

Aula 02 - Importando a Biblioteca Pandas

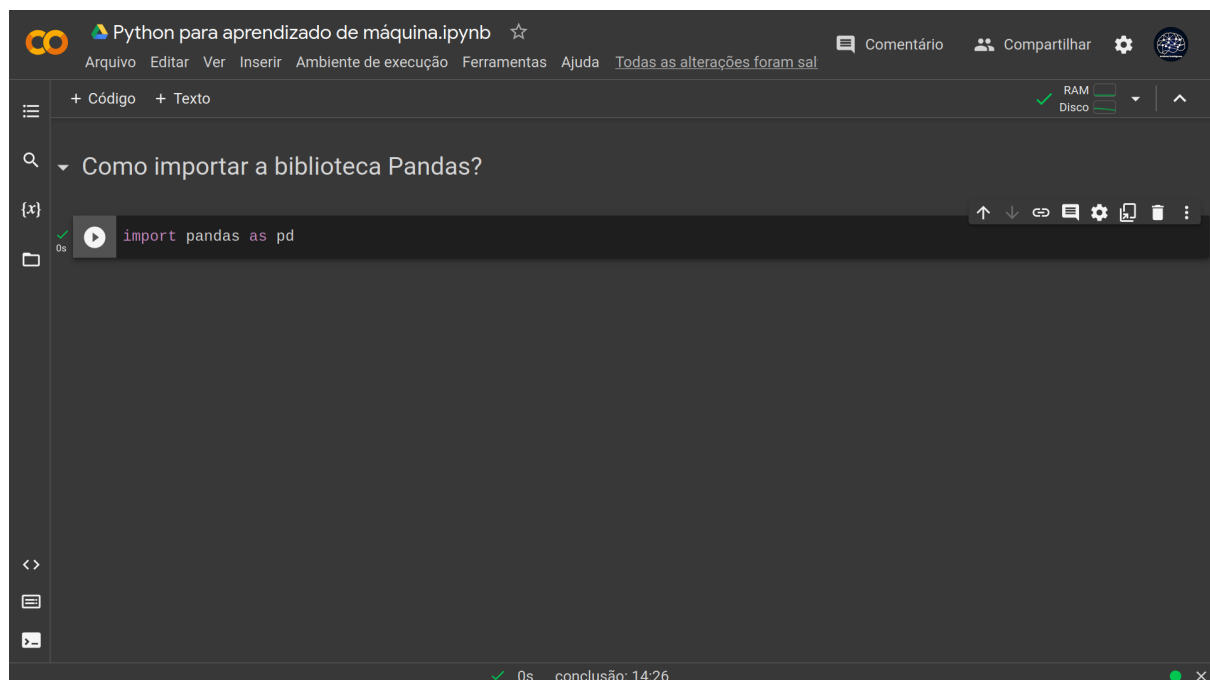
Como que a gente faz a instalação da biblioteca Pandas? Bom, se a gente não estivesse usando a ferramenta do google, o google colab para fazer os nossos códigos em Python, a gente precisaria acessar o site oficial do pandas.

<https://pandas.pydata.org/>.

Aqui no site contém uma grande quantidade de informações sobre a biblioteca Pandas, incluindo documentação completa, tutoriais, exemplos de código, referências, tudo sobre o Pandas, a gente pode encontrar nesse site, e até mesmo a maneira que ele é implementado. Além disso, o site também fornece links para recursos adicionais relacionados ao Pandas, como livros, cursos e eventos.

E, caso a gente queira fazer a instalação do Pandas no nosso computador, basta que a gente venha nessa parte install Pandas e a gente siga este tutorial. Mas como a gente tá usando a plataforma do google colab, a gente não precisa fazer a instalação de nada, pois tudo que a gente precisa, todas bibliotecas, ou pelo menos a maior parte, já está dentro da própria plataforma do colab.

Então, para a gente começar a usar o Pandas basta importar a biblioteca usando o comando import.



Mas por que a gente precisa usar o pandas quando a gente está trabalhando com aprendizado de máquina?

Bom, como muitos devem conhecer as tabelas do Excel são amplamente utilizadas para armazenar dados, mas quando estamos lidando com análise de dados ou aprendizado de máquina, precisamos trabalhar com conjuntos de dados muito grandes e complexos. Nesses casos, torna-se inviável usar ferramentas como o Excel, que têm limitações em termos de tamanho e capacidade de processamento, é aí que o Pandas se torna essencial.

O Pandas permite que a gente carregue, que a gente manipule e analise grandes conjuntos de dados de maneira eficiente, além de oferecer uma ampla variedade de funções e métodos que tornam o trabalho com dados mais simples e intuitivo. Com o Pandas, é possível executar tarefas complexas de análise de dados com apenas algumas linhas de código, o que economiza tempo e aumenta a produtividade. Além do mais, a gente pode ver como esses dados se comportam de maneira visual através dos gráficos.

Antes da gente começar a fazer qualquer código, a gente precisa entender outros dois tipos de dados, estruturas de dados principais no Pandas são Series e DataFrame.

Primeiramente, vamos entender o que é uma serie.

No módulo anterior eu já falei sobre as estruturas de dados que o python já possui, como o tipo int, string, boolean, tipo lista e dentre outros.

A Series é uma matriz unidimensional que contém um conjunto de dados e um índice correspondente a cada elemento.

As sereis ou séries em português são também uma estrutura de dados fundamentais da biblioteca Pandas. Nesse caso, o python não possui essa estrutura de dados nativamente, apenas quando a gente implementa a biblioteca pandas.

Basicamente, uma série é uma coleção de valores do mesmo tipo (como números ou strings) que são indexados por uma lista de rótulos. Os rótulos são usados para acessar e manipular os valores na série.

Aqui está um exemplo de como criar uma série simples em Pandas:

Então vamos ver como a gente pode representar uma série. Depois que a gente importou a biblioteca do pandas a gente pode usar a seguinte função para criar uma série:

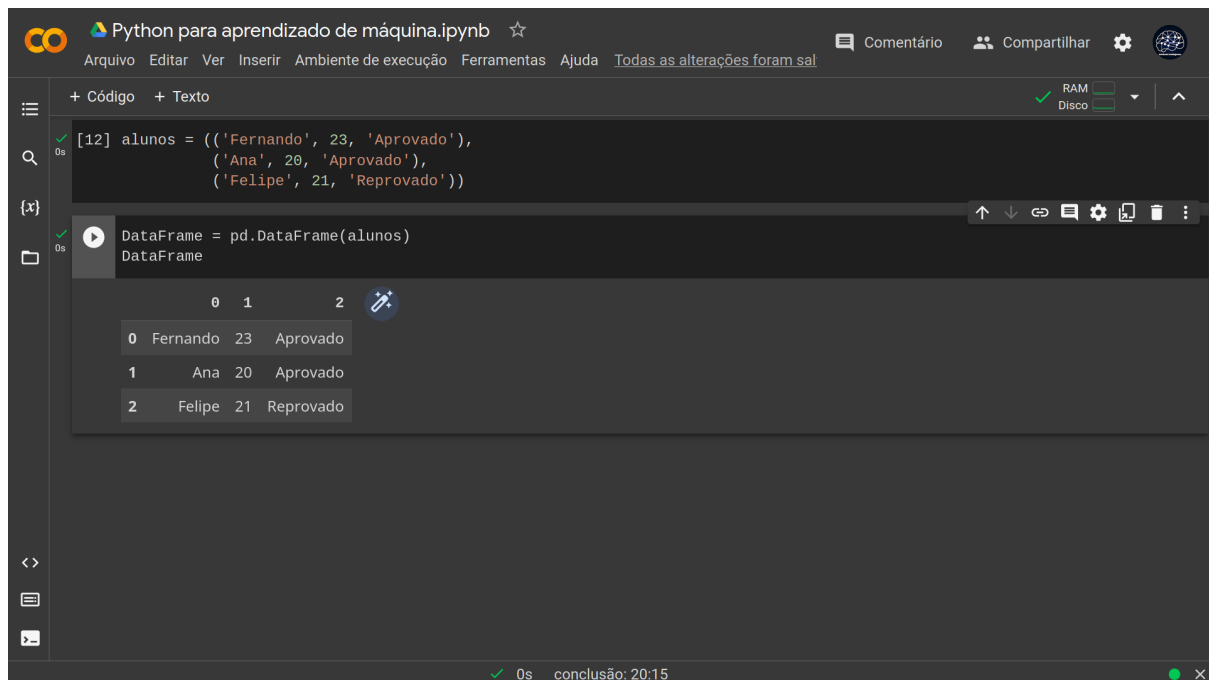
A gente pode usar aquelas três estruturas de dados que a gente já viu pra criar uma série, a gente pode usar uma lista, um dicionário ou até mesmo uma tupla.

Então vamos criar aqui, por exemplo, uma lista.

Depois de criar a lista o que a gente vai fazer é passar essa lista como parâmetro para a função series do pandas.

E a gente usa o pd que a gente usou quando chamou a biblioteca do pandas, o pd é como se fosse um apelido que facilita na hora da gente usar a biblioteca no código.

Mas como o DataFrame é bidimensional a gente pode representar várias linhas ou colunas assim como em uma tabela no Excel.



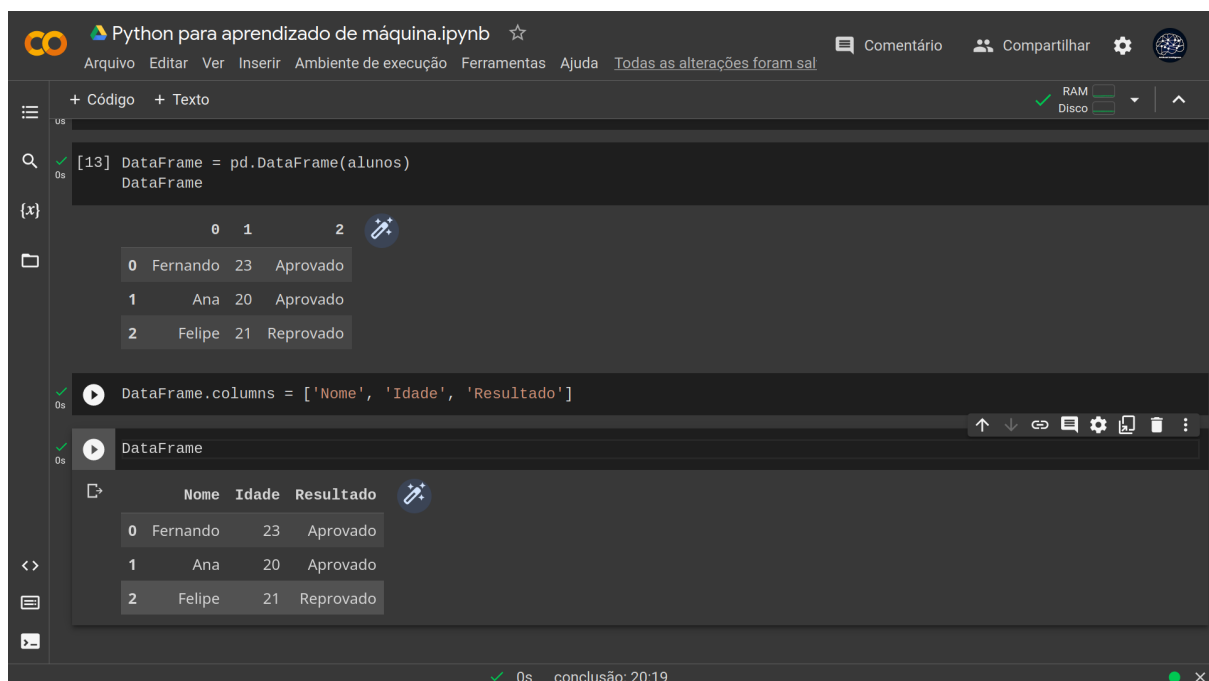
The screenshot shows a Jupyter Notebook interface with the title "Python para aprendizado de máquina.ipynb". The code cell [12] contains the following Python code:

```
[12] alunos = (('Fernando', 23, 'Aprovado'),  
              ('Ana', 20, 'Aprovado'),  
              ('Felipe', 21, 'Reprovado'))  
  
DataFrame = pd.DataFrame(alunos)  
DataFrame
```

The output of the code is a DataFrame with 3 rows and 3 columns. The columns are indexed 0, 1, and 2. The data is as follows:

	0	1	2
0	Fernando	23	Aprovado
1	Ana	20	Aprovado
2	Felipe	21	Reprovado

E existem diversas Funções que a gente pode usar em um DataFrame, como por exemplo, colocar o nome de cada coluna.



The screenshot shows a Jupyter Notebook interface with the title "Python para aprendizado de máquina.ipynb". The code cell [13] contains the following Python code:

```
[13] DataFrame = pd.DataFrame(alunos)  
DataFrame
```

The output of the code is a DataFrame with 3 rows and 3 columns, indexed 0, 1, and 2. The data is as follows:

	0	1	2
0	Fernando	23	Aprovado
1	Ana	20	Aprovado
2	Felipe	21	Reprovado

The next code cell contains the following Python code:

```
DataFrame.columns = ['Nome', 'Idade', 'Resultado']  
  
DataFrame
```

The output of the code is a DataFrame with 3 rows and 3 columns. The columns are named 'Nome', 'Idade', and 'Resultado'. The data is as follows:

	Nome	Idade	Resultado
0	Fernando	23	Aprovado
1	Ana	20	Aprovado
2	Felipe	21	Reprovado

Em resumo, o Pandas é uma ferramenta poderosa e essencial para qualquer pessoa que trabalhe com dados em Python, tornando mais fácil e eficiente a análise e manipulação de dados em diversas áreas de aplicação. Mas por esta aula é isso, e muito obrigado por assistir até aqui.