

III SEMANA DE ENGENHARIA, TECNOLOGIA E COMPUTAÇÃO



AutoML

Alan Cândido de Souza



**INSTITUTO
FEDERAL**
Minas Gerais

BambuÍ, 27 de setembro de 2018



APRESENTAÇÃO

- Engenheiro Eletricista – FASA (Montes Claros-MG)
- Mestrando em Inteligência Computacional (UFMG)



III SETC



ROTEIRO PARTE 1:

- Motivação
- Conceitos básicos de aprendizado de máquina
- Apresentação dos modelos
- Mãos na massa: #LetsCode



III SETC



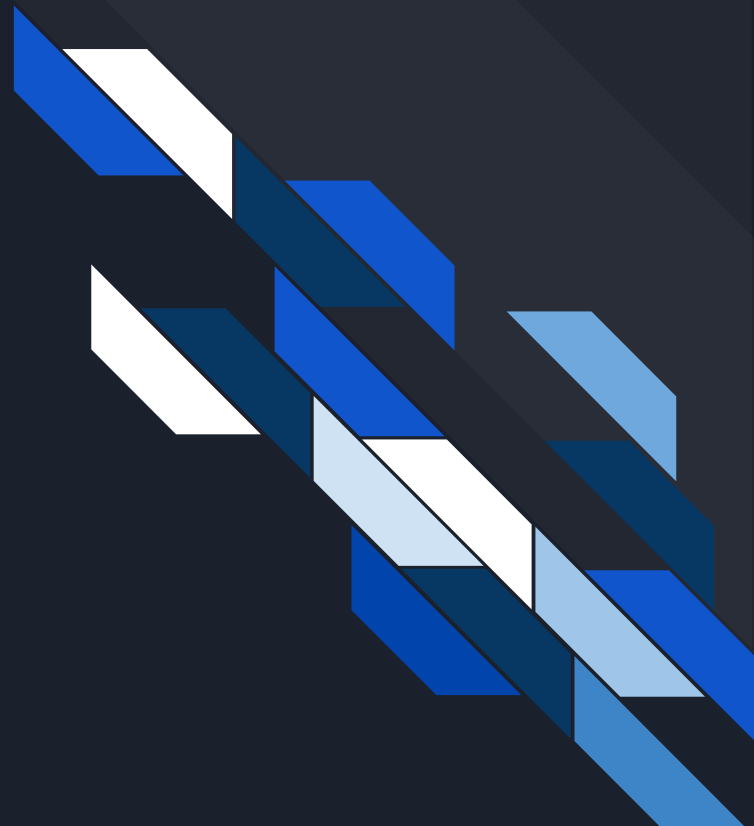
ROTEIRO PARTE 2:

- Introdução ao Aprendizado de Máquina automatizado
- H2O AutoML
- Mãos na massa: #LetsCode



PARTE 1:

Motivação





Por que AutoML?

- Problema hipotético:

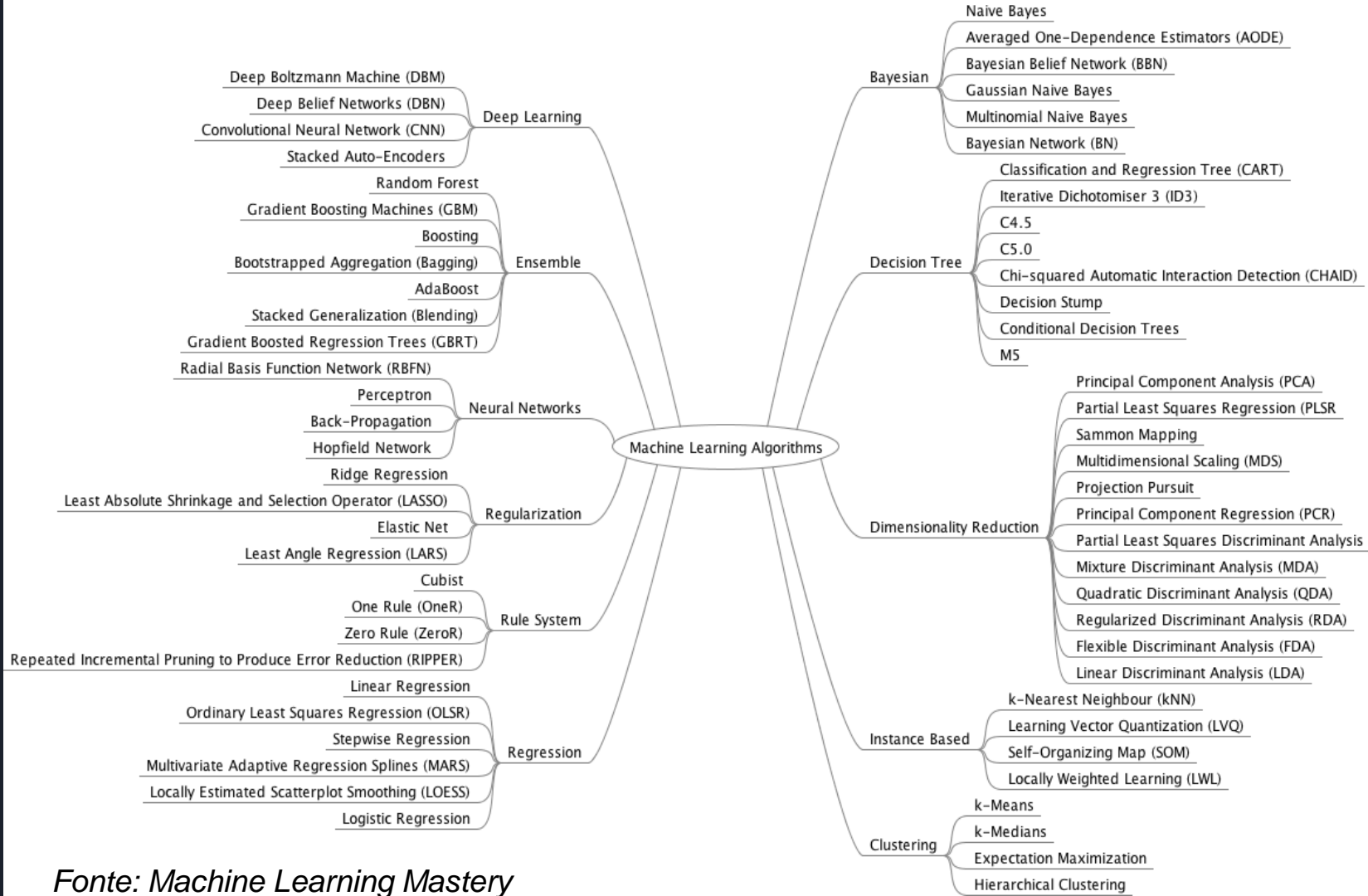
Dataset de diagnóstico de pacientes em uma tomografia.

- Qual modelo usar?





III SETC



Fonte: Machine Learning Mastery



Vantagens do AutoML

- Rápida experimentação (poucas linhas de código)
- Identificação dos melhores modelos
- Configuração de parâmetros
- **Warning:** só mais uma ferramenta do seu toolbox



Aprendizado de Máquina





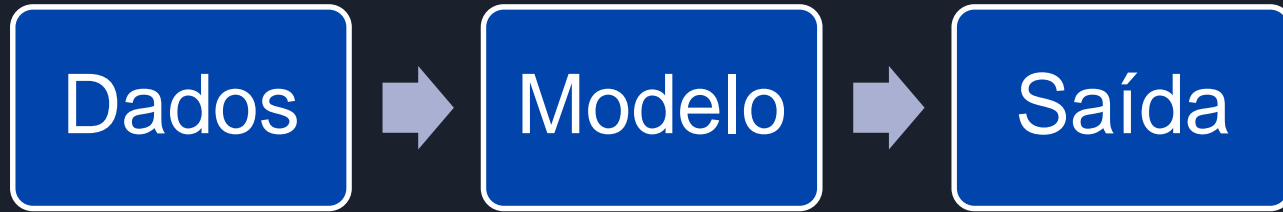
Aprendizado de Máquina

- Conceitos Básicos
- Tipos de problemas
- Métricas de desempenho
- Parâmetros





Conceitos Básicos:



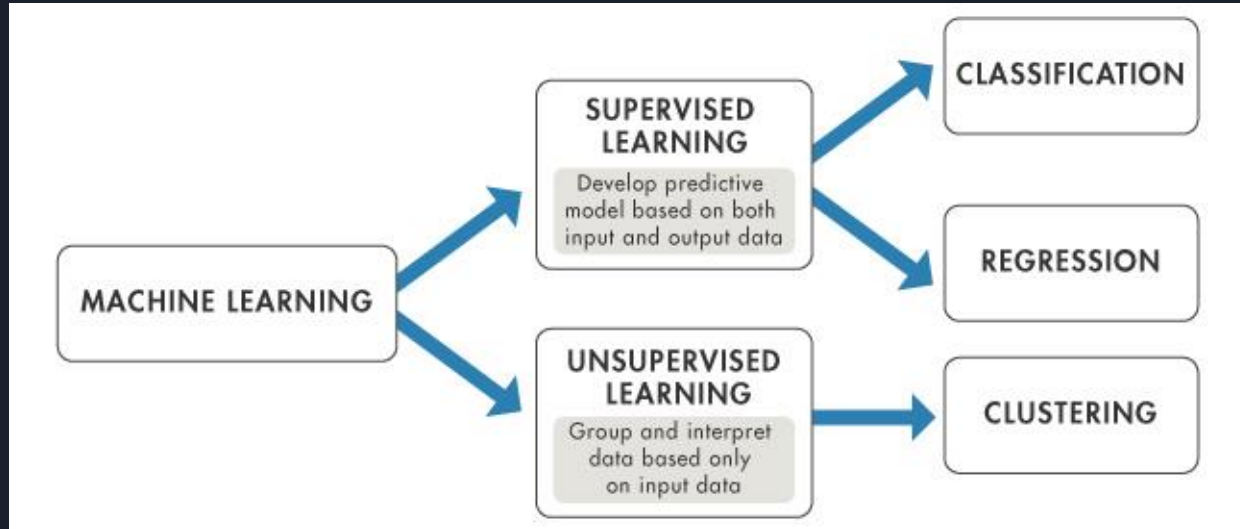


Tipos de aprendizado:

- Aprendizado Supervisionado
- Aprendizado não-supervisionado
- Outros (*Reinforcement Learning*, etc)



Tipos de Problemas:





Métricas de desempenho:

- O quão bom é o seu algoritmo?





Métricas de regressão:

- MSE (*Mean Squared Error*)
- Outras métricas (*deviance*, R^2 , etc...)



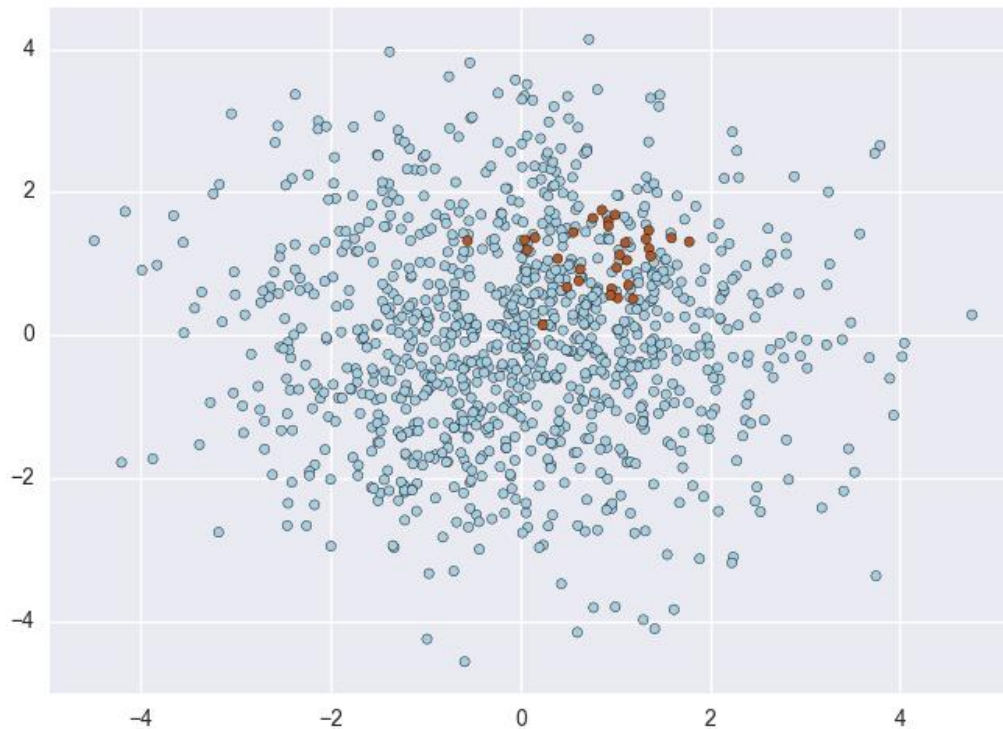


Métricas de classificação:

- *Confusion Matrix (Misclassification)*
- Erro médio por classe
- *Logloss*
- *MSE*
- *AUC (Classificação Binomial)*



Dados desbalanceados





AUC-ROC

- TPR (Taxa de verdadeiros positivos- *Recall*):

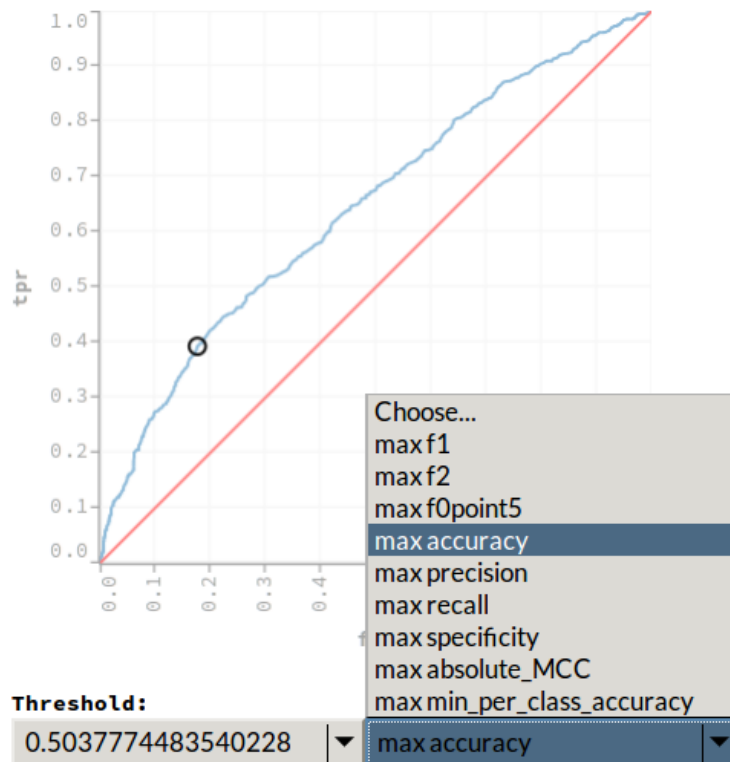
$$\frac{TP}{TP + FN}$$

- FPR (Taxa de falsos positivos):

$$\frac{FP}{FP + TN}$$



▼ ROC CURVE - VALIDATION METRICS , AUC = 0.644682



Selected mark(s):

threshold	0.5038
f1	0.4804
f2	0.4237
f0point5	0.5546
accuracy	0.6419
precision	0.6183
recall	0.3928
specificity	0.8234
absolute_MCC	0.2411
min_per_class_accuracy	0.3928
tns	970
fns	521
fps	208
tps	337
tnr	0.8234
fnr	0.6072
fpr	0.1766
tpr	0.3928
idx	151

Actual/Predicted		0	1	Error	Rate
CM	0	970	208	0.1766	208 / 1178
	1	521	337	0.6072	521 / 858
	Total	1491	545	0.3581	729 / 2036





Estágios do aprendizado

- Treinamento
- Validação
- Teste



III SETC

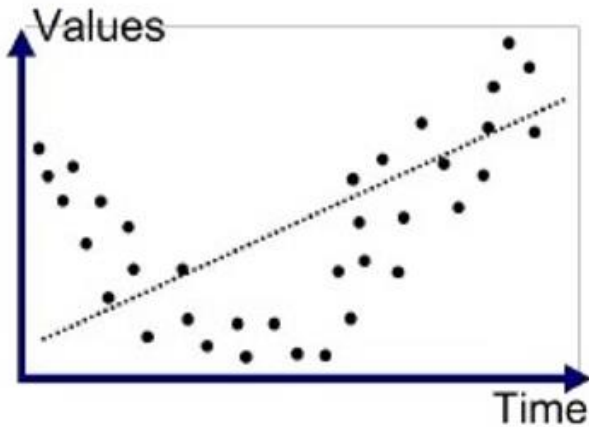


Treinamento:

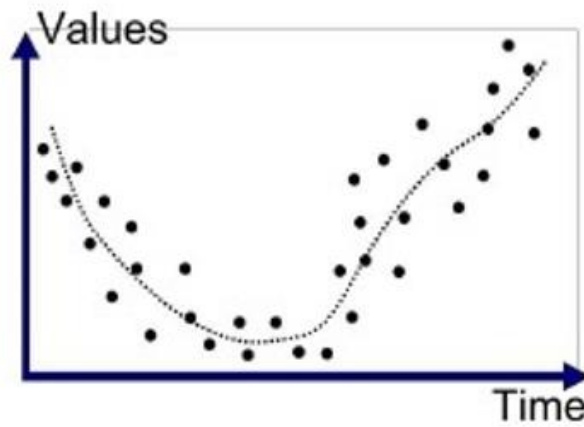
- Processo iterativo (definição do número de *epochs*)
- *Scoring* (acompanhamento do modelo)
- *Early stoping*



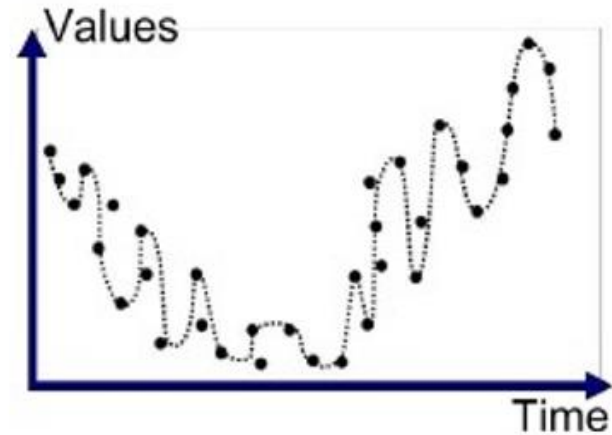
Underfitting e overfitting



Underfitted



Good Fit/Robust




Overfitted

Validação Cruzada



Modelos de Aprendizizado de Máquina





Modelos de aprendizado supervisionado

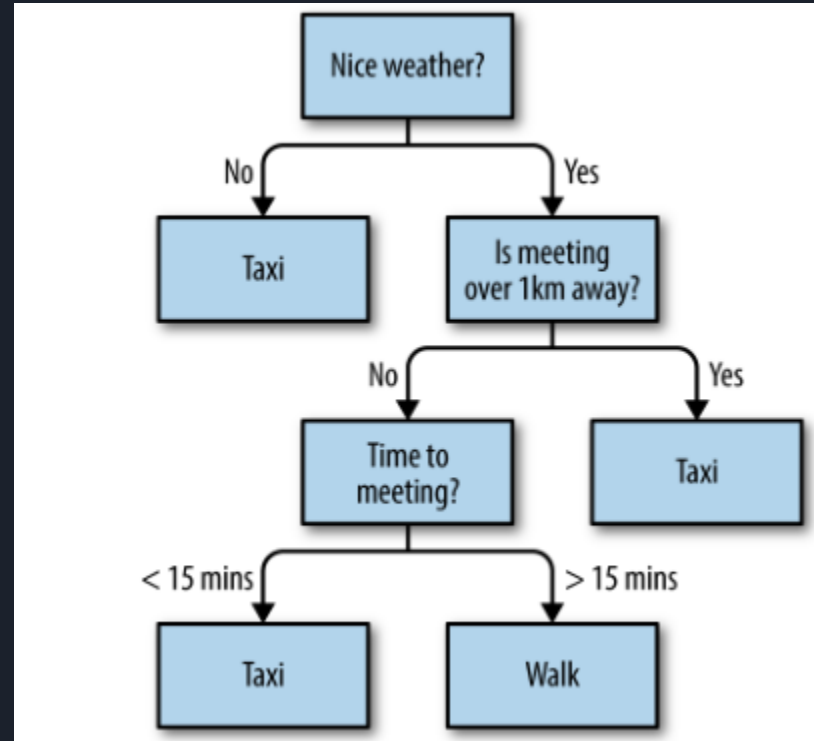
- Modelos baseados em árvores de decisão (*Random Forest e GBM*)
- Modelos Lineares
- Redes Neurais Artificiais



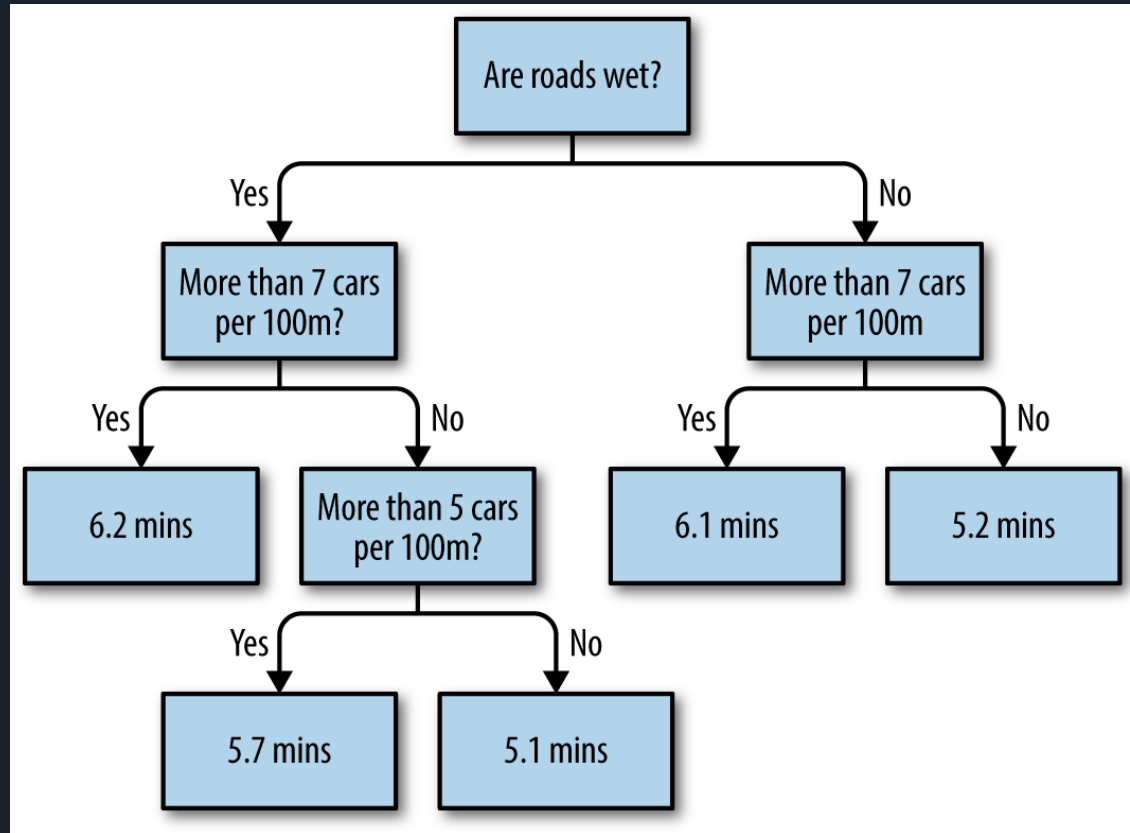
Modelos baseados em árvores de decisão



Árvores de Decisão: Classificação



Árvores de Decisão: Regressão





Random Forest

- Modelo *ensemble* (Combina várias árvores)
- Atribui os dados de forma aleatória
- Classificação: a resposta mais popular
- Regressão: a média dos valores



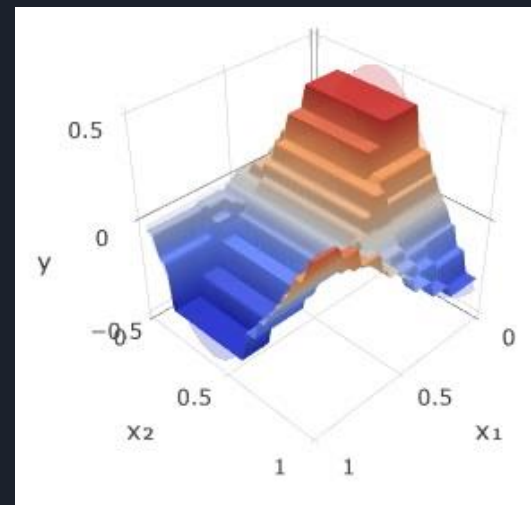
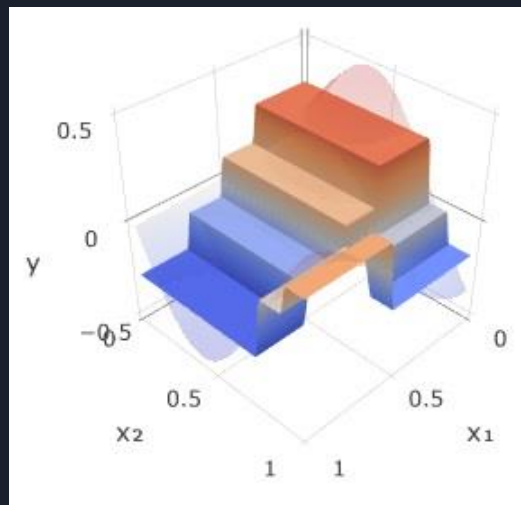
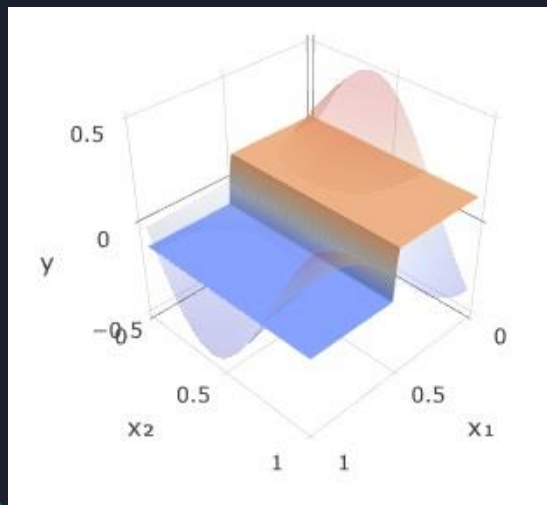


Principais parâmetros:

- *Número de árvores do modelo*
- Profundidade máxima (controla a complexidade de cada árvore)



Random Forest:



Conjunto de árvores de decisão com profundidade 1, 3, e 6 respectivamente



III SETC

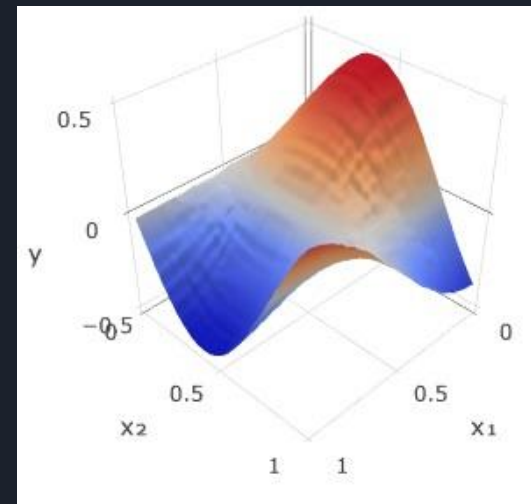
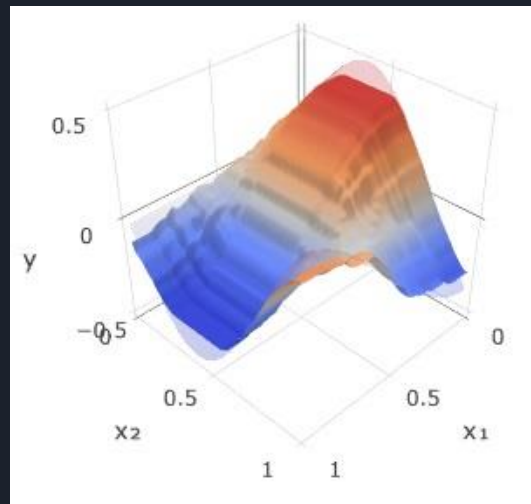
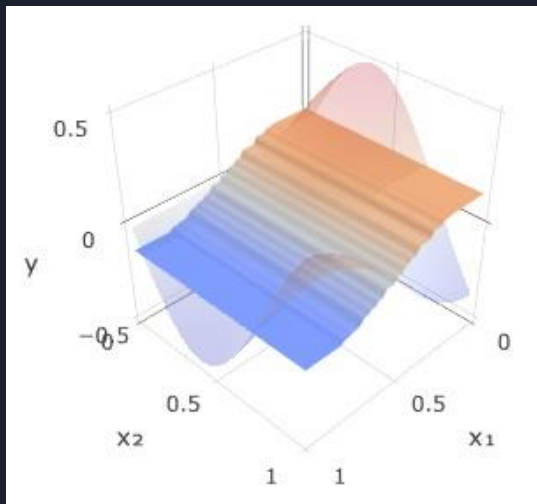


Gradient Boosting Machines (GBM)

- Modelo *ensemble* (assim como *Random Forest*)
- Bosting: melhora a interpretação de cada árvore em cada iteração a partir da atribuição de pesos (gradiente) nos dados mais “difíceis”
- Problemas com *outliers* e *overfitting* (algoritmos de *poda*)



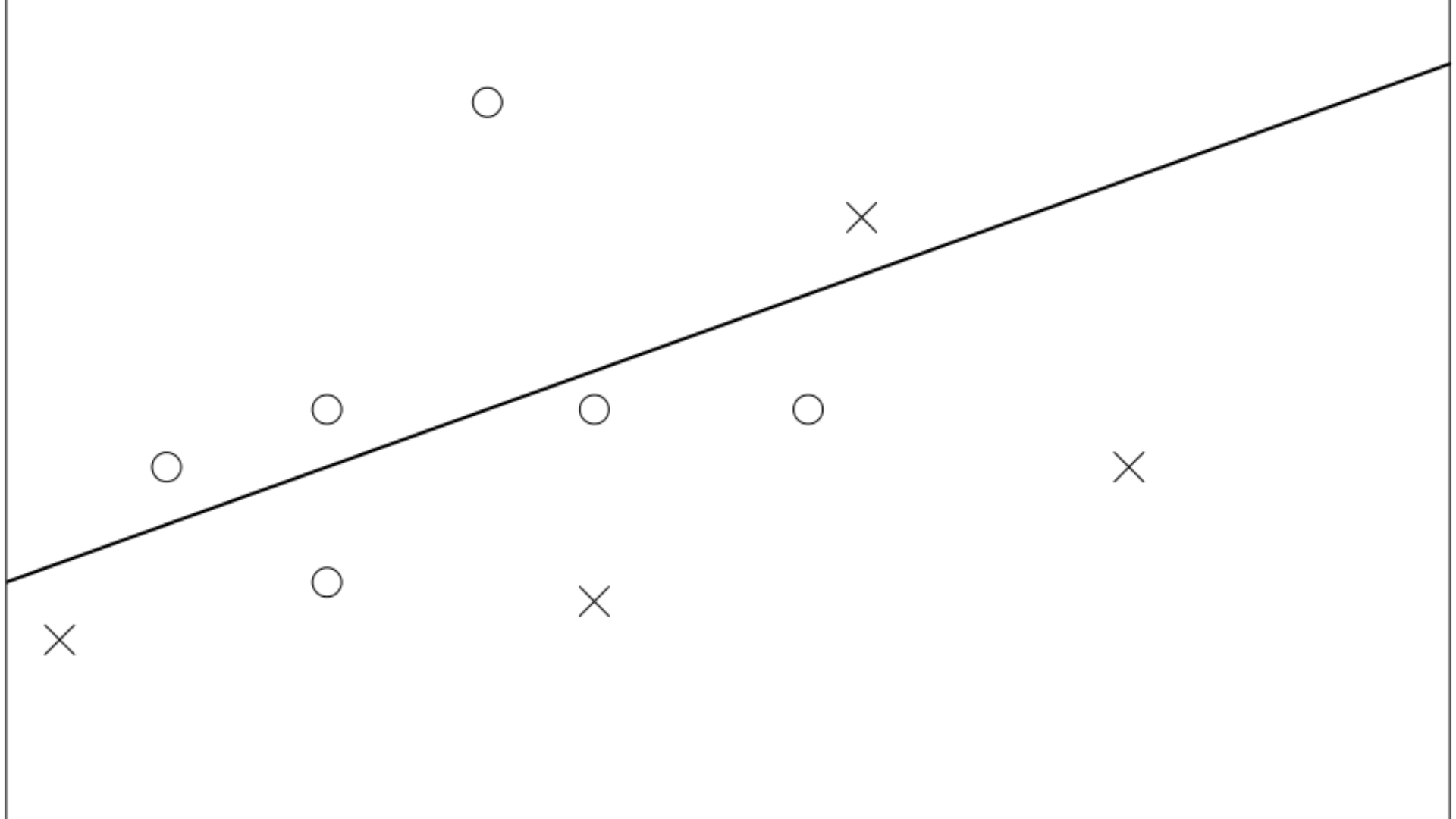
Gradient Boosting Machines:

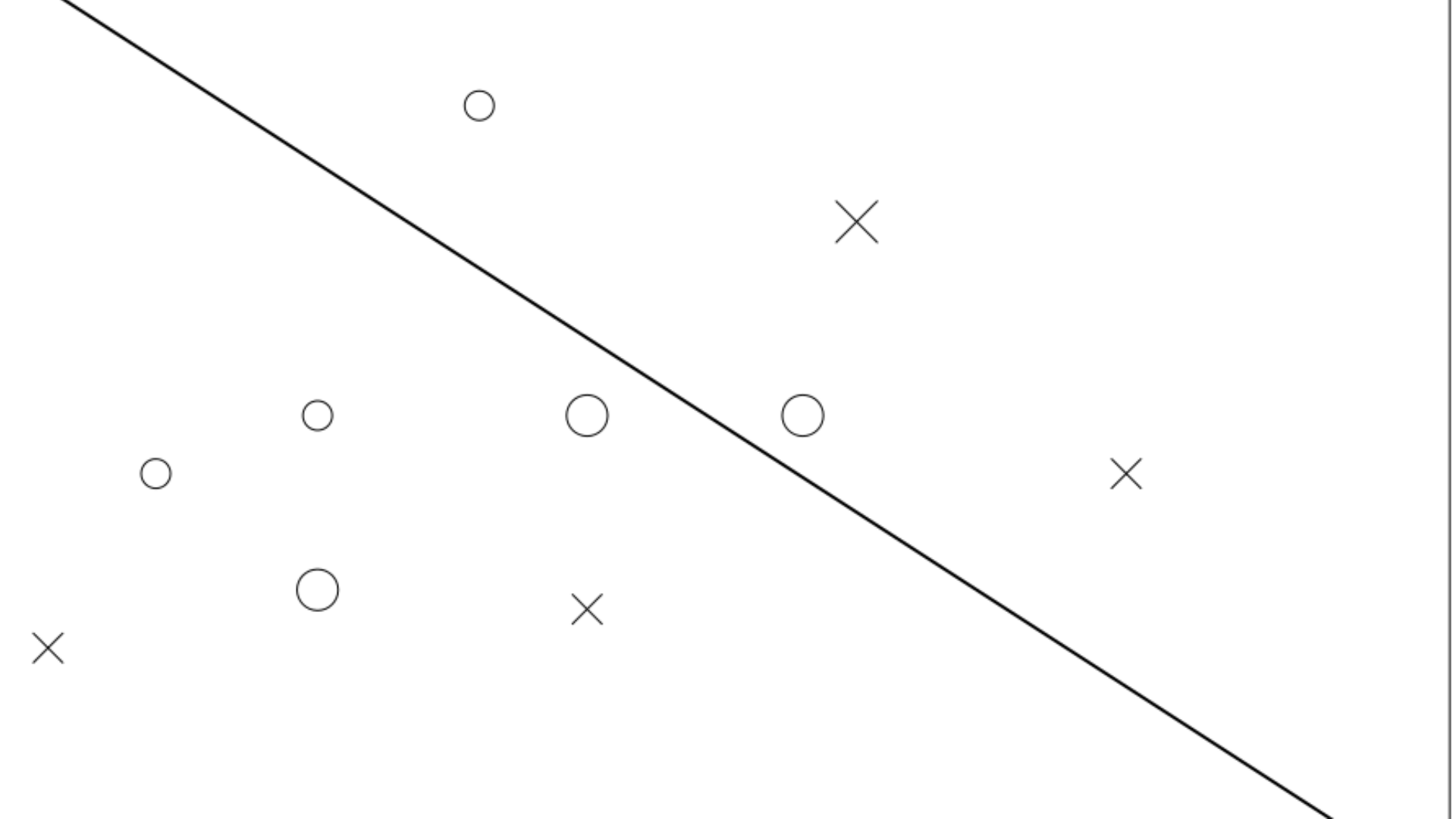


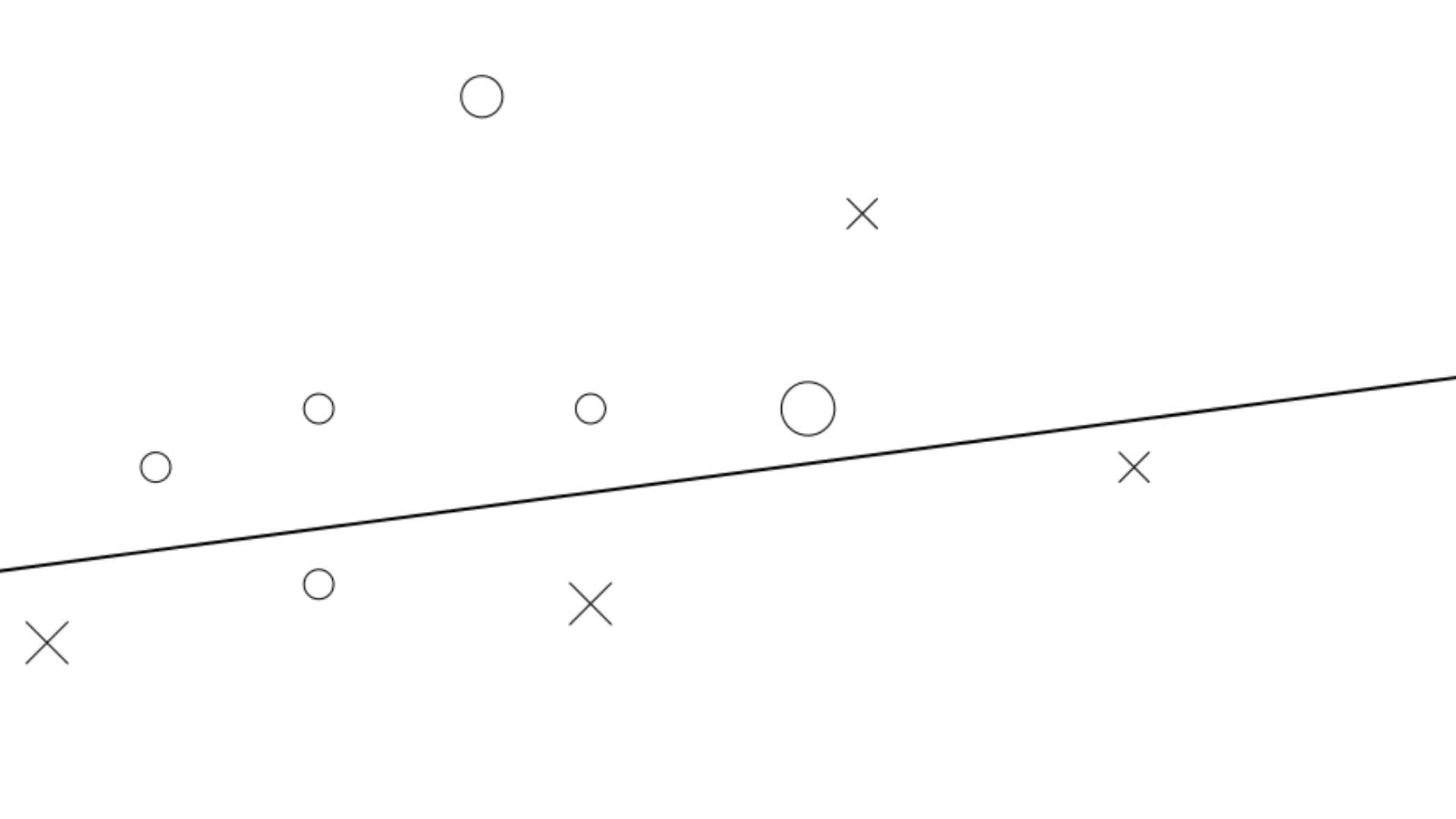
Gradient tree boosting com profundidade 1, 3, e 6 respectivamente



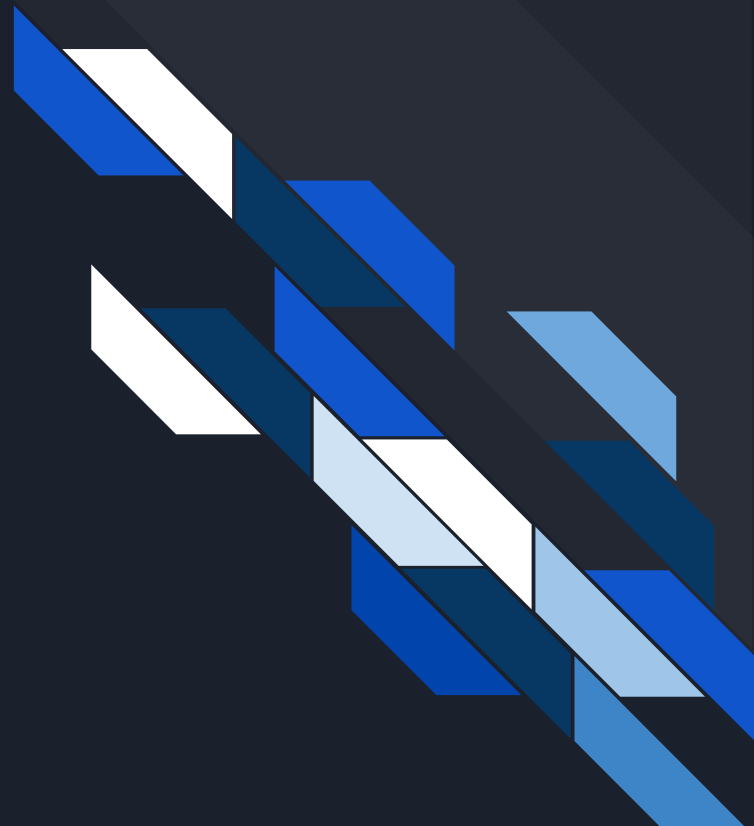
III SETC







Modelos Lineares





Generalized Linear Models (GLM):

- *Aprendizado estatístico*
- Regressão Linear
- Regressão Logística
- *Linear Discriminant Analysis*
- Flexibilidade para trabalhar com grandes quantidades de dados



Regressão Linear

$$\hat{y} = x^T \beta + \beta_0$$

$$\max_{\beta, \beta_0} \frac{-1}{2N} \sum_{i=1}^N (x_i^T \beta + \beta_0 - y_i)^2 - \lambda (\alpha \|\beta\|_1 + \frac{1}{2} (1 - \alpha) \|\beta\|_2^2)$$



$$D = \sum_{i=1}^N (y_i - \hat{y}_i)$$



Regressão Logística:

- Classificação Binomial:
- Modela a probabilidade de determinado dado de entrada pertencer à uma das classes



Regressão Logística:

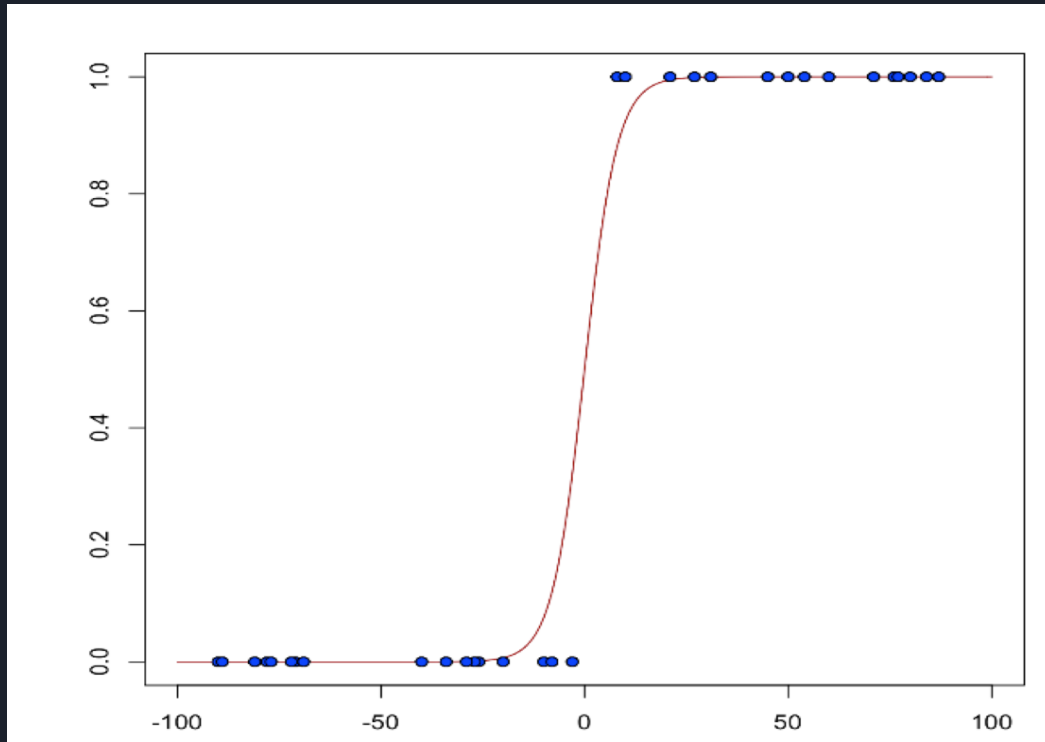
$$\hat{y} = \frac{e^{x^T \beta + \beta_0}}{1 + e^{x^T \beta + \beta_0}}$$

$$\max_{\beta, \beta_0} \frac{1}{N} \sum_{i=1}^N y_i (x_i^T \beta + \beta_0) - \log(e^{x_i^T \beta + \beta_0}) - \lambda (\alpha \|\beta\|_1 + \frac{1}{2} (1 - \alpha) \|\beta\|_2^2)$$



$$D = -2 \sum_{i=1}^N (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

Regressão Logística:



III SETC

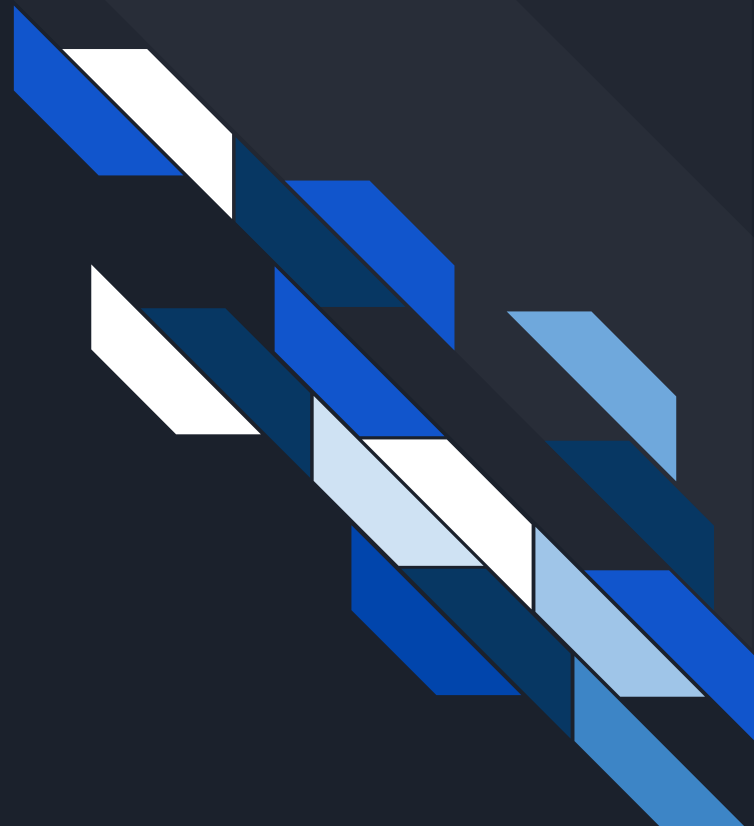


Regularização:

- *L1: Lasso Regression*
- *L2: Ridge Regression*
- *Elastic Net*



Redes Neurais

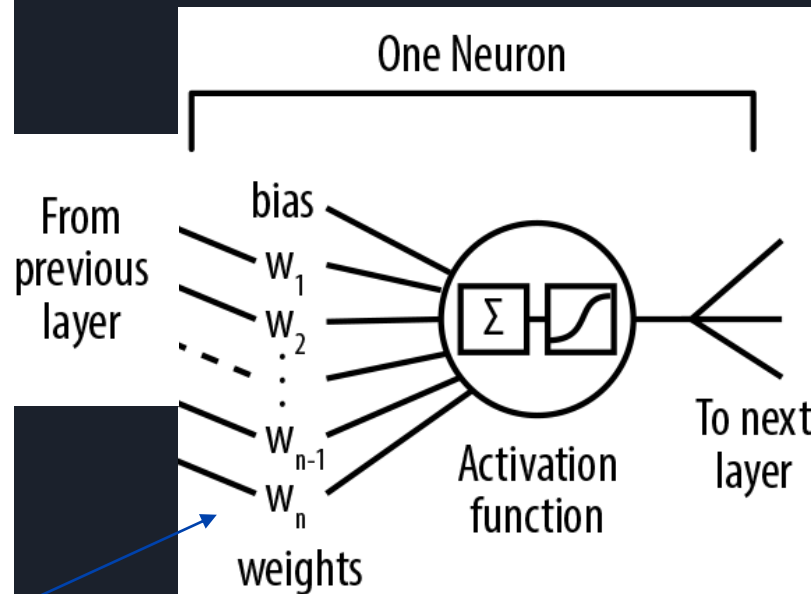
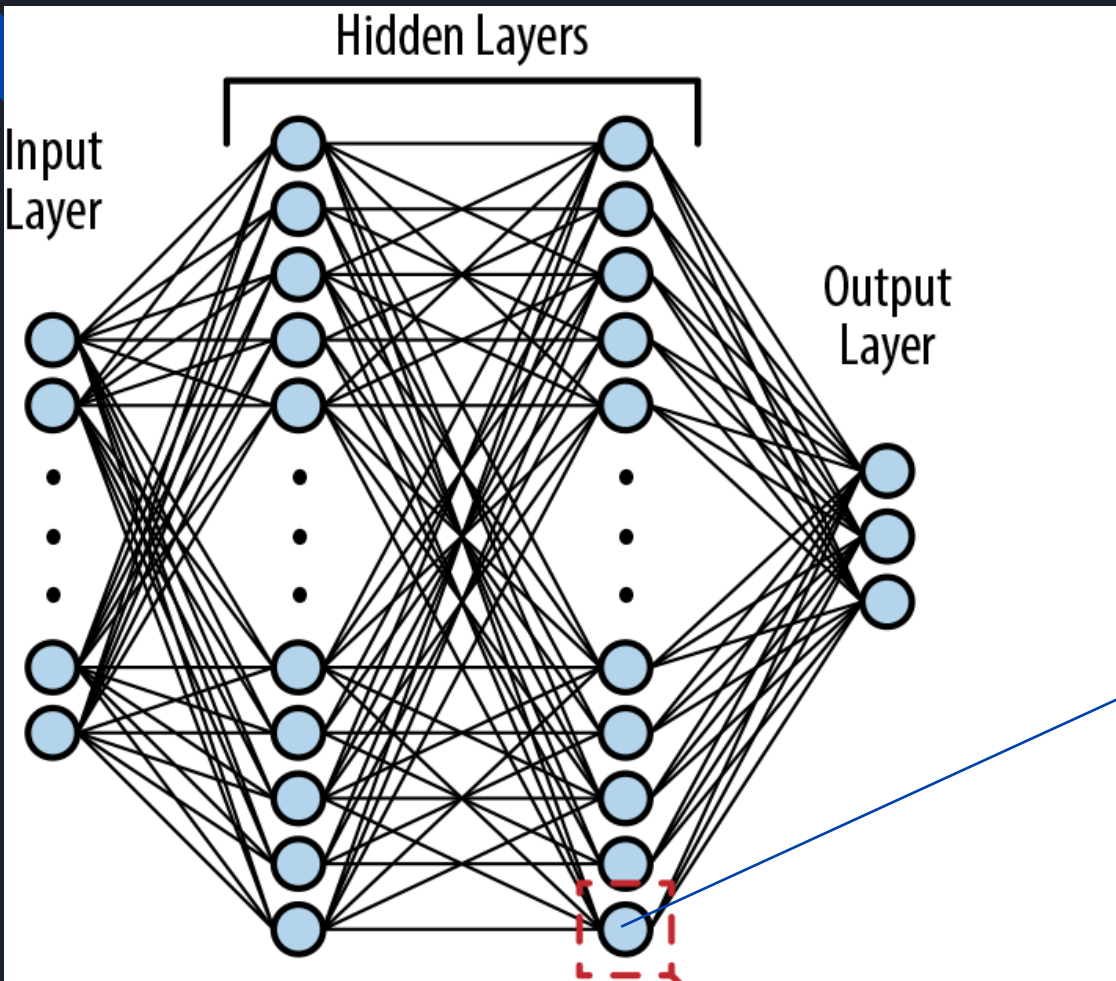




Deep Learning

- Redes neurais (simulação do cérebro humano)
- Organizado por camadas
- Transforma dados (*inputs*) em saídas (*outputs*)
- Apresenta bons resultados para problemas difíceis (reconhecimento de padrões em imagens por exemplo)

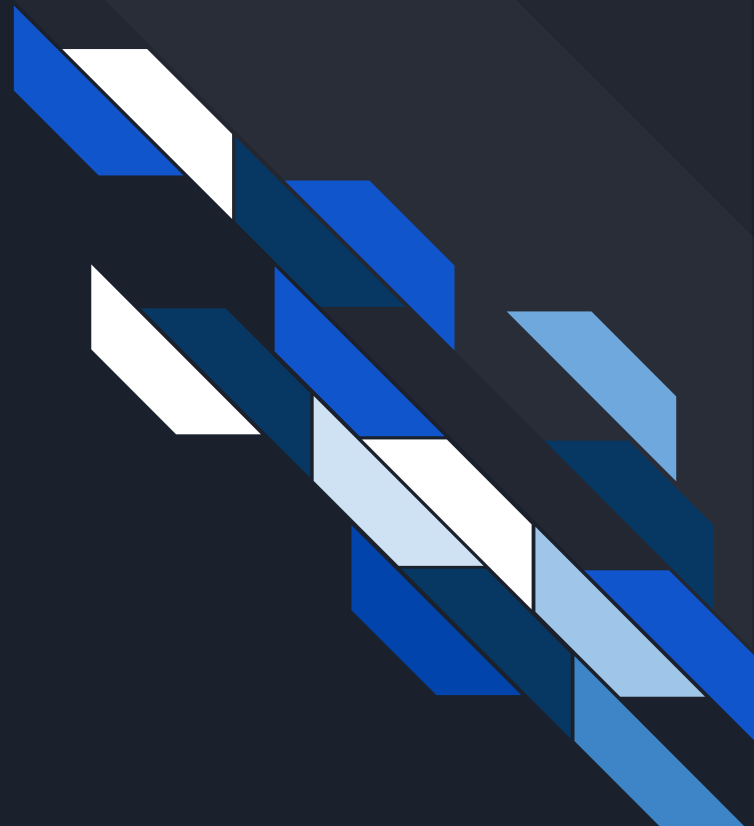




Fonte:

MÃOS NA MASSA

PARTE 1:





H2O Frame work

- Implementada em Java
- *Open source*
- Escalável para *big data*
- *APIs disponíveis em R, python, scala, e interface web (Flow)*





Google Colaboratory

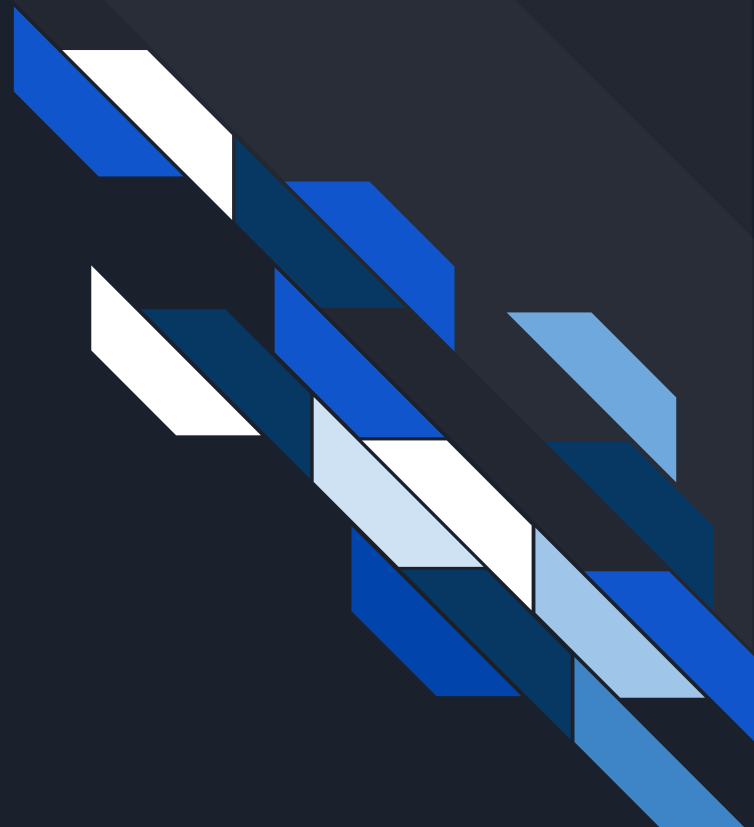
<https://colab.research.google.com/>



III SETC

PARTE 2:

Introdução ao Aprendizado de Máquina Automatizado



Aspectos de AutoML:



Data Prep

Model
Generation



Ensembles



III SETC

Fonte: H2O World 2017



Preparação dos dados

- Normalização $\sim N(0,1)$
- Remoção de Nas (*missing data*)
- *One-hot encoding* de variáveis categóricas
- Extração de características (PCA)
- *Feature Engineering*





Geração de Modelos:

- *Grid Search (Random e Cartesian)*
- Ajuste de parâmetros via *Early stopping*
- *Bayesian Hyperparameter Optimization*



Ensembles:



III SETC



Ensembles:

- *Bagging/Averaging*
- *Staking/Super Learning*
- *Ensemble selection*





H2O AutoML

Random Staking:

Combinação entre o *Grid Search* aleatório e os *Stacked Ensembles*



III SETC



Stacked Ensembles:

- Define n “*Base learners*” (outros modelos de ML)
- Especifica um “*metalearner*” (apenas mais um alg.)
- Implementa validação cruzada (*K-fold*) nos *base learners*





Stacked Ensembles:

- Coleta os valores obtidos dos *base learners*
- Treina um novo algoritmo (*metalearner*) para encontrar a combinação ótima dos *base learners*
- Como adiciona-se apenas mais uma etapa, o custo computacional é pequeno



Stacked Ensembles:

$$n \left\{ \begin{bmatrix} p_1 \end{bmatrix} \cdots \begin{bmatrix} p_L \end{bmatrix} \begin{bmatrix} y \end{bmatrix} \right\} \rightarrow n \left\{ \begin{bmatrix} \overbrace{\quad\quad\quad}^L \\ Z \end{bmatrix} \begin{bmatrix} y \end{bmatrix} \right\}$$





Random Grid + Stacking:

- *Random Grid Search* permite comparar o ajuste de diferentes parâmetros
- Permite ainda gerar diferentes configurações de um mesmo modelo





Random Grid + Stacking:

- No *stacked ensemble* é recomendável observar se os modelos de base têm boa performance sozinhos e se geram erros não correlacionados entre si





Disponível em vários pacotes

- Auto-Keras
- Auto-sklearn
- H2O AutoML
- tpot
- (...)



Disponível em vários pacotes

Benchmarking Automatic Machine Learning Frameworks

Adithya Balaji^{*1} Alexander Allen^{*1}

Abstract

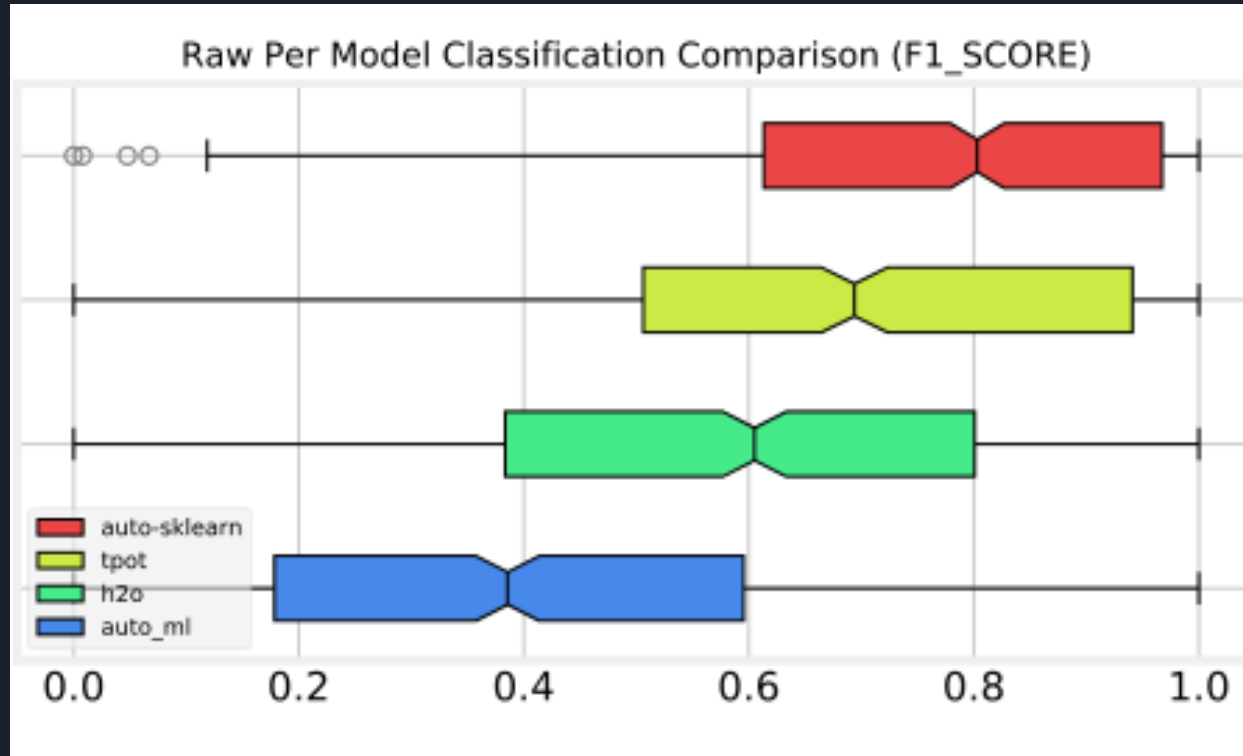
AutoML serves as the bridge between varying levels of expertise when designing machine learning systems and expedites the data science process. A wide range of techniques is taken to address this, however there does not exist an objective comparison of these techniques. We present a benchmark of current open source AutoML solutions using open source datasets. We test auto-sklearn, TPOT, auto_ml, and H2Os AutoML solu-

dardized techniques to the data developed over the years and collected in open source machine learning libraries such as scikit-learn. However, the methods that are used to automate the application and assessment of these techniques widely differ. These methods cannot be assessed on the rigor of their theory alone or by the individual performance of the constituent algorithms. Thus, they must be experimentally assessed as a whole across a variety of data. We perform a quantitative assessment on the most mature open source solutions available for AutoML.

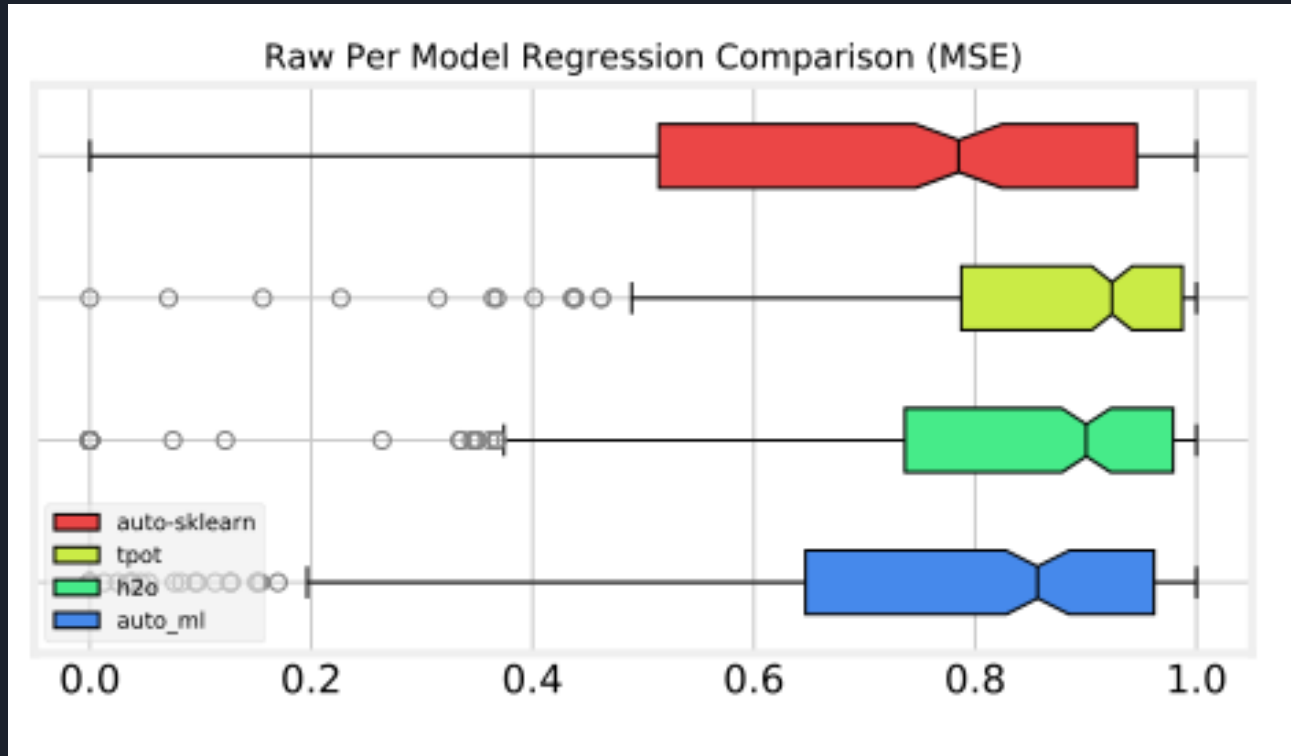


III SETC

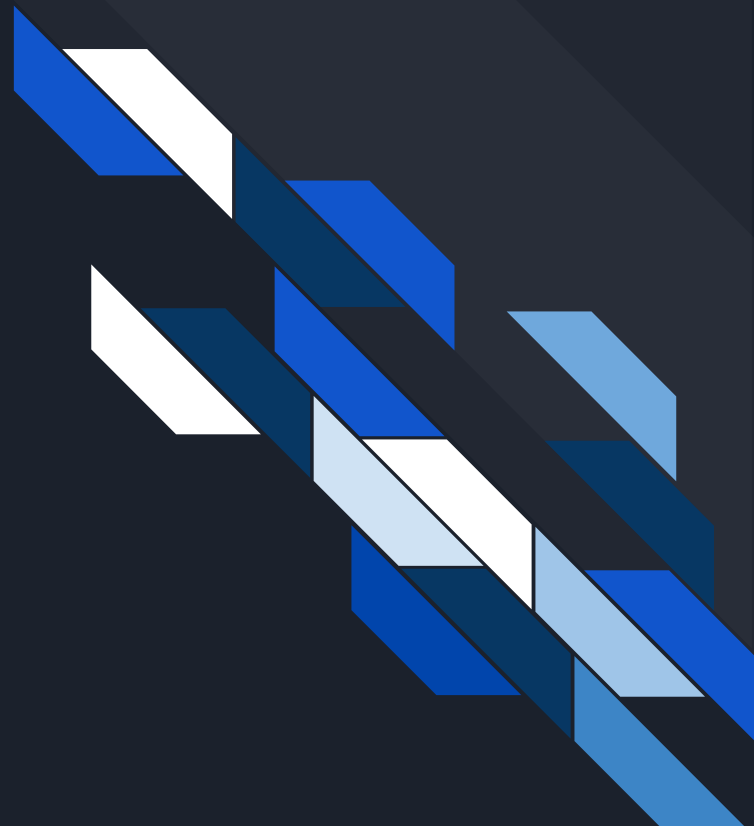
Comparação entre modelos:



Comparação entre modelos:



H2O AutoML





H2O AutoML

- Pré-processamento dos dados
- *Random Grid Search* em parâmetros pré-definidos
- *Early Stopping*
- *Random Forest, GBM, GLM, Deep Learning*
- Rankeamento dos modelos



H2O AutoML em R

```
library(h2o)

h2o.init()

train <- h2o.importFile("train.csv")

aml <- h2o.automl(y = "response_colname",
                 training_frame = train,
                 max_runtime_secs = 600)

lb <- aml@leaderboard
```





H2O AutoML em Python

```
import h2o
from h2o.automl import H2OAutoML

h2o.init()

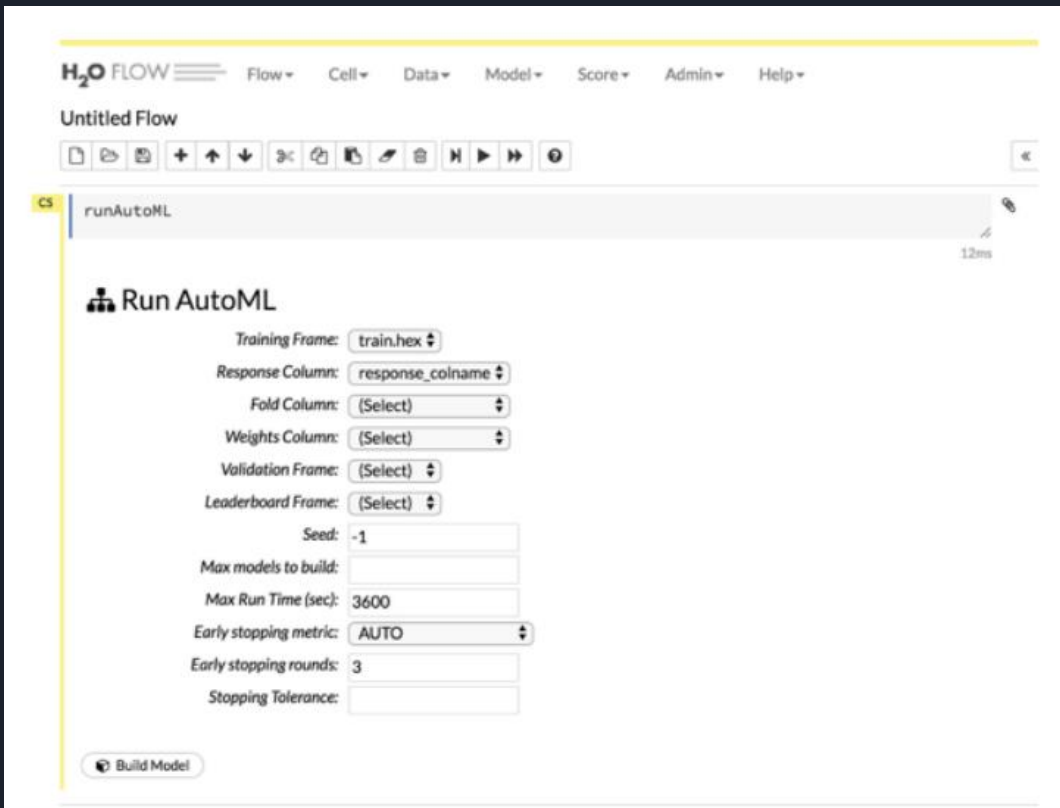
train = h2o.import_file("train.csv")

aml = H2OAutoML(max_runtime_secs = 600)
aml.train(y = "response_colname",
          training_frame = train)

lb = aml.leaderboard
```



H2O AutoML em FLOW



The screenshot displays the H2O FLOW web interface. At the top, the navigation bar includes the H2O FLOW logo and menu items: Flow, Cell, Data, Model, Score, Admin, and Help. Below the navigation bar, the title 'Untitled Flow' is shown. A toolbar with various icons for file operations and execution is visible. The main workspace contains a single cell named 'runAutoML' with a duration of 12ms. The 'Run AutoML' configuration panel is expanded, showing the following settings:

- Training Frame: train.hex
- Response Column: response_colname
- Fold Column: (Select)
- Weights Column: (Select)
- Validation Frame: (Select)
- Leaderboard Frame: (Select)
- Seed: -1
- Max models to build: (empty field)
- Max Run Time (sec): 3600
- Early stopping metric: AUTO
- Early stopping rounds: 3
- Stopping Tolerance: (empty field)

At the bottom of the configuration panel, there is a 'Build Model' button.



H2O AutoML Leaderboard

model_id	auc	logloss
StackedEnsemble_AllModels_0_AutoML_20171121_012135	0.788321	0.554019
StackedEnsemble_BestOfFamily_0_AutoML_20171121_012135	0.783099	0.559286
GBM_grid_0_AutoML_20171121_012135_model_1	0.780554	0.560248
GBM_grid_0_AutoML_20171121_012135_model_0	0.779713	0.562142
GBM_grid_0_AutoML_20171121_012135_model_2	0.776206	0.564970
GBM_grid_0_AutoML_20171121_012135_model_3	0.771026	0.570270
DRF_0_AutoML_20171121_012135	0.734653	0.601520
XRT_0_AutoML_20171121_012135	0.730457	0.611706
GBM_grid_0_AutoML_20171121_012135_model_4	0.727098	0.666513
GLM_grid_0_AutoML_20171121_012135_model_0	0.685211	0.635138





Referências:

- Cook, Darren. *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI.* " O'Reilly Media, Inc.", 2016.
- Documentação H2O: <http://docs.h2o.ai>
- Tutoriais H2O: <https://github.com/h2oai/h2o-tutorials>
- Vídeos: <https://www.youtube.com/user/0xdata>



III SETC

contato: alansouza@ufmg.br

MÃOS NA MASSA

PARTE 2:

