em: <https://pandas.pydata.org/docs/ >. Acesso em: 04 jul. 2024.

PYTHON. **Python 3.12.4 documentation**. 2024. Disponível em: <https://docs.python.org/3/>. Acesso em: 04 jul. 2024.

RUSSELL, S.; NORVIG, P. **Artificial intelligence**: a modern approach, global edition. 4. ed. Londres: Pearson Education, 2021.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM journal of research and development**, v. 44, n. 1.2, p. 206–226, 2000.

SCIKIT-LEARN. **Scikit-learn**. 2024. Disponível em: <https://scikit-learn.org/stable/index.html>. Acesso em: 04 jul. 2024.

SECURITY INFORMED. **AI Computer Vision for Operation Centers**: Moving From Reactive to Proactive Situational Awareness. 2021. Disponível em: <https://www.securityinformed.com/news/userful-explains-moving-reactive-proactive-situational-co-1611850725-ga.1627378657.html>. Acesso em: 04 jul. 2024.

STAT LEARNING. **An introduction to statistical learning**. 2024. Disponível em: <https://www.statlearning.com>. Acesso em: 04 jul. 2024.

PALAVRAS-CHAVE

Palavras-chave: Inteligência Artificial. Aprendizado supervisionado. Python. Aprendizado de Máquina. Machine Learning. Classificação. Regressão. Conjunto de treinamento. Conjunto de teste. Métricas de avaliação.