



**Data Science  
Academy**

[www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)

## Introdução à Lógica de Programação

A Aula de hoje é especial. É o momento “Eureka”. O momento onde tudo começa a fazer sentido.

Obs: Se nunca ouviu a expressão “Eureka”, aqui está o significado: [https://pt.wikipedia.org/wiki/Eureka\\_\(exclama%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Eureka_(exclama%C3%A7%C3%A3o))

Até aqui, em todos os capítulos deste curso de Introdução à Lógica de Programação, estudamos alguns importantes algoritmos com exemplos de implementação em Python. Mas o que todos esses algoritmos têm em comum? Em todos os casos precisamos programar cada detalhe do algoritmo, cada passo, cada iteração, cada resultado. Essa é a forma como criamos softwares, como seu sistema operacional, um sistema ERP, ou mesmo um navegador. Precisamos programar o software para que ele saiba o que fazer até mesmo quando o usuário passar o mouse por um botão.

O problema é que programar todos os detalhes dá muito, muito trabalho. Um software médio pode ter facilmente milhares ou milhões de linhas de código e requerer diversos profissionais. Estima-se que o software de um avião de última geração tenha 2 milhões de linhas de código.

Imagine que você queira criar um software para prever se uma pessoa vai ou não comprar um livro. Parece simples, não? Na verdade, não. Teríamos que programar cada detalhe do software, item a item, instrução a instrução. Seria tão trabalhoso que tornaria esse tipo de aplicação inviável economicamente. E se o software tivesse que prever a compra de diversos itens e não apenas de livros? Dá para imaginar quão complexo seria.

E se ao invés disso, um “pedaço de software” fosse capaz de criar um programa para fazer a previsão acima mencionada? Nós faríamos a coleta de dados históricos, aplicaríamos algum tratamento aos dados e então entregaríamos os dados a esse “pedaço de software”, que passaria pelos dados diversas vezes aprendendo os detalhes e relacionamentos nos dados e ao final entregaria um pequeno modelo que é a representação de uma fórmula matemática que explica a relação entre os dados. Com esse modelo pronto e “treinado”, poderíamos entregar novos dados e ele seria capaz de fazer as previsões. E tudo isso sem programar explicitamente os detalhes sobre a previsão de livros. Isso não seria incrível?

Sim.

E isso é o que chamamos de Machine Learning, uma das sub-áreas da Inteligência Artificial.

Um algoritmo de Machine Learning implementado em Python por exemplo, é esse “pedaço de software” que mencionei. Um conjunto de instruções sobre como aprender relacionamentos nos dados e criar um modelo que represente a relação nos dados. Não temos que programar cada detalhe dos dados, pois eles são aprendidos automaticamente durante o treinamento. Para diferentes tipos de problemas, podemos usar diferentes “pedaços de software”.

Isso é o que chamamos de Aprendizado de Máquina, Machine Learning. Não precisamos programar cada detalhe dos dados, pois o algoritmo aprende os detalhes de forma automática.

ÓTIMO, ENTÃO NÃO PRECISO PROGRAMAR. UFA!

Espere. A Eureka é agora!

Esse é o ponto onde as pessoas em geral mais confundem as coisas. Temos na verdade 3 grandes áreas. Observe:

#### 1- Construção do Algoritmo de Machine Learning e Implementação com Uma Linguagem de Programação

Isso requer programação, mas basicamente o que fazemos é pegar o algoritmo criado por alguém (uma das muitas mentes brilhantes que desenvolveram os algoritmos) e então seguir uma sequência com as instruções na linguagem de programação. Muito similar ao que fizemos até aqui neste curso: pegamos o pseudo-código e implementamos em Python, por exemplo. Os alunos do curso de Matemática Para Machine Learning fazem isso em 4 Labs ao longo do curso.

Mas nem isso é necessário. Você pode usar um algoritmo pronto disponível no Scikit-Learn em Python ou no pacote Caret em R, por exemplo. O desenvolvimento a partir do zero só faz sentido se você for pesquisador ou precisar de algum tipo de customização.

Os algoritmos são frutos de trabalhos de Doutorado ou de pesquisas e são descobertas matemáticas e/ou estatísticas que demonstram uma sequência de passos, que uma vez implementados via programação, nos permitem encontrar relacionamentos nos dados e fazer previsões. Esses algoritmos são construídos através de Lógica de Programação.

Temos algoritmos criados na década de 50, como o Perceptron, ou em 2018, como as Capsule Networks.

#### 2- Pré-Processamento de Dados e Treinamento do Algoritmo

Aqui, usamos programação para arranjar e organizar todo o processo. Essa é a parte mais simples de todas, embora seja trabalhosa. Usamos um loop se for necessário repetir uma ação ou usamos um condicional para executar diferentes ações, por exemplo. A programação aqui é para a sequência de atividades: carregar os dados, transformar variáveis, realizar cálculos de conversão, normalizar dados, treinar o algoritmo, avaliar o modelo, etc...Aqui é onde o Cientista de Dados dedica boa parte do seu tempo.

Quanto mais souber programação, mais fácil será o trabalho, mas as etapas aqui são muito mais simples do que na área 1 descrita acima.



A Lógica de Programação aqui não tem um impacto tão grande, pois não estamos desenvolvendo ou programando, mas sim criando scripts que elaboram uma sequência de tarefas.

### 3- Desenvolvimento de Sistemas

A programação é empregada para construir um sistema web ou uma aplicação de cadastro de usuários, por exemplo. Engenheiros de Software e Desenvolvedores atuam nessas tarefas e podem usar ou não Machine Learning ou podem construir aplicações que usem modelos de Machine Learning já treinados. Esses profissionais devem dominar completamente a Lógica de Programação, pois ela é parte essencial do trabalho.

Ou seja, ao contrário do que muitos pensam, o que precisamos fazer com relação à programação quando trabalhamos com Ciência de Dados, na maioria das vezes é referente a atividades de criação de sequências lógicas de atividades. Usamos linguagem de programação não para desenvolver sistemas, mas para estabelecer sequências de ações. O movimento de Cientista de Dados Full Stack prevê que o profissional possa atuar nas 3 áreas descritas acima. E na prática, pode. A única limitação é a que você impõe a si mesmo. Mas hoje, o trabalho do Cientista de Dados está na área 2 descrita acima. Diga-se de passagem, muito, muito trabalho e que requer bastante conhecimento.

Eureka!

Você não precisa ser especialista em programação de computadores para trabalhar como Cientista de Dados. Criar um loop ou um condicional é aplicar lógica e algumas poucas instruções usando uma linguagem de programação. Não há desenvolvimento ou programação efetivamente, mas uma sequência de tarefas e ações.

Você não faz ideia de como é difícil fazer as pessoas compreenderem isso. O pânico causado pela programação não procede, pois o Cientista de Dados não vai programar nada. Não precisa. Vai “apenas” escrever um script descrevendo passos para realizar seu processo de análise ou tratamento dos dados.

É claro, que quanto mais souber de programação, mas fácil será o trabalho e mais problemas poderão ser resolvidos. Aprender a programar é acima de tudo uma atividade cognitiva que desenvolve nossas habilidades como resolvidores de problemas. E isso sim é o mais valioso!

A Lógica de Programação ajuda a compreender o que estamos fazendo e porque estamos fazendo.

A maior dificuldade em programação não está em saber como criar o loop while (pois tem um exemplo pronto na documentação da linguagem), mas sim o raciocínio lógico de porque precisamos disso! É aí onde concentramos os esforços em nossos cursos!

Nas duas últimas aulas aqui do curso de Introdução à Lógica de Programação trarei 2 algoritmos de Machine Learning e vamos desenvolvê-los na unha. Mantenha as suas afiadas.

#aula13