Universidade Federal do Paraná

Laboratório de Estatística e Geoinformação - LEG

Considerações finais

Eduardo Vargas Ferreira

Como obter boas predições

1. **Entenda os dados:** explore as características, crie gráficos para entender a natureza das variáveis etc.;

2. **Decida sobre a validação cruzada:** uma boa estratégia de validação, garante resultados mais confiáveis, p ex.,

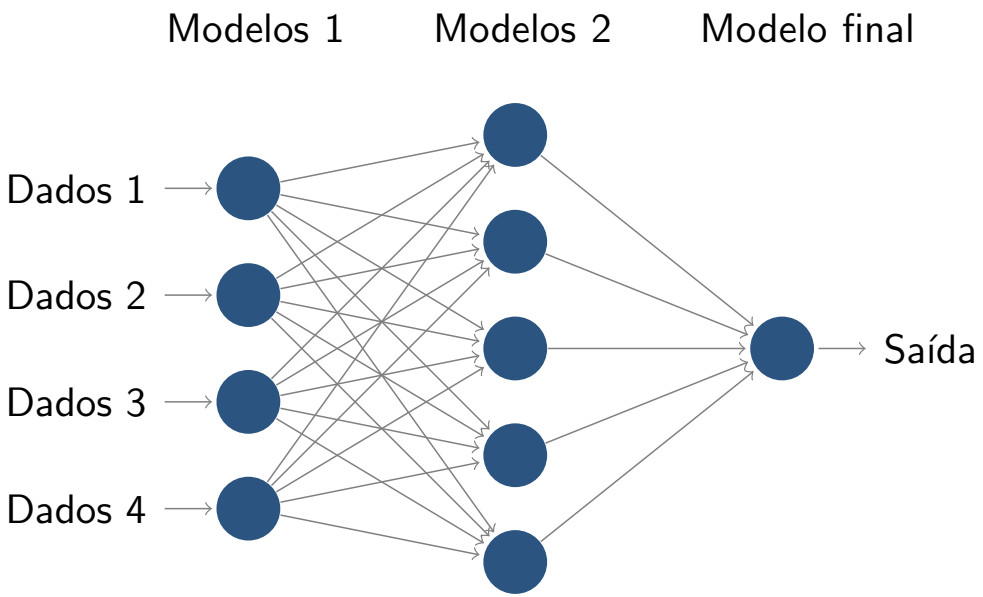
   - ⋆ Repita o processo de validação 10 vezes, e veja como o modelo se comporta. Calcule a médias desses resultados;
   - ⋆ Se os dados mudam rápido com o tempo, ou são assimétricos etc., contemple isso nos exemplos de treinamento e validação.

3. **Feature Engineering:** tente aprimorar a acurácia do modelo, p. ex.

   - ⋆ Tratar outlier, dados faltantes, criar interação, transformar dados contínuos para discretos etc. (pré-processamento);

4. **Combine modelos:** agrupe vários algoritmos, certificando-se que são correlacionados.

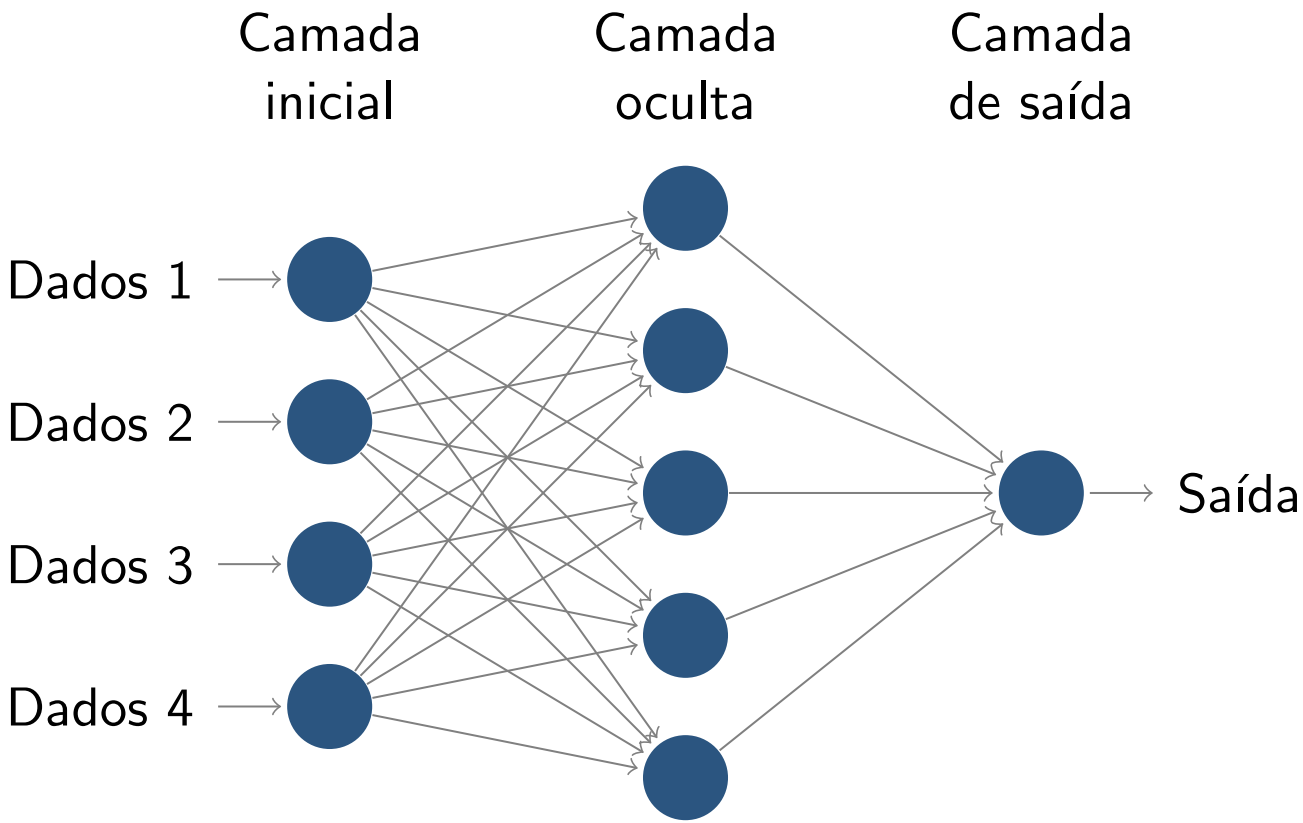
Combinando vários modelos



- As camadas de modelos envolvem técnicas do tipo:
  - ⋆ Regressão Linear;
  - ⋆ Regressão logística;
  - ⋆ KNN;
  - ⋆ Gradiente Boosting;
  - ⋆ Naive Bayes;
  - ⋆ Redes Neurais Artificiais;
  - ⋆ Árvores de decisão;
  - ⋆ Random Forests etc.

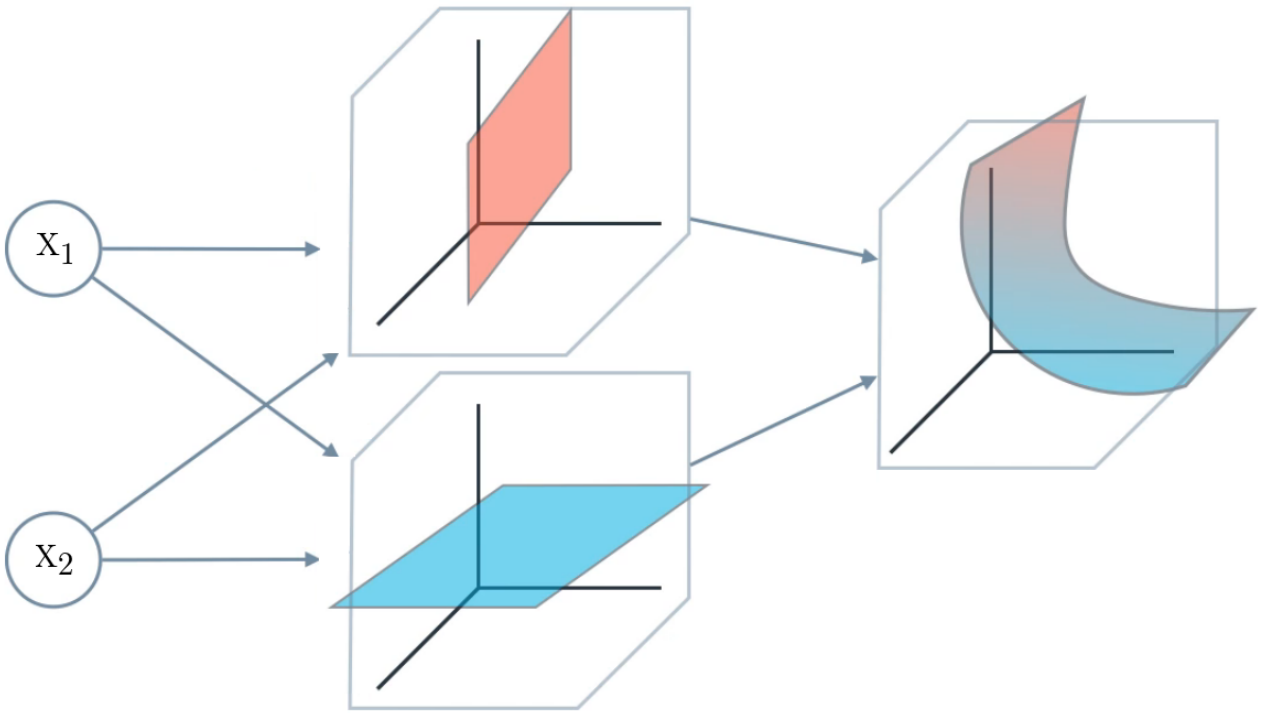
Redes Neurais Artificiais

O que são Redes Neurais Artificiais?



Exemplo

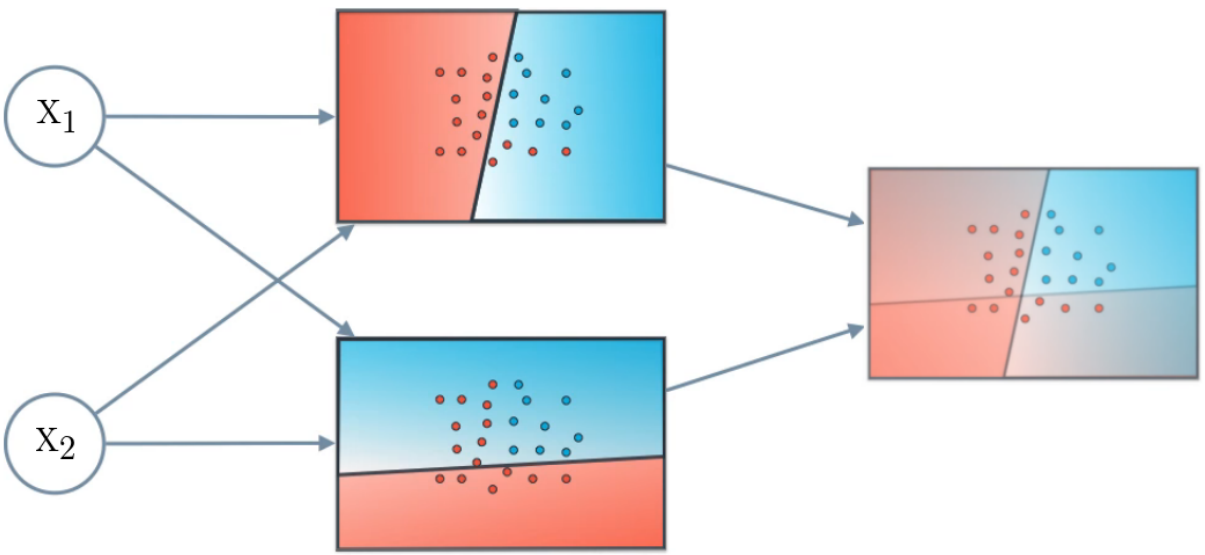
- Em problemas reais, a RNA combina separadores lineares para classificações mais complexas.



Fonte: Luis Serrano

Exemplo

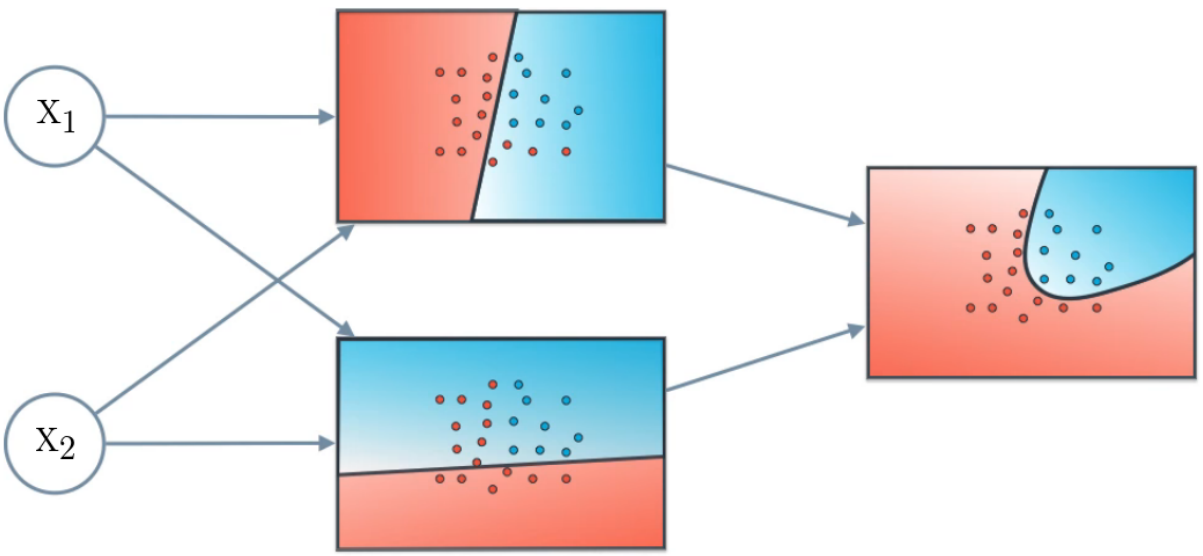
- Em problemas reais, a RNA combina separadores lineares para classificações mais complexas.



Fonte: Luis Serrano

Exemplo

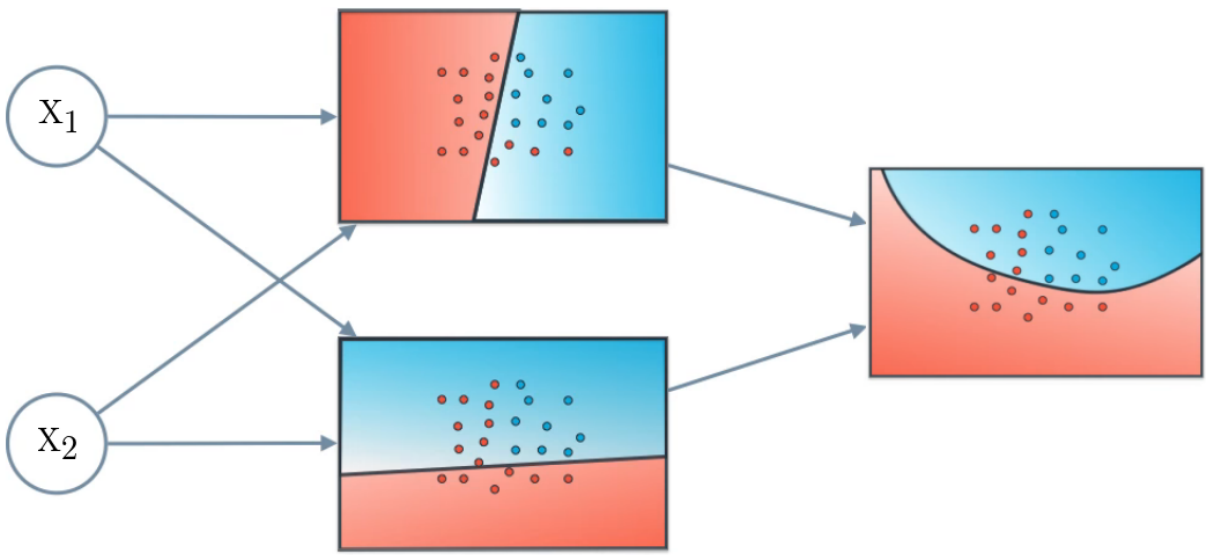
- Em problemas reais, a RNA combina separadores lineares para classificações mais complexas.



Fonte: Luis Serrano

Exemplo

- Dependendo do peso atribuído a cada neurônio, obtemos diferentes regiões.



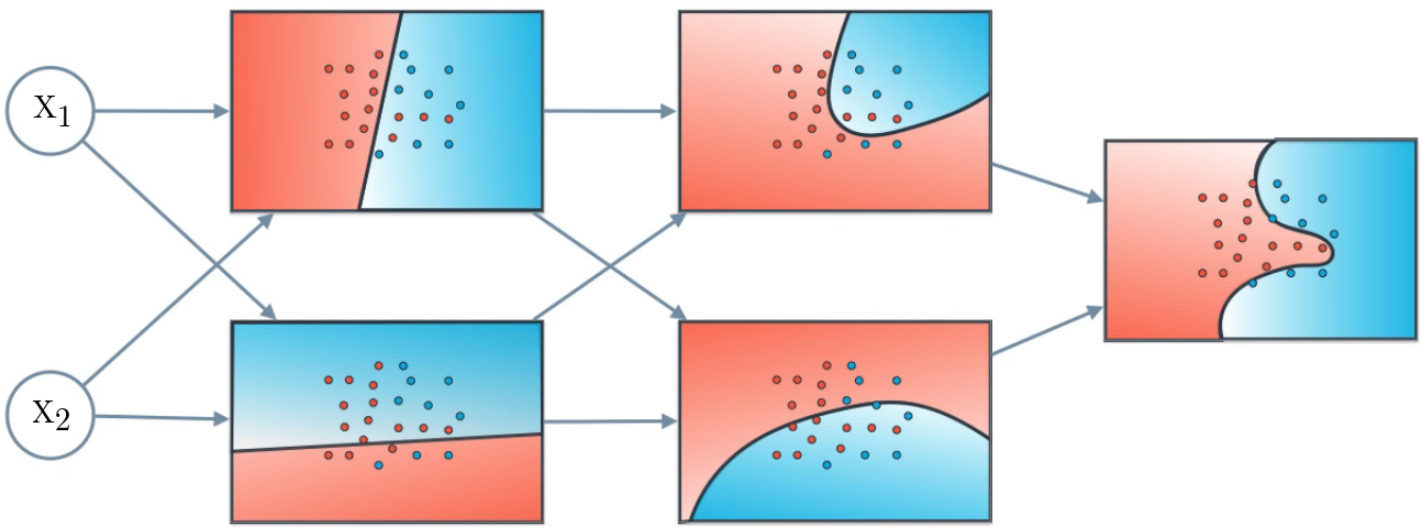
Fonte: Luis Serrano

Quantas camadas devemos utilizar?

1. **Single layer:** capaz de posicionar um hiperplano no espaço das entradas;

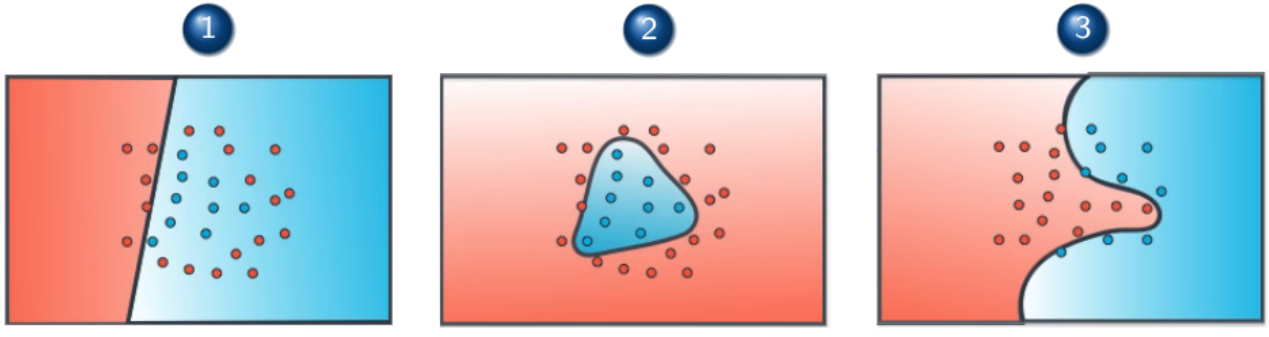
2. **Two layers (one hidden layer):** capaz de descrever uma regra de decisão em somente uma região convexa do espaço;

3. **Three layers (two hidden layers):** a partir de três camadas, somos capazes de generalizar regiões arbitrárias do espaço.

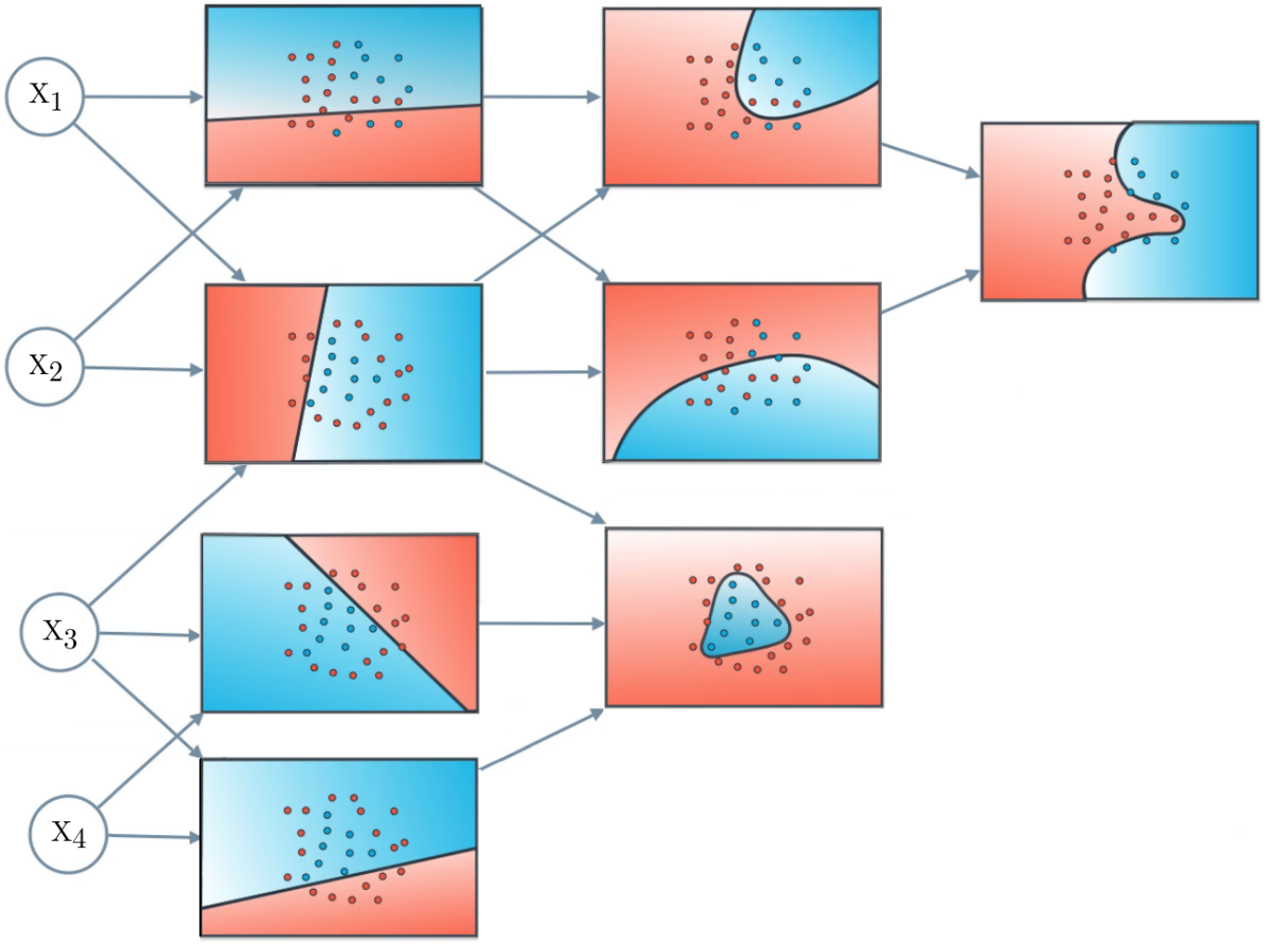


Quantas camadas devemos utilizar?

1. **Single layer:** capaz de posicionar um hiperplano no espaço das entradas;
2. **Two layers (one hidden layer):** capaz de descrever uma regra de decisão em somente uma região convexa do espaço;
3. **Three layers (two hidden layers):** a partir de três camadas, somos capazes de generalizar regiões arbitrárias do espaço.



Exemplo

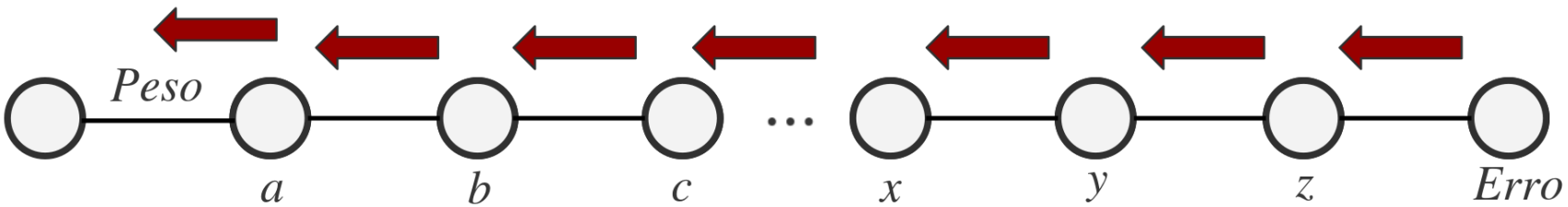


Fonte: Luis Serrano

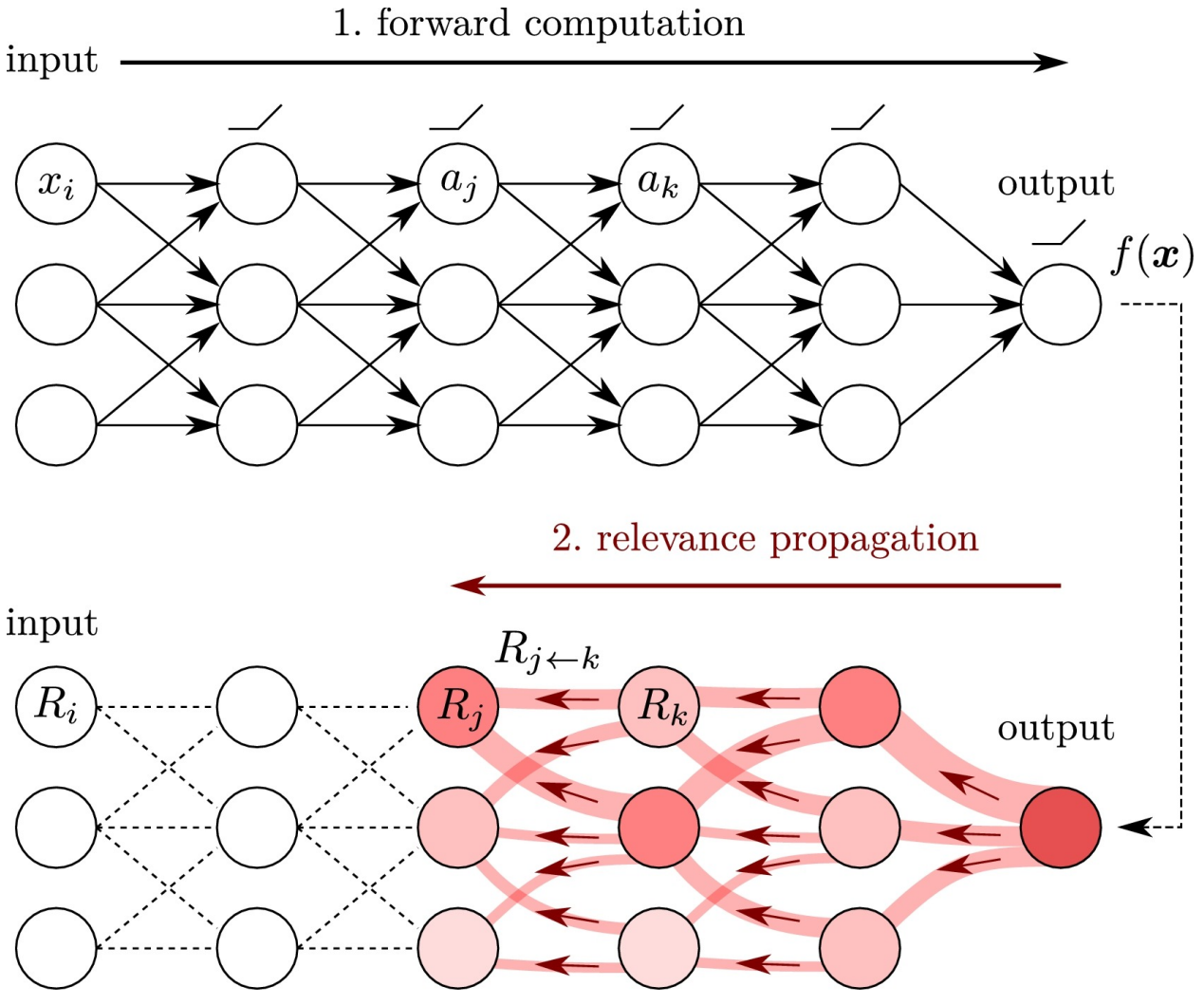
Backpropagation algorithm

- Considerando que ao final da rede temos um erro. Desejamos encontrar os pesos que minimize essa quantidade;

$$\frac{\partial Erro}{\partial Peso} = \frac{\partial a}{\partial Peso} \times \frac{\partial b}{\partial a} \times \frac{\partial c}{\partial b} \times \frac{\partial d}{\partial c} \times \cdots \times \frac{\partial y}{\partial x} \times \frac{\partial z}{\partial y} \times \frac{\partial Erro}{\partial z}$$

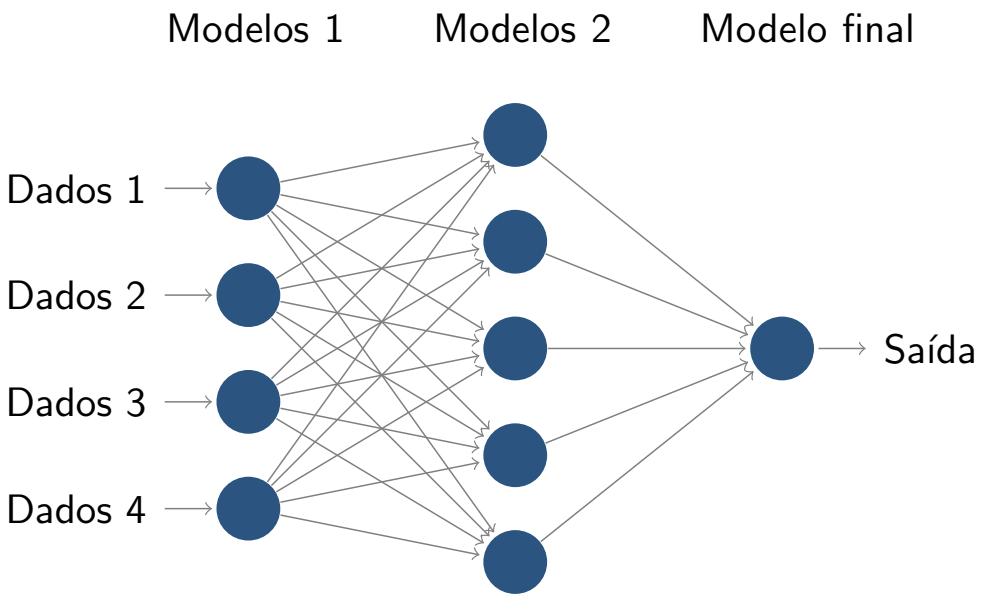


Backpropagation algorithm



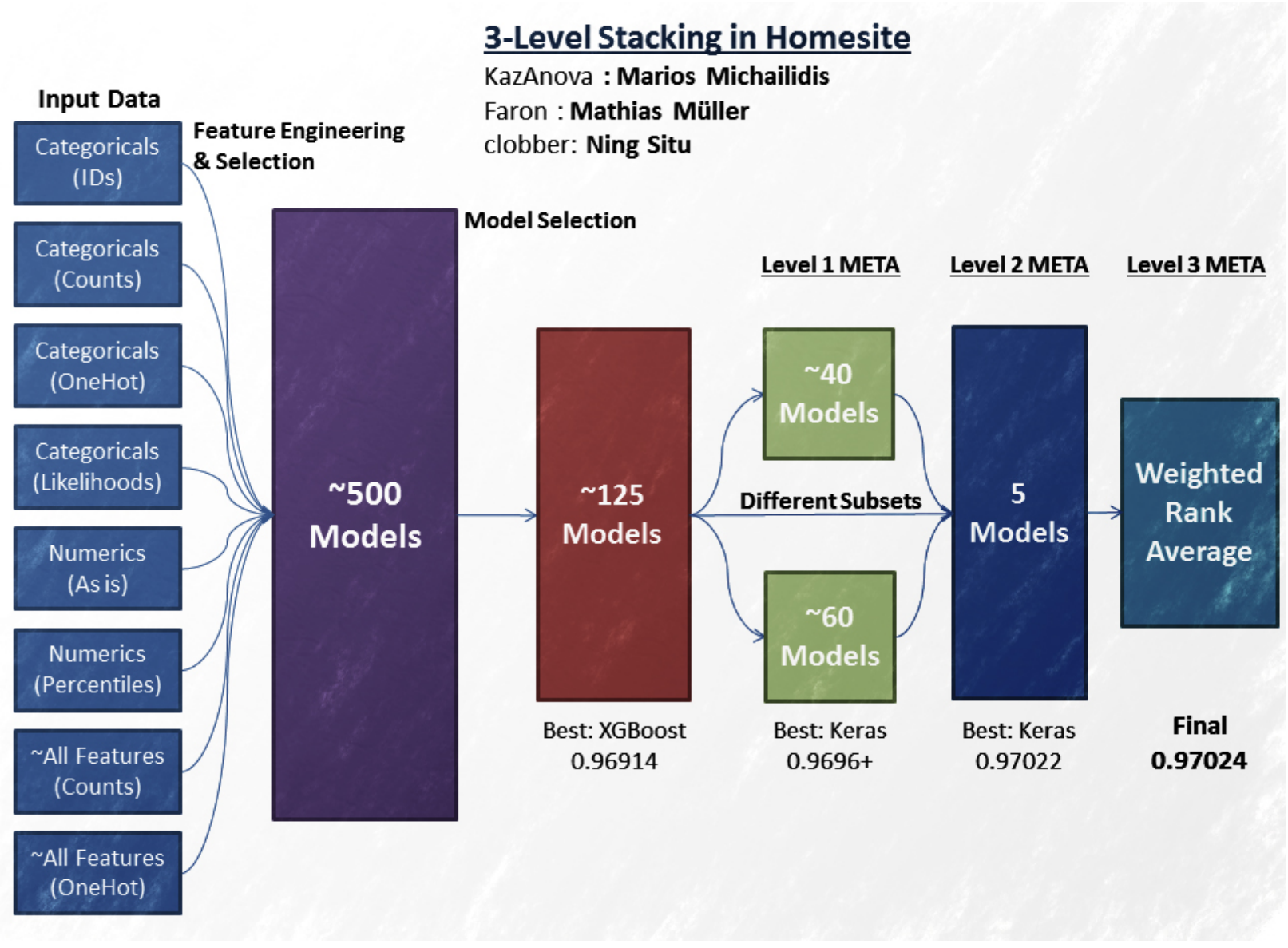
Fonte: Bosse et al. (2018)

Combinando vários modelos



- As camadas de modelos envolvem algoritmos do tipo:
  - ⋆ Regressão Linear;
  - ⋆ Regressão logística;
  - ⋆ KNN;
  - ⋆ Gradiente Boosting;
  - ⋆ Naive Bayes;
  - ⋆ Redes Neurais Artificiais;
  - ⋆ Árvores de decisão;
  - ⋆ Random Forests etc.

Exemplo


3-Level Stacking in Homesite
KazAnova : Marios Michailidis
Faron : Mathias Müller
clobber: Ning Situ
Input Data
Categoricals (IDs)
Categoricals (Counts)
Categoricals (OneHot)
Categoricals (Likelihoods)
Numerics (As is)
Numerics (Percentiles)
~All Features (Counts)
~All Features (OneHot)
Feature Engineering & Selection
~500 Models
Model Selection
~125 Models
Level 1 META
~40 Models
Different Subsets
~60 Models
5 Models
Level 2 META
Weighted Rank Average
Level 3 META
Best: XGBoost 0.96914
Best: Keras 0.9696+
Best: Keras 0.97022
Final 0.97024

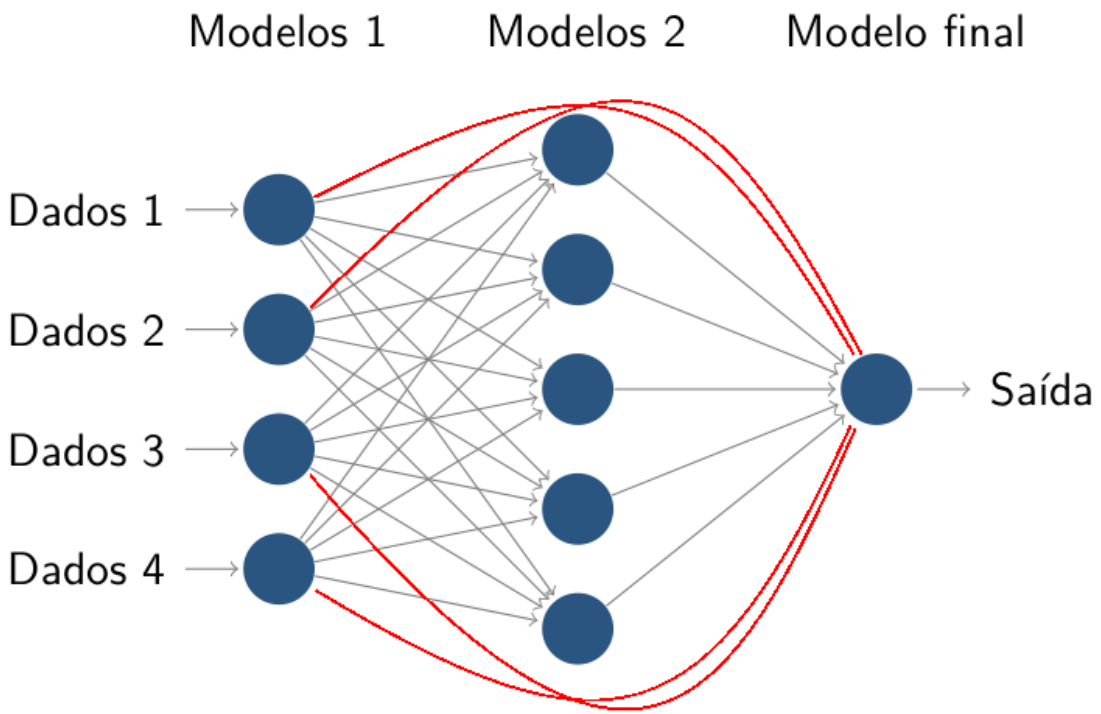
Linhas de comando

Linhas de comando

- Os algoritmos disponíveis requerem algumas especificações. Precisamos entendê-las, para obter resultados promissores, por exemplo:

Comando	Explicação
`task`	Pode ser regressão ou classificação.
`metric`	Métrica de saída na validação cruzada, para cada modelo-neurônio. Pode ser `logloss`, `AUC`, `rmse` etc.
`stackdata`	`TRUE` se saída do modelo na camada k$-$1 entrará também nas camadas $k+1$, $k+2$ etc.
`bins`	Parâmetro que permite que os classificadores sejam usados em problemas de regressão.
`threads`	Número de modelos a serem executados em paralelo.
`folds`	Número de *folds* no treinamento e teste.

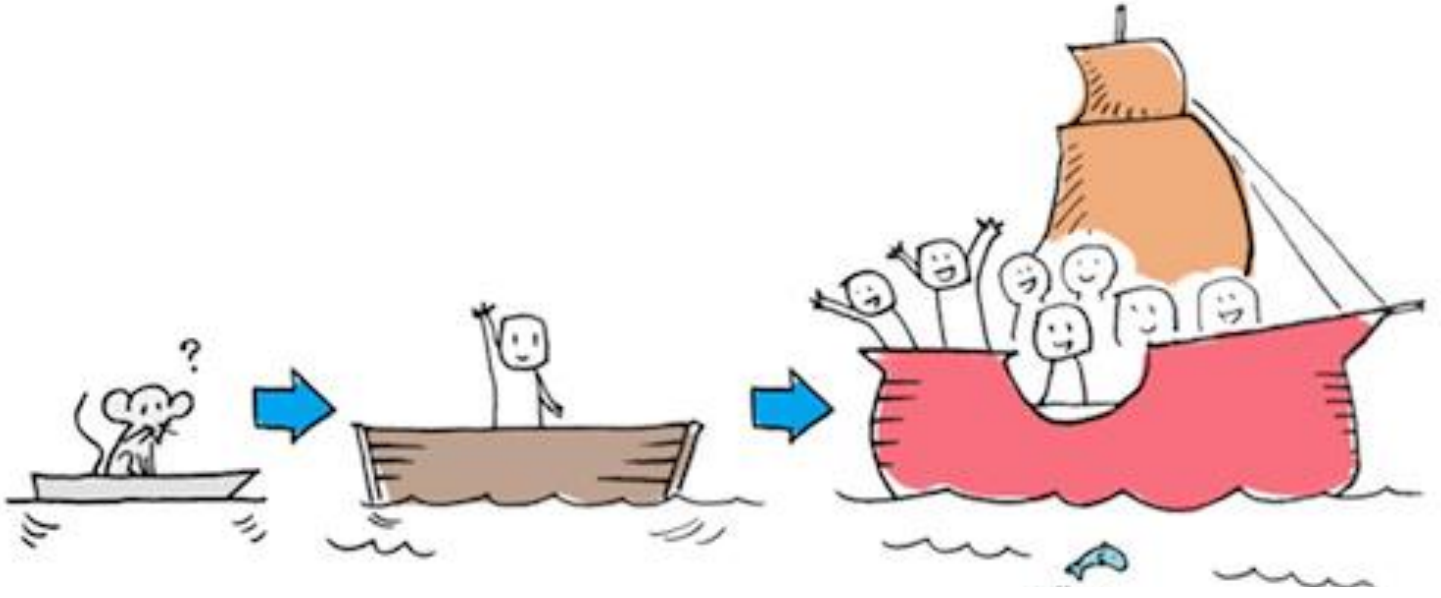
- `TRUE` se saída do modelo na camada k−1 entrará também nas camadas $k+1$, $k+2$ etc.



Desenvolver, testar e implementar

Desenvolver, testar e implementar

- É comum durante o treinamento dos modelos, obtermos determinado resultado, e na prática, o resultado ser outro;

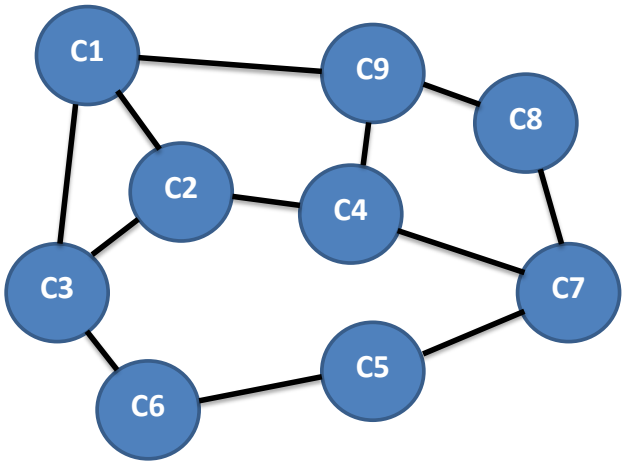


- Para aumentar a convicção de que os modelos farão um bom trabalho, devemos ser capazes de treiná-los em condições próximas das reais;

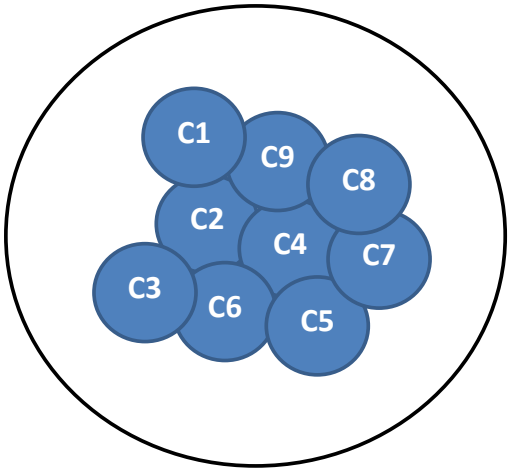
Cultura da Ciência de Dados

Cultura da Ciência de Dados

- A grande dificuldade durante o processo de execução de um projeto de *Machine Learning* é a **mudança cultural**;



Muitas culturas que não estão integradas



Uma cultura dentro de microculturas

"Dando-se oportunidade de escolha entre mudar e provar que não é necessário mudar, a maioria das pessoas prefere a segunda alternativa". **John Galbraith**

Obrigado pela atenção!