Python® para Data Science

Para leigos

Edição de Bolso

John Paul Mueller
e Luca Massaron

ALTA BOOKS
EDITORA
Rio de Janeiro, 2020

Sumário

1 Começando

NESTA PARTE...

Entenda como o Python facilita o data science.

Descubra os atributos do Python comumente usados no data science.

Crie uma configuração própria do Python.

Trabalhe com o Google Colab.

NESTE CAPÍTULO

- **Descobrindo maravilhas e explorando o data science**
- **Conectando Python e data science**
- **Começando com o Python**

Capítulo **1**

Combinando Data Science e Python

O data science pode parecer uma daquelas tecnologias que você nunca usaria, mas isso está errado. Sim, data science envolve o uso de técnicas avançadas de matemática, estatística e big data, mas ajuda a tomar melhores decisões, criar sugestões para opções baseadas em escolhas anteriores e fazer com que robôs vejam objetos. Na verdade, as pessoas usam data science de tantas formas diferentes que não se pode olhar para lugar nenhum ou fazer o que quer que seja sem sentir os efeitos do data science em sua vida. Resumindo: é o data science que está por trás das cortinas na experiência das maravilhas da tecnologia. Sem data science, muito do que entendemos como comum e esperado hoje não seria possível. É por isso que cientista de dados é a profissão mais atraente do século XXI.

LEMBRE-SE

Para que o data science seja viável para alguém que não é um gênio da matemática, são necessárias ferramentas. Você pode usar qualquer quantidade de ferramentas para realizar tarefas de data science, mas o Python é especialmente adequado para

facilitar o trabalho com data science. Por um lado, ele fornece um número incrível de bibliotecas relacionadas à matemática que ajudam na realização de tarefas com uma compreensão quase perfeita do que acontece exatamente. Contudo, a função do Python vai além de suportar vários estilos de código (paradigmas de programação) e facilitar seu trabalho. Portanto, sim, você pode usar outras linguagens para escrever aplicações de data science, mas o Python reduz a carga de trabalho, então é uma escolha natural para quem não quer trabalhar demais, mas quer trabalhar bem.

Este capítulo o apresenta ao Python. Embora o objetivo deste livro não seja fornecer um tutorial completo sobre o Python, explorar algumas questões básicas sobre ele permitirá que você pegue o ritmo. (Se precisar de um bom tutorial introdutório, adquira o livro *Começando a Programar em Python Para Leigos* [Alta Books]. Ele oferece indicações de tutoriais e outros recursos necessário para preencher as lacunas que você possa ter em seu aprendizado do Python.)

ESCOLHENDO UMA LINGUAGEM DE DATA SCIENCE

Há muitas linguagens de programação no mundo, e a maioria foi criada para realizar tarefas específicas ou até para facilitar o trabalho de determinadas profissões. Escolher a ferramenta correta facilita sua vida. É como usar um martelo para apertar um parafuso em vez de uma chave de fenda. Sim, o martelo funciona, mas, definitivamente, a chave de fenda é muito mais fácil de usar e faz um trabalho melhor. Os cientistas de dados usam apenas algumas linguagens, pois elas facilitam o trabalho com os dados. Com isso em mente, aqui estão as principais linguagens para o trabalho com data science, em ordem de preferência:

- **Python (uso geral):** Muitos cientistas de dados preferem usar o Python porque ele fornece muitas bibliotecas, como NumPy, SciPy, MatPlotLib, pandas e Scikit-learn, para facilitar significativamente as tarefas com data science. O Python também é uma linguagem precisa, que facilita o uso de multiprocessamento em grandes conjuntos de dados — re-

duzindo o tempo exigido para analisá-los. A comunidade de data science também evoluiu com IDEs especializados, como o Anaconda, que implementam o conceito Jupyter Notebook, que facilita muito o trabalho com cálculos de data science. Além de tudo isso a favor do Python, ele também é uma linguagem excelente para se criar glue code com linguagens como C/C++ e Fortran. A documentação do Python mostra como criar as extensões necessárias; e a maioria de seus usuários depende da linguagem para ver padrões, como dar permissão para que um robô veja um grupo de pixels como um objeto. Ele também é aplicável a todos os tipos de tarefas científicas.

- **R (uso especial estatístico):** Em muitos aspectos, o Python e o R compartilham os mesmos tipos de funcionalidade, mas as implementam de modo diferente. Dependendo de qual fonte é visualizada, o Python e o R têm mais ou menos o mesmo número de proponentes, e algumas pessoas usam ambas de modo intercambiável (ou, às vezes, em dupla). Diferentemente do Python, o R fornece um ambiente próprio, então você não precisa de produtos de terceiros, como o Anaconda. No entanto, o R não se mistura com outras linguagens com a mesma facilidade que o Python.
- **SQL (gestão de banco de dados):** O mais importante a ser lembrado sobre a Structured Query Language (SQL) é que ela foca os dados, e não as tarefas. Nada opera sem uma boa gestão de dados — eles são o negócio. Grandes organizações usam algum tipo de banco de dados relacional, normalmente acessível com SQL, para armazenar os dados. A maioria dos produtos de Sistemas de Gerenciamento de Banco de Dados (SGBD) depende de SQL como linguagem principal, e os SGBD, em geral, têm um grande número de atributos de análises de dados e de data science incorporados. Como você acessa os dados nativamente, muitas vezes há um ganho de velocidade significativo ao realizar tarefas de data science dessa forma. Os Administradores de Bancos de Dados (DBAs) usam SQL para gerenciar ou manipular os dados, em vez de necessariamente realizar análises detalhadas deles. No entanto, o cientista de dados também pode usar SQL para várias tarefas de data science e disponibilizar os scripts resultantes para as necessidades dos DBAs.

(continua)

(continuação)

- **Java (uso geral):** Alguns cientistas de dados realizam outros tipos de programação que exigem uma linguagem popular, amplamente adaptada e de uso mais geral. Além de fornecer acesso a um grande número de bibliotecas (cuja maioria não é tão útil para data science, mas funciona para outras necessidades), o Java suporta a orientação a objetos melhor do que qualquer outra linguagem desta lista. Além disso, é muito tipado e tende a ser executado com mais rapidez. Consequentemente, algumas pessoas o preferem para o código finalizado. O Java não é uma boa opção para experimentação nem para consultas *ad hoc*.
- **Scala (uso geral):** Como o Scala usa a Máquina Virtual Java (JVM), tem algumas vantagens e desvantagens em relação ao Java. Contudo, como o Python, o Scala é compatível com o paradigma da programação funcional, que usa cálculos lambda como base (veja detalhes em *Programação Funcional Para Leigos*). Além disso, o Apache Spark é escrito em Scala, o que significa que você tem um bom suporte para grupo ao usar essa linguagem — pense no suporte de um conjunto de dados gigantesco. Algumas armadilhas do uso do Scala são a dificuldade de configurá-lo corretamente, a dificuldade de aprendizado e a falta de um conjunto abrangente de bibliotecas específicas de data science.

Definindo o Trabalho Mais Atraente do Século XXI

A certa altura, o mundo via todos os que trabalhavam com estatística como um tipo de contador, ou talvez um cientista louco. Muitas pessoas consideram a estatística e a análise de dados algo chato. Entretanto, o data science é uma dessas profissões em que quanto mais você aprende, mais quer aprender. Responder a uma pergunta muitas vezes gera mais perguntas ainda mais interessantes do que a que acabou de ser respondida. Contudo, o que torna o data science tão atraente é que ele é visto em todas as situações e usado de infinitas maneiras.

Considerando a emergência do data science

Data science é uma expressão relativamente nova. William S. Cleveland a cunhou em 2001 como parte de um artigo sobre o data science como plano de ação para expandir as áreas técnicas do campo da estatística. Foi apenas um ano mais tarde que o Conselho Internacional de Ciência de fato o reconheceu e criou um comitê para ele. A Universidade de Columbia entrou em cena em 2003 ao iniciar a publicação do *Journal of Data Science*.

LEMBRE-SE

No entanto, a base matemática por trás do data science tem muitos séculos, pois ele é praticamente um método para ver e analisar estatística e probabilidade. O primeiro uso relevante do termo "estatística" data de 1749, mas ela com certeza é muito mais antiga. As pessoas usaram estatística para reconhecer padrões por milhares de anos. Por exemplo, o historiador Tucídides (em seu *História da Guerra do Peloponeso*) descreve como os atenienses calcularam a altura do muro de Plateias no século V a.C. por meio da contagem de tijolos em uma seção sem reboco do muro. Como a contagem precisava ser exata, os atenienses tiraram a média da contagem feita por vários soldados.

O processo de quantificar e entender a estatística é, de certa forma, novo, mas a ciência em si é antiga. Uma tentativa anterior de registrar a importância da estatística aparece no século IX, quando Alcindi escreveu o *Manuscrito para Decifrar Mensagens Criptográficas*. Nele, descreve como usar uma combinação de análises estatísticas e de frequência para decifrar mensagens criptografadas. Mesmo no começo, a estatística já era aplicável à ciência para tarefas aparentemente impossíveis de completar. O data science continua esse processo, e, para algumas pessoas, realmente parece mágica.

Esboçando as competências centrais

Como é válido para qualquer um que tenha as atribuições mais complexas hoje em dia, o cientista de dados necessita de uma ampla gama de habilidades para realizar as tarefas necessárias. Na verdade, são tantas

as habilidades diferentes exigidas que os cientistas de dados geralmente trabalham em equipe. Alguém bom em reunir dados pode se juntar a um analista e a alguém com o dom de apresentar informações. Seria difícil encontrar uma única pessoa com todas as habilidades necessárias. Com isso em mente, a lista a seguir descreve áreas em que um cientista de dados se destaca (sendo que, quanto mais competências tiver, melhor):

- **Captura de dados:** Não importa o tipo de habilidade matemática que você tenha se, primeiro, não conseguir dados para analisar. A coleta de dados começa com a administração da fonte de dados por meio de habilidades de gestão de banco de dados. Contudo, os dados brutos não são particularmente úteis em muitas situações — você também deve entender o domínio dos dados para que possa observá-los e formular as perguntas que devem ser feitas. Por fim, deve ter habilidades de modelagem de dados para compreender como os dados se conectam e se eles são estruturados.
- **Análise:** Depois de obter os dados com os quais trabalhará e entender suas complexidades, pode começar a analisá-los. Isso é feito utilizando-se habilidades básicas de ferramentas estatísticas, como aquelas que quase todo mundo aprende na escola. Entretanto, o uso de truques matemáticos e algoritmos especializados torna os padrões nos dados mais óbvios ou os ajuda a chegar a conclusões que não seriam possíveis apenas por meio da revisão dos dados.
- **Apresentação:** A maioria das pessoas não entende muito bem os números. Elas não conseguem ver os padrões que os cientistas de dados veem. É importante apresentar esses padrões na forma gráfica para ajudar os outros a visualizar o significado dos números e como aplicá-los de modo significativo. Mais importante ainda, a apresentação deve contar uma história específica, para que o impacto dos dados não seja perdido.

Conectando data science, big data e IA

Curiosamente, o ato de mover dados para que alguém realize análises é uma especialidade chamada de Extração, Transformação e Carregamento

(da sigla em inglês, ETL). O especialista em ETL usa linguagens de programação como o Python para extrair os dados de várias fontes. As corporações tendem a não manter os dados em um local de fácil acesso, então encontrar os dados requeridos para realizar análises leva tempo. Depois que o especialista em ETL encontra os dados, uma linguagem de programação ou outra ferramenta os transforma em um formato comum para propósitos de análise. O processo de carregamento é diversificado, mas este livro conta com o Python para realizar a tarefa. Em uma grande operação real, você pode usar ferramentas como Informatica, MS SSIS ou Teradata para realizá-la.

Criando o Pipeline do Data Science

O data science é metade arte e metade engenharia. Reconhecer padrões em dados, considerar quais perguntas fazer e determinar quais algoritmos funcionam melhor são partes do lado artístico. No entanto, para que esse lado se concretize, a parte da engenharia se apoia em um processo especial para alcançar objetivos específicos. Esse processo é o pipeline de data science, que requer que o cientista de dados siga determinados passos na preparação, na análise e na apresentação dos dados.

Preparando os dados

Os dados acessados de várias fontes não vêm em um pacote bonito, pronto para a análise — é bem pelo contrário. Os dados brutos não só podem variar consideravelmente no formato, como você também pode ter que os transformar para que todas as fontes de dados sejam coesas e favoráveis à análise. A transformação pode exigir mudança nos tipos de dados, na ordem em que aparecem e até na criação de entradas de dados com base nas informações fornecidas pelas entradas existentes.

Realizando análise de dados exploratória

A matemática por trás da análise de dados depende dos princípios de engenharia em que os resultados são comprováveis e coerentes. Contudo,

o data science fornece acesso a uma grande variedade de métodos estatísticos e algoritmos que o ajudam a descobrir padrões nos dados. Uma única abordagem normalmente não funciona. É preciso usar um processo iterativo para retrabalhar os dados a partir de diversas perspectivas. O uso de tentativa e erro faz parte da arte do data science.

Aprendendo com os dados

Ao iterar com vários métodos de análises estatísticas e aplicar algoritmos para detectar padrões, você começa a aprender com os dados. Eles podem não contar a história que você achava que contariam, ou podem ter muitas histórias para contar. A descoberta faz parte da vida do cientista de dados. Na verdade, é a parte divertida, pois não é possível saber com antecedência o que exatamente os dados revelarão.

LEMBRE-SE

É claro que a natureza imprecisa dos dados e a descoberta de padrões que parecem ser aleatórios significa que devemos manter a mente aberta. Se tiver ideias preconcebidas do que os dados contêm, não encontrará as informações que de fato estão neles. Assim, perde a fase de descoberta do processo, que se traduz em oportunidades perdidas para você e para as pessoas que dependem de você.

Visualizando

Visualizar significa ver os padrões nos dados e ser capaz de reagir a eles. Também significa ser capaz de ver quando os dados não fazem parte do padrão. Pense em si mesmo como um escultor de dados — removendo os que estão fora do padrão (os outliers) para que os outros vejam a obra-prima das informações escondida. Sim, você consegue vê-la, mas até que os outros também consigam, ela permanece apenas em seu campo de visão.

Obtendo insights e produtos de dados

Pode parecer que o cientista de dados simplesmente procura métodos únicos para visualizar os dados. Contudo, o processo não acaba até que se tenha uma compreensão clara do significado desses dados. Os

insights obtidos a partir da manipulação e da análise de dados o ajudam a realizar tarefas reais. Por exemplo, os resultados de uma análise podem ser utilizados para tomar decisões de negócios.

Em certos casos, o resultado de uma análise cria uma resposta automatizada. Por exemplo, quando um robô vê uma série de pixels obtidos de uma câmera, os pixels que formam um objeto têm um significado especial, e a programação do robô dita algum tipo de interação com esse objeto. Mas, até que o cientista de dados crie uma aplicação que carregue, analise e visualize os pixels da câmera, o robô não vê nada.

Entendendo o Papel do Python no Data Science

Dadas as fontes de dados certas, as exigências de análise e as necessidades de apresentação, o Python se aplica a todas as partes do pipeline de data science. Na verdade, é exatamente isso o que você fará neste livro. Cada exemplo usa o Python para lhe explicar outra parte da equação do data science. De todas as linguagens disponíveis para realizar tarefas de data science, o Python é a mais flexível e apta, pois suporta muitas bibliotecas de terceiros dedicadas à tarefa.

Trabalhando com uma linguagem multiúso, simples e eficiente

Há muitas maneiras de realizar tarefas de data science. Este livro trata apenas de um dos diversos métodos à disposição. Entretanto, o Python é uma das poucas soluções que, de forma isolada, resolve problemas complexos de data science. Em vez de ter de usar várias ferramentas para realizar uma tarefa, pode-se usar apenas uma única linguagem, o Python, para fazer o trabalho. A diferença desta é o grande número de bibliotecas científicas e matemáticas criadas por terceiros. A inserção dessas bibliotecas amplia o Python e lhe possibilita realizar facilmente tarefas que outras linguagens teriam problemas para dar conta.

DICA

As bibliotecas do Python são seu principal atrativo; mas ele oferece mais do que um código reutilizável. O mais importante a ser considerado é que essa linguagem é compatível com quatro estilos de programação:

- **Funcional:** Trata cada declaração como uma equação matemática e evita qualquer forma de estado ou dados mutáveis. A principal vantagem dessa abordagem é não ter nenhum efeito colateral. Além disso, esse estilo de programação é mais adequado para o processamento paralelo, pois não há estado a ser considerado. Muitos desenvolvedores preferem este tipo de programação para recursão e para cálculo lambda.
- **Imperativa:** Realiza cálculos como uma mudança direta no estado do programa. Esse estilo é particularmente útil ao manipular estruturas de dados e produz um código elegante, mas ainda simples.
- **Orientada a objetos:** Depende de campos de dados tratados como objetos e manipulados apenas por métodos prescritos. O Python não suporta por completo esta forma de programação, pois não consegue implementar atributos como a ocultação de dados. No entanto, esse é um estilo de programação útil para aplicações complexas, pois suporta o encapsulamento e o polimorfismo, e também favorece a reutilização do código.
- **Procedural:** Trata de tarefas como iterações passo a passo, em que as tarefas comuns são colocadas em funções que são chamadas quando necessário. Esse estilo de programação favorece a iteração, o sequenciamento, a seleção e a modularização.