

# Introdução a Ciencia de Dados

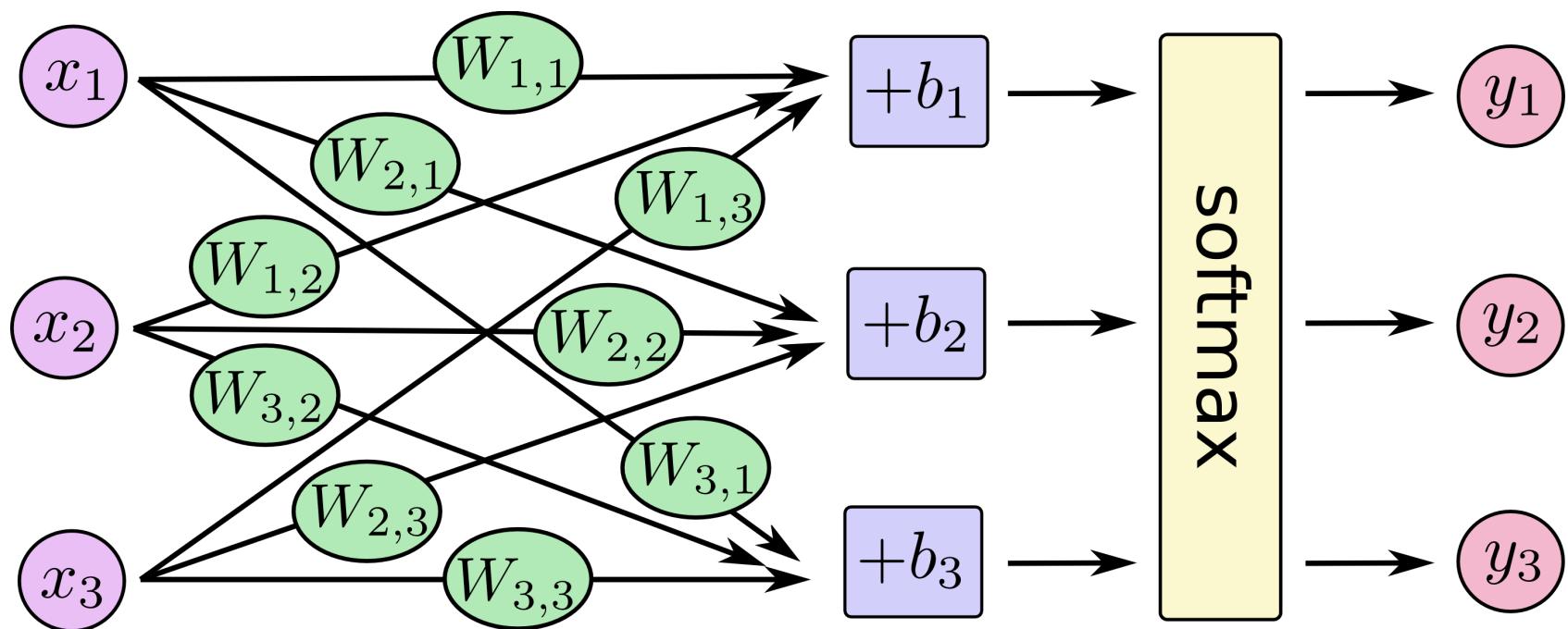
Hitoshi Nagano, Ph.D.



# **Redes Neurais Artificiais**

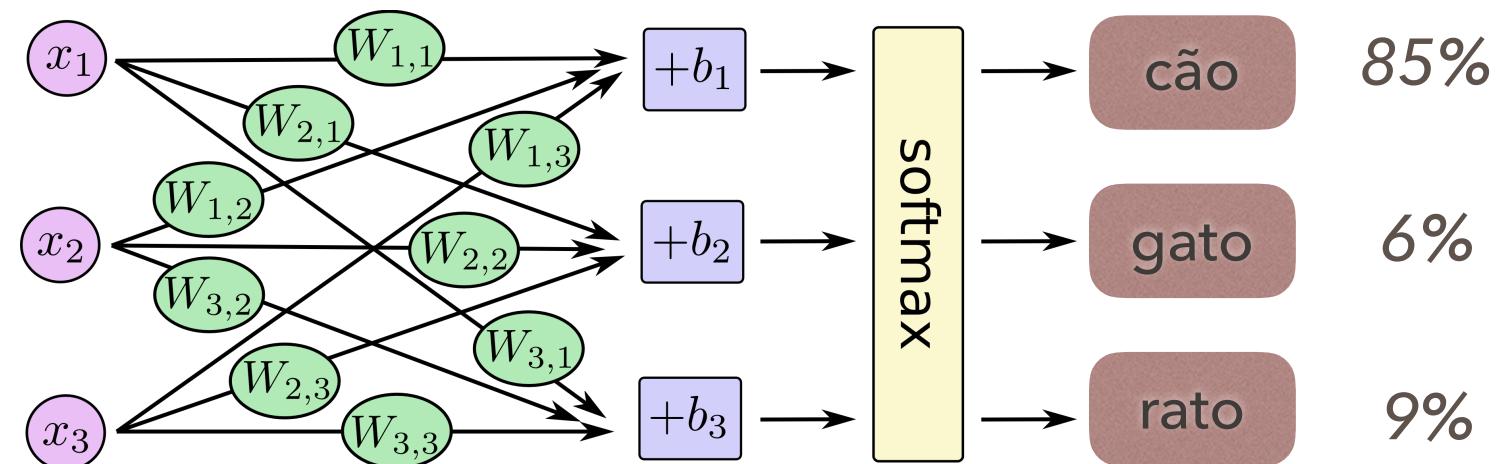
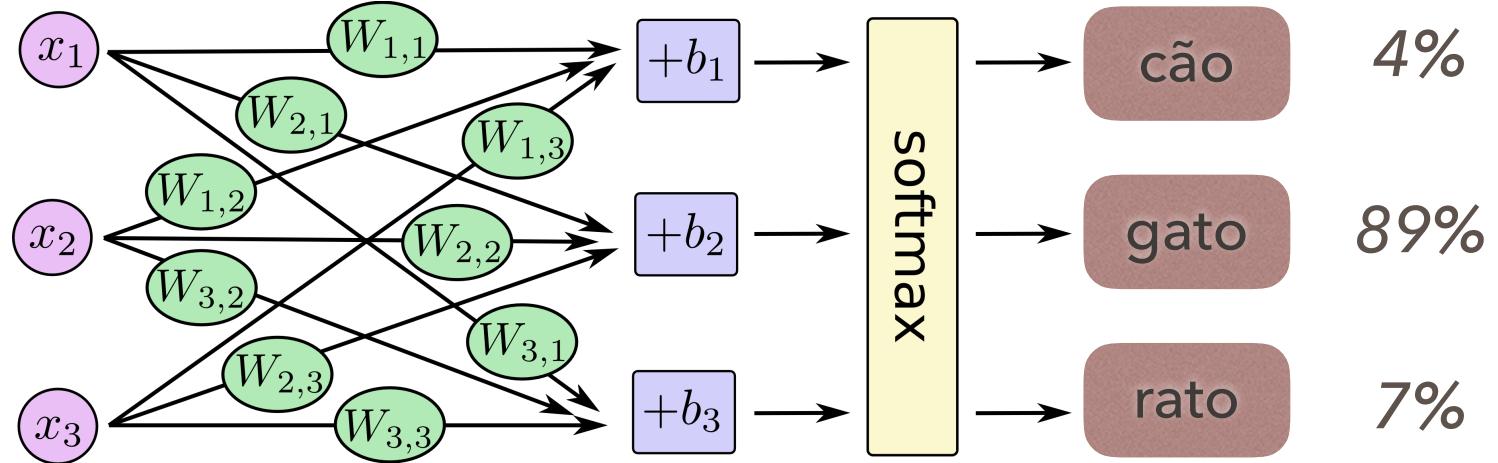
- arquitetura
- forward propagation
- otimização
- backpropagation
- tensorflow & keras

- livros
  - <http://neuralnetworksanddeeplearning.com>
  - [http://www.deeplearningbook.org/lecture\\_slides.html](http://www.deeplearningbook.org/lecture_slides.html)
- cursos
  - <http://cs231n.github.io/>
  - <https://web.stanford.edu/class/cs224n/>
- blogs
  - <http://colah.github.io/>

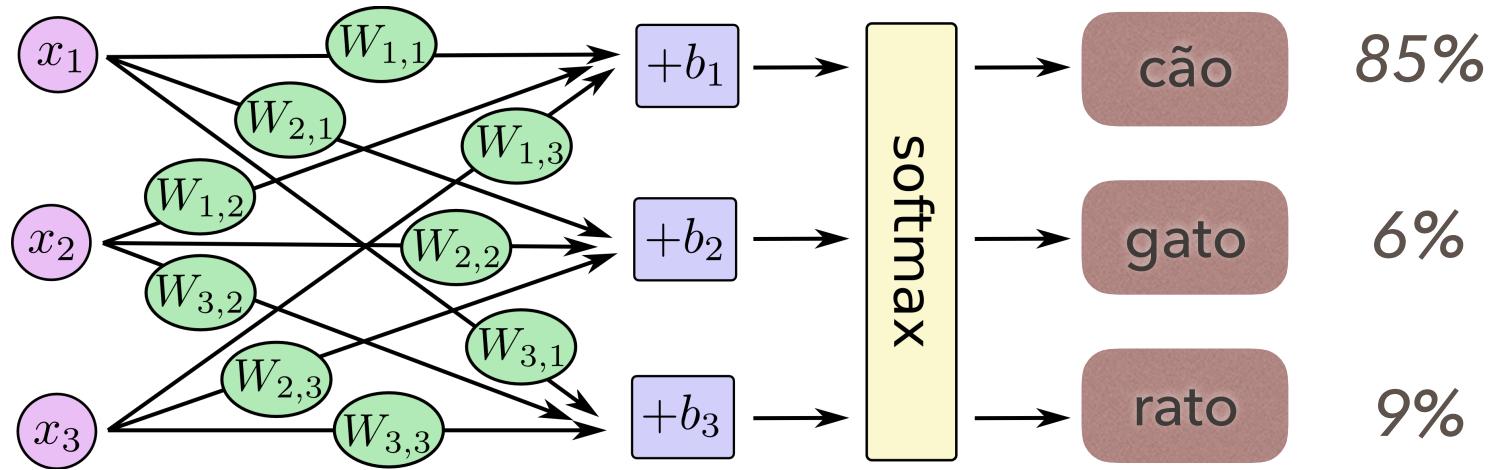


[https://www.tensorflow.org/get\\_started/mnist/beginners](https://www.tensorflow.org/get_started/mnist/beginners)

# qual é o problema?



# qual é o problema?

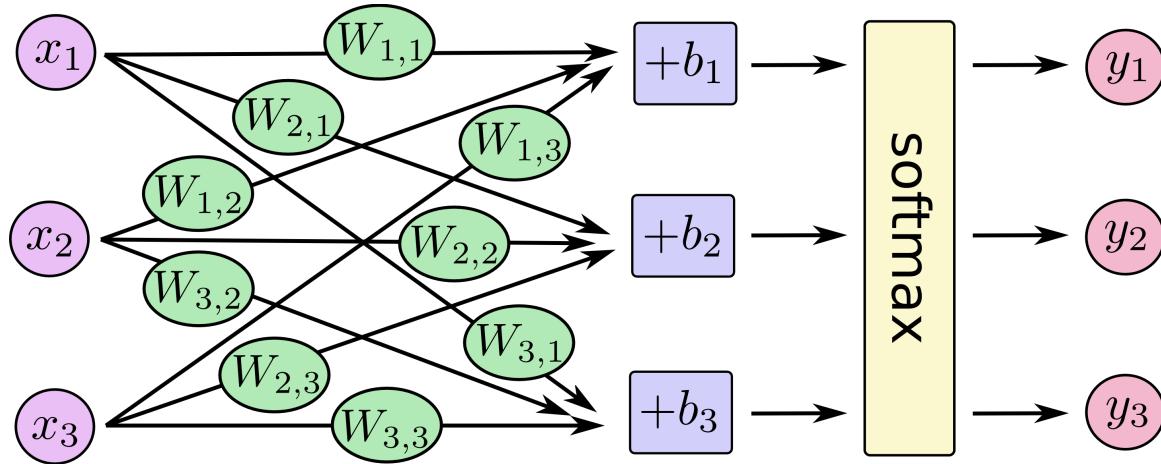


**X**

	pixel1	pixel2	pixel3	...	pixel n
	2	133	25	...	77
	3	3	55	...	89
	254	255	253	...	255
	8	8	10	...	21

**y**

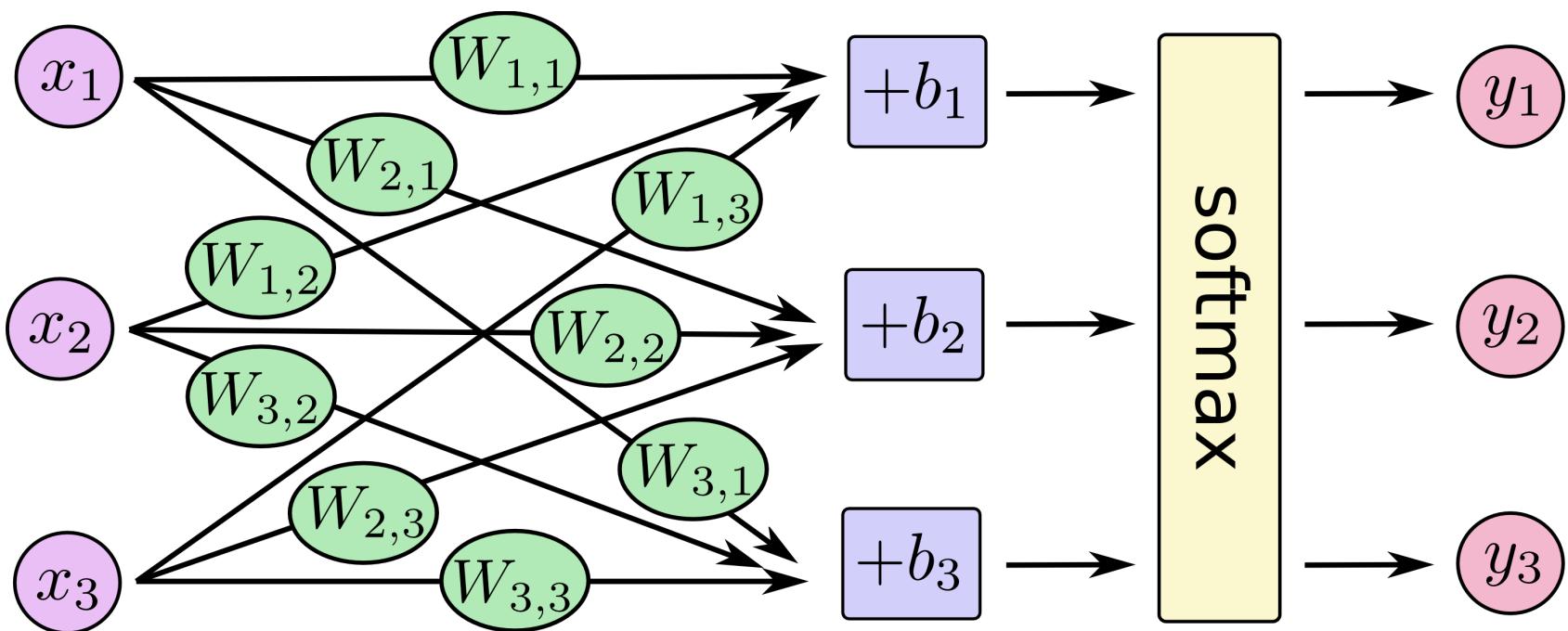
	cão	gato	rato
	0	1	0
	1	0	0
	0	0	1
	0	1	0

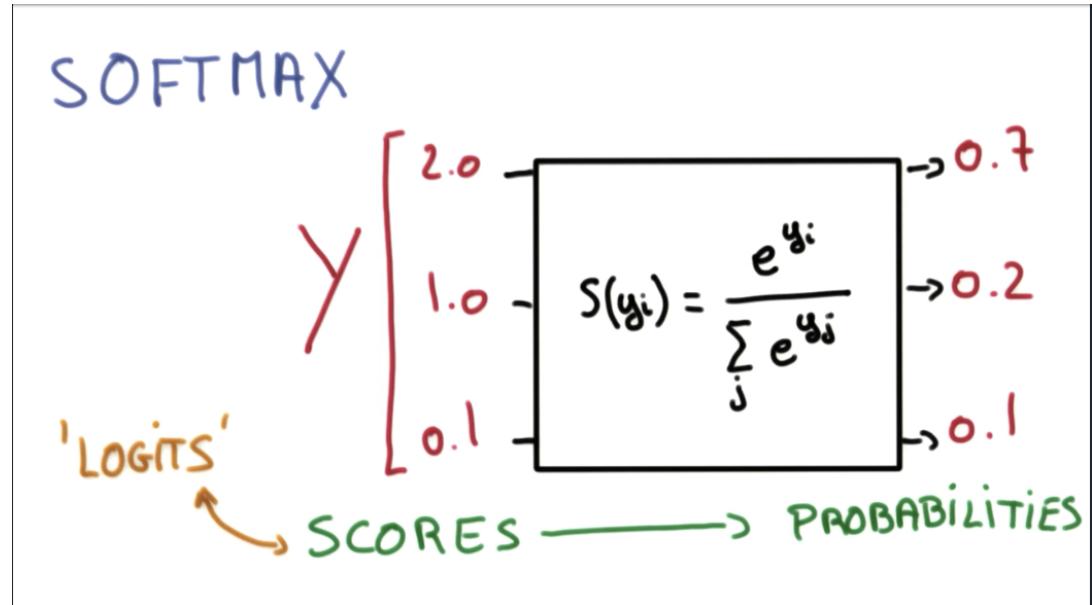


$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \left( \begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

$$y = \text{softmax}(Wx + b)$$

E... o que é esse softmax?

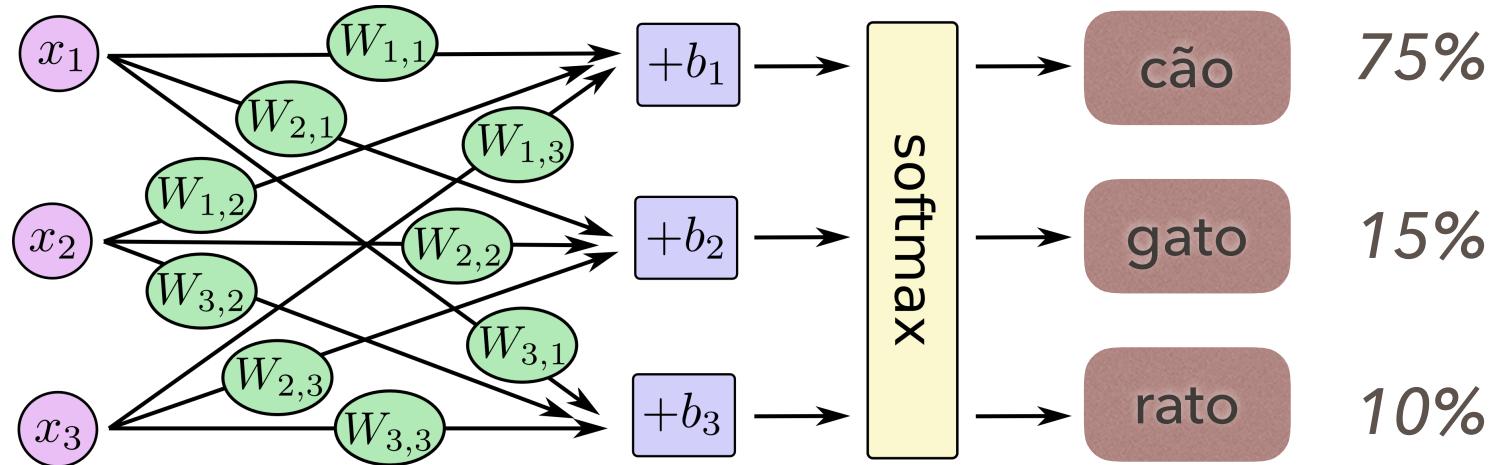
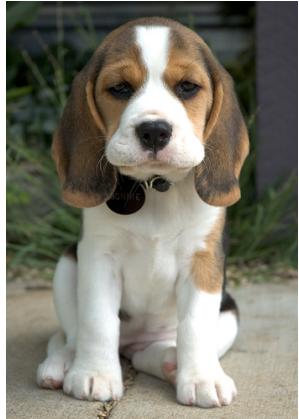




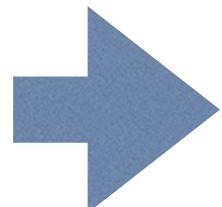
Fonte: Udacity

	scores	e^scores	probabilidades
y_0	2	7.4	0.7
y_1	1	2.7	0.2
y_2	0.1	1.1	0.1
fator de normalização →		11.2	

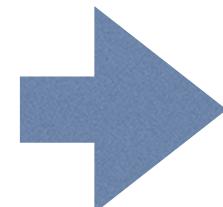
## ESSE MODELO É “BOM”?



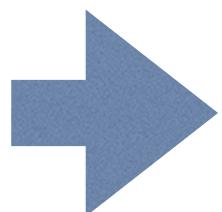
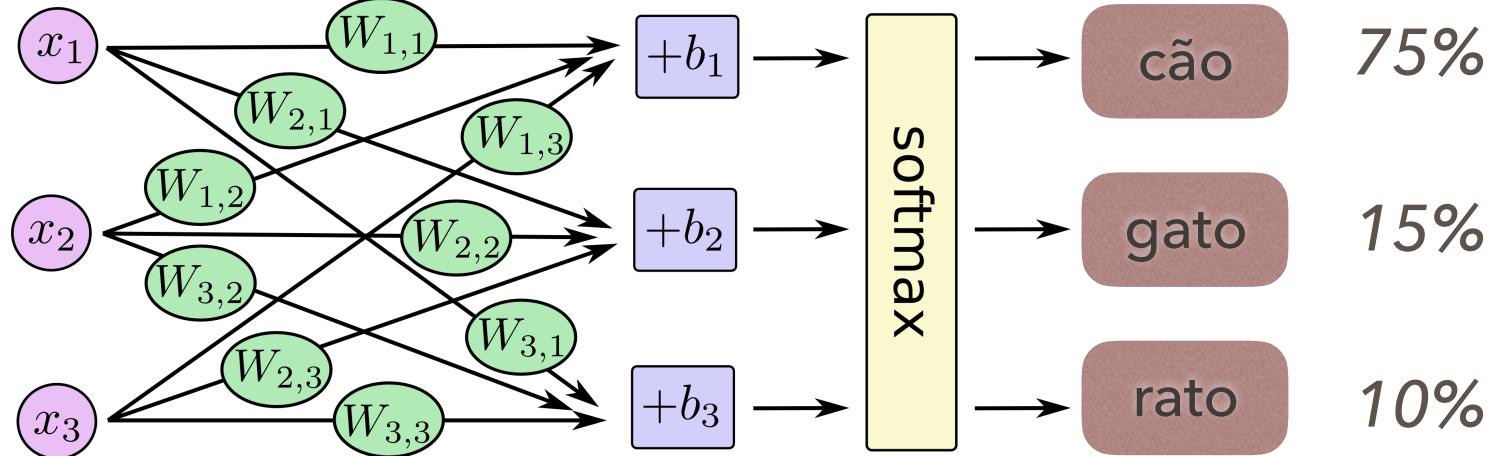
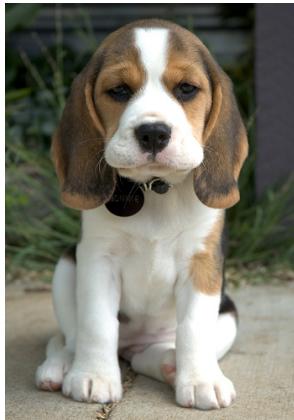
*errra menos  
acerta mais*



*“bons”  
 $W$  &  $b$*



*“bom”  
modelo*

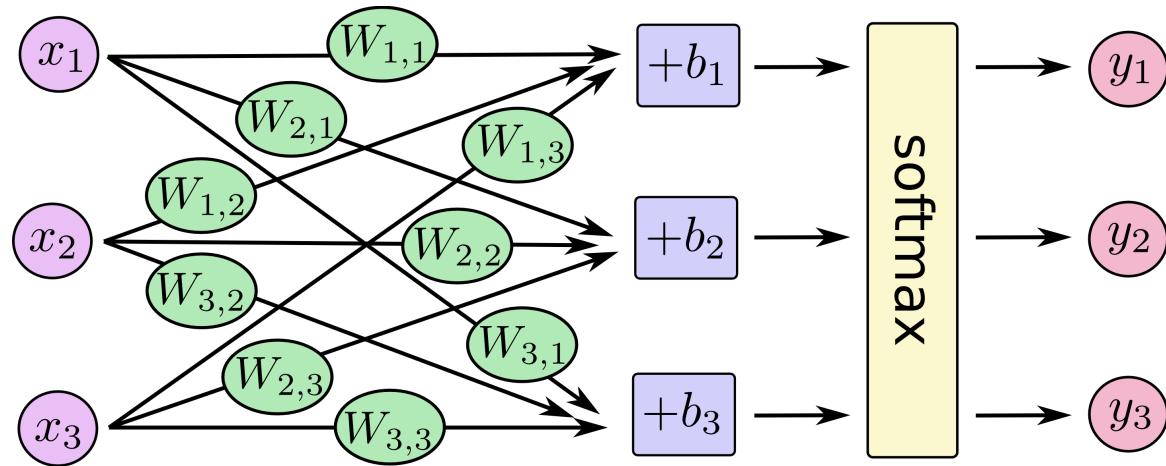


## GRADIENT DESCENT

[HTTP://WWW.KDNUGGETS.COM/2017/04/SIMPLE-UNDERSTAND-GRADIENT-DESCENT-ALGORITHM.HTML](http://www.kdnuggets.com/2017/04/simple-understand-gradient-descent-algorithm.html)

<https://docs.google.com/spreadsheets/d/1TFHcFix5zN5ikqFkJC6mlXg83LxGJAeVJl3BddvYsPM/edit?usp=sharing>

# como quantificar “BOM”???



$X$

$y$

$y\_pred$

	pixel1	pixel2	pixel3	...	pixel n	cão	gato	rato	cão	gato	rato
	2	133	25	...	77	0	1	0	0.06	0.85	0.09
	3	3	55	...	89	1	0	0	0.77	0.20	0.03
	254	255	253	...	255	0	0	1	0.32	0.15	0.53
	8	8	10	...	21	0	1	0	0.40	0.20	0.30

# como quantificar “BOM”???

*X*

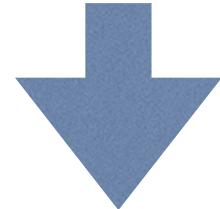
	pixel1	pixel2	pixel3	...	pixel n
	2	133	25	...	77
	3	3	55	...	89
	254	255	253	...	255
	8	8	10	...	21

*y*

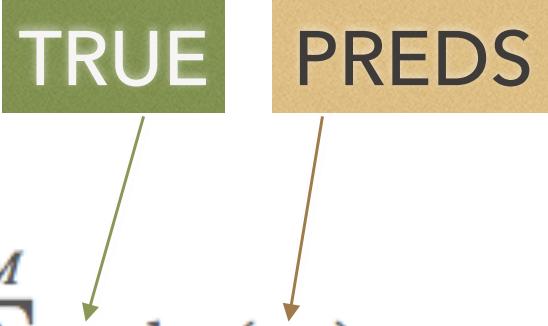
cão	gato	rato
0	1	0
1	0	0
0	0	1
0	1	0

*y\_pred*

cão	gato	rato
0.06	0.85	0.09
0.77	0.20	0.03
0.32	0.15	0.53
0.40	0.20	0.30



CROSS-ENTROPY



The diagram illustrates the components of the logloss formula. Two boxes are shown: a green box labeled "TRUE" and a yellow box labeled "PREDS". Two arrows point from these boxes to the corresponding variables in the formula: one arrow points from "TRUE" to  $y_{i,j}$ , and another arrow points from "PREDS" to  $p_{i,j}$ .

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

<https://www.kaggle.com/wiki/LogLoss>

**PREDS**

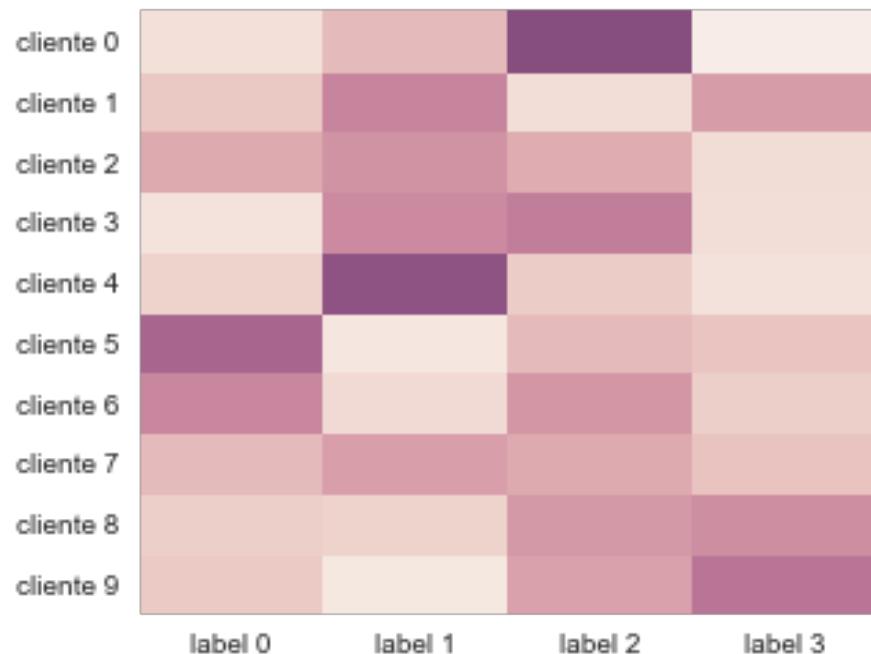
	label 0	label 1	label 2	label 3
<b>cliente 0</b>	0.069	0.213	0.698	0.020
<b>cliente 1</b>	0.164	0.426	0.081	0.329
<b>cliente 2</b>	0.278	0.370	0.269	0.083
<b>cliente 3</b>	0.055	0.405	0.460	0.080
<b>cliente 4</b>	0.124	0.666	0.145	0.066
<b>cliente 5</b>	0.569	0.045	0.214	0.172
<b>cliente 6</b>	0.421	0.091	0.353	0.135
<b>cliente 7</b>	0.215	0.323	0.279	0.182
<b>cliente 8</b>	0.139	0.123	0.347	0.390
<b>cliente 9</b>	0.155	0.036	0.311	0.498

**TRUE**

	label 0	label 1	label 2	label 3
<b>cliente 0</b>	0	1	0	0
<b>cliente 1</b>	0	0	0	1
<b>cliente 2</b>	1	0	0	0
<b>cliente 3</b>	0	0	0	1
<b>cliente 4</b>	0	1	0	0
<b>cliente 5</b>	0	0	1	0
<b>cliente 6</b>	0	0	1	0
<b>cliente 7</b>	0	0	1	0
<b>cliente 8</b>	1	0	0	0
<b>cliente 9</b>	0	0	0	1

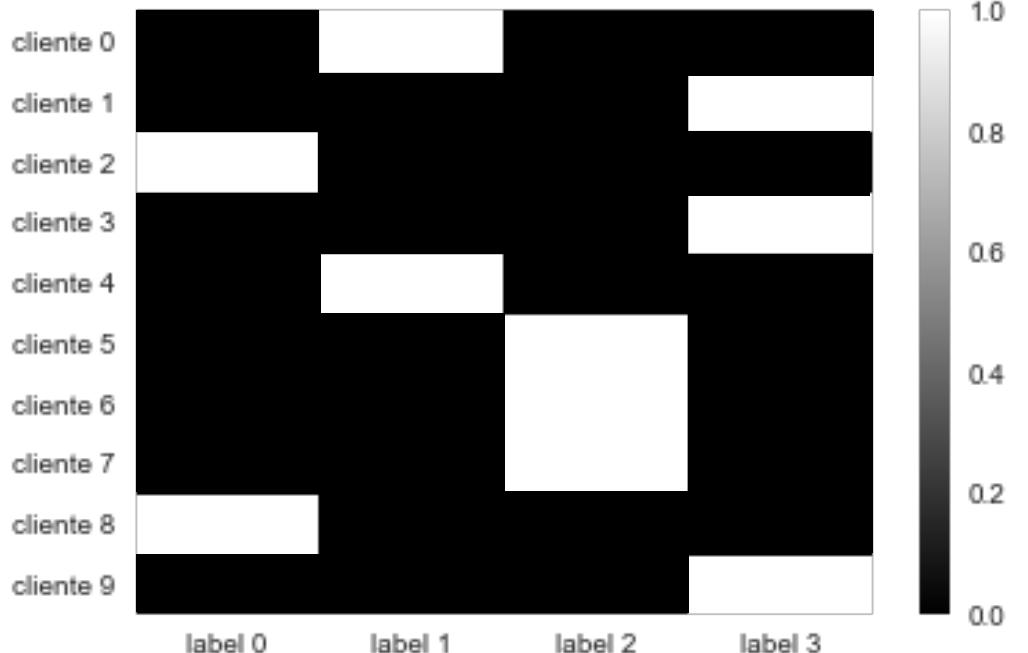
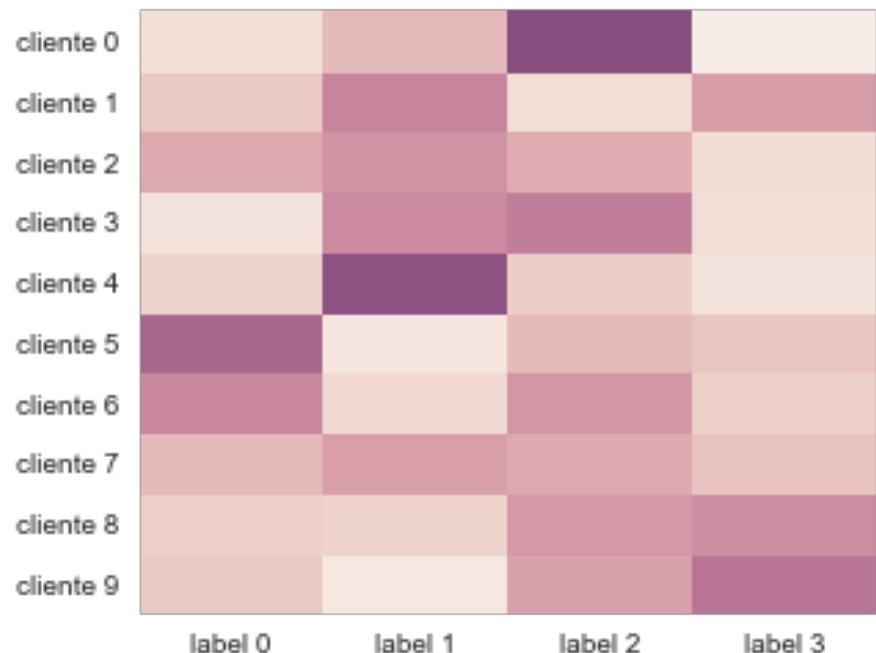
PREDS

TRUE



PREDS

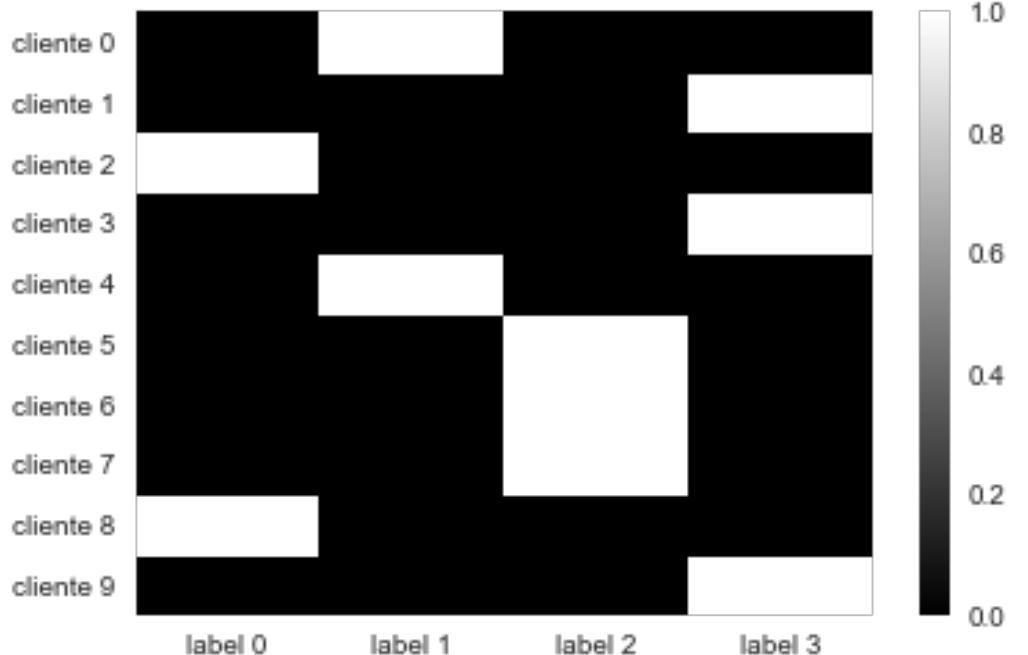
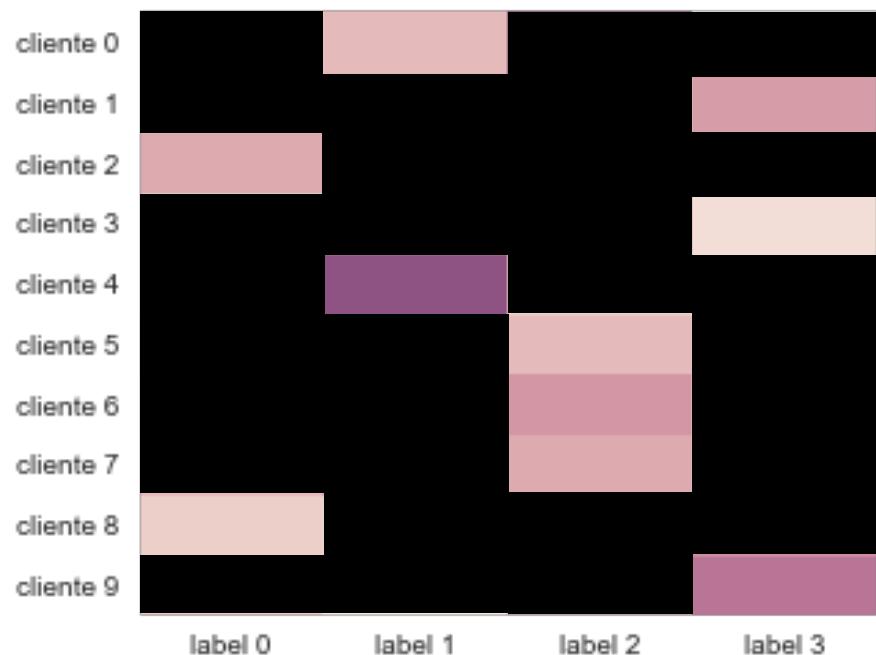
TRUE



$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

PREDS

TRUE



$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

$y$

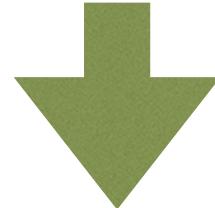
$y_{pred}$

cão	gato	rato	cão	gato	rato
0	1	0	0.06	0.85	0.09
1	0	0	0.77	0.20	0.03
0	0	1	0.32	0.15	0.53
0	1	0	0.40	0.20	0.40

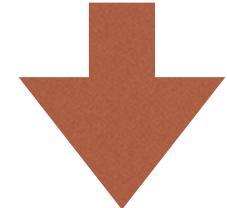
$y$

$y_{pred}$

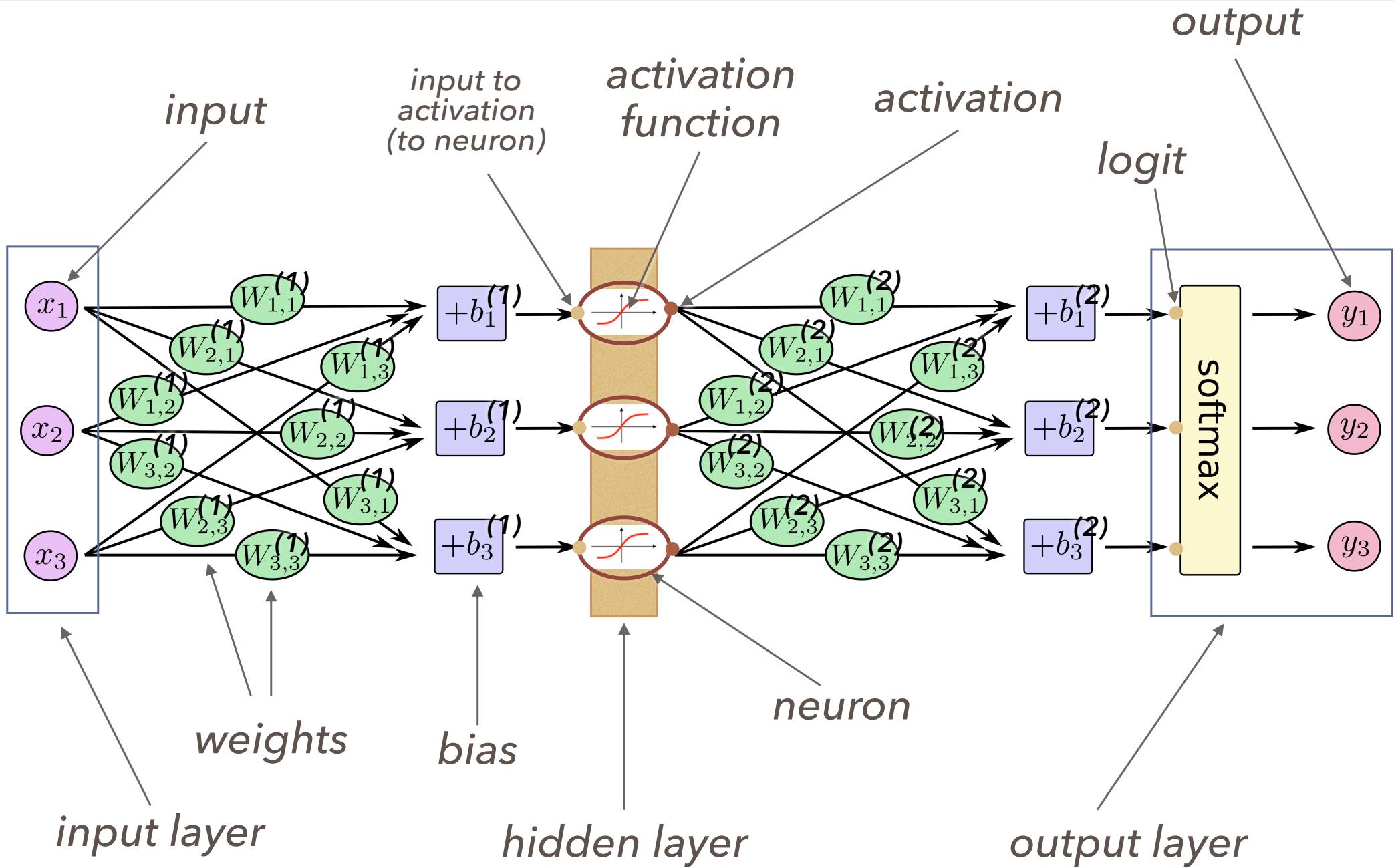
cão	gato	rato	cão	gato	rato
0	1	0	0.60	0.15	0.25
1	0	0	0.24	0.20	0.56
0	0	1	0.32	0.03	0.65
0	1	0	0.40	0.15	0.45

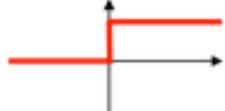
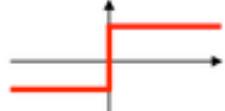
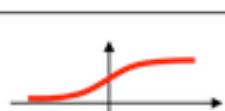


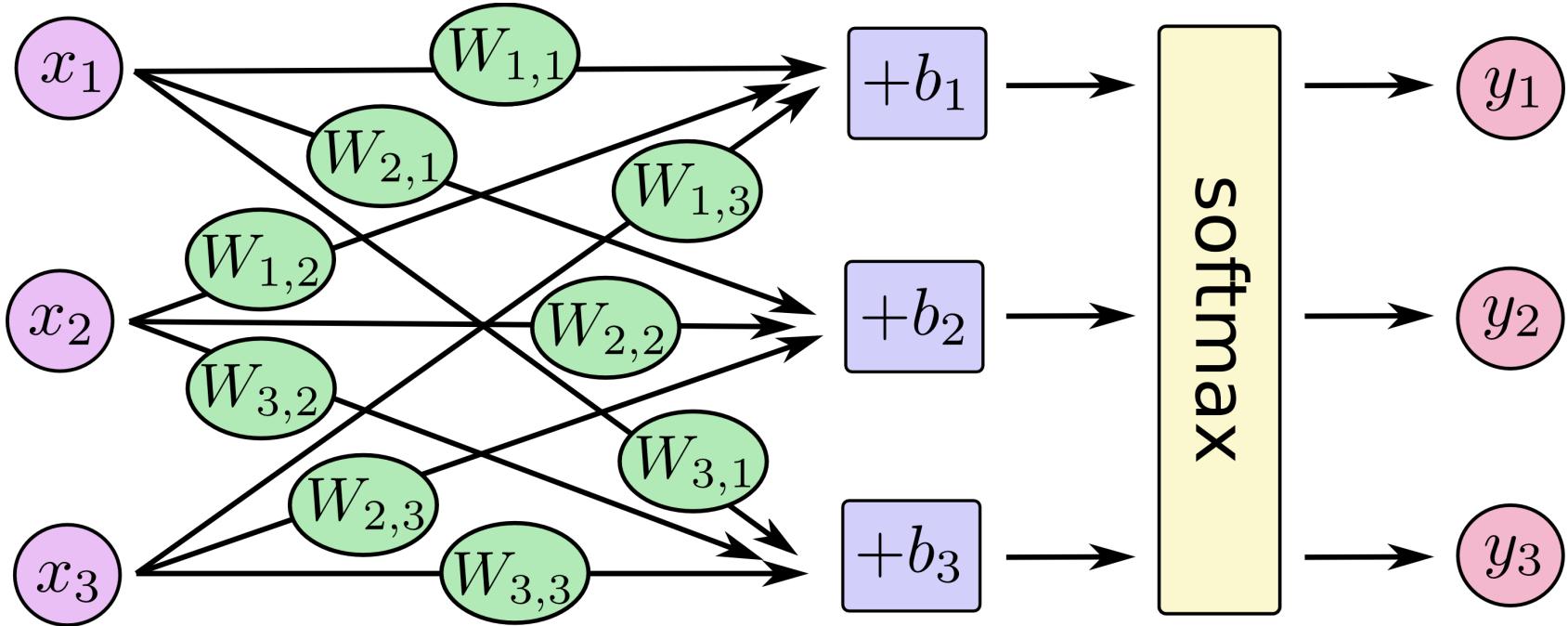
CROSS-ENTROPY  
MENOR



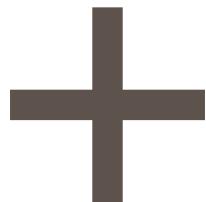
CROSS-ENTROPY  
MAIOR

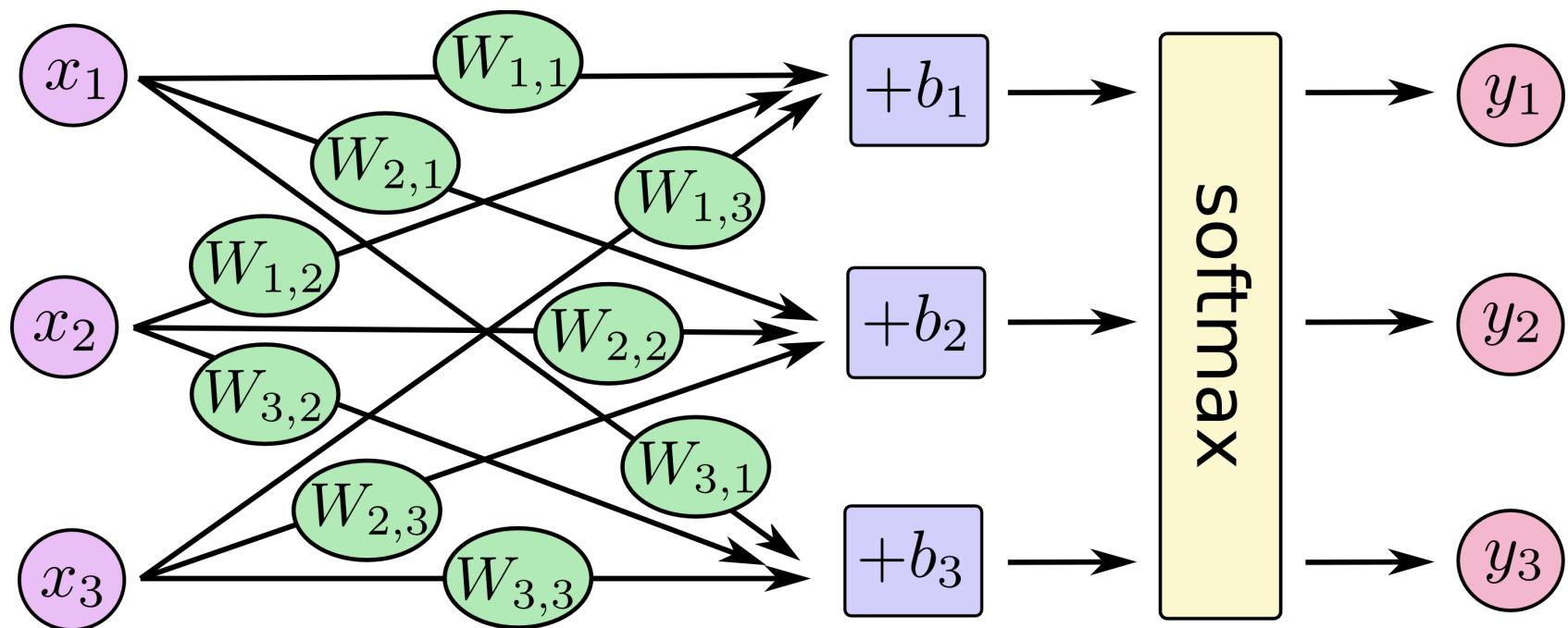


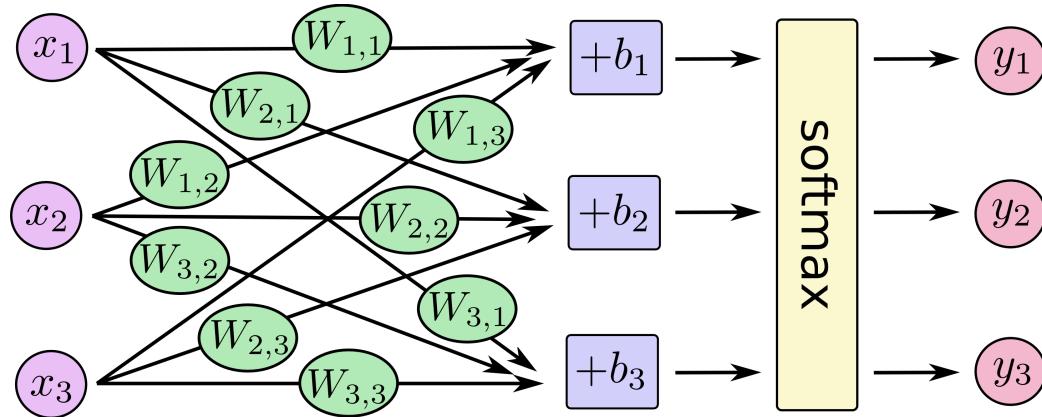
Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer NN	



*Este setup também é conhecido por  
MULTINOMIAL LOGISTIC CLASSIFICATION*







- arquitetura:
  - dimensão da entrada
  - dimensão da saída  
(numero de classes)
  - dimensões escondidas
    - # de pesos & bias
  - função de ativação
- operação:
  - tamanho do batch
  - qtd de epochs
- objetivos:
  - função de custo & métrica
  - otimizador

DADOS:

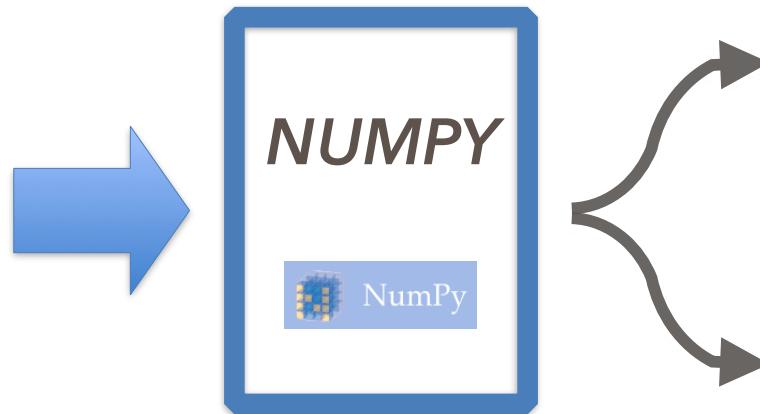
CSV

SQL

...



*indices  
slicing, indexing  
limpeza  
get\_dummies  
merge, concat  
groupby*



*algebra linear  
multiplicacao matrizes  
prod. interno  
transform. linear*



scikit  
learn

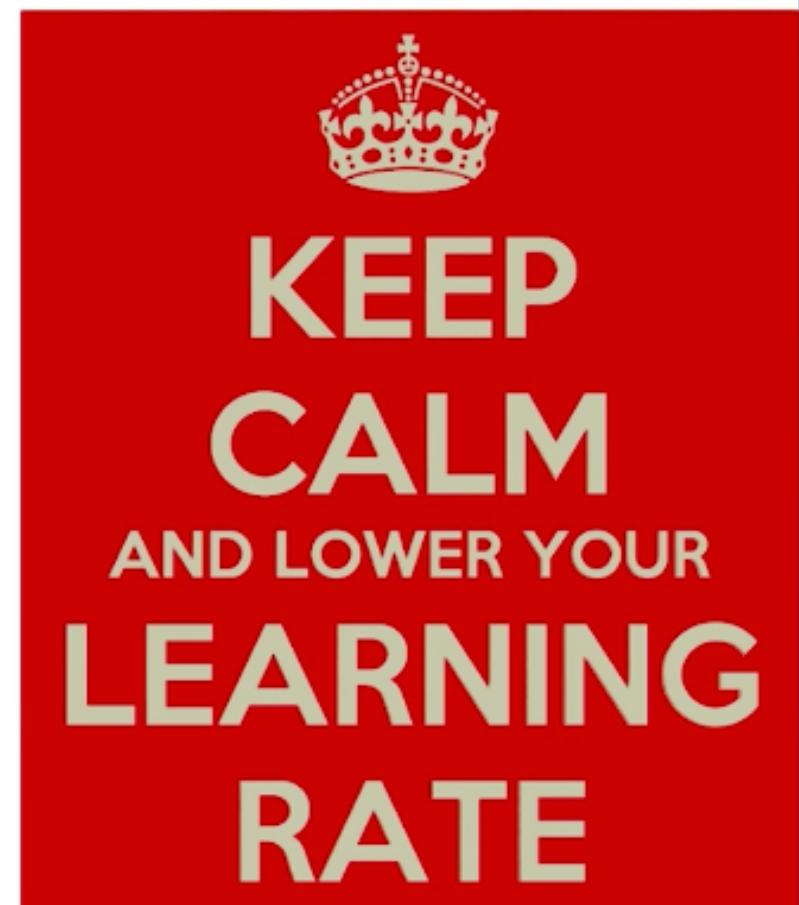
TensorFlow

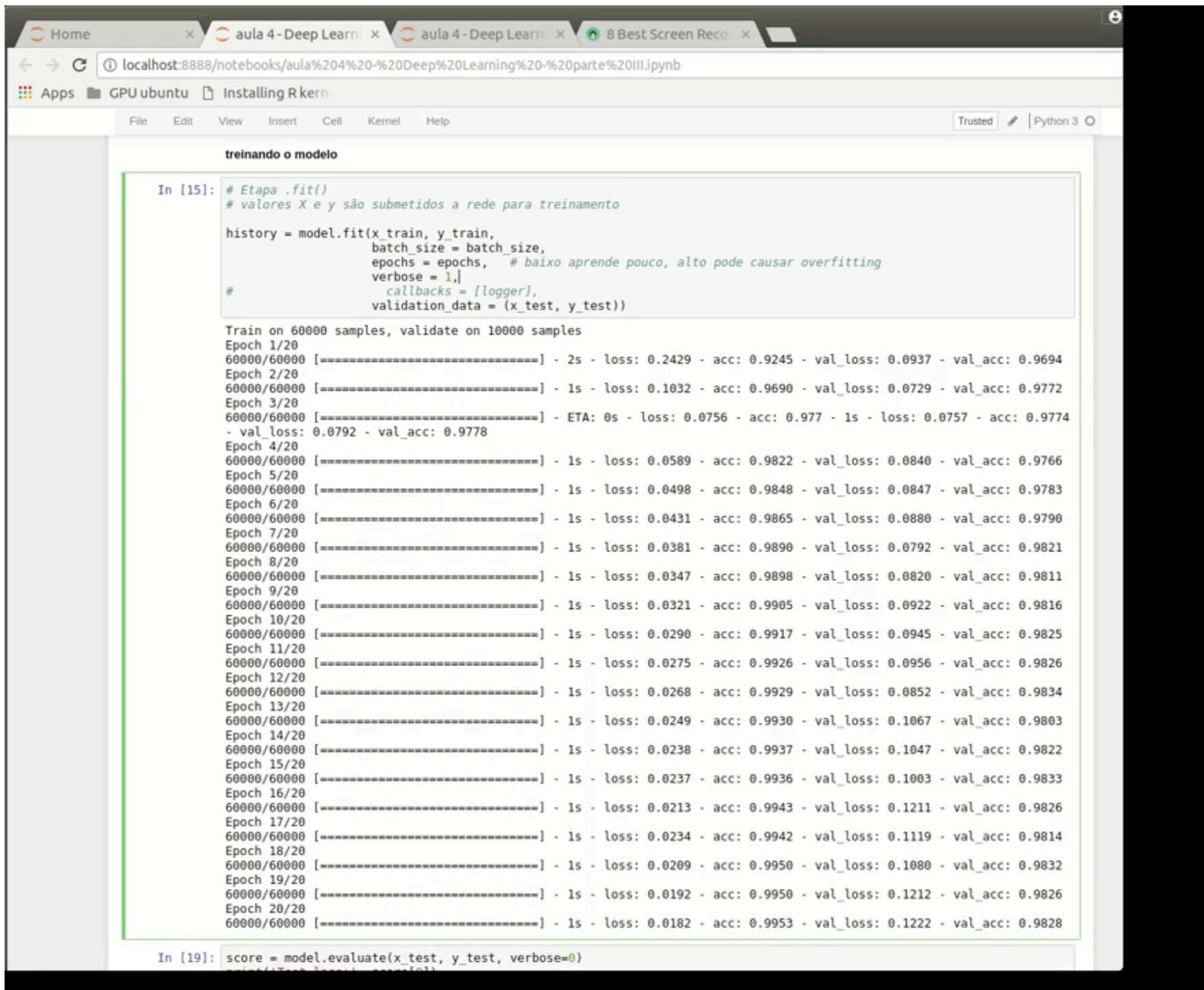


## SGD 'BLACK MAGIC'

MANY HYPER-PARAMETERS

- INITIAL LEARNING RATE
- LEARNING RATE DECAY
- MOMENTUM
- BATCH SIZE
- WEIGHT INITIALIZATION





A screenshot of a Jupyter Notebook interface. The title bar shows three tabs: 'aula 4-Deep Learn', 'aula 4-Deep Learn', and '8 Best Screen Reco'. The notebook file path is 'localhost:8888/notebooks/aula%204%20-%20Deep%20Learning%20-%20parte%20III.ipynb'. The toolbar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. The status bar indicates 'Trusted' and 'Python 3'. The main area contains code for training a model and its output logs.

**treinando o modelo**

```
In [15]: # Etapa .fit()
# valores X e y são submetidos a rede para treinamento

history = model.fit(x_train, y_train,
                     batch_size = batch_size,
                     epochs = epochs, # baixo aprende pouco, alto pode causar overfitting
                     verbose = 1,
                     callbacks = [logger],
                     validation_data = (x_test, y_test))

Train on 60000 samples, validate on 10000 samples
Epoch 1/20
60000/60000 [=====] - 2s - loss: 0.2429 - acc: 0.9245 - val_loss: 0.0937 - val_acc: 0.9694
Epoch 2/20
60000/60000 [=====] - 1s - loss: 0.1032 - acc: 0.9690 - val_loss: 0.0729 - val_acc: 0.9772
Epoch 3/20
60000/60000 [=====] - ETA: 0s - loss: 0.0756 - acc: 0.977 - 1s - loss: 0.0757 - acc: 0.9774
- val_loss: 0.0792 - val_acc: 0.9778
Epoch 4/20
60000/60000 [=====] - 1s - loss: 0.0589 - acc: 0.9822 - val_loss: 0.0840 - val_acc: 0.9766
Epoch 5/20
60000/60000 [=====] - 1s - loss: 0.0498 - acc: 0.9848 - val_loss: 0.0847 - val_acc: 0.9783
Epoch 6/20
60000/60000 [=====] - 1s - loss: 0.0431 - acc: 0.9865 - val_loss: 0.0880 - val_acc: 0.9790
Epoch 7/20
60000/60000 [=====] - 1s - loss: 0.0381 - acc: 0.9890 - val_loss: 0.0792 - val_acc: 0.9821
Epoch 8/20
60000/60000 [=====] - 1s - loss: 0.0347 - acc: 0.9898 - val_loss: 0.0820 - val_acc: 0.9811
Epoch 9/20
60000/60000 [=====] - 1s - loss: 0.0321 - acc: 0.9905 - val_loss: 0.0922 - val_acc: 0.9816
Epoch 10/20
60000/60000 [=====] - 1s - loss: 0.0290 - acc: 0.9917 - val_loss: 0.0945 - val_acc: 0.9825
Epoch 11/20
60000/60000 [=====] - 1s - loss: 0.0275 - acc: 0.9926 - val_loss: 0.0956 - val_acc: 0.9826
Epoch 12/20
60000/60000 [=====] - 1s - loss: 0.0268 - acc: 0.9929 - val_loss: 0.0852 - val_acc: 0.9834
Epoch 13/20
60000/60000 [=====] - 1s - loss: 0.0249 - acc: 0.9930 - val_loss: 0.1067 - val_acc: 0.9803
Epoch 14/20
60000/60000 [=====] - 1s - loss: 0.0238 - acc: 0.9937 - val_loss: 0.1047 - val_acc: 0.9822
Epoch 15/20
60000/60000 [=====] - 1s - loss: 0.0237 - acc: 0.9936 - val_loss: 0.1003 - val_acc: 0.9833
Epoch 16/20
60000/60000 [=====] - 1s - loss: 0.0213 - acc: 0.9943 - val_loss: 0.1211 - val_acc: 0.9826
Epoch 17/20
60000/60000 [=====] - 1s - loss: 0.0234 - acc: 0.9942 - val_loss: 0.1119 - val_acc: 0.9814
Epoch 18/20
60000/60000 [=====] - 1s - loss: 0.0209 - acc: 0.9950 - val_loss: 0.1080 - val_acc: 0.9832
Epoch 19/20
60000/60000 [=====] - 1s - loss: 0.0192 - acc: 0.9950 - val_loss: 0.1212 - val_acc: 0.9826
Epoch 20/20
60000/60000 [=====] - 1s - loss: 0.0182 - acc: 0.9953 - val_loss: 0.1222 - val_acc: 0.9828
```

```
In [19]: score = model.evaluate(x_test, y_test, verbose=0)
          print('Test accuracy:', score[1])
```



# SISTEMAS DE RECOMENDAÇÃO



amazon  
Prime

Books ▾



WORLD BOOK DAY

#Love

Departments ▾

Browsing History ▾

Hitoshi's Amazon.com

Today's Deals

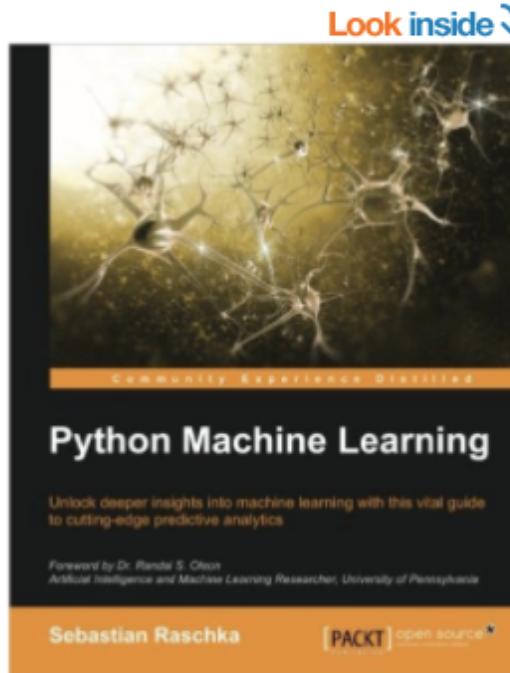
EN  
🌐 ▾Hello, Hitoshi  
Account & Lists ▾

Orders

Prime ▾

Books Advanced Search New Releases Best Sellers The New York Times® Best Sellers Children's Books Textbooks Textbook Rentals Sell Us Your Books

Books &gt; Computers &amp; Technology &gt; Computer Science



Look inside ↓

## Python Machine Learning

Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics

Forward by Dr. Randal S. Olson  
Artificial Intelligence and Machine Learning Researcher, University of Pennsylvania

Sebastian Raschka

PACKT open source\*

Flip to back



See all 3 images

# Python Machine Learning Paperback – September 23,

2015

by Sebastian Raschka ▾ (Author)

★★★★★ 5 ★ 101 customer reviews

▶ See all 2 formats and editions

Kindle

\$31.56

Paperback

\$40.49 ✓Prime

Read with Our Free App

38 Used from \$28.75

24 New from \$35.00

Unlock deeper insights into Machine Learning with this vital guide to cutting-edge predictive analytics

## About This Book

- Leverage Python's most powerful open-source libraries for deep learning, data wrangling, and data visualization
- Learn effective strategies and best practices to improve and optimize machine learning systems and algorithms
- Ask – and answer – tough questions of your data with robust statistical

▼ Read more

Report incorrect product information.

Share

 Buy New

Qty: 1

List Price

Save: \$

FREE Shipping for Prime members once available [Details](#)**In Stock.**Ships from and sold by Amazon.com  
Gift-wrap available.

Add to Cart

Turn on 1-Click ordering for this item

Want it Tuesday, April 25? Order within **7 hrs 5 mins** and choose **Day Shipping** at checkout. Details

Ship to:

Hitoshi Nagano- Eden Prairie, MN 55344



Buy Used ✓Prime

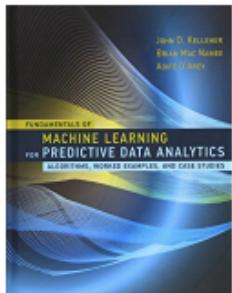
Add to List

World Book Day is April 23

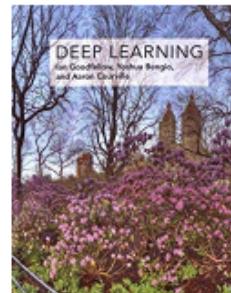
Join us in celebrating reading! [Learn more](#)

## Customers who bought this item also bought

Page 1 of 25



Fundamentals of Machine Learning for Predictive Data Analytics:...  
 > John D. Kelleher  
 ★★★★★ 23  
 Hardcover  
 \$70.00 ✓Prime



Deep Learning (Adaptive Computation and Machine Learning series)  
 > Ian Goodfellow  
 ★★★★★ 63  
 Hardcover  
 \$72.00 ✓Prime



Data Science from Scratch: First Principles with Python  
 > Joel Grus  
 ★★★★★ 80  
**#1 Best Seller** in Data Mining  
 Paperback  
 \$28.97 ✓Prime



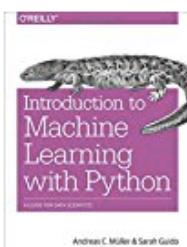
Hands-On Machine Learning with Scikit-Learn and TensorFlow:...  
 > Aurélien Géron  
 ★★★★★ 3  
**#1 Best Seller** in Computer Vision & Pattern...  
 Paperback  
 \$30.10 ✓Prime



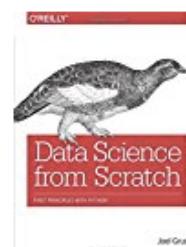
## Your recently viewed items and featured recommendations

Inspired by your browsing history

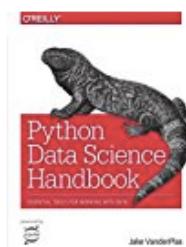
Page 1 of 10



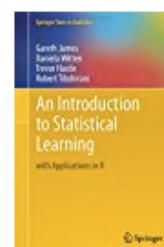
Introduction to Machine Learning with Python:...  
 Andreas C. Müller  
 ★★★★★ 14  
 Paperback  
 \$34.38 ✓Prime



Data Science from Scratch: First...  
 > Joel Grus  
 ★★★★★ 80  
 Paperback  
 \$28.97 ✓Prime



Python Data Science Handbook: Essential...  
 > Jake VanderPlas  
 ★★★★★ 12  
 Paperback  
 \$45.47 ✓Prime



An Introduction to Statistical Learning with Applications in R  
 > Gareth James  
 ★★★★★ 139  
 Hardcover  
 \$70.35 ✓Prime



Practical Statistics for Data Scientists: 50 Essential Concepts  
 > Peter Bruce  
 Paperback  
 \$22.85 ✓Prime

- técnicas, ferramentas e sistemas...
- ... para sugerir ítems a pessoas

- aumentar vendas - volume
- aumentar vendas - diversidade
- aumentar satisfação
- aumentar fidelização
- entender melhor o que o cliente quer

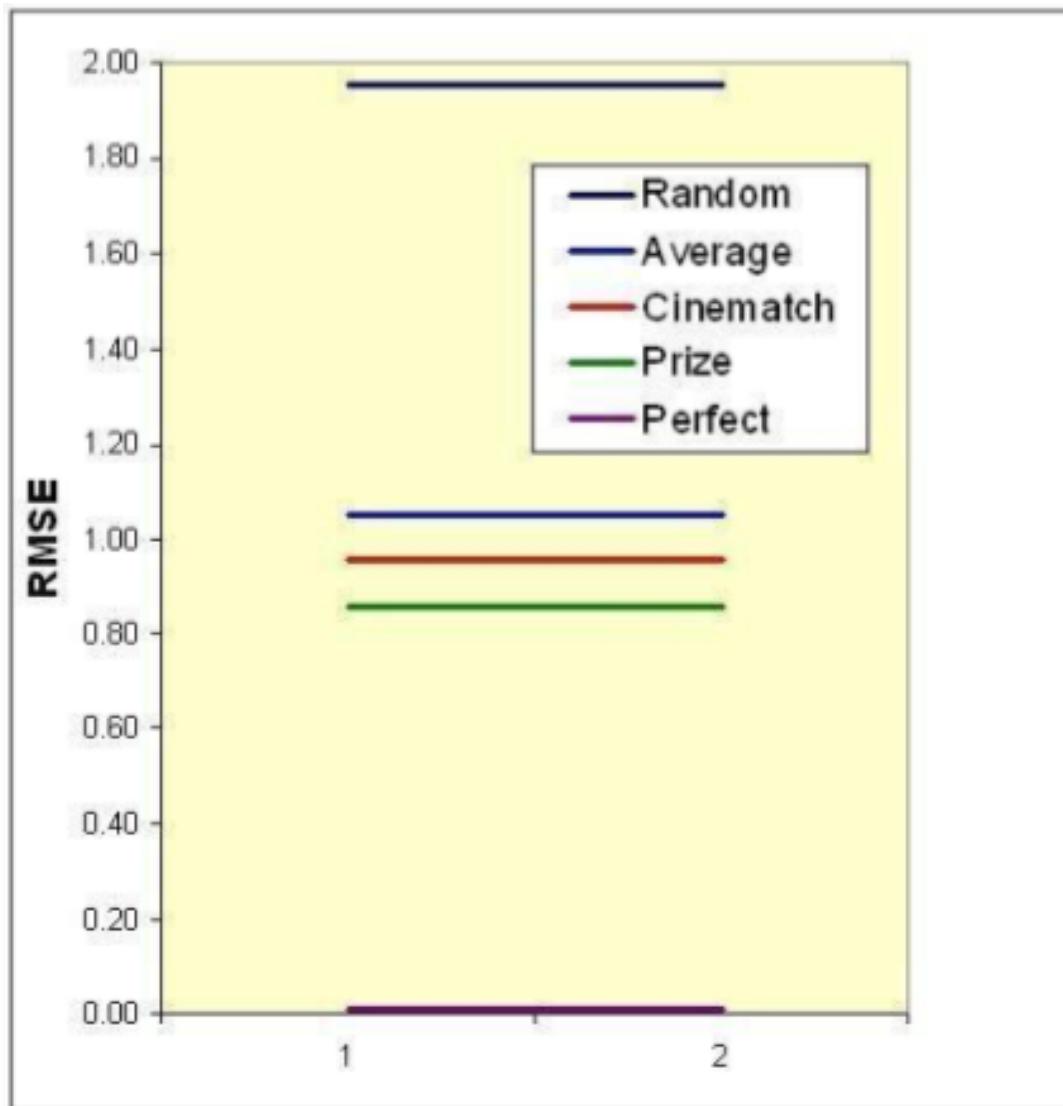
- **Cauda Longa:** “We are leaving the age of information and entering the age of recommendation” - Chris Anderson: “The Long Tail...”
- **Big Data:** “People read 10MB a day, hear 400MB a day and see 1MB every second” - The economist Nov/2006
- **Paradoxo da Escolha:** “I want a pair of jeans”; “Do you want stone-washed, acid-washed, or distressed? Do you want button-fly or zipper-fly? Regular or faded? - Barry Schwartz : “Paradox of choice, why more is less”

- **Netflix:** mais de 60% dos filmes assistidos foram recomendados
- **Google news:** SR geram 38% mais *clickthrough*
- **Amazon:** 35% das vendas vem dos SR

- recomendação: item → pessoa
- busca: pessoa → item
- a recomendação “acha” a pessoa
- busca e recomendação: dois lados de uma mesma moeda

- Função que prediz se uma pessoa vai gostar de um item
- dados:
  - similaridade: outras pessoas & feedback passado
  - similaridade: outros itens & feedback passado
  - contexto
  - ...

## complexidade VS resultado



# tipos de dados (expressando preferências)

Avaliações

4.8 ★★★★★ (5)      100% dos clientes recomendam este produto      Avaliar      avaliações mais recentes ▾

Muito bom!!  
★★★★★  
De fácil manuseio, e bem simples para quem está começando o Inglês :) Além da rapidez na entrega :D  
Luiz Phelipe      25/05/2015

Valeu a pena ter comprado  
★★★★★  
Era um produto do jeito que eu esperava e chegou antes da data  
Jessicaprof      18/04/2015

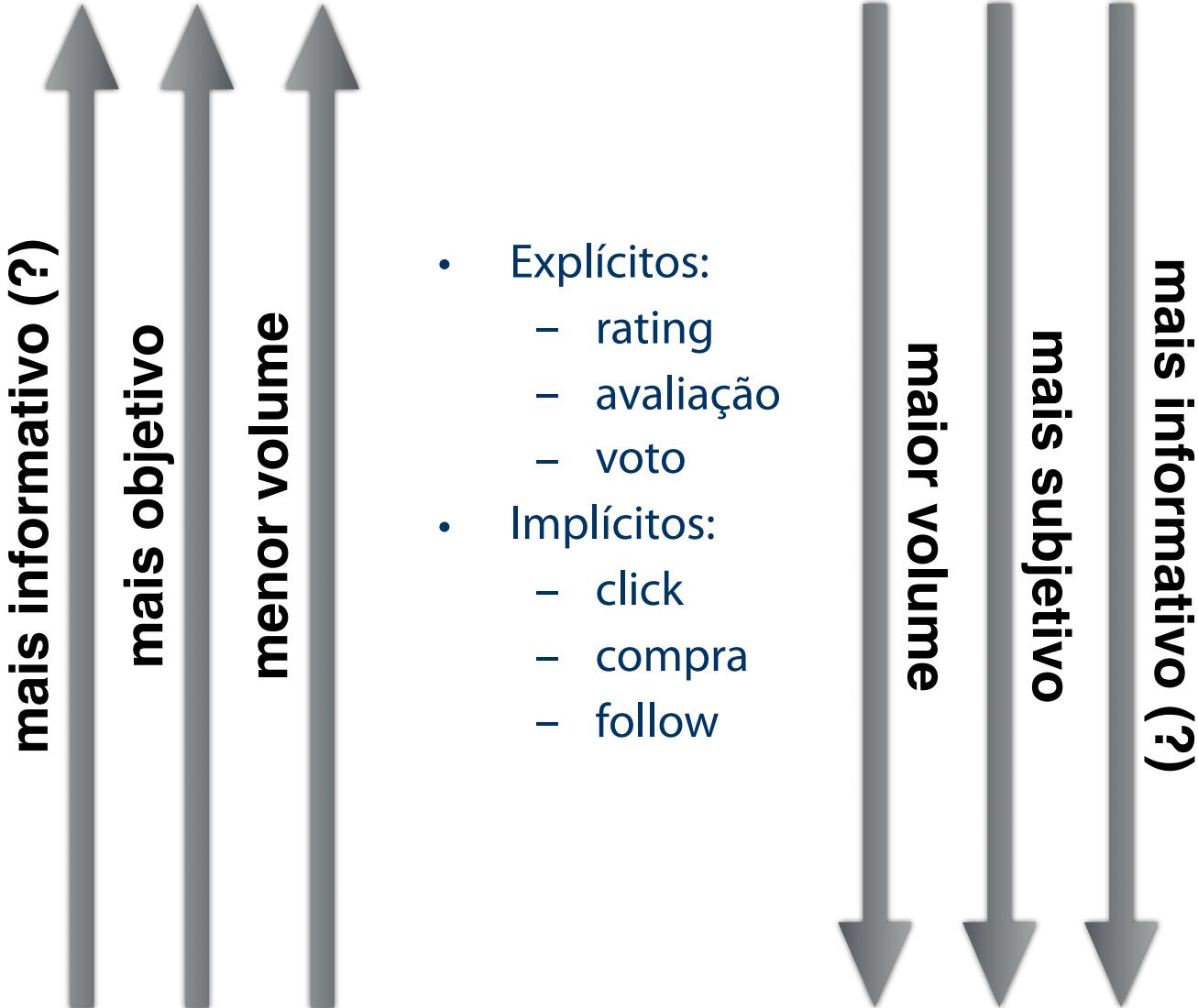
22

I'm considering MongoDB for my next big project, how can I do reporting?

My understanding is that I can't do the same kinds relational database. The reporting I had in mind in "tables" with strict criteria.

3

Is this easily doable in MongoDB, or is it going to



- qual é melhor ?
- para recomendações
- explicito: avaliações  
implícito: intenções
- gostar != recomendar

 Clip slide

## Approaches to Recommendation

- non-personalized
- Collaborative Filtering: Recommend items based only on the users past behavior
  - **User-based:** Find similar users to me and recommend what they liked
  - **Item-based:** Find similar items to those that I have previously liked
- Content-based: Recommend based on item features

Clip slide

## What works

- Depends on the **domain** and particular **problem**
- However, in the general case it has been demonstrated that the best isolated approach is CF.
  - Other approaches can be hybridized to improve results in specific cases (cold-start problem...)
- What matters:
  - **Data preprocessing**: outlier removal, denoising, removal of global effects (e.g. individual user's average)
  - “Smart” **dimensionality reduction** using MF/SVD
  - **Combining methods**



Xavier Amatriain – July 2014 – Recommender Systems

 Clip slide

## Serendipity

- Unsought finding
- Don't recommend items the user already knows or **would have found anyway.**
- Expand the user's taste into neighboring areas by improving the obvious
- Collaborative filtering can offer controllable serendipity (e.g. controlling how many neighbors to use in the recommendation)



Xavier Amatriain – July 2014 – Recommender Systems

**relembrando**

**Algebra Linear**



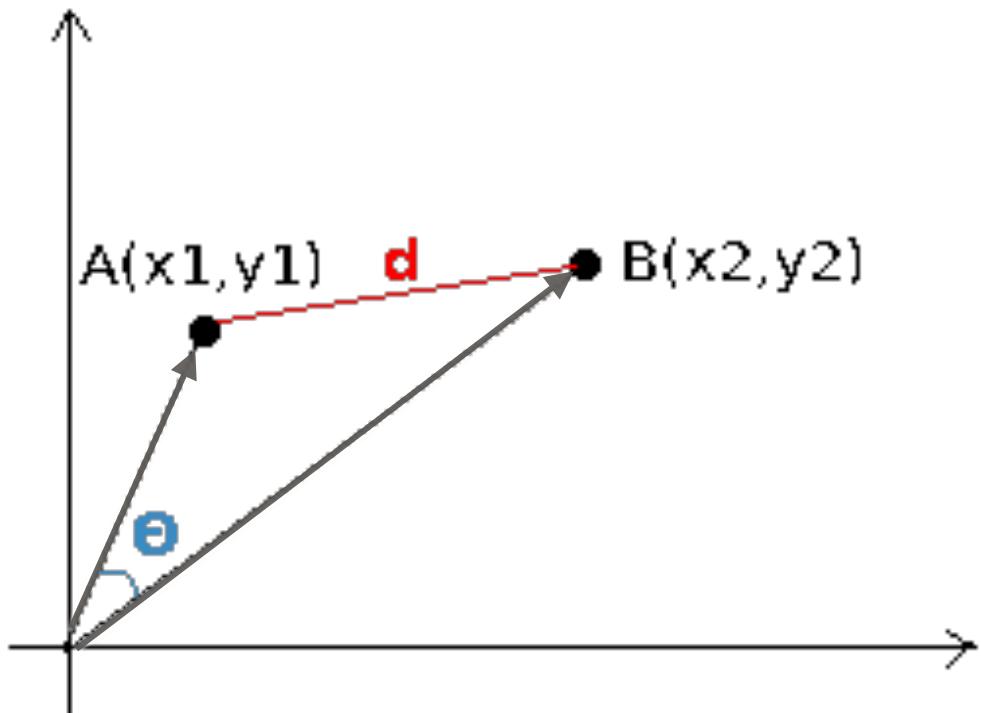
- para recomendar, podemos encontrar:
  - alguém similar
  - produto similar
- ... e fazer a recomendação
- como medimos similaridade?

- distância euclideana
- cosine similarity (similaridade de cosseno)
- pearson's r

- vetor
- produto interno
- angulo e cosseno do angulo
- distancia euclideana
- norma L2

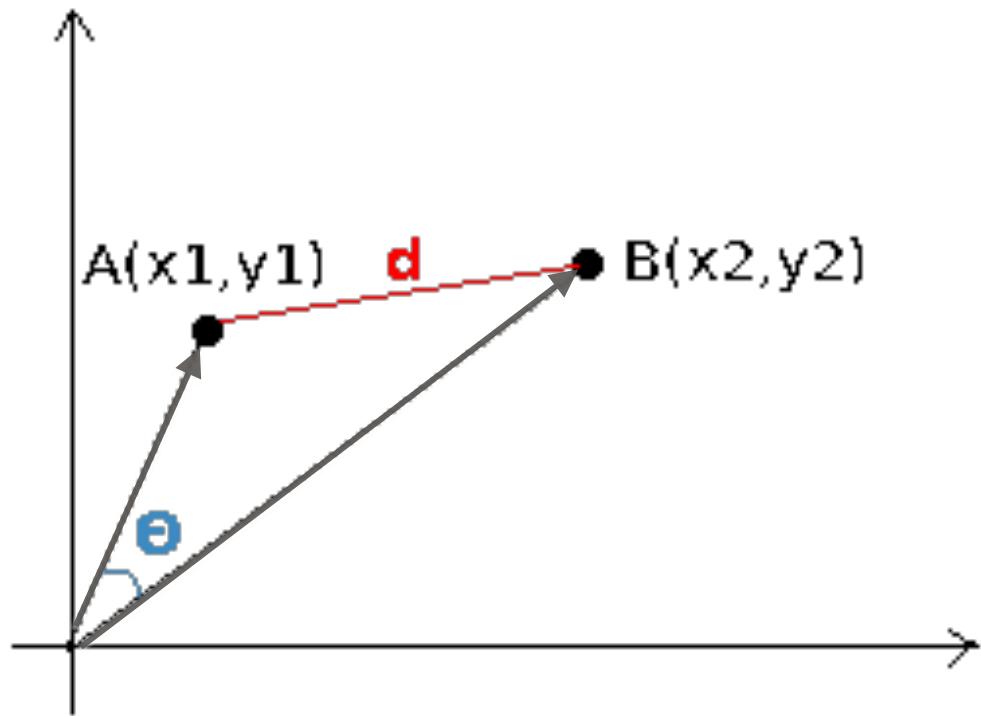
# algebra linear !

## Distancia euclideana



	x	y
A	1	3
B	4	4

# algebra linear ! similaridade cosseno



	x	y
A	1	3
B	4	4



	<b>x</b>	<b>y</b>	<b>z</b>
<b>A</b>	1	3	1
<b>B</b>	4	4	6
<b>C</b>	2	2	3

## DISTANCIA EUCLIDEANA

$$d(x,y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

## SIMILARIDADE COSSENO

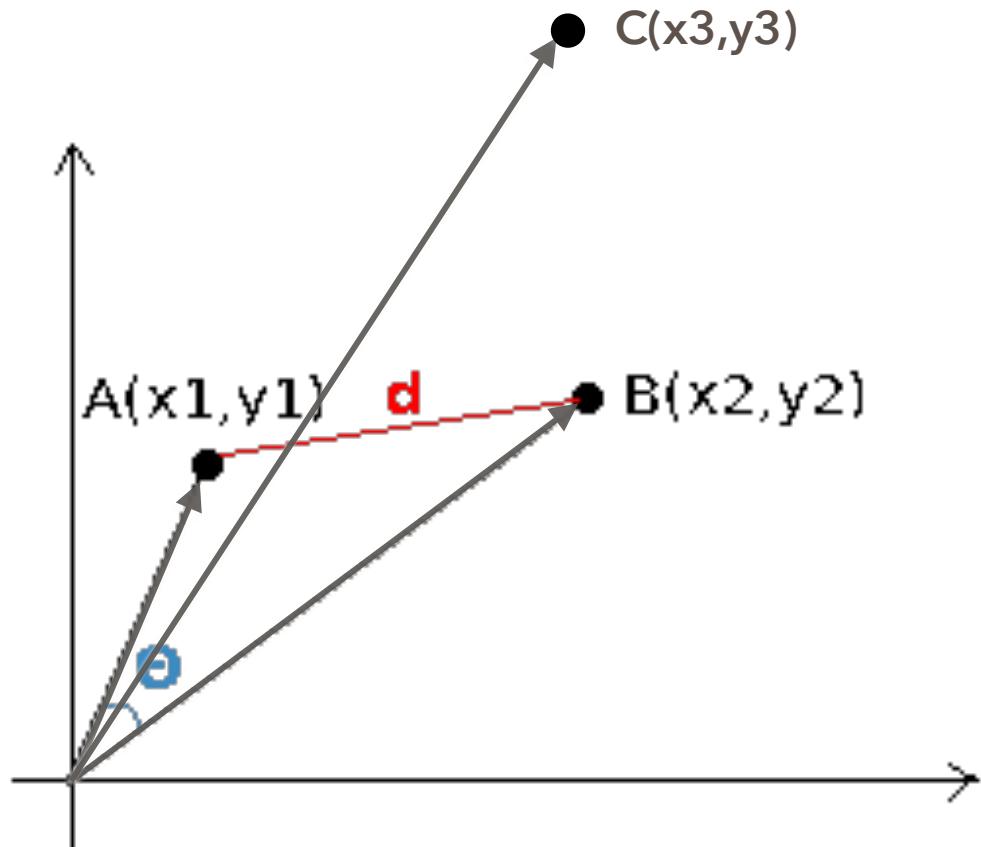
$$\cos(\theta) = \frac{\sum x_i * y_i}{\sqrt{\sum x_i^2} * \sqrt{\sum y_i^2}}$$

## PEARSON'S R

$$\rho_{xy} = \frac{\sum((x_i - \bar{x}) * (y_i - \bar{y}))}{\sqrt{\sum(x_i - \bar{x})^2} * \sqrt{\sum(y_i - \bar{y})^2}}$$

# algebra linear !

## Exercicio



A	1	3
B	4	4
C	4	7



# Exercício

## Filtro Colaborativo



## UB Collaborative Filtering

- A collection of user  $u_i$ ,  $i=1, \dots, n$  and a collection of products  $p_j$ ,  $j=1, \dots, m$
- An  $n \times m$  matrix of ratings  $v_{ij}$ , with  $v_{ij} = ?$  if user  $i$  did not rate product  $j$
- Prediction for user  $i$  and product  $j$  is computed

$$v_{ij}^* = K \sum_{v_{kj} \neq ?} u_{jk} v_{kj} \quad \text{or} \quad v_{ij}^* = v_i + K \sum_{v_{kj} \neq ?} u_{jk} (v_{kj} - v_k)$$

- Similarity can be computed by Pearson correlation

$$u_{ik} = \frac{\sum_j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_j (v_{ij} - v_i)^2 \sum_j (v_{kj} - v_k)^2}} \quad \text{or} \quad \cos(u_i, u_j) = \frac{\sum_{k=1}^m v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^m v_{ik}^2 \sum_{k=1}^m v_{jk}^2}}$$



Xavier Amatriain – July 2014 – Recommender Systems

## User-based CF Example

[Clip slide](#)

	SHERLOCK	HOBBIT	AVENGERS	DOWNTON ABBEY	WALKING DEAD	
1	2		2	4	5	
2			4			1
3			5		2	
4		1		5		4
5			4			2
6	4	5		1		

$\text{sim}(u, v)$

NA

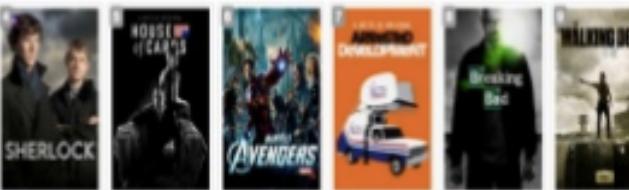
NA

## User-based CF Example

The Netflix logo, featuring the word "NETFLIX" in its signature red, bold, sans-serif font.

Xavier Amatriain – July 2014 – Recommender Systems

## User-based CF Example



sim(u,v)

	SHERLOCK	HOUSE OF CARDS	THE AVENGERS	COMMUNITY	BREAKING BAD	WALKING DEAD	sim(u,v)
1	2		2	4	5		NA
2		5	4			1	0.87
3			5		2		1
4			1	5		4	-1
5	3.51*	3.81*	4	2.42*	2.48*	2	
6	4	5		1			NA

**NETFLIX**

Xavier Amatriain – July 2014 – Recommender Systems