Python® para Data Science

Para leigos

1 Começando com Data Science e Python

NESTA PARTE...

Entenda como o Python facilita o data science.

Defina os atributos do Python comumente usados no data science.

Crie uma configuração própria do Python.

Trabalhe com o Google Colab em dispositivos alternativos.

NESTE CAPÍTULO

- » Descobrindo as maravilhas do data science
- » Explorando o funcionamento do data science
- » Criando a conexão entre 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 decisões melhores, a criar sugestões para opções baseadas em escolhas anteriores e a 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 aceitamos 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, o Python 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 — reduzindo 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 (o Capítulo 3 ensina a usar o Jupyter Notebook, então não se preocupe com isso agora). 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. A maioria dos usuários do Python 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. Os negócios não podem operar 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 geralmente 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) normalmente 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.

- **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 fortemente 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. As próximas seções fornecem mais detalhes sobre o porquê de o data science ser um campo de estudos tão incrível.

Considerando a emergência do data science

Data science é um termo relativamente novo. William S. Cleveland o cunhou em 2001 como parte de um artigo intitulado "Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics" [Data Science: Um Plano de Ação para Expandir as Áreas Técnicas do Campo da Estatística, em tradução livre]. Foi apenas um ano mais tarde que o Conselho Internacional de Ciência realmente reconheceu o data science 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 certamente 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 relativamente 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, Alcindi 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 de um cientista de dados

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, você 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 praticamente todo o 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 graficamente esses padrões 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 facilmente acessível, 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.


LEMBRE-SE

O data science não é necessariamente um meio para um fim. Ele pode, na verdade, ser apenas um passo no caminho. À medida que um cientista de dados trabalha com vários conjuntos de dados e descobre fatos interessantes, esses fatos podem agir como ideias para outros tipos de análises e aplicações de IA. Por exemplo, considere que seus hábitos de compra sugiram de quais livros você pode gostar ou onde gostaria de passar as férias. Hábitos, como os de compra, também ajudam a entender outras atividades, às vezes menos inofensivas. Os livros *Aprendizado de Máquina Para Leigos* e *IA Para Leigos* (Alta Books), ambos de John Mueller e Luca Massaron, explicam esses e outros usos do data science. No momento, considere que o que aprender com este livro poderá ter um efeito definitivo em um plano de carreira que pode seguir diversas outras direções.

Entendendo o papel da programação

Um cientista de dados precisa conhecer várias linguagens de programação para alcançar objetivos específicos. Por exemplo, você pode precisar de conhecimentos de SQL para extrair dados de bancos de dados relacionais. O Python pode ajudá-lo a realizar as tarefas de carregamento, transformação e análise de dados. Contudo, você pode escolher um produto como o MATLAB (que tem a própria linguagem de programação) ou o PowerPoint (que conta com VBA) para apresentar as informações para outras pessoas. (Adquira o livro *MATLAB Para Leigos* (Alta Books) se tiver interesse em comparar o uso do MATLAB com o do Python.) Os conjuntos de dados imensos com que os cientistas de dados contam muitas vezes requerem vários níveis de processamento redundante para transformá-los em dados processados úteis. Realizar essas tarefas manualmente consome muito tempo e propicia erros, então a programação é o melhor método para alcançar o objetivo de uma fonte de dados útil e coerente.

Dada a quantidade de produtos que a maioria dos cientistas de dados usa, não é possível utilizar apenas uma linguagem de programação. Sim, o Python pode carregar, transformar e analisar dados, e até apresentá-los ao usuário final, mas funciona somente quando a linguagem fornece a funcionalidade requerida. Você pode precisar escolher outras linguagens para a tarefa, e essa escolha depende de diversos critérios. Considere o seguinte:

- Como você pretende usar o data science em seu código (há várias tarefas a serem consideradas, como análise de dados, classificação e regressão).
- Sua familiaridade com a linguagem.

- A necessidade de interação com outras linguagens.
- A disponibilidade de ferramentas para a melhoria do ambiente de desenvolvimento.
- A disponibilidade de APIs e bibliotecas para facilitar a realização de tarefas.

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 do data science. 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, análise e apresentação dos dados. As próximas seções o ajudam a entender melhor o pipeline do data science, para que você compreenda como o livro o emprega durante a apresentação dos exemplos.

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 transformá-los 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 originalmente, ou podem ter

muitas histórias para contar. A descoberta faz parte da vida do cientista de dados. Na verdade, é a parte divertida do data science, 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 aparentemente aleatórios significa que devemos manter a mente aberta. Se você tiver ideias preconcebidas do que os dados contêm, não encontrará as informações que realmente estão neles. Você 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 você 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 capaz, pois suporta muitas bibliotecas de terceiros dedicadas à tarefa. As próximas seções detalham por que o Python é uma escolha tão boa para tantas (se não para a maioria) necessidades de data science.

Considerando a transformação do perfil dos cientistas de dados

Algumas pessoas veem o cientista de dados como um nerd inacessível que usa a matemática para fazer milagres. É ele quem está por trás da cortina em uma experiência parecida com a do *Mágico de Oz*. No entanto, essa perspectiva está mudando. Em muitos aspectos, o mundo agora vê o cientista de dados como um assistente de desenvolvedor ou como um novo tipo de desenvolvedor. A ascensão de aplicações que aprendem, de todos os tipos, é a essência dessa mudança. Para que uma aplicação aprenda, ela precisa ser capaz de manipular grandes bancos de dados e descobrir padrões deles. Além disso, deve ser capaz de criar dados com base nos antigos — fazendo uma espécie de previsão embasada. Os novos tipos de aplicações afetam as pessoas de maneiras que pareceriam ficção científica poucos anos atrás, e é claro que as mais notáveis dessas aplicações definem os comportamentos de robôs que no futuro interagirão muito mais próximos das pessoas do que fazem atualmente.

A partir de uma perspectiva de negócios, a necessidade de combinar o data science e o desenvolvimento de aplicações é óbvia: os negócios devem realizar vários tipos de análises nos bancos gigantescos de dados que coletam — para interpretar as informações e usá-las para prever o futuro. Mas, na verdade, o impacto maior da junção desses dois ramos da ciência — o data science e o desenvolvimento de aplicações — será sentido em termos da criação de tipos totalmente novos de aplicações, alguns que nem conseguimos imaginar com clareza atualmente. Por exemplo, novas aplicações que auxiliem alunos a aprenderem com maior precisão ao analisar suas tendências de aprendizado e criar métodos instrucionais que funcionem para esse aluno específico. Essa combinação de ciências também resolveria uma grande variedade de problemas médicos que parecem impossíveis de solucionar hoje — não apenas mantendo as doenças afastadas, mas também resolvendo problemas; por exemplo, como criar dispositivos protéticos verdadeiramente utilizáveis que se apresentam e atuam como a coisa real.

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 de Python 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 o Python 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 desta abordagem é que não tem nenhum efeito colateral. Além disso, este 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 ao estado do programa. Este 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 totalmente esta forma de programação, pois não consegue implementar atributos como a ocultação de dados. No entanto, este 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. Este estilo de programação favorece a iteração, o sequenciamento, a seleção e a modularização.

Aprendendo Rapidamente a Usar o Python

É hora de tentar usar o Python para ver o pipeline de data science em ação. As próximas seções fornecem um panorama breve do processo que será explorado detalhadamente no restante do livro. Você não realizará as tarefas das próximas seções. Na verdade, só instalará o Python no Capítulo 3, então, por enquanto, apenas siga o texto. Este livro usa uma versão específica do Python e um IDE chamado de Jupyter Notebook, portanto, aguarde até o Capítulo 3 para

instalá-los (ou, se preferir, pule e instale-os já). Não se preocupe em entender todos os aspectos do processo agora. O objetivo destas seções é explicar o fluxo do uso do Python para realizar tarefas de data science. Muitos dos detalhes podem parecer difíceis de entender agora, mas o restante do livro o ajudará.


LEMBRE-SE

Os exemplos deste livro dependem de uma aplicação baseada na web chamada de Jupyter Notebook. As capturas de telas que verá aqui e nos próximos capítulos refletem a aparência do Jupyter Notebook no Firefox em um sistema com Windows 7. Sua visualização conterá os mesmos dados, mas a interface pode ser diferente, dependendo da plataforma (como o uso de um notebook, em vez de um desktop), do sistema operacional e do navegador usados. Não se preocupe se vir pequenas diferenças entre sua exibição e as capturas de tela no livro.


DICA

Você não precisa digitar o código-fonte deste capítulo. Na verdade, é muito mais fácil fazer o download dos arquivos (veja detalhes sobre o download do código-fonte na Introdução). O código-fonte deste capítulo está no arquivo `P4DS4D2_01_Quick_Overview.ipynb`, uma referência ao título original *Python for Data Science For Dummies*, 2a edição.

Carregando dados

Antes de começar, carregue alguns dados. O livro mostra vários tipos de métodos para realizar essa tarefa. Neste caso, a Figura 1-1 mostra como carregar um conjunto de dados chamado de Boston, que contém os preços de moradias e outros fatos sobre as casas na área de Boston. O código posiciona todo o conjunto de dados na variável `boston` e depois posiciona partes desses dados nas variáveis `X` e `y`. Pense nas variáveis como se fossem caixas. Elas são importantes, pois possibilitam o trabalho com os dados.

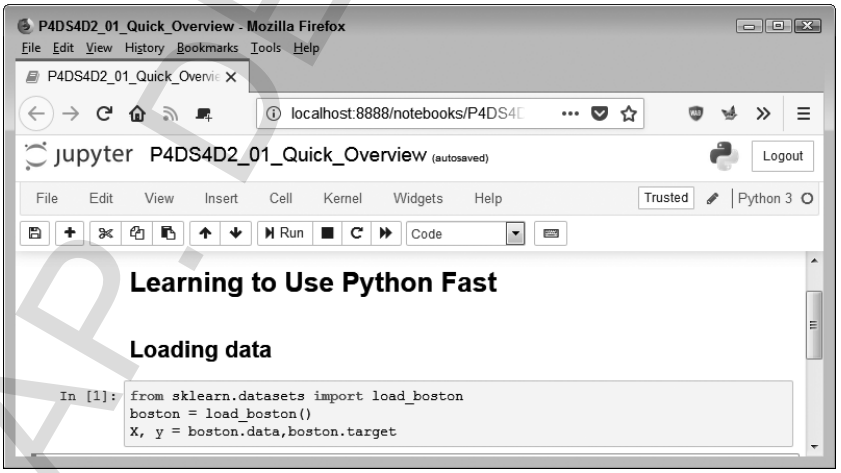

FIGURA 1-1: Carregando dados em variáveis para poder manipulá-los.

Treinando um modelo

Agora que temos dados para trabalhar, podemos fazer algo com eles. Há diversos tipos de algoritmos incorporados no Python. A Figura 1-2 mostra um modelo de regressão linear. Repito: não se preocupe com o funcionamento exato, os capítulos posteriores abordarão os detalhes da regressão linear. O importante a ser notado na Figura 1-2 é que o Python possibilita a realização da regressão linear usando apenas duas declarações e posiciona o resultado em uma variável chamada de `hypothesis`.

Visualizando um resultado

Não vale a pena realizar nenhum tipo de análise a não ser que você obtenha benefícios como resultado. Este livro mostra várias maneiras de visualizar a saída, mas a Figura 1-3 começa com algo simples. Neste caso, você vê o coeficiente resultante da análise de regressão linear.

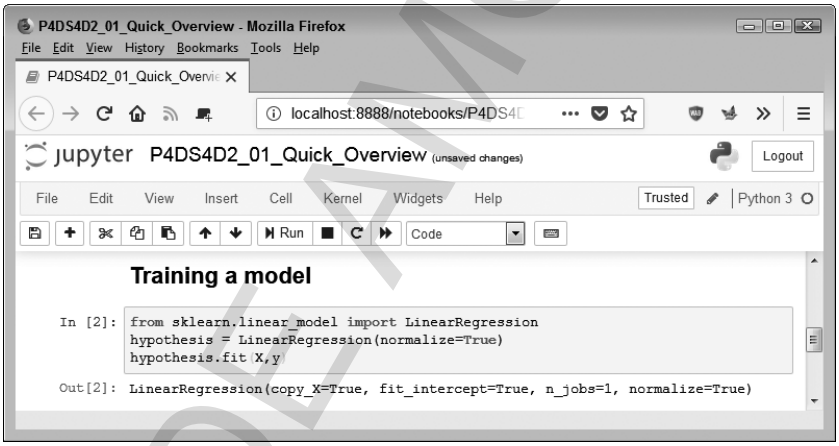

FIGURA 1-2: Usando a variável para treinar um modelo de regressão linear.

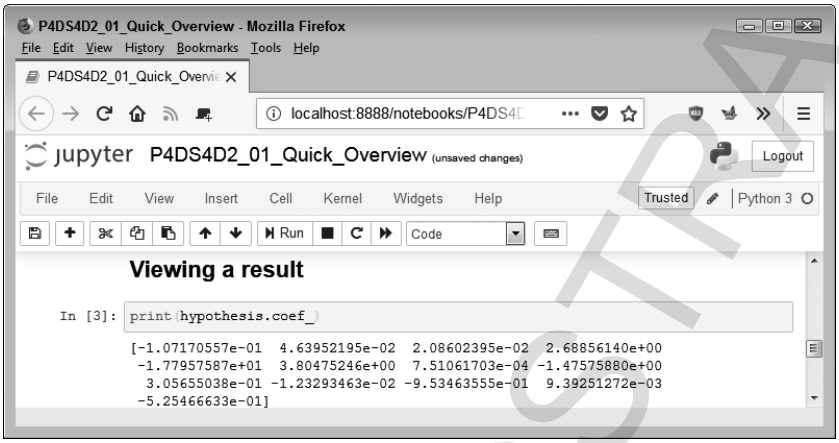

FIGURA 1-3: Exibindo um resultado como uma resposta ao modelo.

DICA

Uma das razões para o uso do Jupyter Notebook neste livro é que o produto ajuda a criar saídas bem formatadas como parte da criação da aplicação. Observe novamente a Figura 1-3 e verá um relatório que poderia ser simplesmente impresso e oferecido a um colega. A saída não é adequada para muitas pessoas, mas aqueles com experiência com Python e data science a acharão útil e muito informativa.