Relatório 4 - Principais Bibliotecas e Ferramentas Python para Aprendizado de Máquina (I)

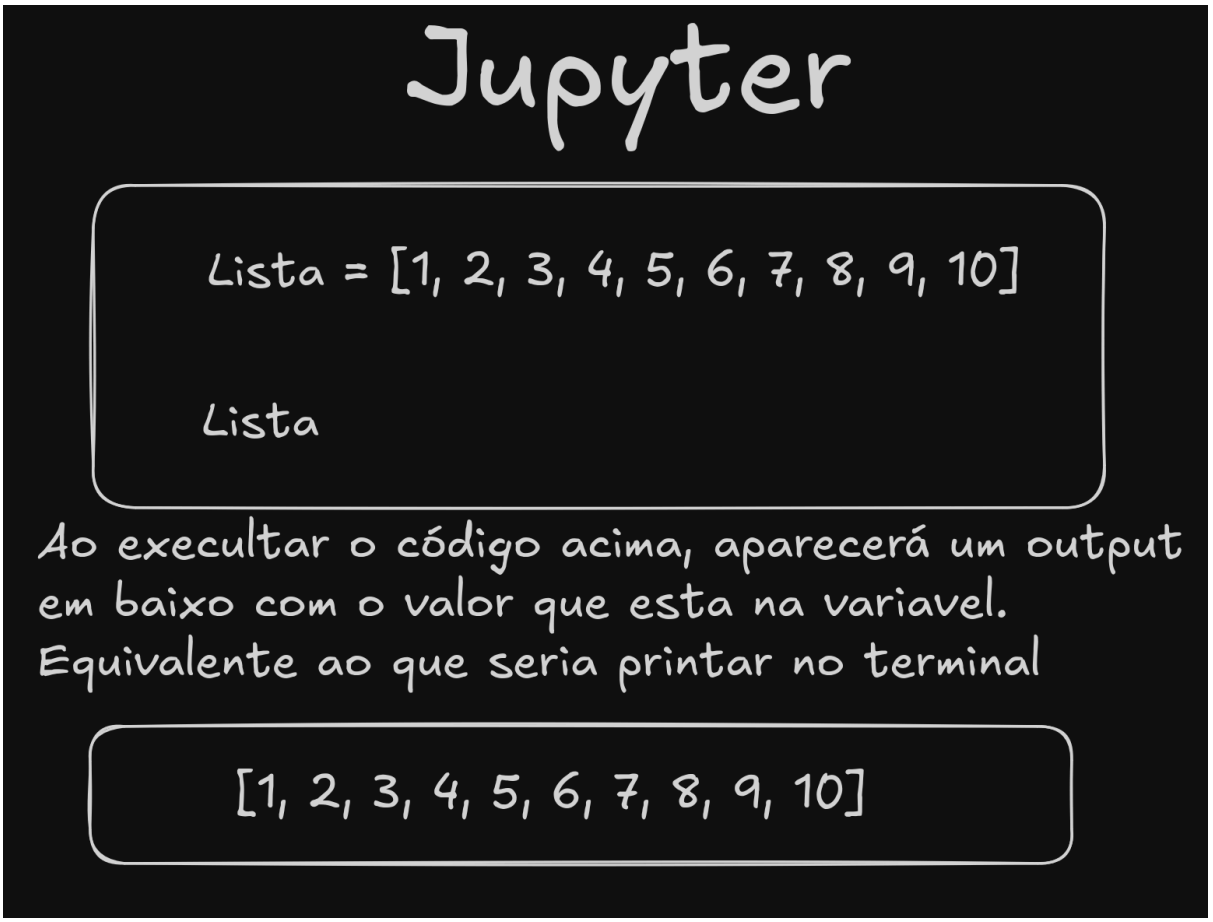
Felipe Fonseca

Descrição da atividade

Jupyter

A primeira parte do card foi o curso de como utilizar o notebook Jupyter para programar. Ele é basicamente um software no qual é possível criar os códigos, mas rodar eles em uma ordem diferente da que está estabelecida, além disso ele permite a visualização muito fácil de qualquer variável, seja o retorno de uma função, uma variável do tipo do data frame ou qualquer outra coisa.

O notebook jupyter facilita para o degub de código e principalmente para a amostra dos resultados, invés de ter que colocar prints ou algo do tipo para conseguirmos ver o que está acontecendo com as variáveis, ele permite vermos isso diretamente, o que é algo revolucionário.

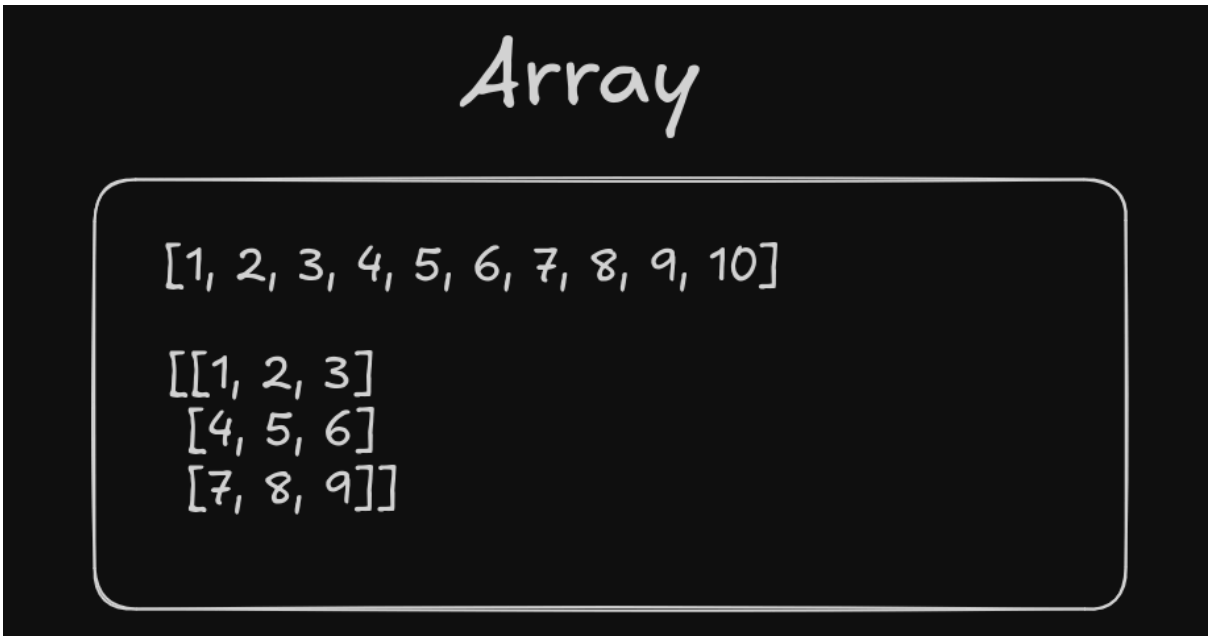


Numpy

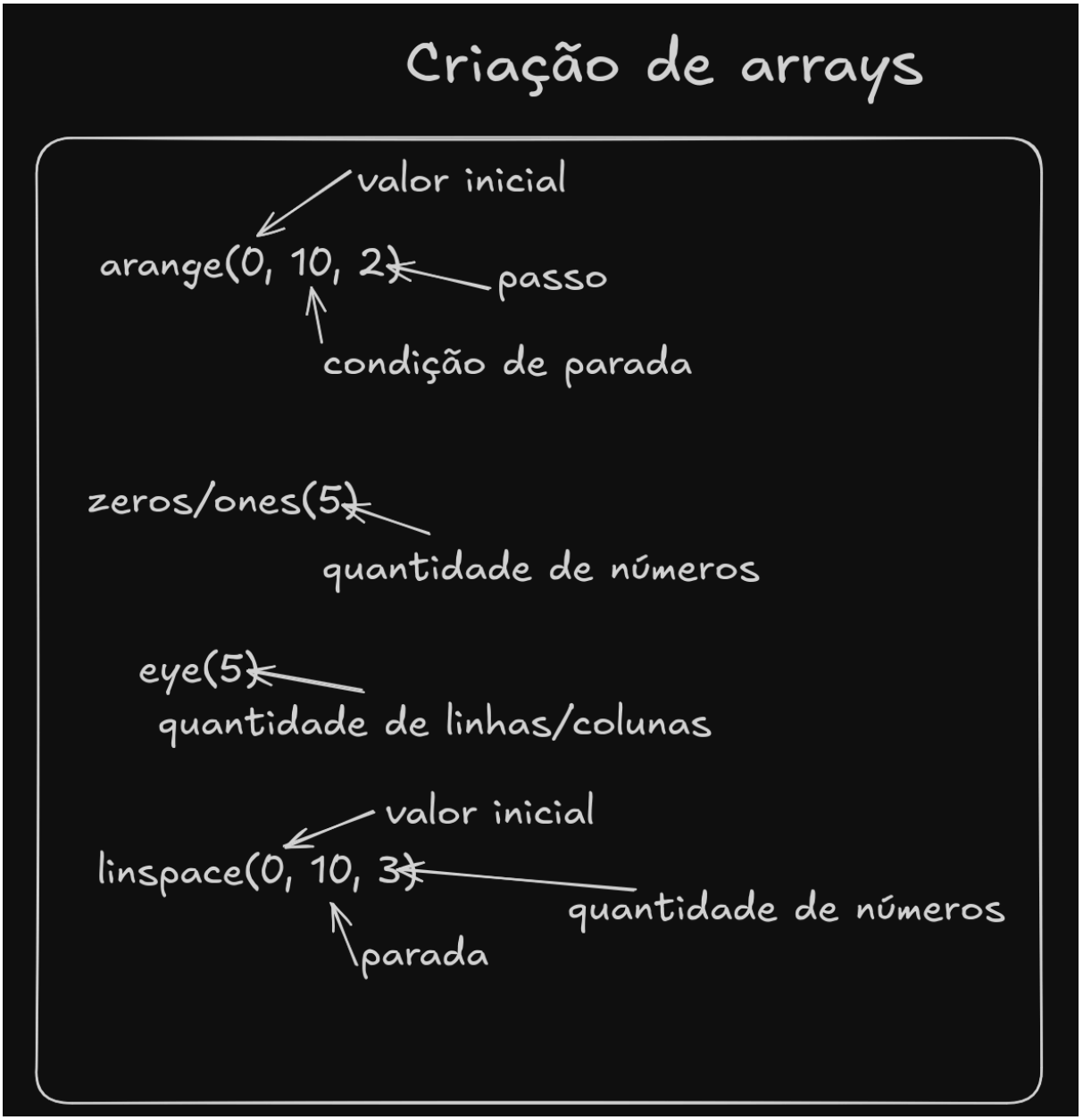
Numpy é uma biblioteca rápida e eficiente, escrita em C, que tem como foco Álgebra Linear, tendo várias funções e métodos que são úteis para tal. Uma convenção é sempre importar o Numpy como np.

Arrays

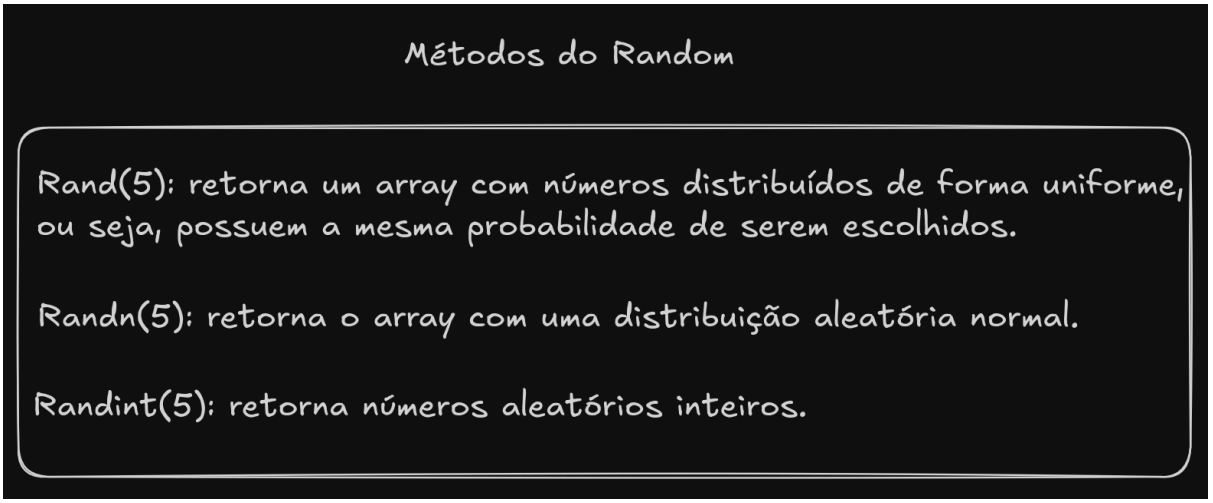
Começando pelas Arrays, um array no numpy serve para tratar e cuidar de N dados que estiverem ali, possuindo vários métodos para o tratamento desses dados. Pode-se criar uma array do numpy através da função Array. A array no numpy não possui um limite de tamanho, ela pode se comportar como uma lista ou como uma matriz, e até como um objeto de 3d (possuindo 3 dimensões invés de apenas linhas e colunas.)



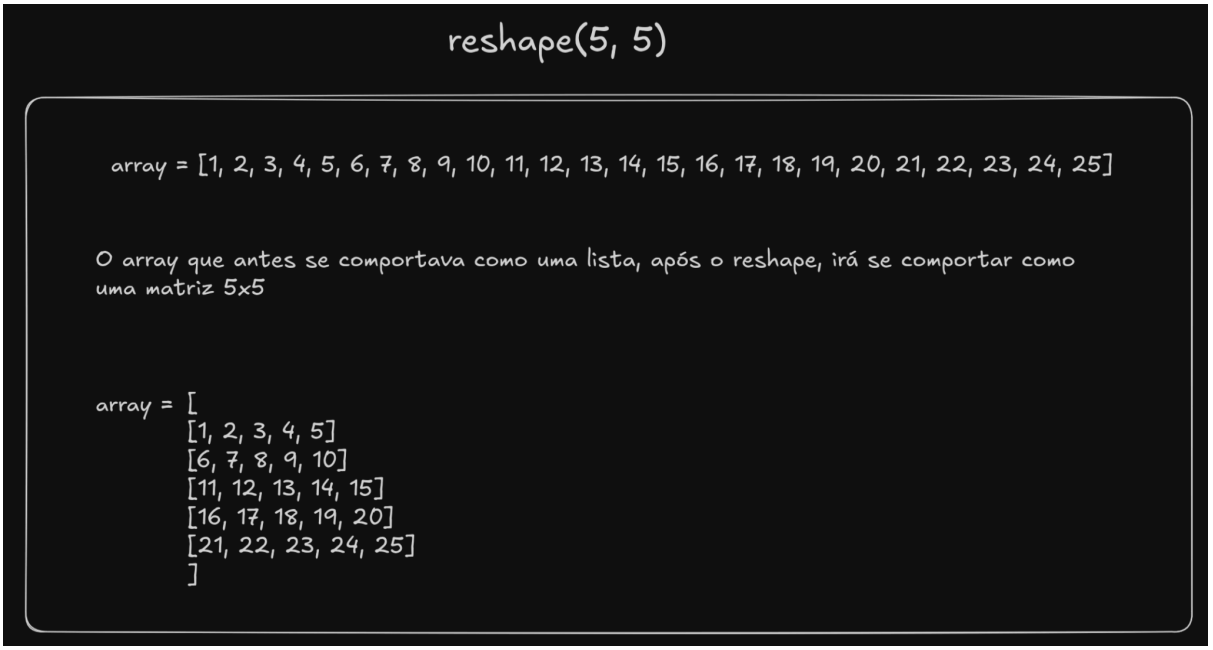
Algumas outras funções podem ser utilizadas para criar um array também, como por exemplo a função arange, que funciona igual a função range do python. Duas funções que possuem funcionamento similar é a função **zeros** e a função **ones**, ambas você define a quantidade de elementos, e ele retornará um array com a quantidade especificada, porém contendo apenas zeros (caso tenha usado a função zeros) ou 1 (caso tenha usado a função ones). Também existe a função eye, que retorna uma matriz identidade com o tamanho especificado (matriz identidade é uma matriz quadrada). A função linspace é a função que eu achei mais interessante dessas de criação de array, ela retorna um array na qual o espaçamento entre os elementos será sempre igual. Primeiro você define um ponto de partida, depois o ponto de chegada, e a quantidade de elementos que você quer no array, e então ele definirá os valores.



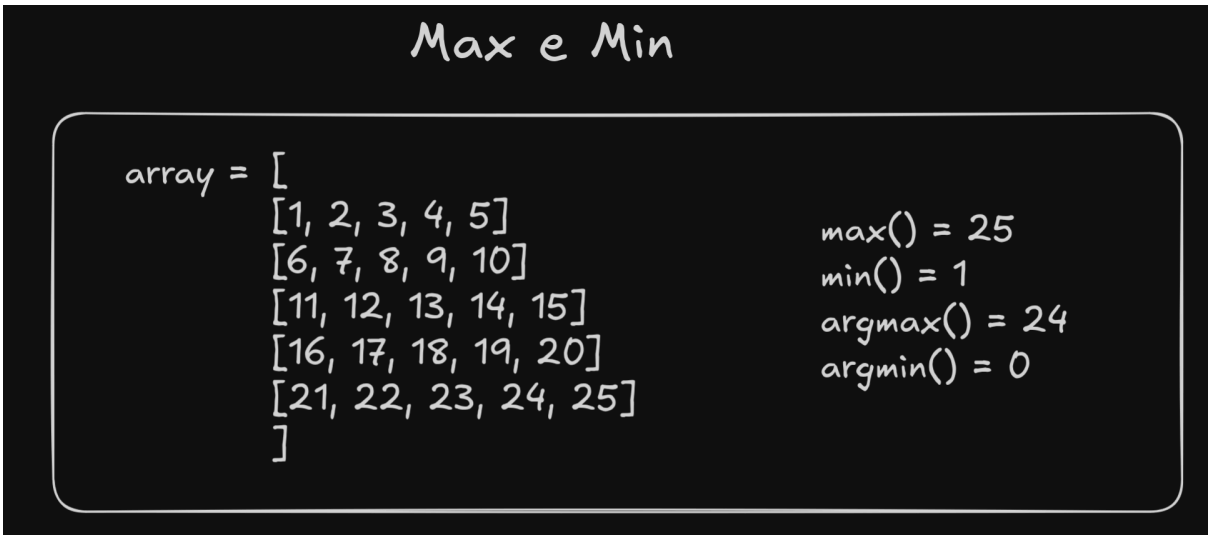
A função random serve para gerar valores aleatórias e possui vários métodos:



A função reshape serve para formatar o array, podendo transformar um array que está se comportando como lista, para um array que se comporte como uma matriz.[



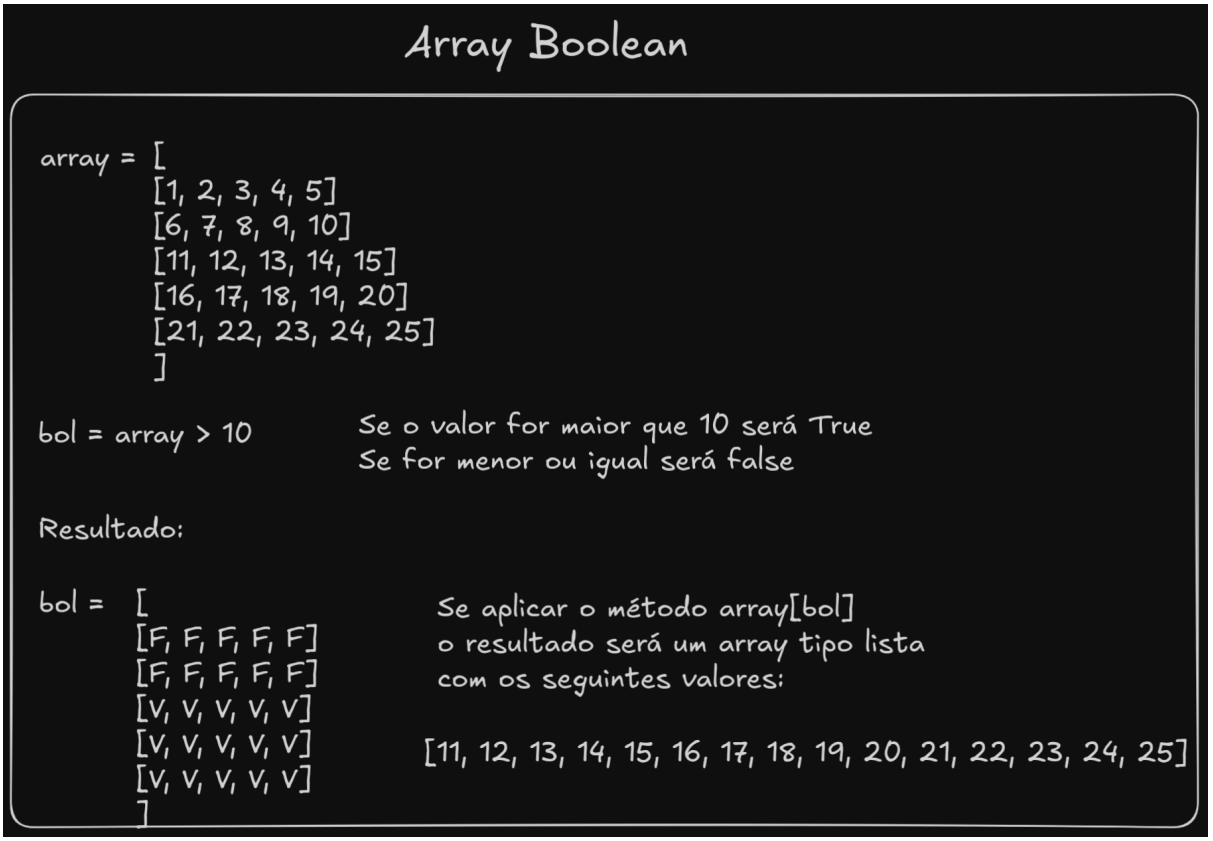
Através de funções como max() e min() é possível achar o maior/menor valor do array, e através de argmax() e argmin() é possível achar o index desse maior/menor valor



Indexação

As formas de indexação de um array funciona igual a de uma lista normal do python, tanto para achar um único valor, quanto para achar uma lista de valores.

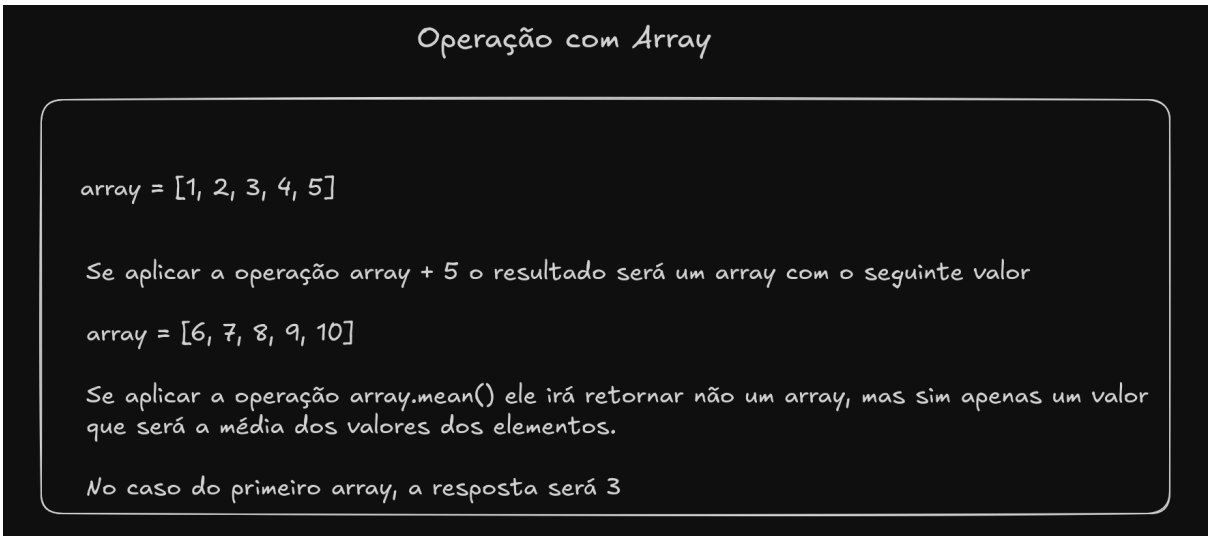
Um funcionamento que eu achei interessante foi o método de criar um array do tipo boolean com valores de verdadeiro ou falso para uma condição aplicada de forma escalar nos elementos de um array. Dessa forma, é possível criar um novo array apenas com os valores onde essa condição foi aplicada.



Operações

É possível fazer operações de array com array, e também é possível fazer operações de array com um escalar, utilizando operados comuns como - ou +, mas também com funções, por exemplo utilizando a função sqrt() que calcula a raiz quadrada do elemento. Se for utilizada em um array, a função irá retornar um array com a raiz quadrada sendo aplicada em seus elementos.

Algumas funções interessantes são a função mean(), que retorna a média de todos os valores presentes no array, ou então a função std, que calcula o desvio padrão do array.



Pandas

Séries

Uma série no pandas se comporta basicamente da mesma forma que um dicionário do python, conteúdo um “index” e um valor para cada um desse index. A seleção de um valor também funciona da mesma forma.

Um fator apresentado na videoaula foi a possibilidade de fazer operação de soma entre duas séries, na qual ela faz a verificação através do index. O retorno da operação será uma nova série contendo todos os index das duas séries, os index que aparecem nas duas terão os valores somados e os que aparecem em apenas uma série ficará com um valor vazio.

Dataframe

Data frame é a principal estrutura de dados utilizada no pandas, ela pode ser comparada a uma planilha do excel, onde possuem linhas e colunas e pode armazenar grandes quantidades de dados.

Para fazer a busca de uma coluna, é a mesma coisa de buscar o valor de um dicionário, você apenas coloca o valor/nome da coluna e consegue fazer a busca. O retorno dessa busca é justamente uma série, e essa série terá como nome o mesmo nome da coluna, e os index serão os index/nome das linhas, contendo os valores de cada um. É também possível passar uma lista de colunas.

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund Sort ascending
0	The Santa Anna National Bank	Santa Anna	Texas	5520	Coleman County State Bank	June 27, 2025	10549
1	Pulaski Savings Bank	Chicago	Illinois	28611	Millennium Bank	January 17, 2025	10548
2	The First National Bank of Lindsay	Lindsay	Oklahoma	4134	First Bank & Trust Co.	October 18, 2024	10547
3	Republic First Bank dba Republic Bank	Philadelphia	Pennsylvania	27332	Fulton Bank, National Association	April 26, 2024	10546
4	Citizens Bank	Sac City	Iowa	8758	Iowa Trust & Savings Bank	November 3, 2023	10545
5	Heartland Tri-State Bank	Elkhart	Kansas	25851	Dream First Bank, N.A.	July 28, 2023	10544

	Bank Name
0	The Santa Anna National Bank
1	Pulaski Savings Bank
2	The First National Bank of Lindsay
3	Republic First Bank dba Republic Bank
4	Citizens Bank
5	Heartland Tri-State Bank

	Bank Name	City
0	The Santa Anna National Bank	Santa Anna
1	Pulaski Savings Bank	Chicago
2	The First National Bank of Lindsay	Lindsay
3	Republic First Bank dba Republic Bank	Philadelphia
4	Citizens Bank	Sac City
5	Heartland Tri-State Bank	Elkhart

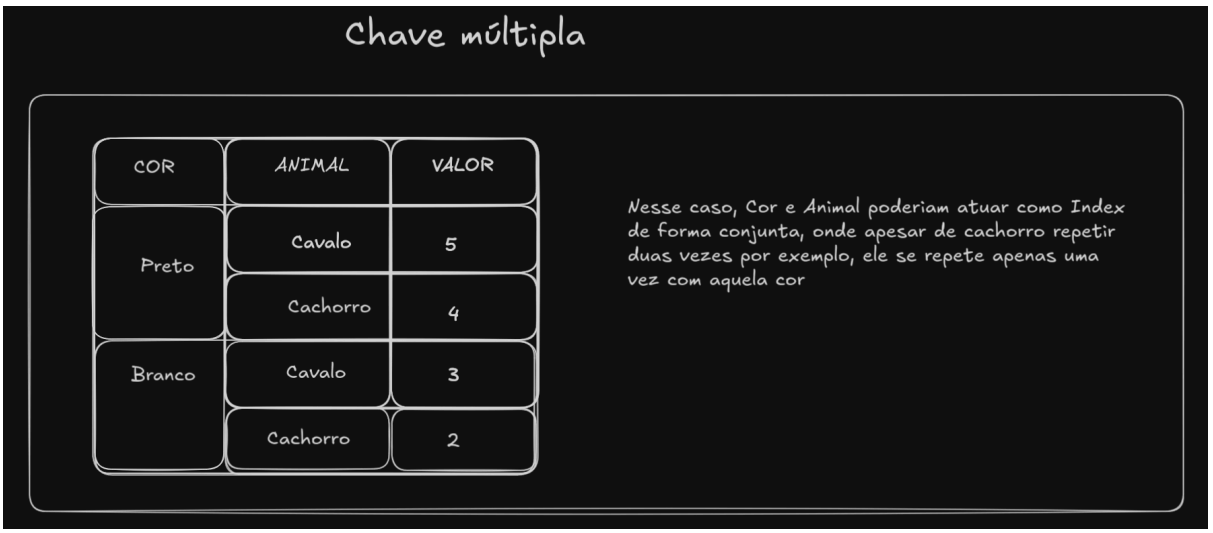
Para criar uma nova coluna é preciso basicamente “fingir que ela já existe”, e aplicar um valor a ela. Para deletar a coluna, você pode fazer de duas formas diferentes. Ou utilizando o comando drop e especificando a coluna, ou então utilizando o comando del. A diferença é que o drop é uma função do pandas, enquanto del é um comando do python. O recomendo creio que seja utilizar sempre o drop que é próprio do pandas e trata melhor os dados.

Para fazer a busca por um valor exato (linhas e colunas) usa o comando loc ou o comando iloc, e então passa as linhas e colunas como referência. A diferença é que no loc você passa o valor exato da linha/coluna, enquanto no iloc você passa o index como se fosse uma lista do python. Outra forma de fazer busca no dataframe é através de uma condição para os valores da coluna, passando uma expressão que retorna apenas as linhas na qual a expressão é verdadeira..

Assim como em uma matriz no numpy, também é possível criar um dataframe boleean para verificar os valores presentes e então criar um novo dataframe com base nisso. A diferença é que aqui no dataframe não tem como simplesmente sumir com os valores, então eles ficam definidos como NaN.

No pandas também é possível resetar o index para o padrão através de reset_index(), o que fará com que o index antigo vire uma nova coluna e agora o novo index será uma enumeração padrão (0 até n), ou então até definir uma coluna como novo index através do comando set_index().

Ainda sobre o index, é possível fazer uma indexação múltipla, contendo basicamente dois valores de index, como se fosse duas chaves primárias do sql.



Tratamento de dados

É possível tratar os dados de várias formas diferentes, uma que foi apresentada no curso foi o tratamento de valor nulos, utilizando de funções como dropna() ou fillna() que removem ou substituem os valores nulos respectivamente.

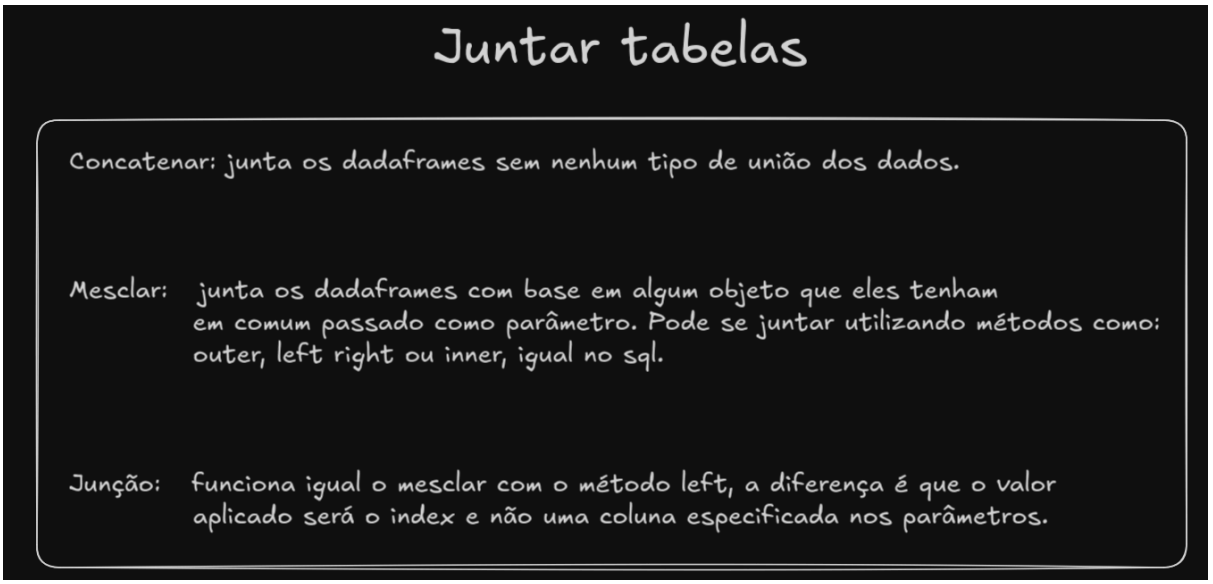
Agrupamento

O dataframe permite o agrupamento através de algum tipo do dado, onde esse dado aparecerá como novo index. Por padrão, o que será definido é apenas um objeto, e será necessário mais alguma operação para conseguir que esse grupo realmente mostre alguma coisa.

Alguns métodos são interessantes de serem utilizados com o agrupamento, como por exemplo o método sum(), que fará a soma dos valores do agrupamento, ou então a função mean() que irá calcular a média dos valores, ou até a função count(), que conta quantas vezes o objeto agrupado apareceu.

Junções

É possível juntar dataframes com o pandas através de 3 métodos diferentes:



Operações

As operações normalmente são aplicadas nas colunas inteiras, e podem ser funções como por exemplo unique() que filtra a tabela mostrando os elementos sem repetição, ou a função munique() que conta a quantidade desses elementos. A função value_counts() também funciona de forma semelhante, filtrando os elementos e mostrando a quantidade de vezes que eles aparecem na coluna. Outra coisa que achei interessante foi a função apply, onde você pode aplicar uma função aos valores de uma coluna.

Importação/Exportação

Outra coisa muito interessante do pandas é a capacidade de importar arquivos csv ou arquivos do excel para o dataframe, da mesma forma que é possível exportar o dataframe para esses arquivos. Também é possível ler e importar para um dataframe através de uma url html.

Prática:

Na prática do pandas eu dei uma brincada principalmente com as colunas, utilizando uma base de dados fake com jogadores do São Paulo e do Palmeiras. Utilizei métodos para criar e excluir coluna, resetar e inserir index, criei um multiindex, Utilizei do método head e do sort_values para visualizar bem os dados.Utilizei de condições na hora de fazer pesquisa para pegar apenas alguns dados. Utilizei o dropna para excluir valores nulos de uma tabela em que ela não era necessária. Utilizei um método de concatenar as duas tabelas dos df para formar uma tabela maior e utilizei um método de agrupamento por time para verificar a quantidade de gols que cada um fez. Também utilizei no final a função xs para fazer uma busca pelo sub_index invés do principal ou pelos dois.

Na prática do numpy eu não consegui pensar em nada muito criativo pra brincar, então fiquei apenas testando alguns métodos com os arrays, criando algumas matrizes, testando o reshape etc. Realmente nem eu gostei da prática que eu fiz no numpy, se voltar o card nessa parte pelo menos dá uma ideia do que eu posso praticar aqui porque não consegui pensar em nada.

Conclusões

Sobre o jupyter, sinceramente eu achei uma ferramenta fantástica, é algo revolucionário para a forma que podemos fazer os códigos e programar, a facilidade de ver os dados e compilar em uma ordem diferente do que está escrito é absurda. A única coisa que eu não gostei foi da versão padrão dele, aberta no navegador. Provavelmente para as próximas aulas tentarei configurar ele para rodar no vscode.

Sobre o Numpy eu achei uma biblioteca realmente bem interessante que possui muitos métodos para os tratamentos de dados. Eu gosto muito da otimização também, o fato de ser uma biblioteca extremamente leve certamente é algo que facilita o trabalho para uma grande quantidade de dados.

E sobre o Pandas, essa sim é uma biblioteca boa, eu conhecia ela já anteriormente e já tinha usado ela um pouco. Sinceramente acho mais fácil trabalhar com o pandas do que com o excel por exemplo, é muito fácil fazer o tratamento de vários tipos de dados como os dados vazios, ou então aplicar condições, substituir números, fazer verificações etc.

Referencias

Apenas as videoaulas disponibilizadas no card:

3. Jupyter Notebook

- 8 🎥 Jupyter Notebook
- 9 🎥 Opcional: Ambientes Virtuais

5. Python para Análise de dados - Numpy

- 18 🎥 Bem vindo à seção de Numpy
- 19 🎥 Introdução ao Numpy
- 20 🎥 Criação de vetores e arrays e numpy.random
- 21 🎥 Indexação e Fatiamento de arrays
- 22 🎥 Operações com Numpy Arrays
- 23 🎥 Exercícios
- 24 🎥 Soluções

6. Python para Análise de dados - Pandas

- 25 🎥 Bem vindo à seção Pandas
- 26 🎥 Introdução ao Pandas
- 27 🎥 Series
- 28 🎥 DataFrame - Criação e Fatiamento
- 29 🎥 DataFrame - Seleção condicional, set_index
- 30 🎥 DataFrame - Índices Multiníveis
- 31 🎥 Dados Ausentes
- 32 🎥 GroupBy
- 33 🎥 Concatenar, Juntar e Mesclar
- 34 🎥 Operações
- 35 🎥 Entrada e Saída de dados

7. Python para Análise de dados - Exercícios Pandas/

- 36 🎥 Exercício 1 - Salários SF
- 37 🎥 Soluções - Salários SF
- 38 🎥 Compras de E-comerce
- 39 🎥 Soluções Compras de E-comerce