



BigML

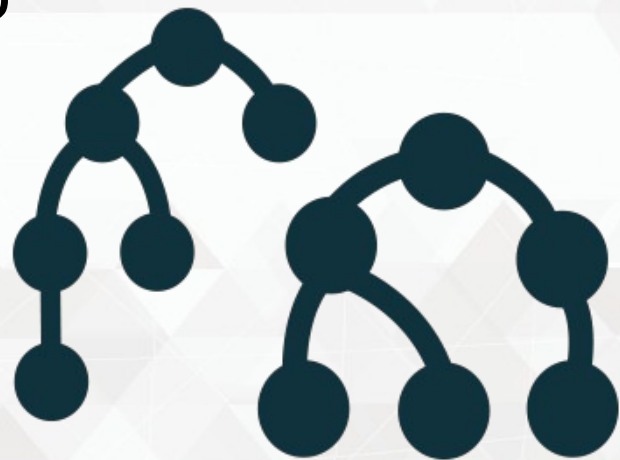
Ensembles – Modelos múltiples

2023

<lapti>

Programa

- Problemas com árvores de decisão
- O que são ensembles
- Tipos
- Criação
- Visualização
- Configuração



O que são ensembles?

- Em vez de criar um modelo unitário
 - ~ Combinar a saída de vários modelos tipicamente “fracos” em um conjunto com mais capacidade
- Questões
 - ~ Porque isso é necessário?
 - ~ Como criamos modelos com menor abrangência?
 - ~ Como combinar esses modelos?

Não existem modelos perfeitos

- Um dado algoritmo de ML pode simplesmente **não ser capaz** de modelar exatamente uma “solução real” para um dataset em particular
 - ~ Tentativa de aproximar uma linha a uma curva
- Mesmo se o modelo é bastante capaz, a “solução real” pode ser elusiva
 - ~ DT/NN podem modelar qualquer fronteira de decisão com dados suficientes
 - Mas a solução é NP-hard (Não-determinístico, tempo Polinomial)
 - ~ Algoritmos práticos envolvem processos randômicos
 - Podem chegar a soluções diferentes, dependendo de vários fatores
 - ~ Em teoria “igualmente boas” soluções
- E pode piorar...

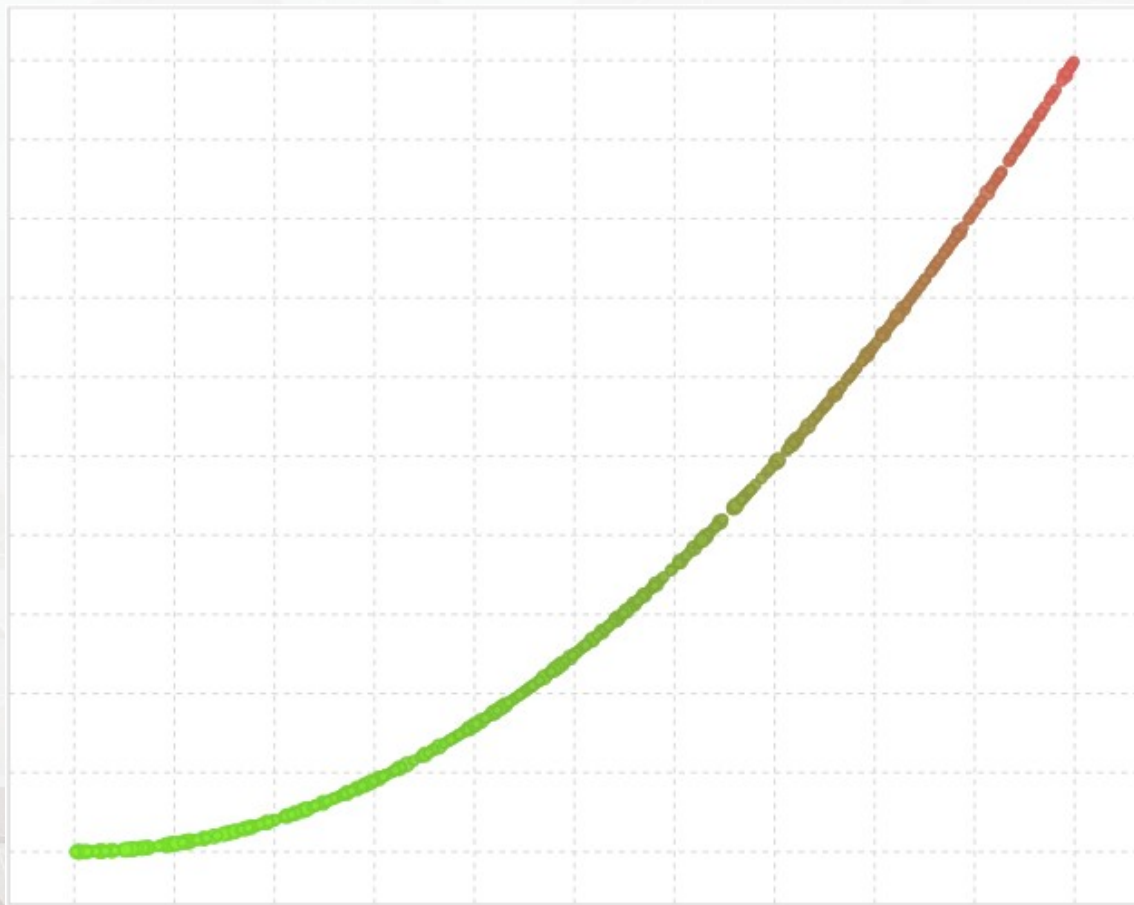
Não existem dados perfeitos

- Sem dados suficientes
 - ~ Sempre trabalhando com dados de treinamento finitos
 - ~ Então, cada “modelo” é uma aproximação da “solução real” e podem existir várias boas aproximações
- Anomalias / outliers
 - ~ O modelo tenta generalizar a partir de dados de treinamento discretos
 - ~ Outliers podem “desviar” o modelo, por overfitting
- Erros em dados
 - ~ O modelo não deve fazer tudo por você
 - ~ E **sempre** existem erros nos seus dados
 - Sério, sempre existe, eu queria estar só exagerando...

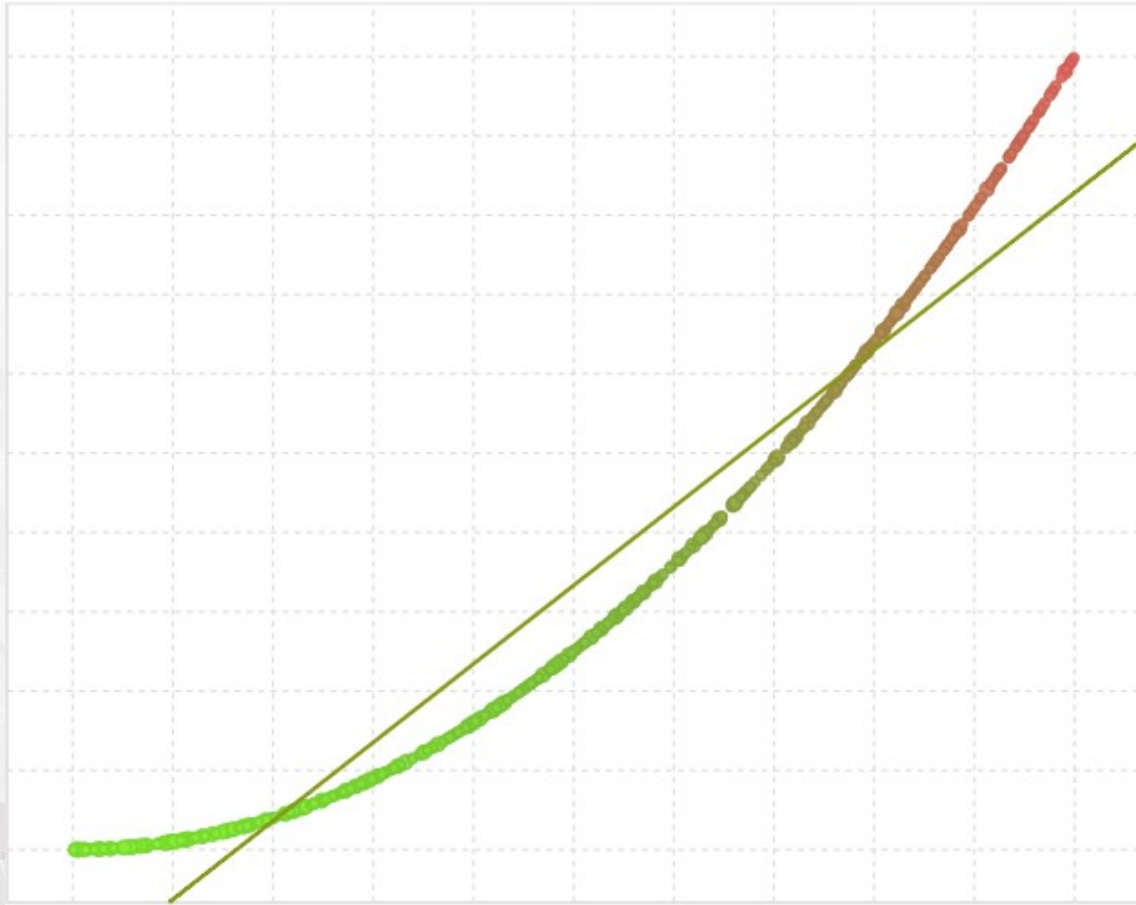
Técnicas de ensemble

- Ideia principal:
 - ~ Pela combinação de vários “bons modelos”, o resultado pode ser mais próximo do “melhor modelo possível”
 - ~ É necessário garantir diversidade
 - Um ensemble de 1000 modelos muito semelhantes não é útil
- Estratégias para dados de treinamento
 - ~ Construir vários modelos, cada um com somente parte dos dados
 - Tanto em linhas quanto em colunas
 - ~ Introduzir aleatoriedade diretamente no algoritmo
 - ~ Adicionar pesos no treinamento para “focar” em modelos adicionais onde erros são elevados
- Estratégias para predição
 - ~ Modelar os erros
 - ~ Modelar a saída de vários algoritmos diferentes

Exemplo



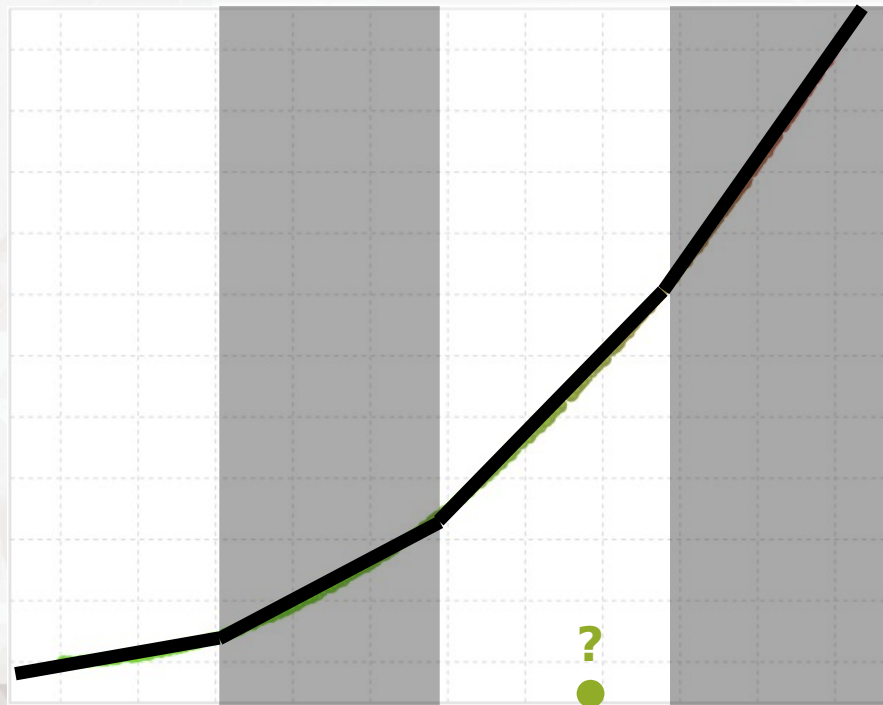
Exemplo



Exemplo

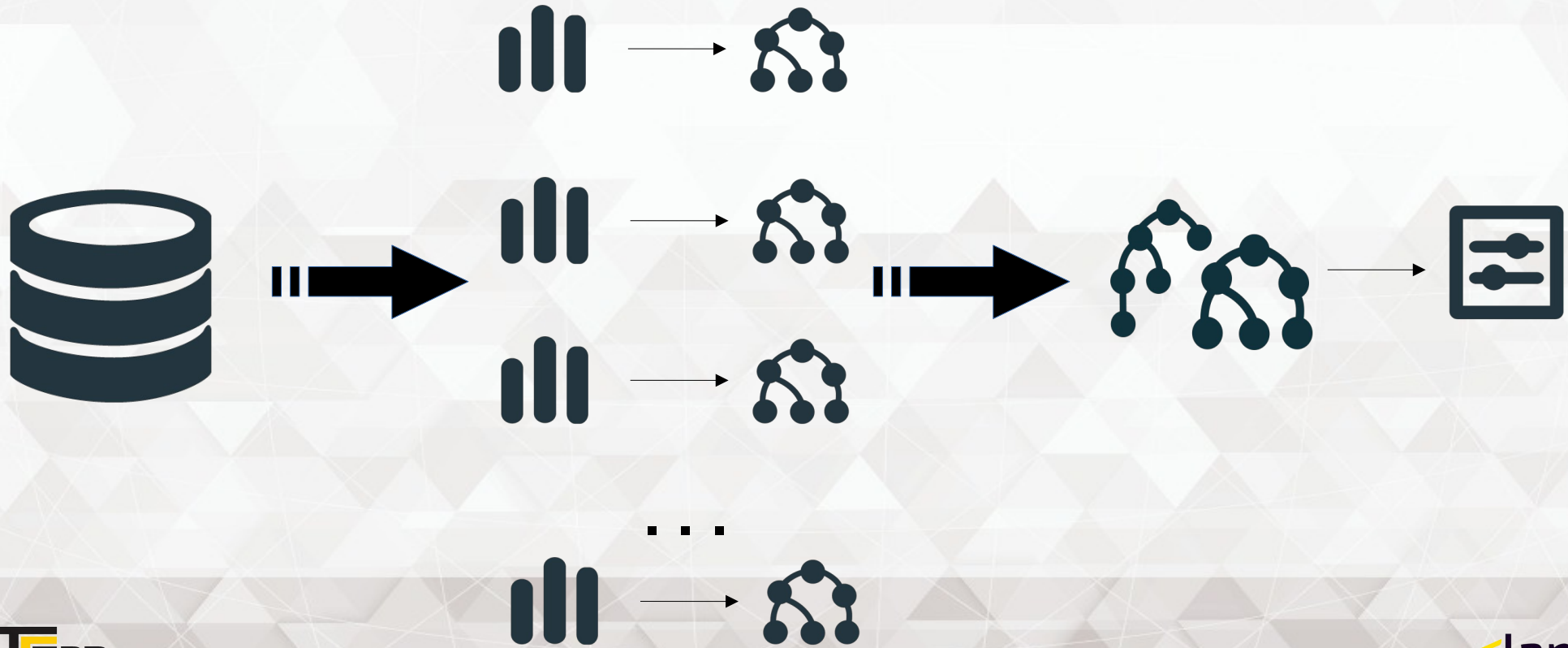
Particionar os dados...

então modelar cada partição...



Para predições, usar o modelo da mesma partição

Decision Forest



Tipos de ensembles

- BigML provê 3 tipos

- ~ Bagging

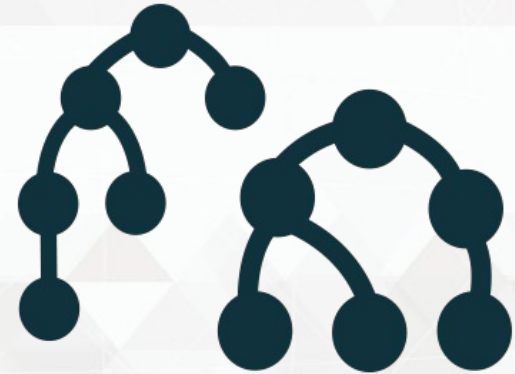
- Divide dataset em porções aleatórias
 - ~ Somente linhas
 - ~ Simples, mas com um desempenho muito bom

- ~ Random Decision Forests

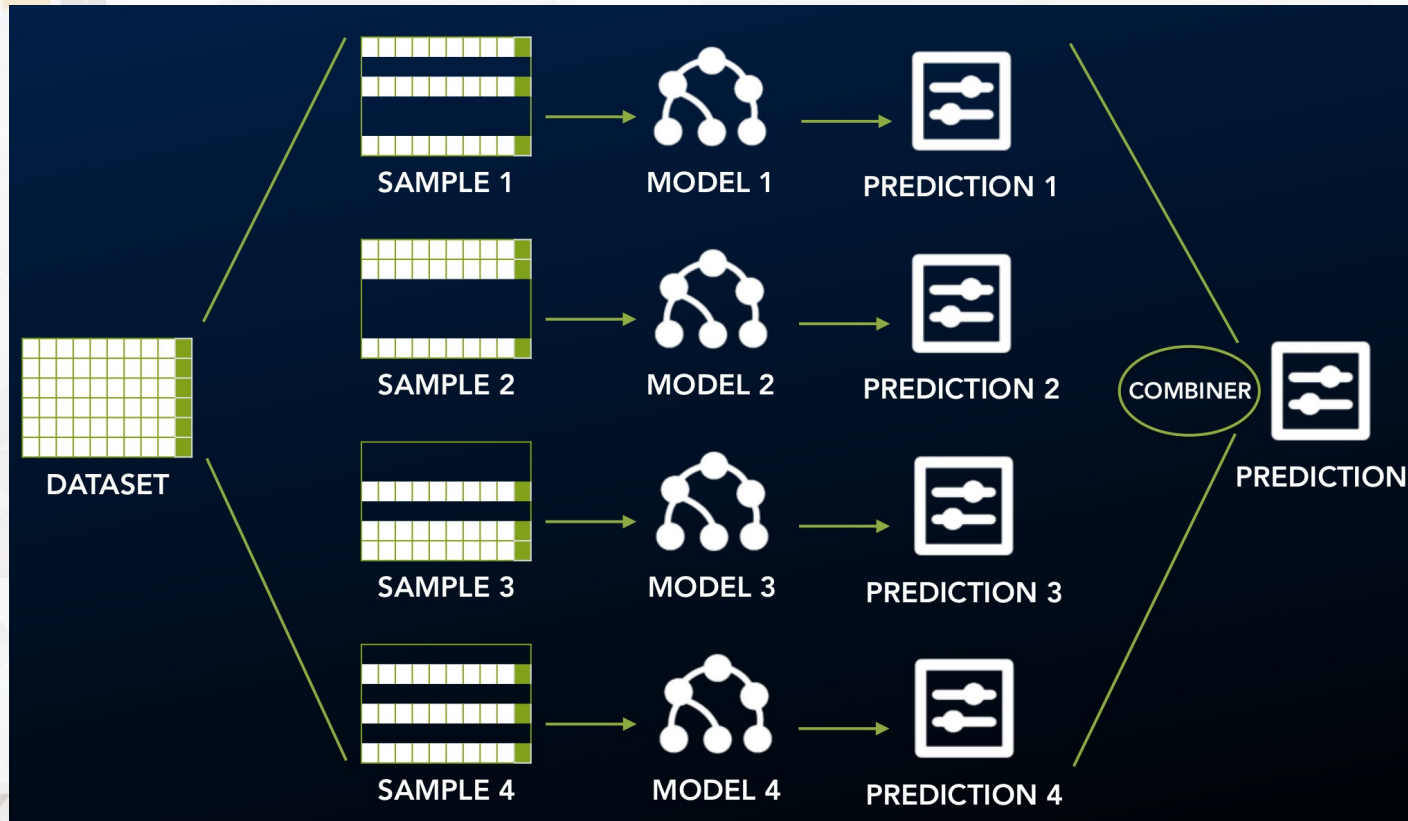
- Divide dataset em porções aleatórias
 - ~ Linhas e também features

- ~ Boosted Trees

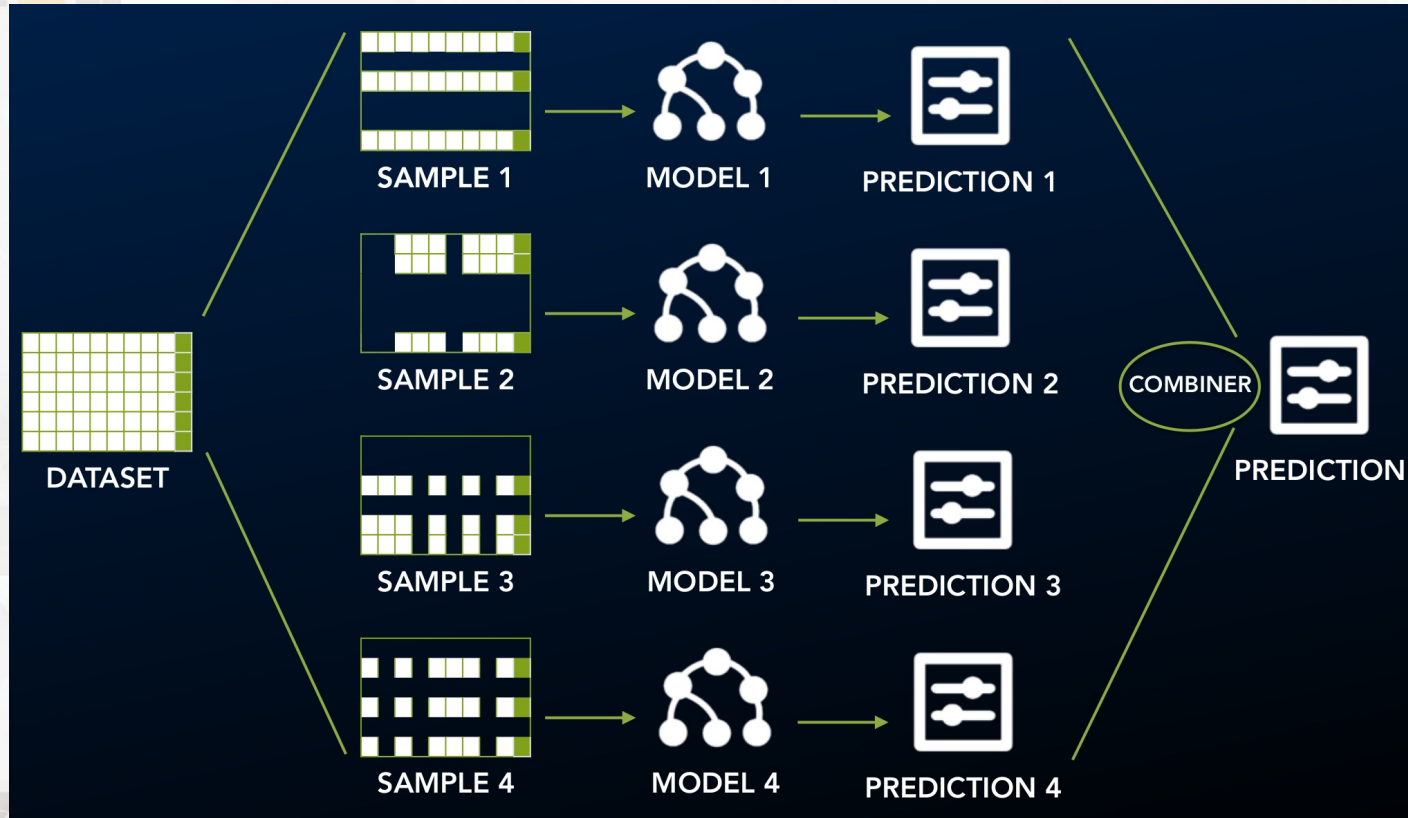
- Gradient Boosted Trees
 - Várias iterações de “weak learners” com resultado combinado
 - Em toda “boosting iteration”, cada modelo tenta corrigir os erros da iteração anterior
 - ~ Otimizando uma função de perda



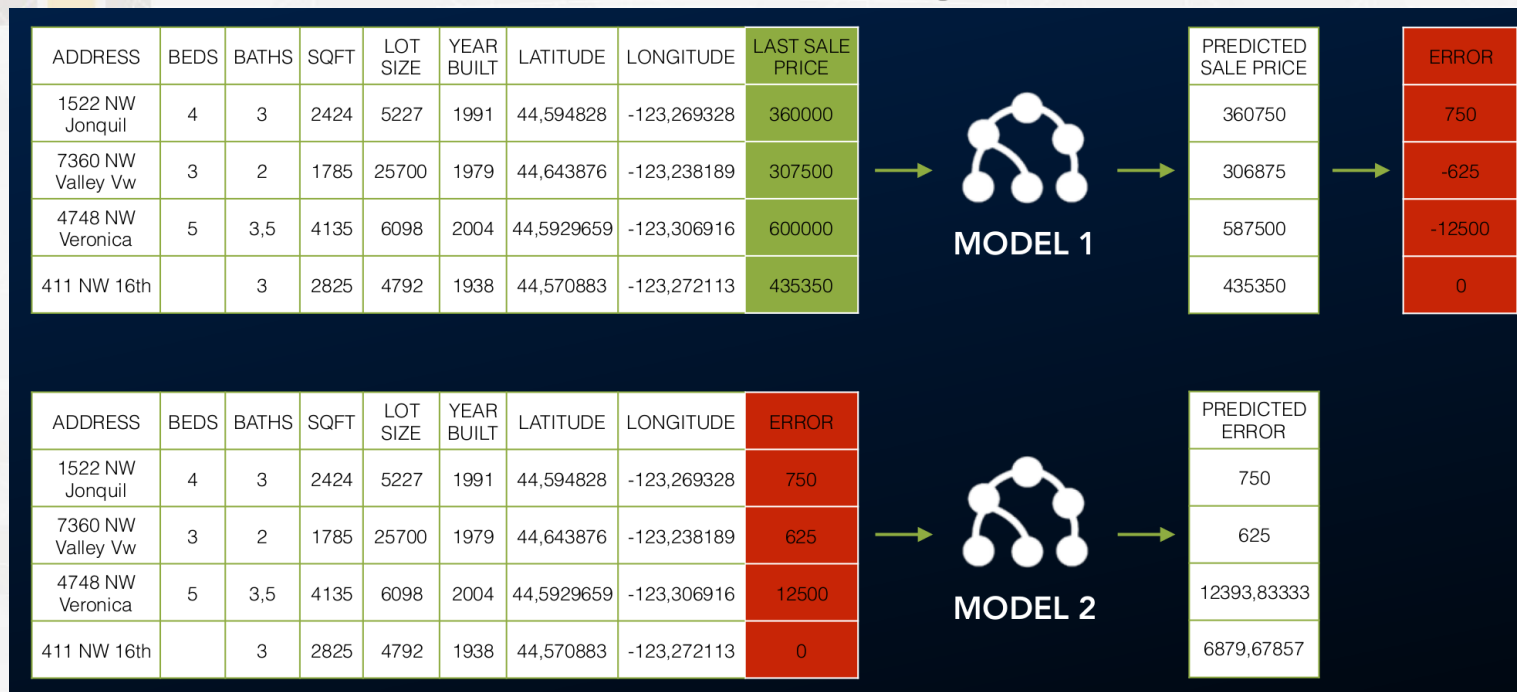
Decision Forest - bagging



Random Decision Forest



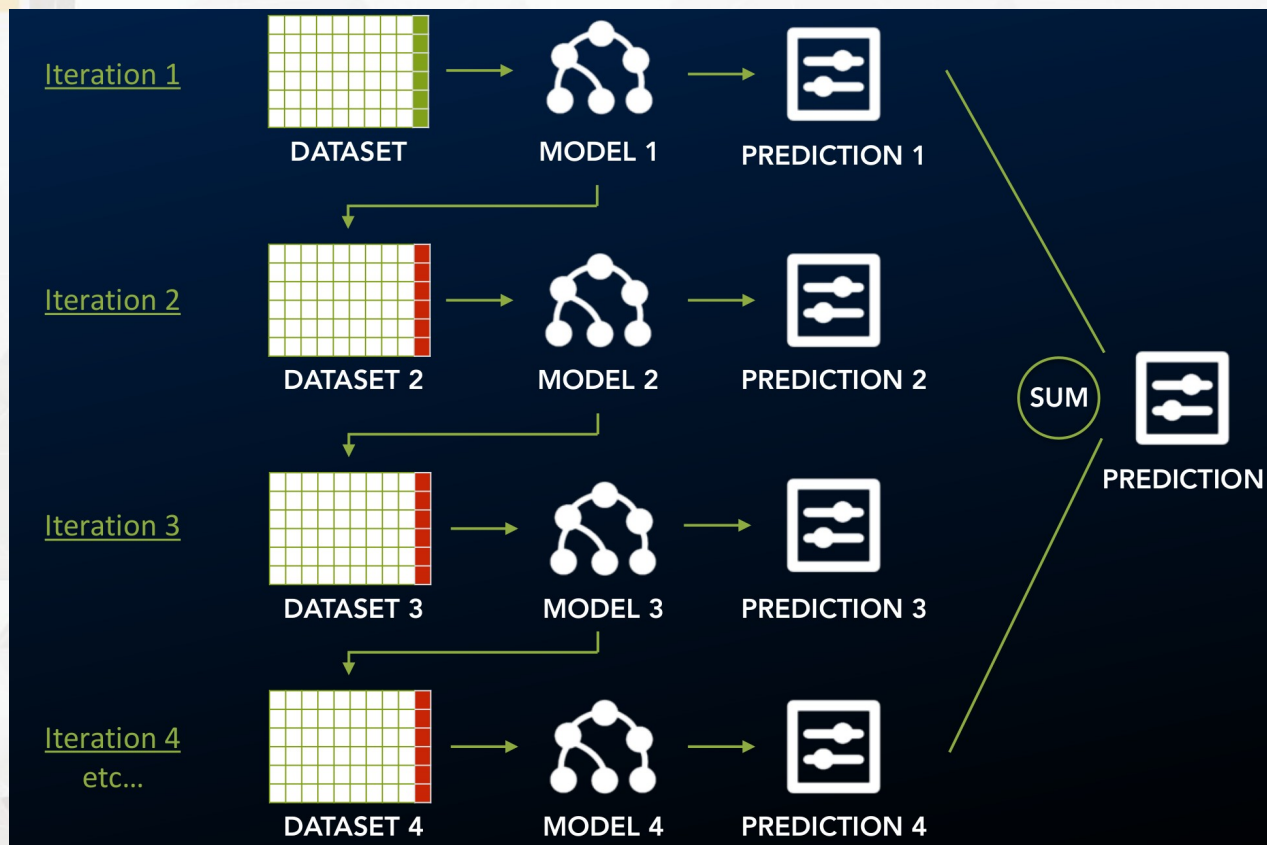
Boosting



Model1: qual é a predição para o preço de venda desta casa?

Model2: quanto de erro Model1 teve?

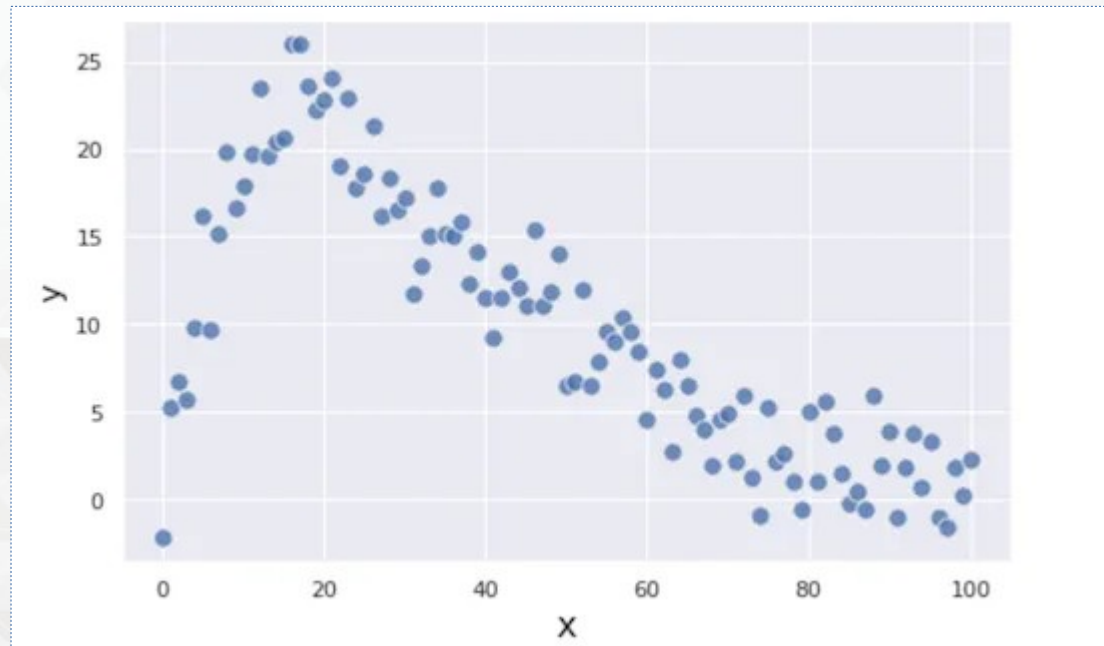
Boosting



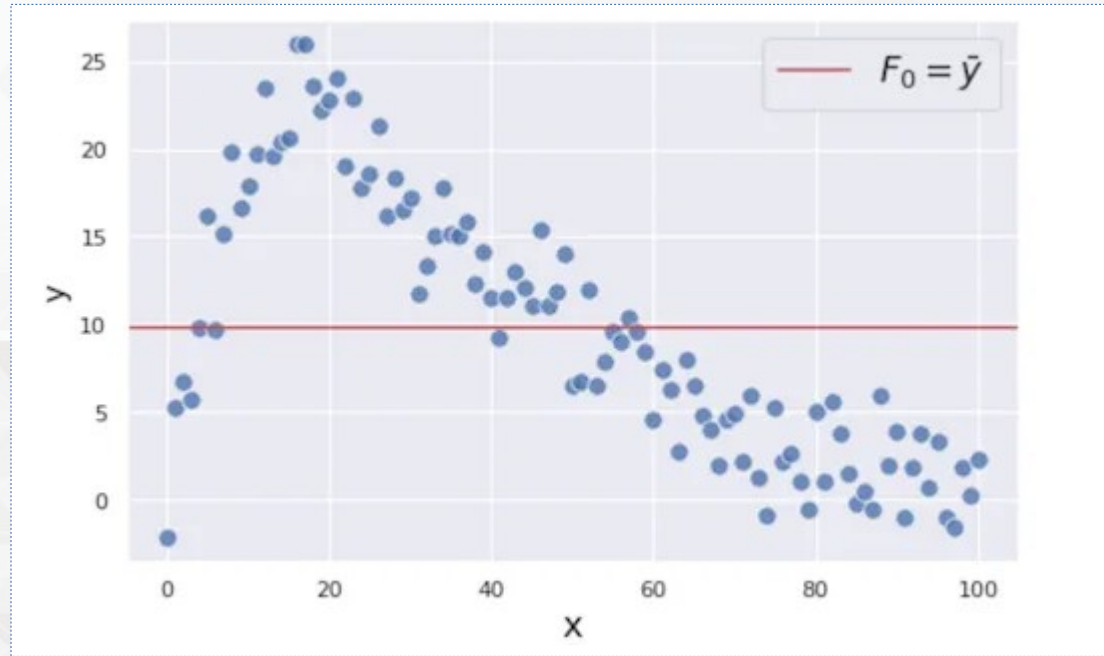
Boosting



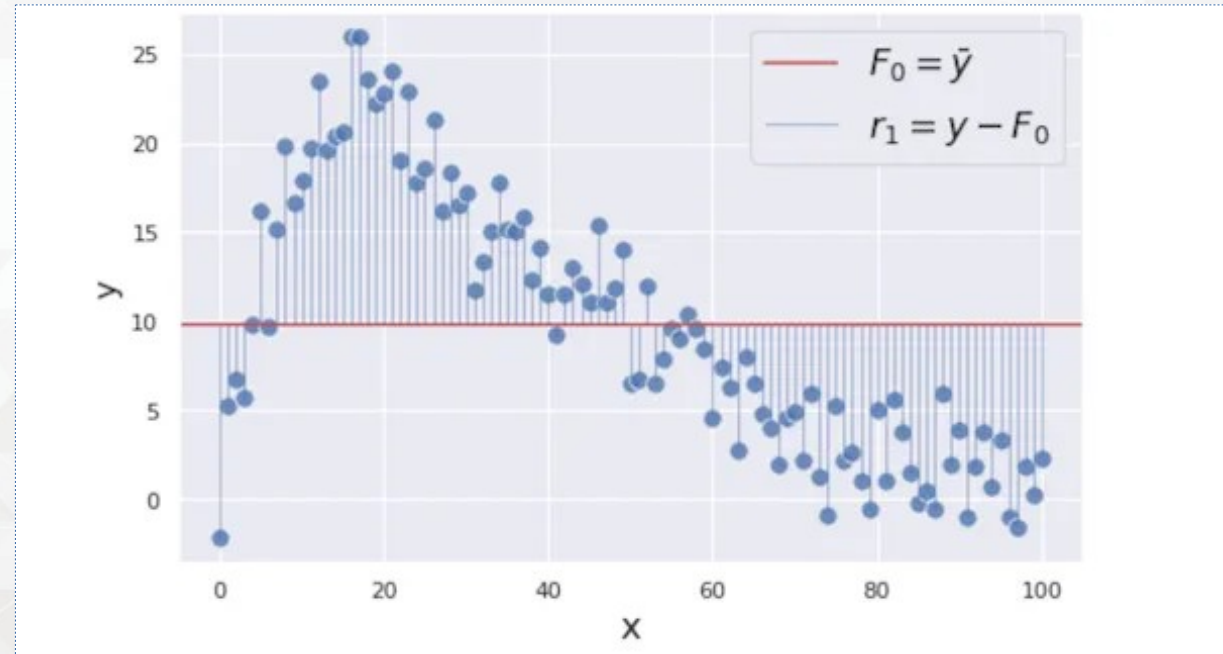
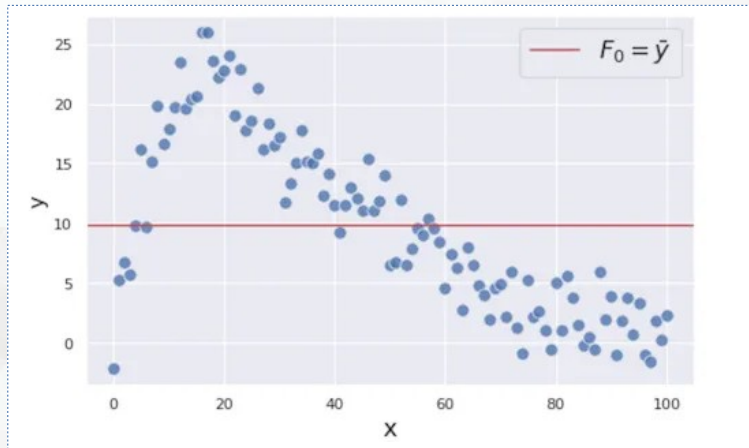
Boosting



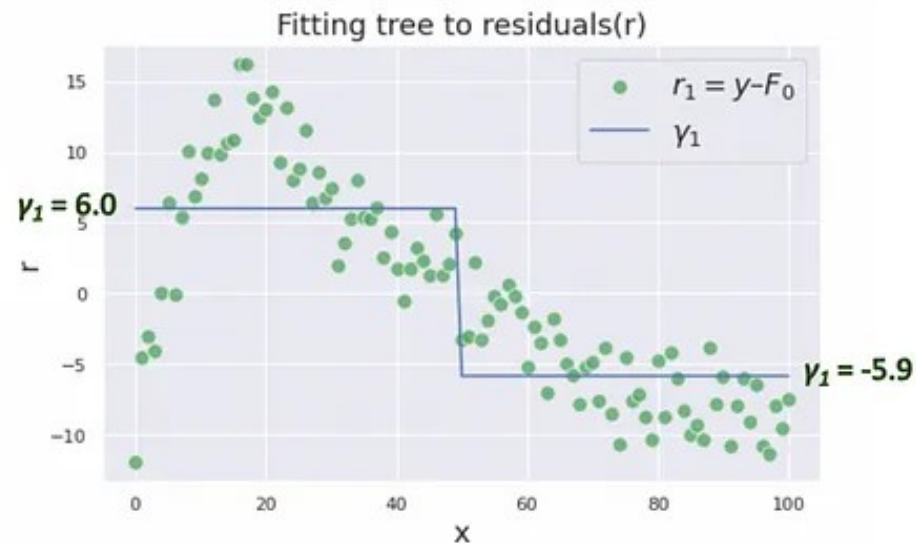
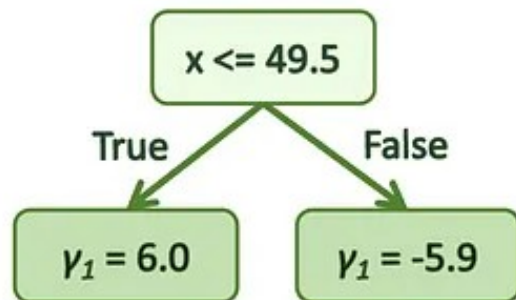
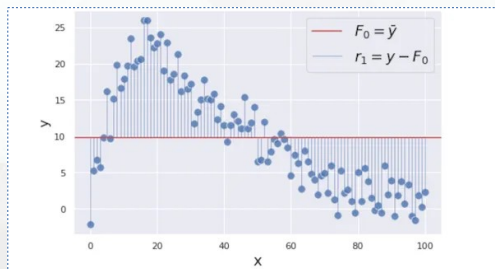
Boosting



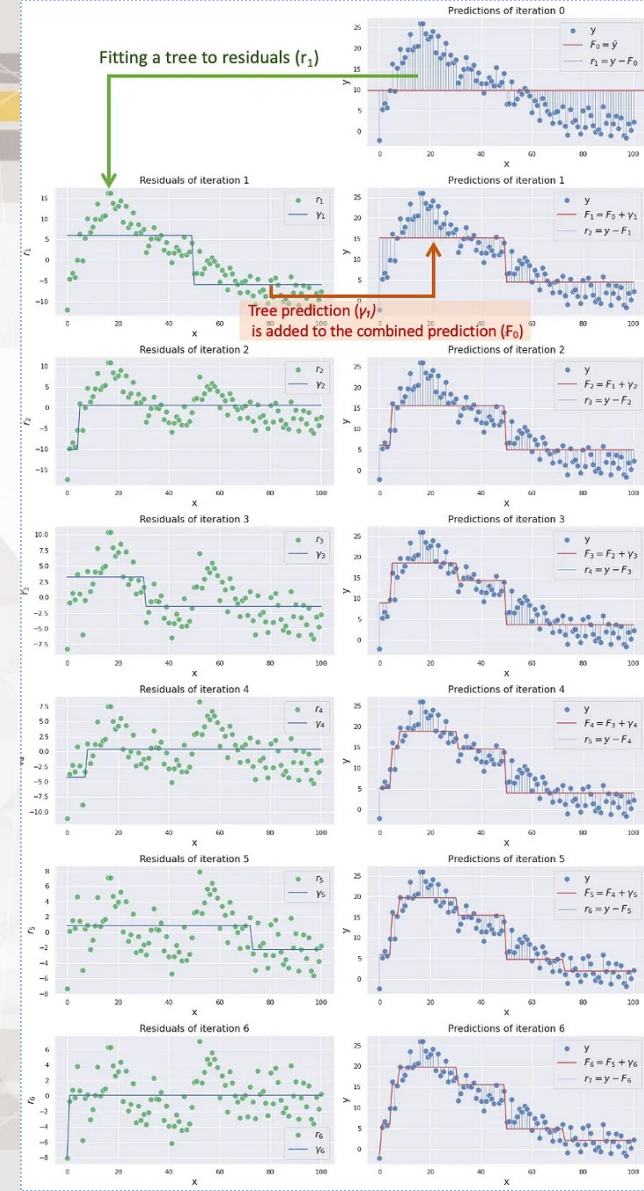
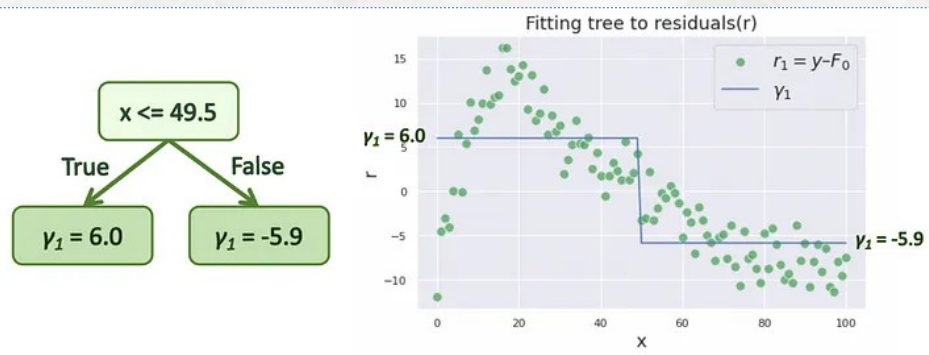
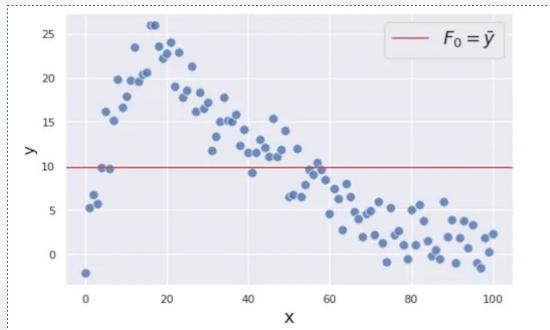
Boosting



Boosting



Boosting



Configurações

- Parâmetros individuais de árvores ainda estão disponíveis
 - ~ Balanceamento de objetivo, missing splits, profundidade de nós, etc
- Número de modelos
 - ~ Quantas árvores criar
- Opções de sampling
 - ~ Determinístico / Randômico
 - ~ Reposição
 - ~ Features consideradas a cada split (bagging / random forest)
- Em tempo de predição
 - ~ Combiner

Configuração Boosting

- Número de iterações
 - ~ Parecido com número de modelos para DF/RDF
- Iterações podem ser limitadas por Early Stopping
 - ~ Early out of bag: testa com amostras out-of-bag
 - ~ Early holdout: testa com uma parte do dataset
 - ~ None: faz todas as iterações
 - Geralmente é melhor usar um alto número de iterações e deixar o Early Stopping trabalhar
- Learning rate
 - ~ Controla quão agressivamente o boosting vai tentar se adequar aos dados (fit)
 - Valores maiores deixam o modelo mais rápido (converge mais rápido), mas pode gerar overfitting
- Sampling e Replacement
- Parâmetros individuais de árvores

Configurações

Sources **Datasets** Supervised ^{NEW} Unsupervised Predictions Tasks WhizzML

Sentiment

ENSEMBLE CONFIGURATION

Objective field: retweet_count 123

☒ Automatic optimization Max. training time: 00:30:00 Ensemble candidates: 128

Type: Decision Forest Number of models: 10 Number of iterations: 64

Advanced configuration

Ensemble name: Sentiment

Reset Create ensemble

Sources **Datasets** Supervised Unsupervised ^{NEW} Predictions ^{NEW} Tasks WhizzML

Diabetes diagnosis dataset | Training (70%)

ENSEMBLE CONFIGURATION

Objective field: Diabetes ABC



Type: Decision Forest
Decision Forest
Boosted Trees

Number of models: 10 Number of iterations: 10

Advanced configuration

Ensemble name: Diabetes diagnosis dataset | Training (70%)'s ensemble

Reset Create ensemble

Name	Type	Count	Missing	Errors	Histogram
Pregnancies	123	537	0	0	
Glucose	123	537	0	0	

Configurações

Sources **Datasets** Supervised Unsupervised ^{NEW} Predictions ^{NEW} Tasks WhizzML

Absenteeism_at_work [filtered] | Training (69%)

ENSEMBLE CONFIGURATION

Objective field: Absenteeism ABC ☐ Automatic optimization

Type: Decision Forest Number of models: 10 Number of iterations: 64

Advanced configuration

Tree:

Statistical pruning: ml tree ensemble STATISTICAL PRUNING AUTO

Missing splits: tree MISSING SPLITS NO

Node threshold: 512 NODE THRESHOLD 512

Randomize: tree RANDOMIZE Default

Random candidates: Default

Boosting:

Weights:

Sources **Datasets** Supervised Unsupervised Predictions Tasks WhizzML

Diabetes diagnosis dataset | Training (70%)

ENSEMBLE CONFIGURATION

Objective field: Diabetes ABC Type: ^{NEW} Boosted Trees Number of models: 10 Number of iterations: 10

Advanced configuration

Tree:

Boosting:

Early stopping: Early out of bag EARLY STOPPING Early out of bag

Learning Rate (LR): 10% LEARNING RATE 10%

Weights:

Tree sampling:

Dataset sampling: 537 instances

Agregação de resultados

- Decision Forests

- ~ Predições de cada árvore são agregadas em uma média para a predição final
- ~ Medidas de qualidade também
 - Confidence, probabilidades, erro esperado (regressões)
- ~ Em classificação, as medidas por classe são calculadas separadamente
 - Classe com maior probabilidade ou confiança é retornada
 - Pode ser calculada por “votação”, baseada no número de árvores decidindo por cada classe

- Boosted Trees

- ~ Modelo é baseado em adição, não em média
- ~ Probabilidade é resultado de classificação
 - Sem confidence
- ~ Peso de cada boosting é utilizado
 - E gerado para cada caso
- ~ Vetor de somas com peso é transformada em probabilidade de classes por uma função softmax

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

Combiner



MODEL 1

VERDADERO	FALSO
80 %	20 %



MODEL 2

VERDADERO	FALSO
40 %	60 %



MODEL 3

VERDADERO	FALSO
60 %	40 %



ENSEMBLE

VERDADERO	FALSO
60 %	40 %

TRUE: $(80 + 40 + 60) / 3 = 60$

FALSE: $(20 + 60 + 40) / 3 = 40$



MODEL 1

SALES	ERROR
200 \$	2,40 \$



MODEL 2

SALES	ERROR
250 \$	2,10 \$



MODEL 3

SALES	ERROR
180 \$	1,45 \$



ENSEMBLE

SALES	ERROR
210 \$	1,98 \$

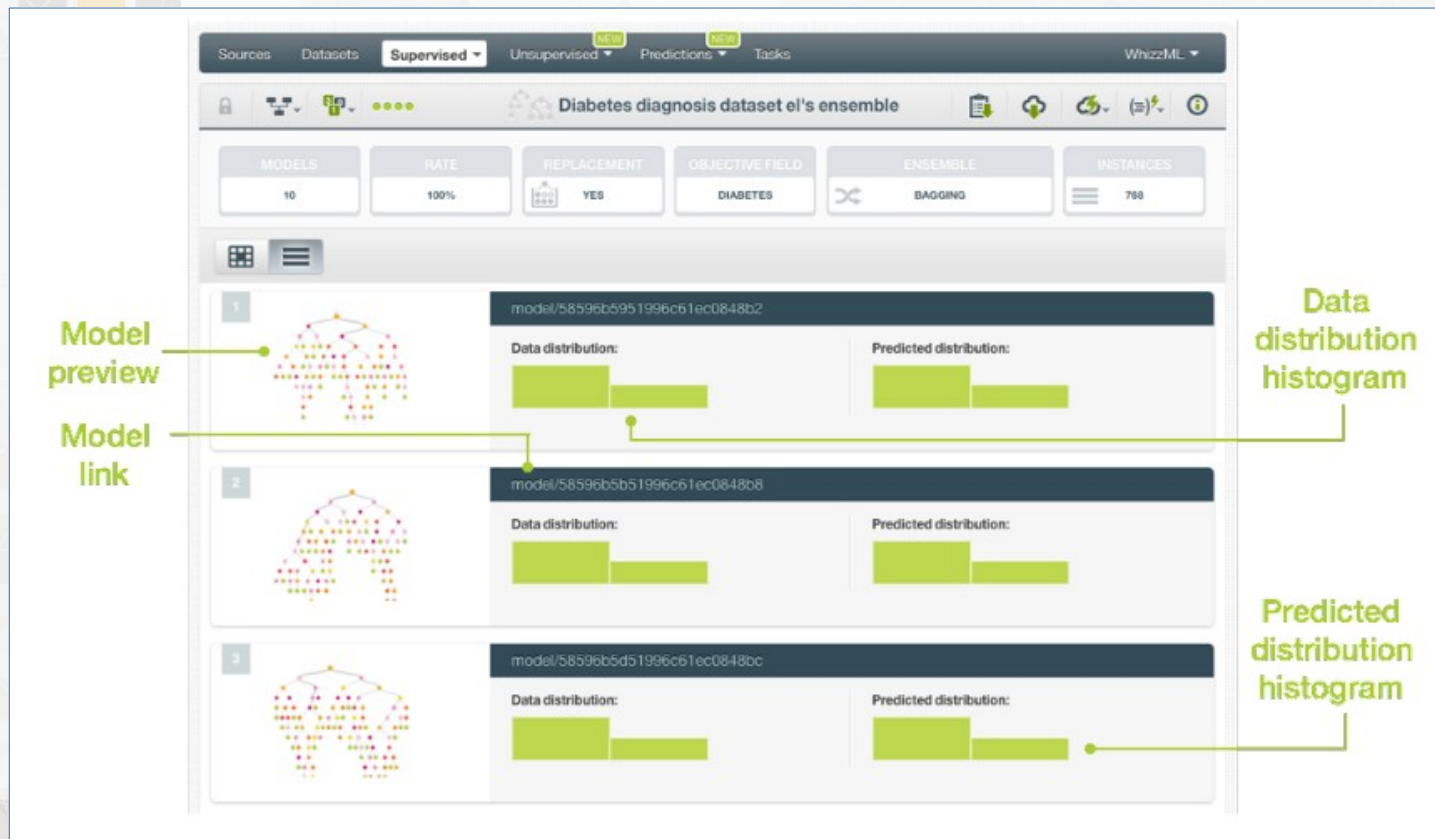
$(200 + 250 + 180) / 3 = \textbf{\$210}$

$(2.4 + 2.1 + 1.45) / 3 = \textbf{\$1.98}$

O que usar?

- **Avaliação é importante**
- Para datasets grandes / complexos
 - ~ DF/RDF com node threshold mais profundo
 - ~ Ou boosting com mais iterações
- Para dados com ruído
 - ~ Boosting pode gerar overfitting
 - ~ RDF são preferíveis
- Para dados “largos”
 - ~ RDF será mais rápido e provavelmente tão eficiente quanto
- Para dados simples
 - ~ Modelo único pode ser adequado, com a vantagem de interpretabilidade
- Classificação com número grande de classes
 - ~ Boosting pode ser lento, DF/RDF mais adequado
- Dados gerais/genéricos
 - ~ DF/RDF provavelmente melhores que modelo único ou boosting
 - ~ Boosting pode ser lento porque modelos são processados serialmente

Visualização



The background features a complex geometric pattern of overlapping triangles in shades of gray and white. At the top, there is a horizontal band with a yellow and gray geometric design, including a stylized 'U' shape. The word 'Obrigado' is centered in a large, black, sans-serif font.

Obrigado

leandro@utfpr.edu.br

<http://lapti.ct.utfpr.edu.br>

<lapti>