# Analytics e Inteligência Artificial

Aula 17

## Aprendizagem Não-Supervisionada

# FIA Business School

NOSSOS DIFERENCIAIS | QUEM SOMOS

**BUSINESS SCHOOL**
Graduação, pós-graduação, MBA, Pós- MBA, Mestrado Profissional, Curso In *Company* e EAD

**CONSULTING**
Consultoria personalizada que oferece soluções baseadas em seu problema de negócio

**RESEARCH**
Atualização dos conhecimentos e do material didático oferecidos nas atividades de ensino

FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO

Líder em Educação Executiva, referência de ensino nos cursos de graduação, pós-graduação e MBA, tendo excelência nos programas de educação. Uma das principais escolas de negócio do mundo, possuindo convênios internacionais com Universidades nos EUA, Europa e Ásia. +8.000 projetos de consultorias em organizações públicas e privadas.

- Único curso de graduação em administração a receber as notas máximas
- A primeira escola brasileira a ser finalista da maior competição de MBA do mundo
- Única *Business School* brasileira a figurar no *ranking* LATAM
- Signatária do Pacto Global da ONU
- Membro fundador da ANAMBA - Associação Nacional MBAs
- Credenciada pela AMBA - *Association of MBAs*
- Credenciada ao *Executive MBA Council*
- Filiada a AACSB - *Association to Advance Collegiate Schools of Business*
- Filiada a EFMD - *European Foundation for Management Development*
- Referência em cursos de MBA nas principais mídias de circulação

# LABDATA FIA

NOSSOS DIFERENCIAIS | QUEM SOMOS

O **Laboratório de Análise de Dados** - LABDATA é um Centro de Excelência que atua nas áreas de ensino, pesquisa e consultoria em análise de informação utilizando técnicas de ***Big Data***, ***Analytics*** e **Inteligência Artificial**.

Profª Drª Alessandra Montini

O LABDATA é um dos pioneiros no lançamento dos cursos de *Big Data* e *Analytics* no Brasil

Os diretores foram professores de grandes especialistas do mercado

+10 anos de atuação

+1000 alunos formados

**Docentes**

- Sólida formação acadêmica: doutores e mestres em sua maioria
- Larga experiência de mercado na resolução de *cases*
- Participação em Congressos Nacionais e Internacionais
- Professor assistente que acompanha o aluno durante todo o curso

**Estrutura**

- 100% das aulas realizadas em laboratórios
- Computadores para uso individual durante as aulas
- 5 laboratórios de alta qualidade (investimento +R$2MM)
- 2 Unidades próximas a estação de metrô (com estacionamento)

# Corpo Diretivo

COORDENADORES DO LABDATA | ATUAÇÃO ACADÊMCIA E PROFISSIONAL

Profª Dra.
**Alessandra Montini**

Diretora do LABDATA-FIA, apaixonada por dados e pela arte de lecionar. Têm muito orgulho de ter criado na FIA cinco laboratórios para as aulas de Big Data e inteligência Artificial. Possui mais de 20 anos de trajetória nas áreas de Data Mining, Big Data, Inteligência Artificial e Analytics. Cientista de dados com carreira realizada na Universidade de São Paulo. Graduada e mestra em estatística aplicada pelo IME-USP e doutora pela FEA-USP. Com muita dedicação chegou ao cargo de professora e pesquisadora na FEA-USP, ganhou mais de 30 prêmios de excelência acadêmica pela FEA-USP e mais de 30 prêmios de excelência acadêmica como professora dos cursos de MBA da FIA. Orienta alunos de mestrado e de doutorado na FEA-USP. Membro do Conselho Curador da FIA, Coordenadora de Grupos de Pesquisa no CNPQ, Parecerista da FAPESP e Colunista de grandes Portais de Tecnologia.

linkedin.com/in/alessandramontini/

Prof. Dr.
**Adolpho Walter Canton**

Diretor do LABDATA-FIA. Consultor em Projetos de *Analytics*, *Big Data* e Inteligência Artificial. Professor FEA – USP. PhD em Estatística Aplicada pela *University of North Carolina at Chapel Hill*, Estados Unidos.

# Currículo – Prof. João Nogueira

FORMAÇÃO ACADÊMICA | EXPERIÊNCIA PROFISSIONAL

- **(2019-Presente)** – Professor nos cursos de Extensão, Pós e MBA em Big Data e Data Mining na Fundação Instituto de Administração (FIA) – www.fia.com.br
- **(2018-Presente)** – Cientista de Dados na Via Varejo – https://viavarejo.com.br
- **(2016-Presente)** – Doutorando em Física Computacional e Estatística pelo Departamento de Física na Universidade Federal do Ceará – https://física.ufc.br
- **(2014-2016)** – Mestre em Física da Matéria Condensada pelo Departamento de Física na Universidade Federal do Ceará - https://física.ufc.br
- **(2012-2013)** – Estudante Intercambista na Universidade de Coimbra – Portugal – https://www.uc.pt
- **(2010-2014)** – Bacharel em Física pela Universidade Federal do Ceará – http://www.ufc.br
- **Contatos:**
  - E-mail: joaonogueira@fisica.ufc.br

# Conteúdo Programático da Disciplina – Projeto de Inteligência Artificial

| Data | Horário | Tema |
|---|---|---|
| 09/03/2021 | 19:00 | Aula 1 - Introdução ao Ambiente de Desenvolvimento |
| 11/03/2021 | 19:00 | Aula 2 - Revisão de Python |
| 16/03/2021 | 19:00 | Aula 3 - Manipulação de Dados |
| 18/03/2021 | 19:00 | Aula 4 - Análise Exploratória de Dados |
| 23/03/2021 | 19:00 | Aula 5 - Projeto da disciplina - Parte 1 - Análise Exploratória de Dados |
| 25/03/2021 | 19:00 | Aula 6 - Introdução, Motivação e Framework de Machine Learning |
| 06/04/2021 | 19:00 | Aula 7 - Analytical Base Table |
| 08/04/2021 | 19:00 | Aula 8 - Aprendizagem Supervisionada - Classificação |
| 13/04/2021 | 19:00 | Aula 9 - Aprendizagem Supervisionada - Classificação |
| 15/04/2021 | 19:00 | Aula 10 - Aprendizagem Supervisionada - Classificação |
| 20/04/2021 | 19:00 | Aula 11 - Projeto da disciplina - Parte 2 - Machine Learning - Classificação |
| 22/04/2021 | 19:00 | Aula 12 - Projeto da disciplina - Parte 2 - Machine Learning - Classificação |
| 27/04/2021 | 19:00 | Aula 13 - Aprendizagem Supervisionada - Regressão |
| 29/04/2021 | 19:00 | Aula 14 - Aprendizagem Supervisionada - Regressão |
| 04/05/2021 | 19:00 | Aula 15 - Projeto da disciplina - Parte 3 - Machine Learning - Regressão |
| 06/05/2021 | 19:00 | Aula 16 - Aprendizagem Não-Supervisionada |
| 11/05/2021 | 19:00 | Aula 17 - Aprendizagem Não-Supervisionada |
| 13/05/2021 | 19:00 | Aula 18 - Projeto da disciplina - Parte 4 - Machine Learning - Clusterização |
| 18/05/2021 | 19:00 | Aula 19 - AutoML |
| 20/05/2021 | 19:00 | Aula 20 - Demonstração de Deploy de Machine Learning |

# Conteúdo da Aula

1. Redução de Dimensionalidade
   i. O que é?
   ii. Projeção
   iii. Manifold
   iv. PCA

2. Exercício Prático

# Material das aulas

- Iremos utilizar o Google Colab para desenvolver os códigos durante as aulas.

- Acesse https://bit.ly/tutorial-colab-projeto para realizar o tutorial de utilização do Google Colab.

# 1. Redução da Dimensionalidade

# 1. Redução da Dimensionalidade

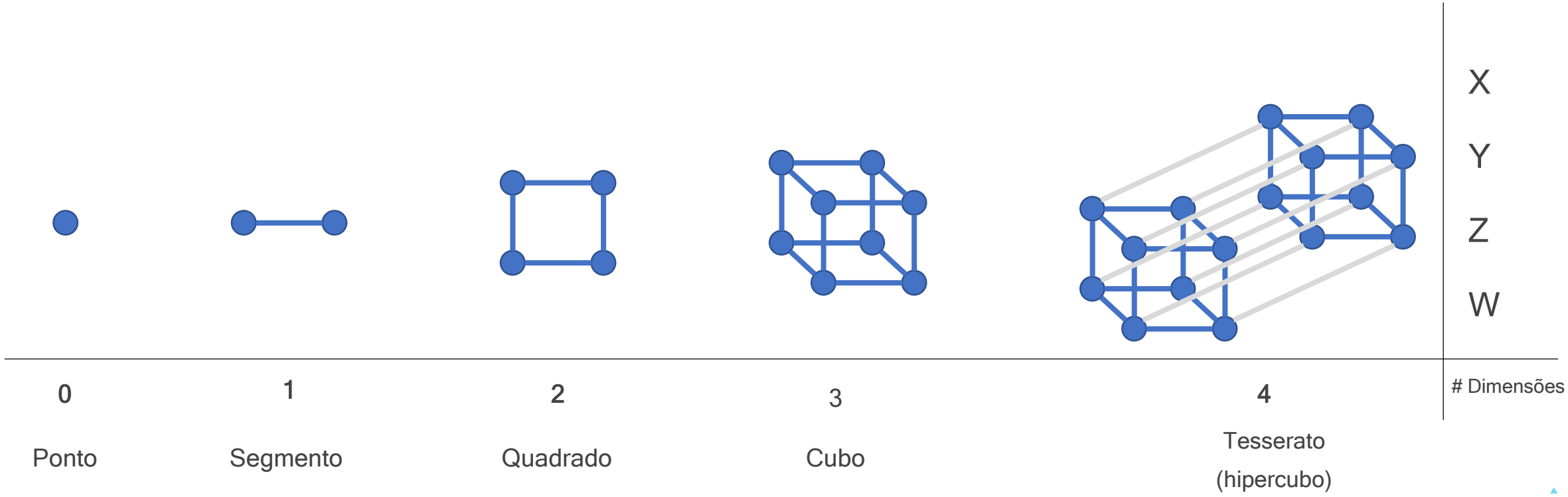
O QUE É?

- Até agora, os modelos que criamos foram baseados em poucas características e mesmo assim tivemos alguns desafios.
  - Encontrar uma solução boa
  - Tempo para otimizar os modelos
- Esses problemas escalam na medida que mais e mais características (features) são adicionadas.
- Agora, imagine agora um conjunto de dados com milhares ou até milhões de características.
- Esse problema é conhecido como a **Maldição da Dimensionalidade**.

# 1. Redução da Dimensionalidade

O QUE É?

- **Maldição da Dimensionalidade**




Hipercubo: https://www.youtube.com/watch?v=BVo2igbFSPE

# 1. Redução da Dimensionalidade

O QUE É?

- Felizmente podemos transformar um problema muito complexo e praticamente sem solução para um problema tratável no mundo real ao reduzir consideravelmente o número de ***features***.
- Além de acelerar o treinamento, a redução de dimensionalidade é extremamente útil para visualizar os dados, nesse caso reduzindo para 2 ou 3 dimensões.
- Para entender melhor a questão da dimensionalidade, vamos olhar para o conjunto de dados **MNIST**.
  - Esse conjunto de dados é composto por 70 mil pequenas imagens de dígitos escritos a mão.
  - Cada imagem é rotulada com o dígito que a representa.

# 1. Redução da Dimensionalidade

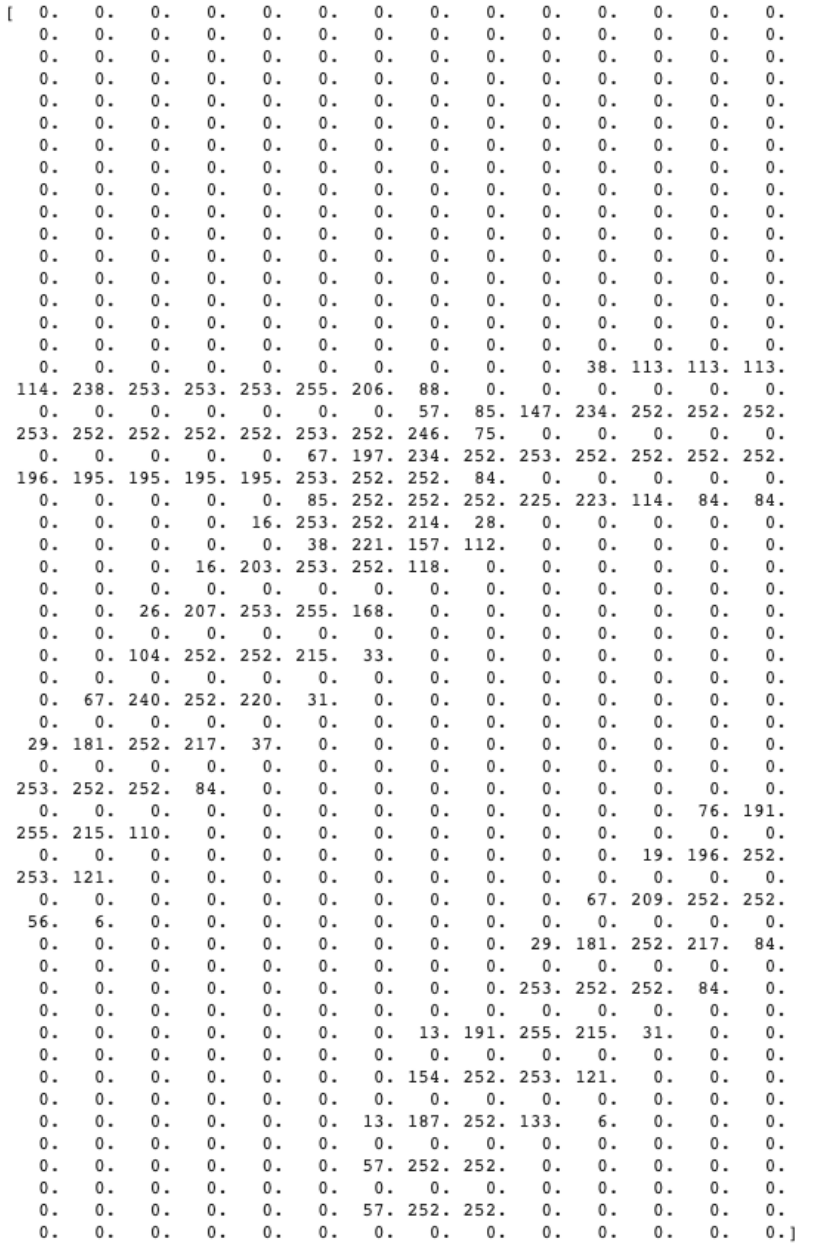
O QUE É?

```
from sklearn.datasets import fetch_openml

X, y = fetch_openml('mnist_784', version=1, return_X_y=True)

print(X.shape)
(70000, 784)

print(y.shape)
(70000, )
```



# 1. Redução da Dimensionalidade

O QUE É?

- Cada número dentro dessa estrutura de dados é uma característica que representa a imagem.

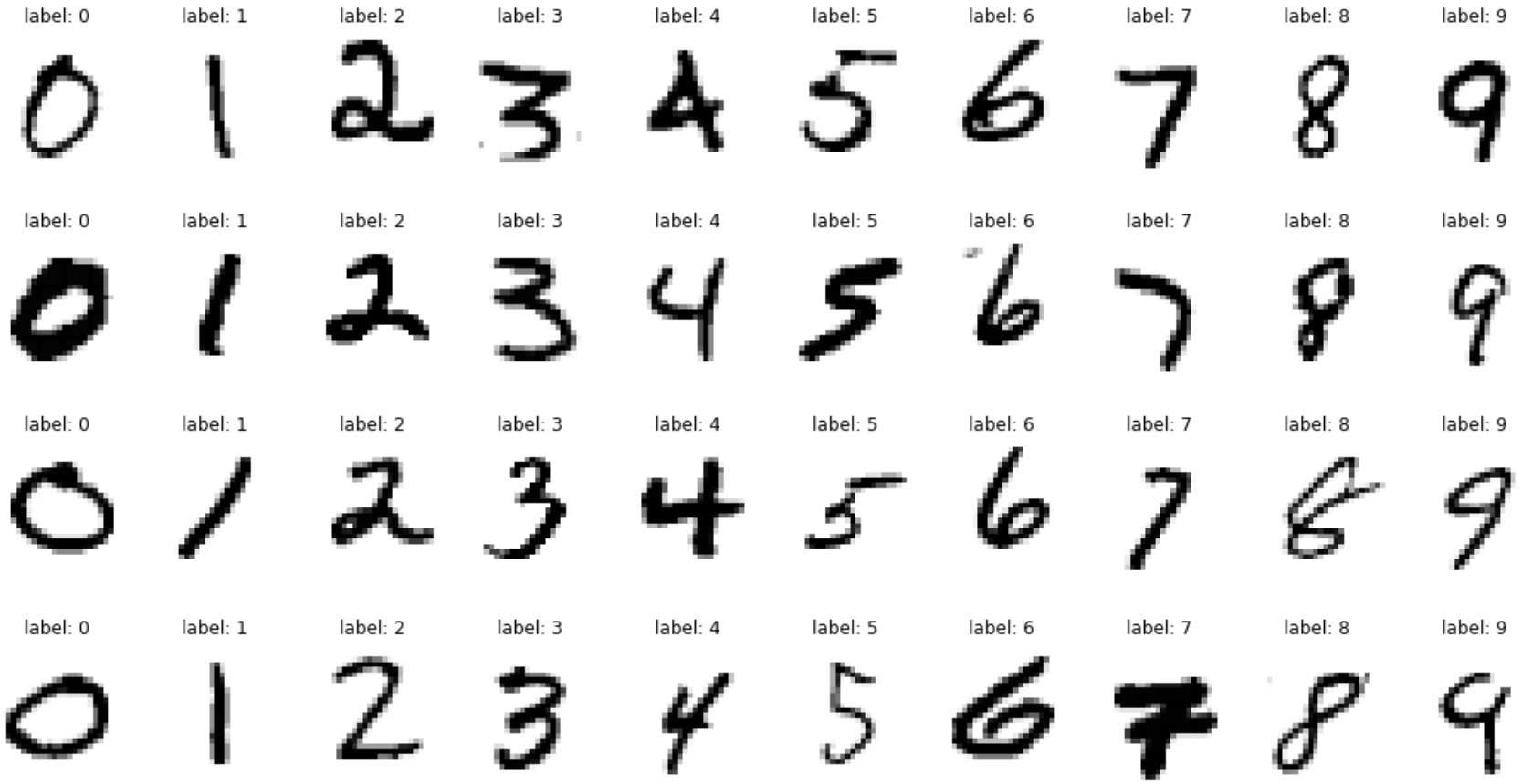
```
[   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  38. 113. 113. 113.
  114. 238. 253. 253. 253. 255. 206.  88.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.  57.  85. 147. 234. 252. 252. 252.
  253. 252. 252. 252. 252. 253. 252. 246.  75.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.  67. 197. 234. 252. 253. 252. 252. 252. 252.
  196. 195. 195. 195. 195. 253. 252. 252.  84.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.  85. 252. 252. 252. 225. 223. 114.  84.  84.
    0.   0.   0.   0.  16. 253. 252. 214.  28.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.  38. 221. 157. 112.   0.   0.   0.   0.   0.
    0.   0.   0.  16. 203. 253. 252. 118.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.  26. 207. 253. 255. 168.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0. 104. 252. 252. 215.  33.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.  67. 240. 252. 220.  31.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   29. 181. 252. 217.  37.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
  253. 252. 252.  84.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  76. 191.
  255. 215. 110.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  19. 196. 252.
  253. 121.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  67. 209. 252. 252.
   56.   6.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.  29. 181. 252. 217.  84.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0. 253. 252. 252.  84.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.  13. 191. 255. 215.  31.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0. 154. 252. 253. 121.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.  13. 187. 252. 133.   6.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.  57. 252. 252.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.  57. 252. 252.   0.   0.   0.   0.   0.
    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
```

- Logo, essa imagem contém **784 *features***.

# 1. Redução da Dimensionalidade

O QUE É?

- Outros exemplos:



# 1. Redução da Dimensionalidade

O QUE É?

- Antes de estudarmos em algoritmos específicos de redução de dimensionalidade, podemos utilizar duas abordagens: Projeção e Manifold Learning.
- Projeção:
  - Na maioria dos problemas, as instâncias de treinamento não se espalham uniformemente em todas as dimensões.
  - Muitas características são quase constantes, enquanto outras estão altamente correlacionadas (como visto no MNIST).
  - Como resultado, todas as instâncias de treinamento estão realmente dentro (ou perto) de um *subespaço* de dimensão bem menor no espaço de alta dimensão, vejamos um exemplo.

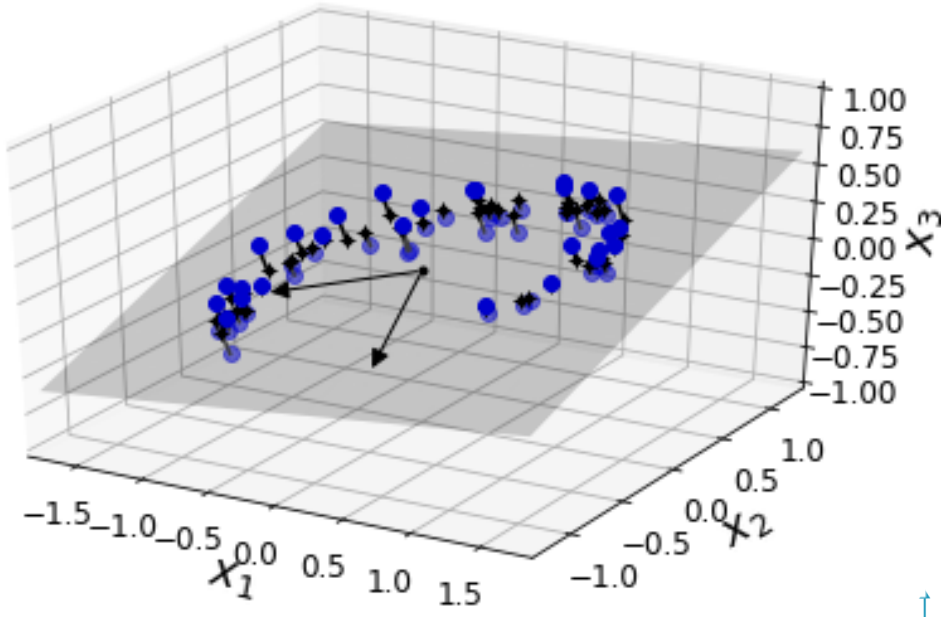
**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Note que todas as instâncias de treinamento estão próximas de um plano: este é um subespaço (2D) de dimensões inferiores em um espaço de alta dimensão (3D).

- Agora se projetarmos perpendicularmente cada instância de treinamento a este subespaço (representado pelas linhas curtas que conectam as instâncias ao plano), obtemos um novo conjunto de dados 2D.



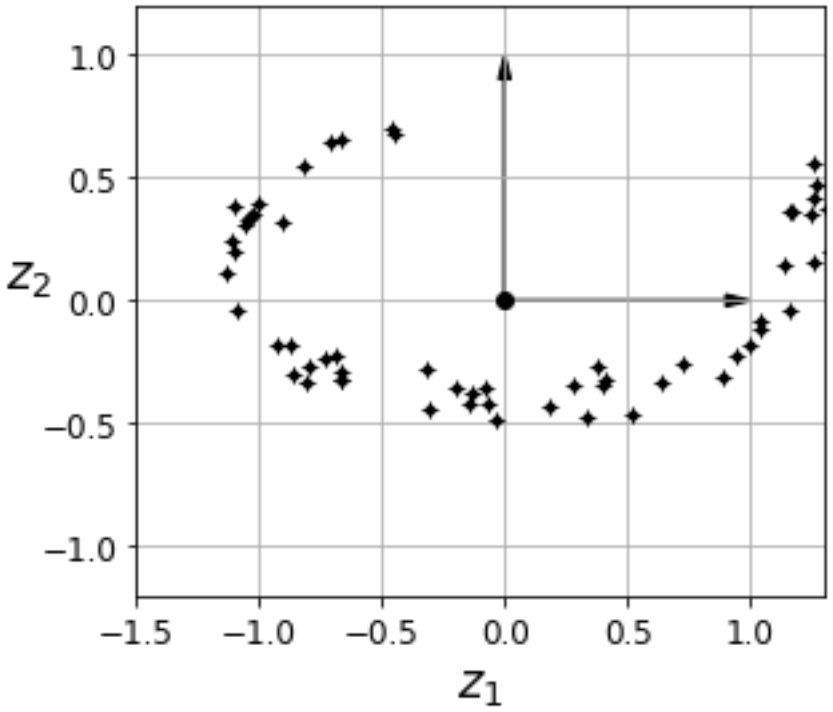

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Com isso, reduzimos a dimensionalidade do conjunto de dados de 3D para 2D.

- Note que os eixos correspondem a novas características (features)  z1 e z2 (coordenadas das projeções do plano).

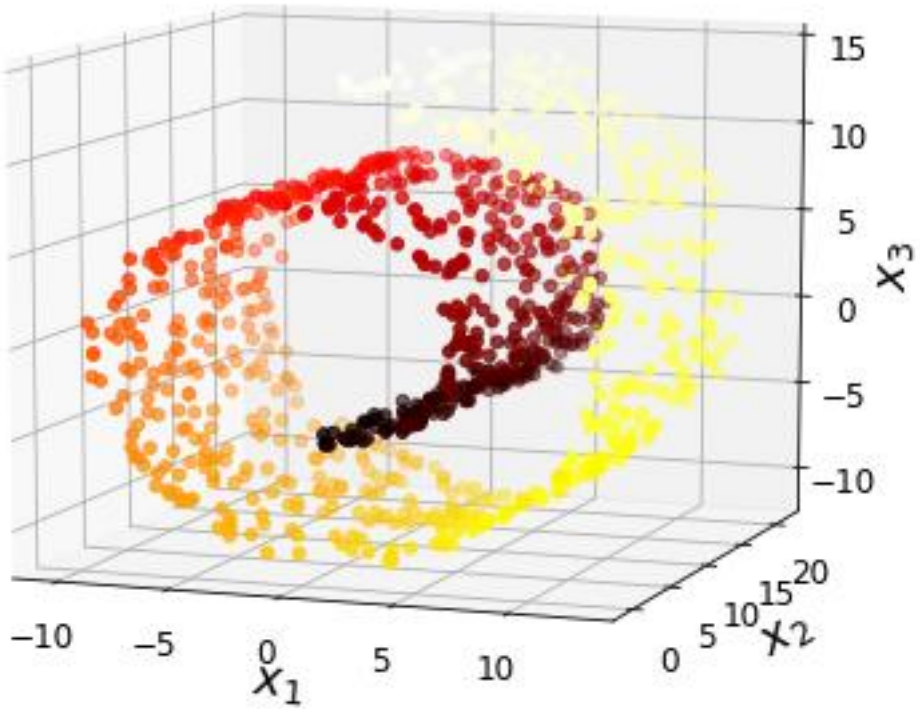



**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

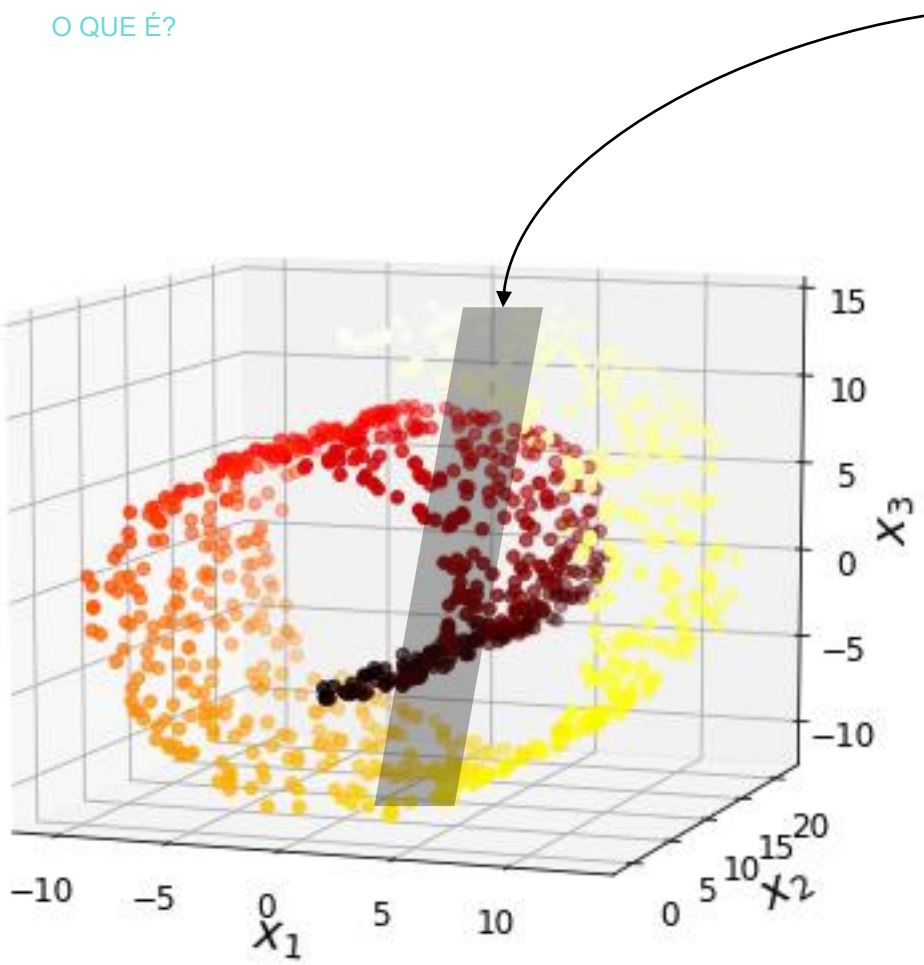
- Nem sempre a projeção é a melhor abordagem para a redução da dimensionalidade, uma vez que o subespaço pode torcer e girar como o famoso conjunto de dados em rolo suiço.



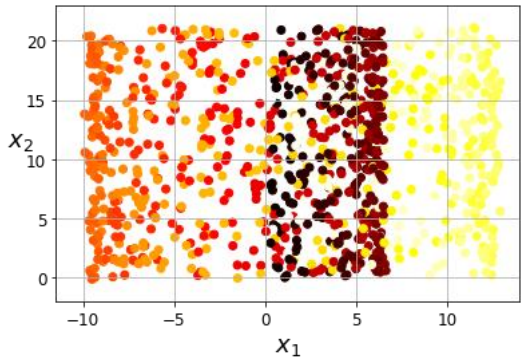

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

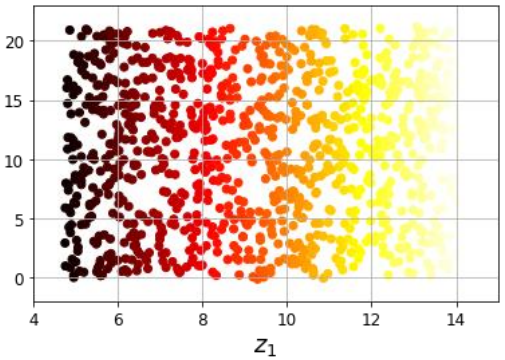
# 1. Redução da Dimensionalidade

O QUE É?




- Nesse caso projetar em **um plano**, comprimira diferentes camadas do rolo suíço (imagem abaixo à esquerda).
- No entanto o que queremos é desenrolar o rolo para obter o conjunto de dados 2D à direita na imagem abaixo.




**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Esse exemplo de rolo suiço é um exemplo de Manifold 2D.
- Manifold 2D é uma forma 2D que pode ser dobrada e torcida em um espaço de dimensões maiores.
- Geralmente um manifold d-dimensional é uma parte de um espaço n-dimensional (em que n > d) que se assemelha localmente a um hiperplano d-dimensional.
- No caso do rolo suiço, d = 2 e n = 3.
    - Se assemelha localmente a um plano 2D, mas é enrolado na terceira dimensão.
- Muitos algoritmos de redução de dimensionalidade funcionam modelando o *manifold* onde estão as instâncias de treinamento, isso é chamado de **Manifold Learning**.

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Esse método de aprendizado se baseia na *manifold assumption*, também conhecida como *manifold hypothesis*, que sustenta que a maioria dos conjuntos de dados de alta dimensão do mundo real está próxima de um *manifold* muito mais baixo.
    - Essa suposição é frequentemente observada empiricamente.
- Voltamos ao exemplo do MNIST:
    - Todas as imagens manuscritas dos dígitos têm algumas semelhanças.
    - Elas são feitas de linhas conectadas, bordas são brancas, estão mais ou menos centralizadas, e assim por diante.
    - Se essas imagens fossem geradas aleatoriamente, somente uma fração minúscula delas seria semelhante a dígitos escritos a mão.

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

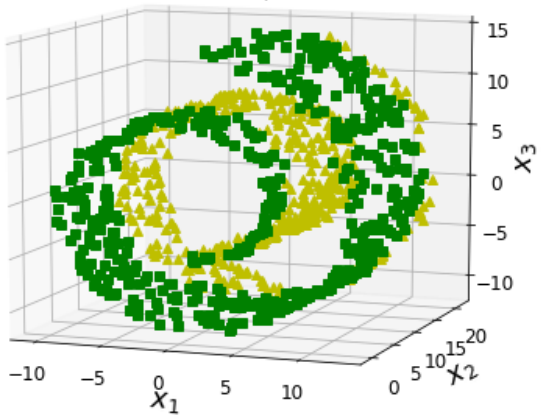
- Podemos dizer que os graus de liberdade disponíveis quando se tenta criar uma imagem numérica são drasticamente inferiores aos graus de liberdade que teria se pudesse gerar outra imagem qualquer.
- Essas restrições tendem a comprimir o conjunto de dados em um *manifold* de dimensão inferior.

- A *manifold assumption* é muitas vezes acompanhada por outra suposição implícita:
  - A tarefa em questão (classificação ou regressão, por exemplo) será mais simples se for expressa em um espaço de dimensão inferior do *manifold.*

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

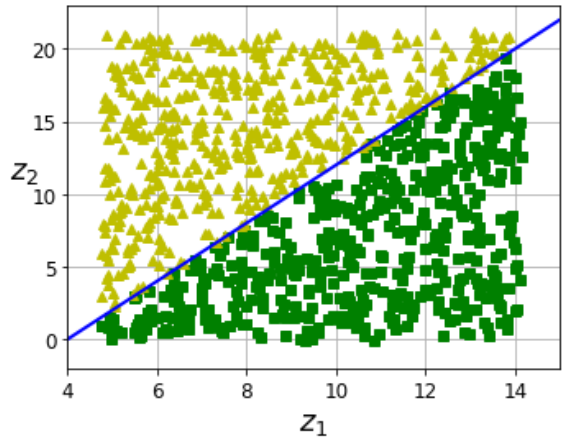
# 1. Redução da Dimensionalidade

O QUE É?




- Como podemos visualizar, o rolo suíço é separado em duas classes no espaço 3D e para encontrar a fronteira de decisão seria bastante complexa.

- Mas, no espaço manifold 2D desenrolado a fronteira de decisão é apenas uma linha reta.

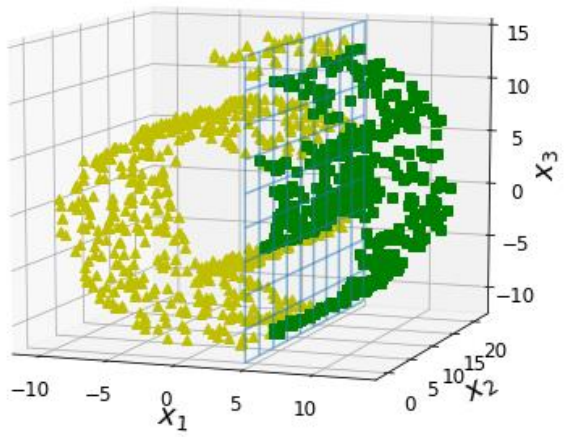



**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

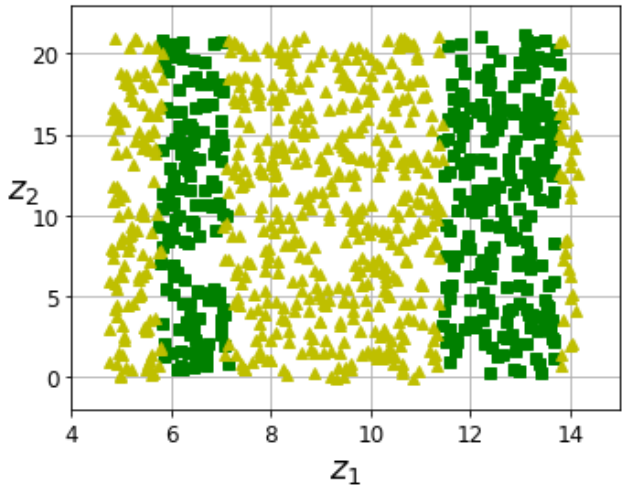
# 1. Redução da Dimensionalidade

O QUE É?

- No entanto essa suposição nem sempre é valida, por exemplo se a fronteira de decisão for em x1 = 5.




- Essa fronteira de decisão parece muito simples no espaço 3D original (um plano vertical), porém no *manifold* desenrolado será mais complexa.




**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Em resumo, se reduzir a dimensionalidade em seu conjunto de treinamento antes de treinar um modelo, ele definitivamente acelerará o treinamento, mas nem sempre poderá levar a uma solução melhor ou mais simples.

**Tudo depende do conjunto de dados.**

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Existem diversos algoritmos de redução de dimensionalidade
  - PCA – *Principal Component Analysis*
  - t-NSE – *t-Distributed Stochastic Neighbor Embedding*
  - UMAP - *Uniform Manifold Approximation and Projection*
  - LLE – Locally Linear Embedding
  - LDA - *Linear Discriminant Analysis*
  - MDS - *Multidimensional scaling*
  - Isomap

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

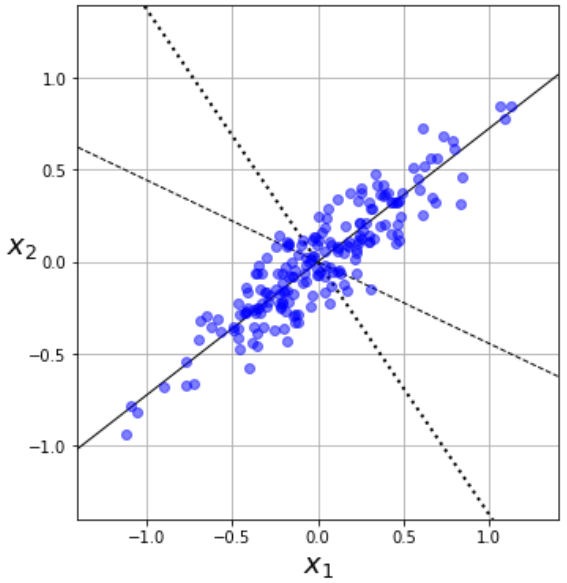
- Vamos estudar o algoritmo PCA – Análise dos Componentes Principais, pois é o algoritmo mais popular para redução de dimensionalidade.
- Seu funcionamento é relativamente simples:
  - Primeiro identifica o hiperplano que se encontra mais próximo dos dados, e
  - Depois projeta os dados sobre ele.

- Considere um simples conjunto de dados 2D com 3 eixos diferentes (ou seja, hiperplanos unidimensionais)

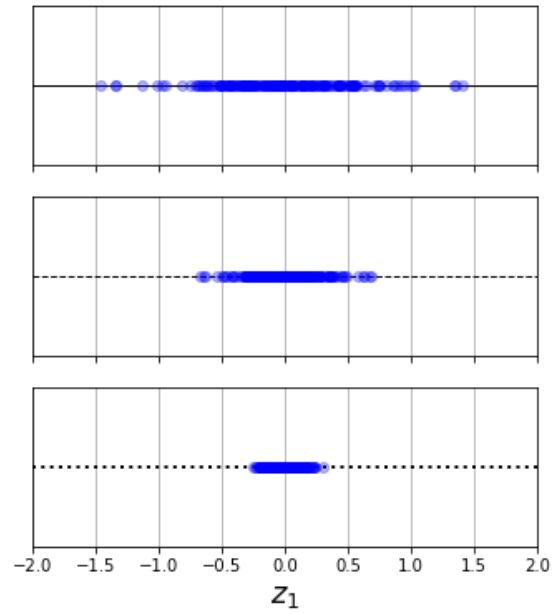
**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?




- O resultado da **projeção** do conjunto de dados em cada um desses eixos.
- A projeção na linha sólida preserva a variância máxima.
- A projeção na linha tracejada preserva uma variância intermediária.
- A projeção na linha pontilhada preserva pouca variância.




**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Parece  razoável selecionar o eixo que preserva a quantidade máxima de variância, pois provavelmente ela perderá menos informações do que as outras projeções.
- O fato de o eixo minimizar a distância quadrática média entre o conjunto original de dados e sua projeção nesse eixo é outra maneira de justificar a escolha.

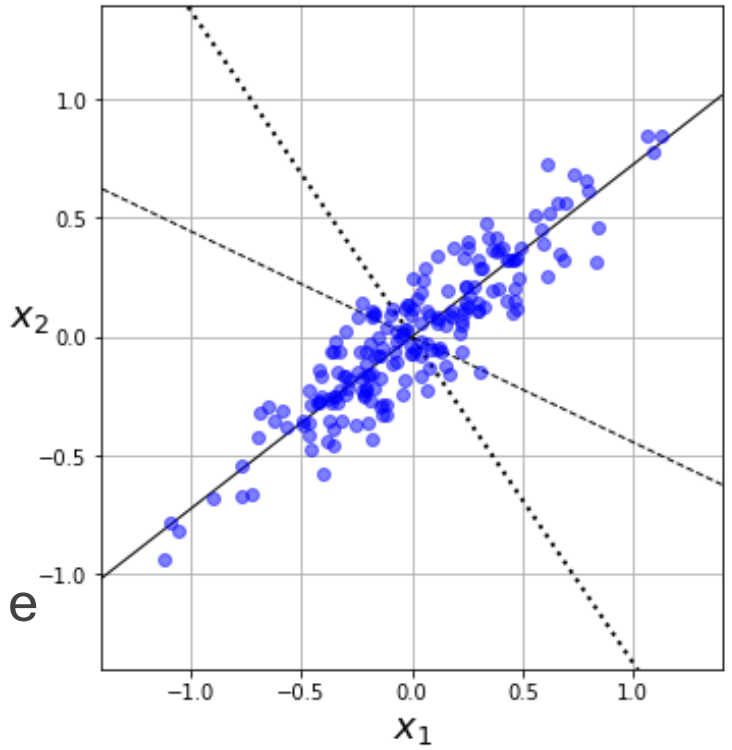
- **Esta é a ideia simples por trás do PCA.**

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- O PCA identifica o eixo que representa a maior quantidade de variância no conjunto de treinamento, no nosso caso é a linha sólida.
- Além disso, ele também encontra um segundo eixo ortogonal ao primeiro, que representa a maior quantidade remanescente da variância. Em nosso caso será a linha pontilhada.
- Se tivéssemos mais dimensões, o PCA também encontraria um terceiro eixo ortogonal ao dois anteriores, e um quarto, um quinto, e assim por diante.




**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- A classe PCA do Scikit-Learn implementa o PCA com a utilização da decomposição SVD como fizemos anteriormente.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X2D = pca.fit_transform(X_train)
```

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Uma informação útil é a taxa de variância explicada de cada componente principal, ele fica disponível no atributo `explained_variance_ratio_`. Esse valor indica a proporção da variância do conjunto de dados que se encontra ao longo do eixo de cada componente principal.

```
print(pca.explained_variance_ratio_)

[0.84248607, 0.14631839]
```

- Indica que 84,24% da variância do conjunto de dados está ao longo do primeiro eixo e 14,63% situa-se ao longo do segundo eixo.  Isso deixa 1,1%, aproximadamente, para o terceiro eixo, por isso é razoável supor que ele carrega pouca informação.

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- O invés de escolher arbitrariamente o número de dimensões a serem reduzidas, é preferível escolher o número de dimensões que adicionam uma fração suficientemente grande de variância (por exemplo 95%).
- A menos que queira reduzir a dimensionalidade para visualização dos dados, nesse caso o número de dimensões deve ser 2 ou 3.
- Para calcular o PCA mantendo a variância em 95%

```
pca = PCA(n_components=0.95)
X_reduced = pca.fit_transform(X_train)
```

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

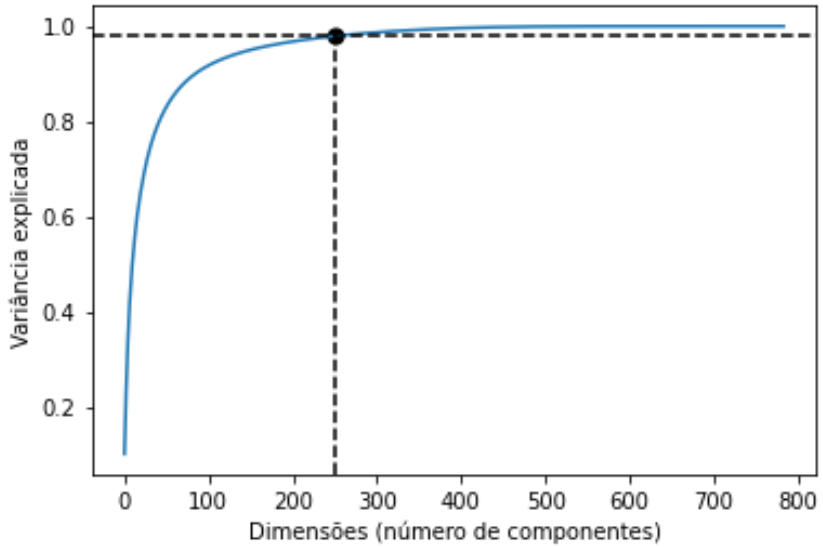
O QUE É?

- Outra opção seria plotar a variância explicada como uma função do número de dimensões.
- Geralmente haverá um cotovelo na curva no qual a variância explicada rapidamente deixa de crescer.

```python
import matplotlib.pyplot as plt

pca = PCA().fit(X_train)
cumsum = np.cumsum(pca.explained_variance_ratio_)
plt.plot(cumsum)

plt.axvline(x=250, ymax=0.95, ls='--', color='black')
plt.axhline(y=0.98, xmax=750, ls='--', color='black')
plt.scatter(250, 0.98, s=50, c='black')
plt.xlabel('Dimensões (número de componentes)')
plt.ylabel('Variância explicada');
```




**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Voltamos ao nosso exemplo do MNIST.
- Temos 784 características e iremos aplicar o PCA para **manter 95% da sua variação**.
- Após aplicar o PCA, cada instância terá pouco mais de 150 características.

- **Desta forma, o conjunto de dados contém menos de 20% do seu tamanho original!**

- Essa é uma taxa de compressão razoável, possibilitando acelerar o processo de treinamento de algum algoritmo de classificação, regressão ou clusterização.

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

- Para o conjunto de dados do MNIST, mesmo após aplicar o PCA e realizar a compressão, seria possível descomprimir os dados e recriar as imagens escritas a mão?
- Sim, podemos utilizar o método `inver_transform()` para descomprimir e voltar para as 784 dimensões.
- Nesse processo, a projeção criada perde um pouco de informação (considerando que descartamos 5% da variância), mas provavelmente o resultado será bem próximo do original.
- A distância quadrática média entre os dados originais e os dados reconstruídos é chamada de *erro de reconstrução.*

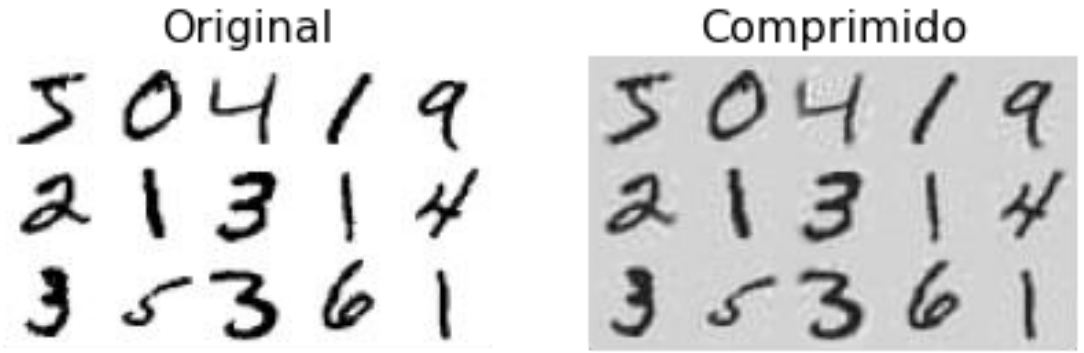
$$X_{recuperado} = X_{d-proj} \cdot W_d^T$$

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

```
pca = PCA(n_components = 154)
X_reduced = pca.fit_transform(X_train)
X_recovered = pca.inverse_transform(X_reduced)
```




Ver no notebook a função de plotagem para a reconstrução da imagem

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 1. Redução da Dimensionalidade

O QUE É?

</> Abra o arquivo **"aula17-parte1-conceitos.ipynb"**

**Fonte**: Capítulo 8: Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow de Aurelien Geron, 2019, O'Reilly

# 2. Exercício Prático

# 1. Redução da Dimensionalidade

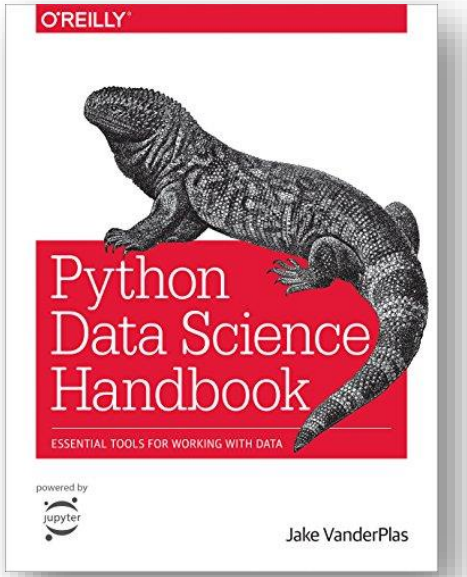
O QUE É?

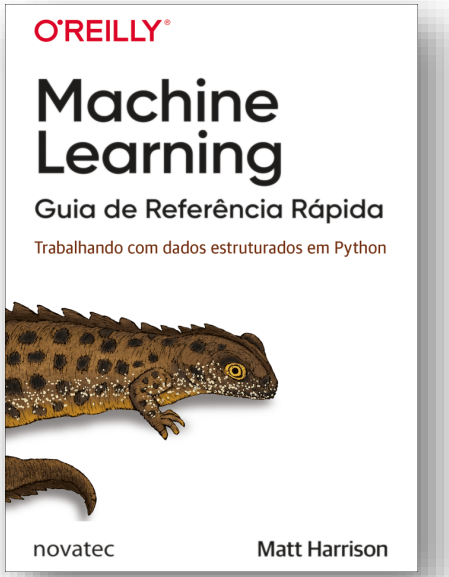
</> Abra o arquivo **"aula17-parte2-exercício-pratico.ipynb"**

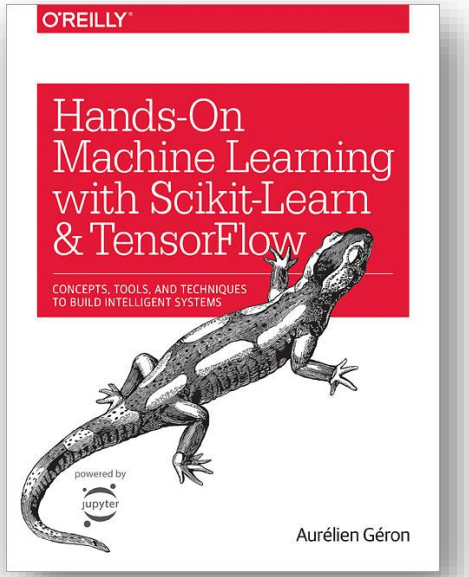
# Referências Bibliográficas

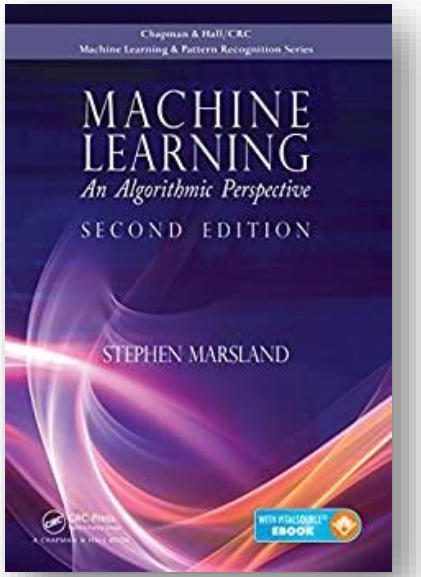
# Referências Bibliográficas

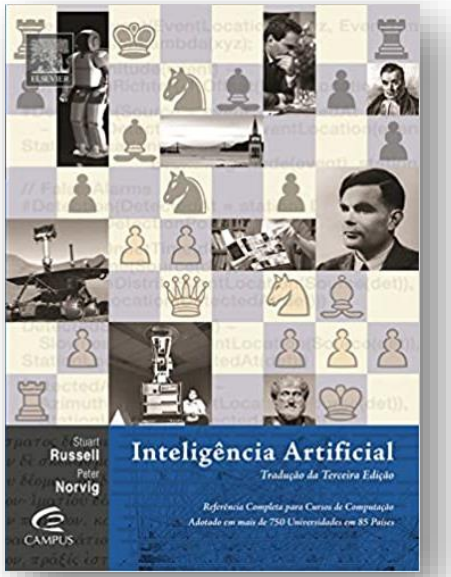
LIVROS



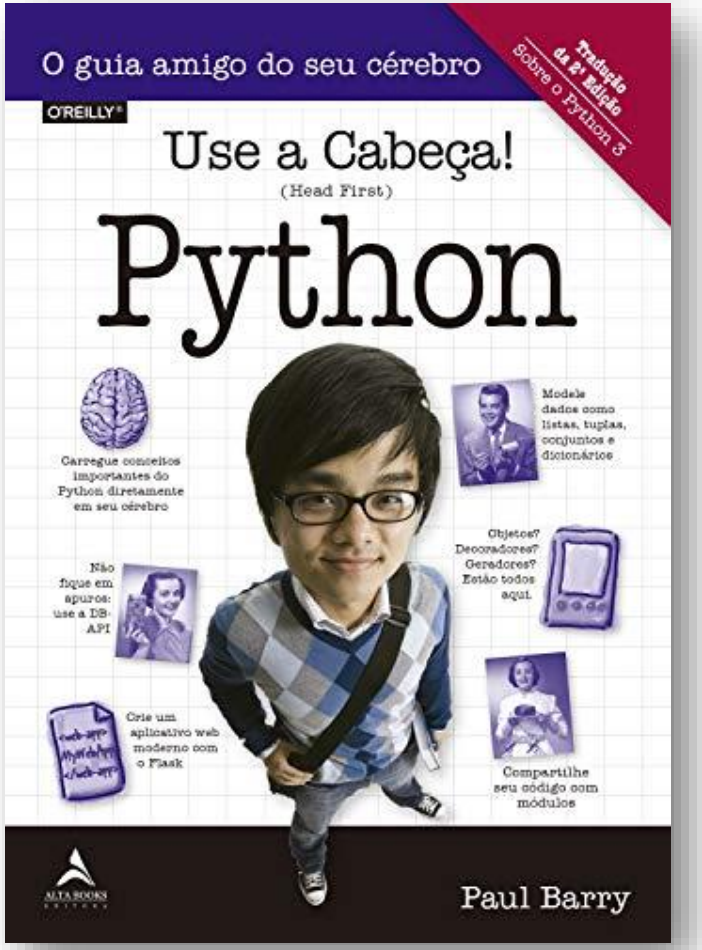



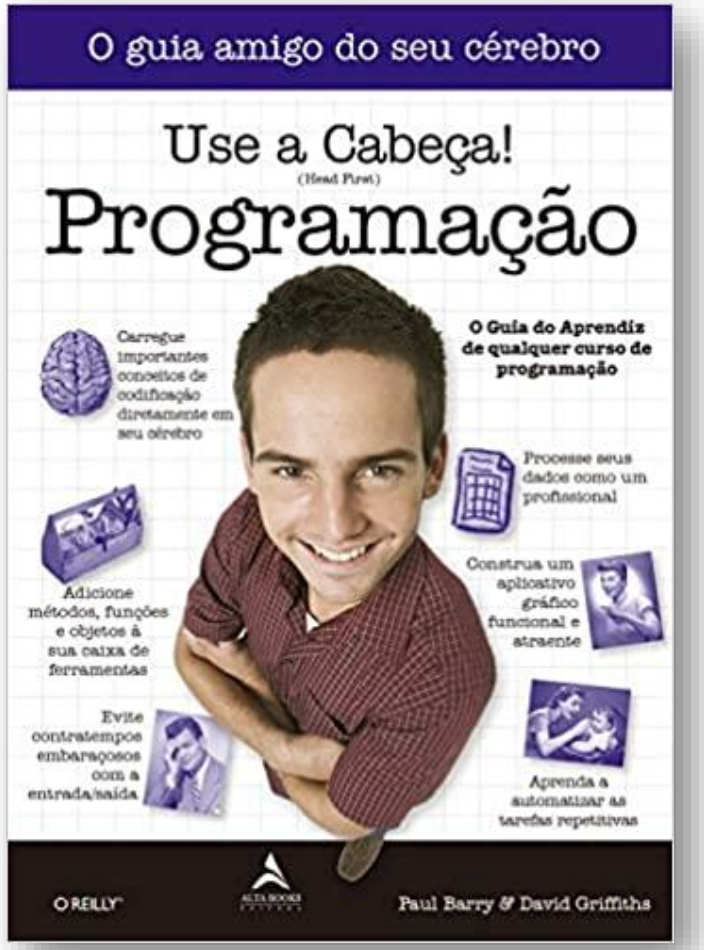



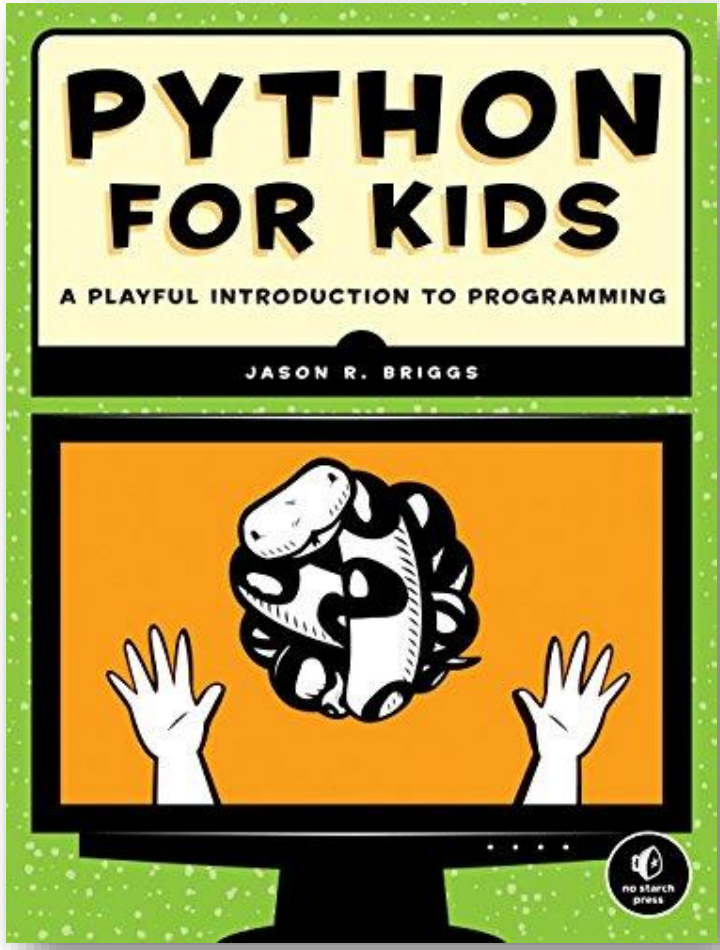



# Referências Bibliográficas

LIVROS

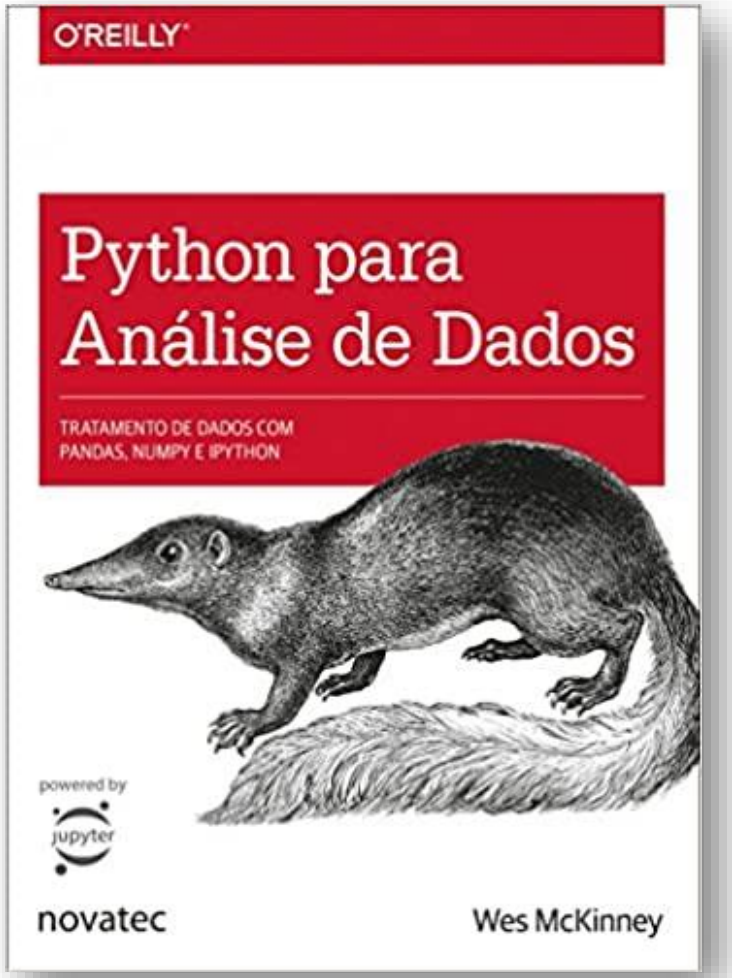


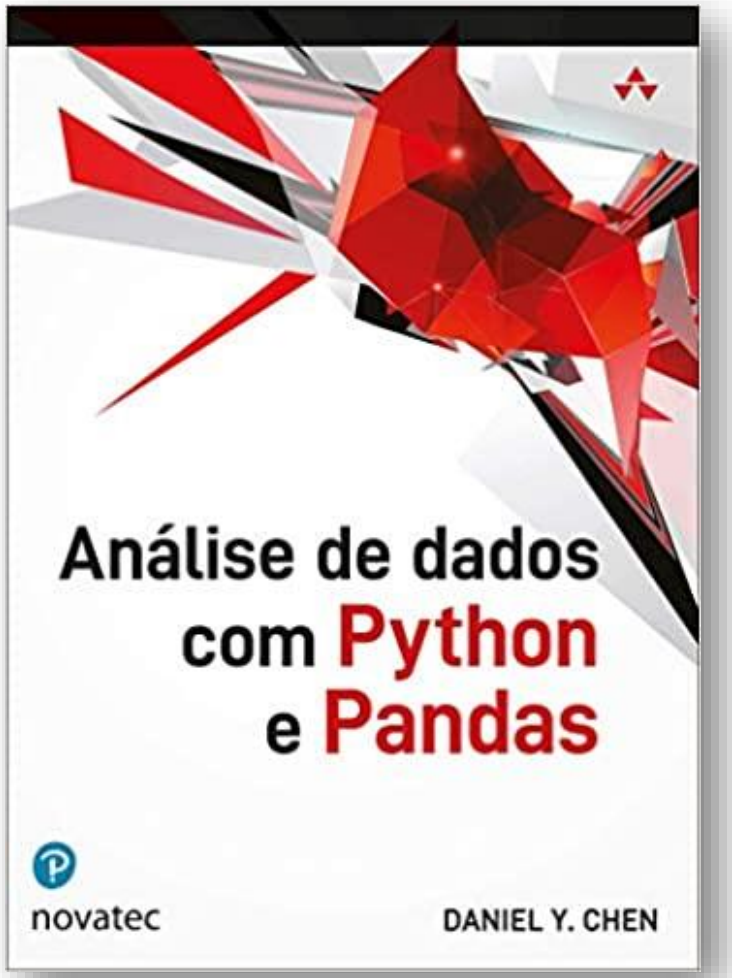



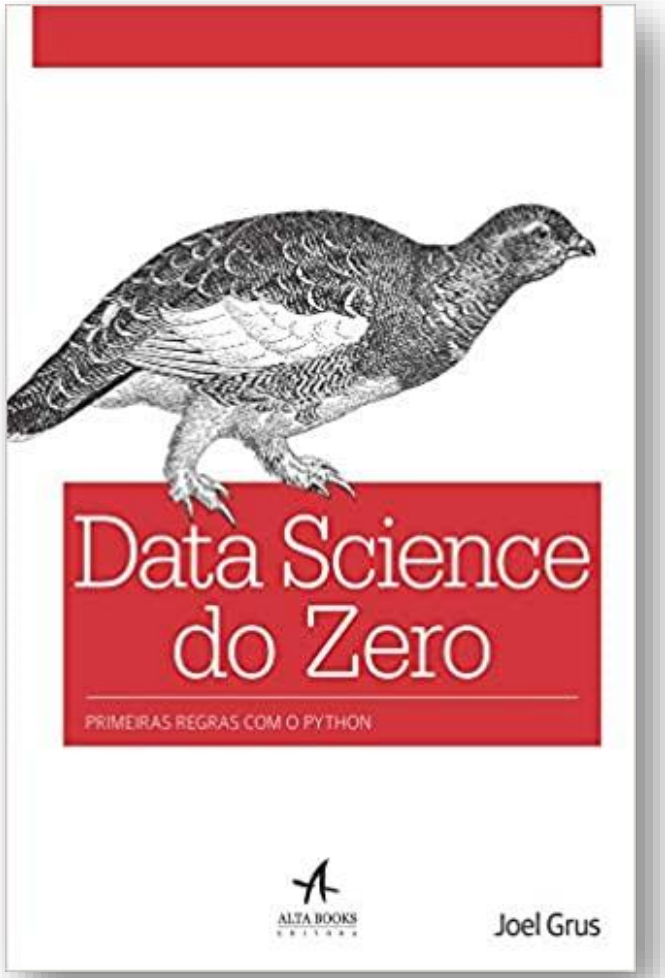



# Referências Bibliográficas

LIVROS

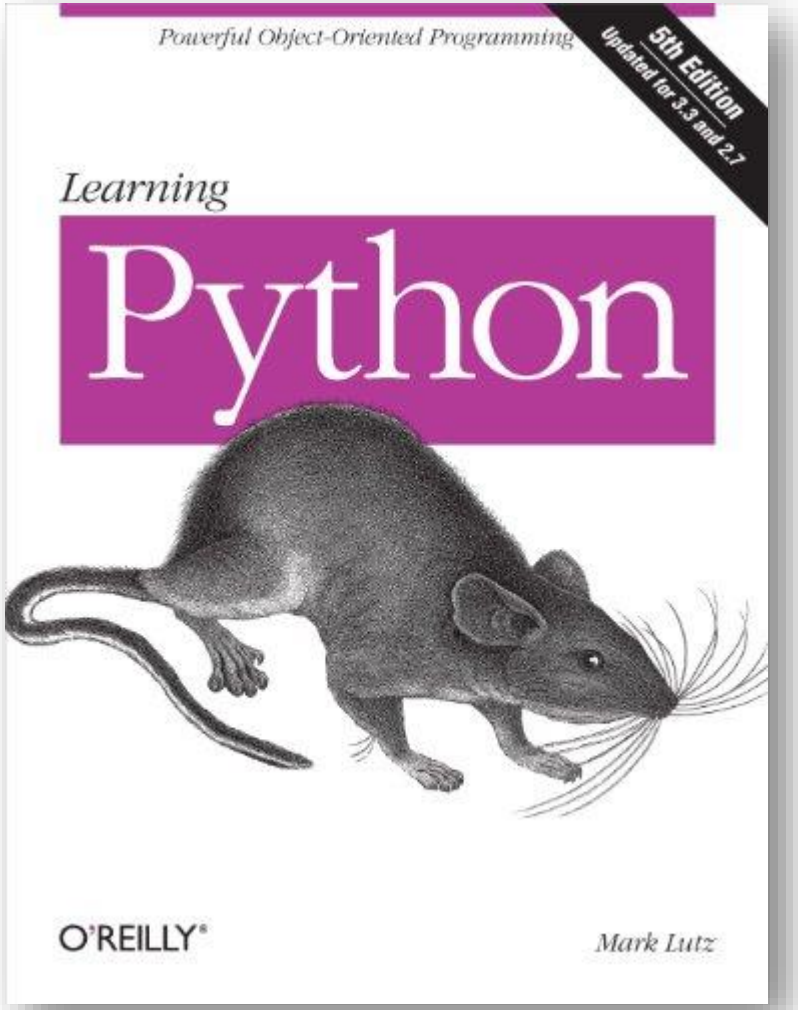


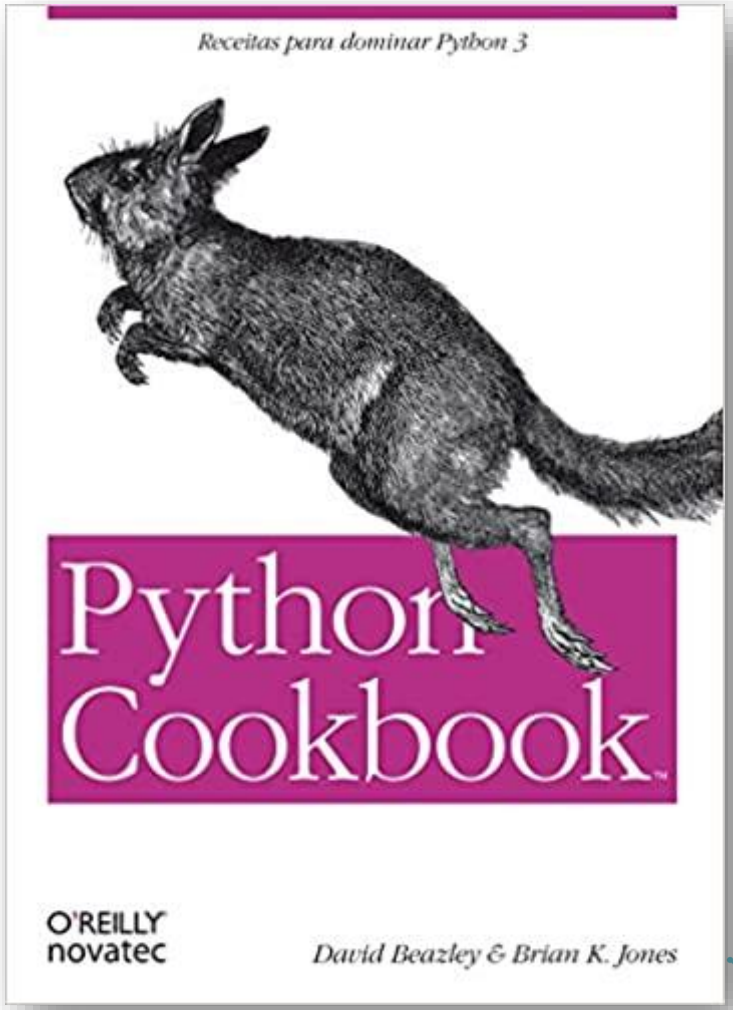



# Referências Bibliográficas

LIVROS







# Referências Bibliográficas

LINKS, ÍCONES, IMAGENS


- As referências de links utilizados podem ser visualizados em http://urls.dinomagri.com/refs
- Tutoriais disponíveis no site oficial do Pandas - http://pandas.pydata.org/pandas-docs/stable/
- Livro de receitas disponíveis no site oficial do Pandas - http://pandas.pydata.org/pandas-docs/stable/cookbook.html

- As imagens foram Icon made by Srip, Pixel perfect, Eucalyp e Prettycons from www.flaticon.com