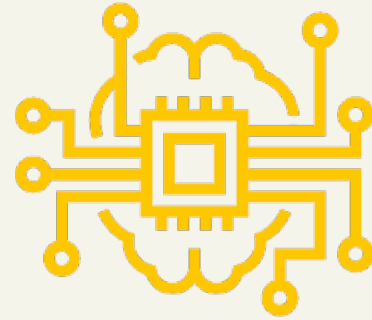


Aprendizado de Máquina
Ensemble Learning

Ricardo Sant'Ana

Introdução



1

Introdução

2

Classificador Bagging

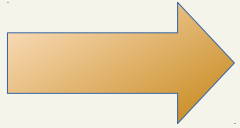
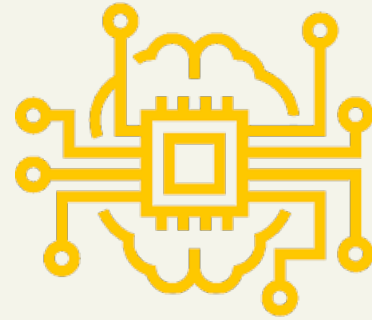
3

Classificador Boosting

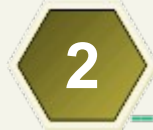
4

Conclusão

Introdução



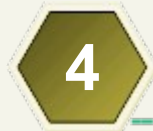
Introdução



Classificador Bagging



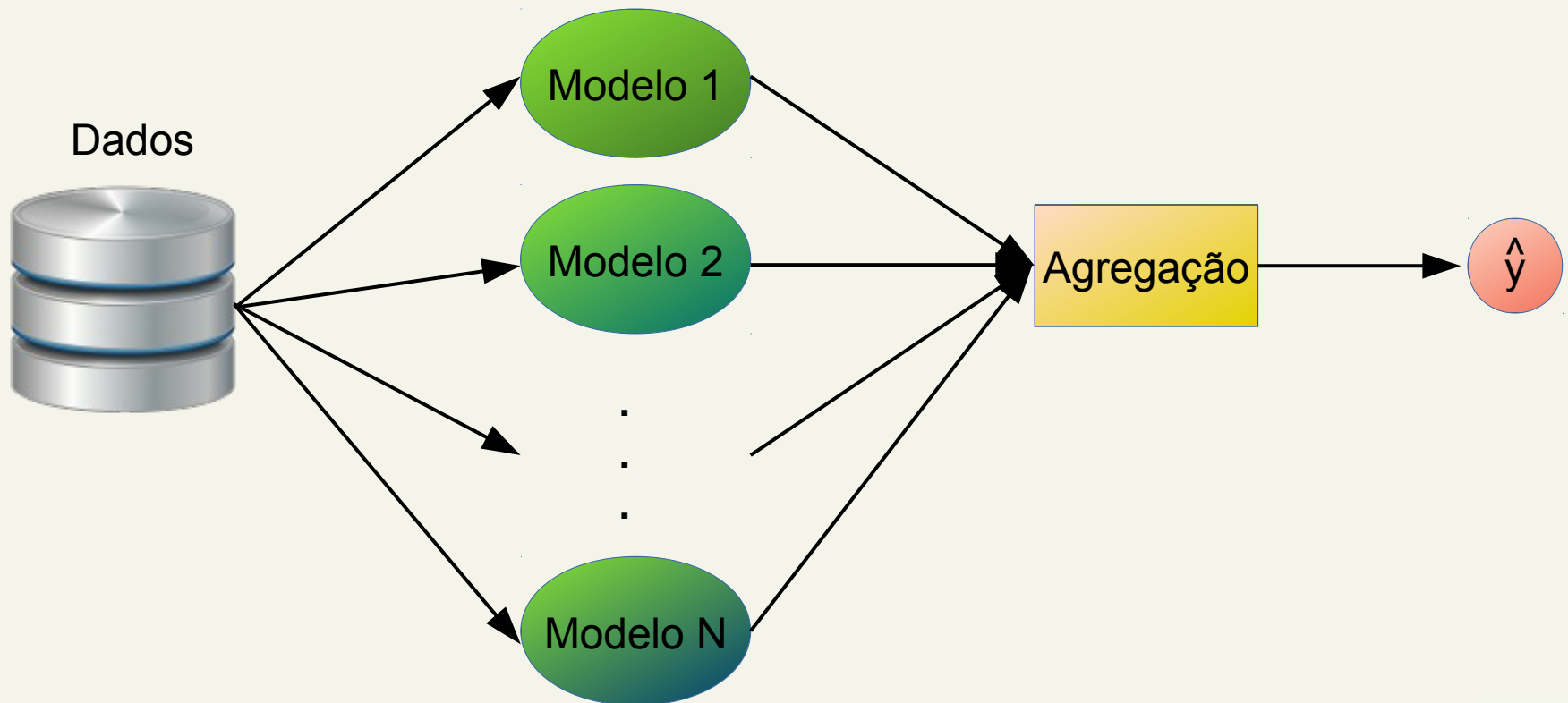
Classificador Boosting



Conclusão

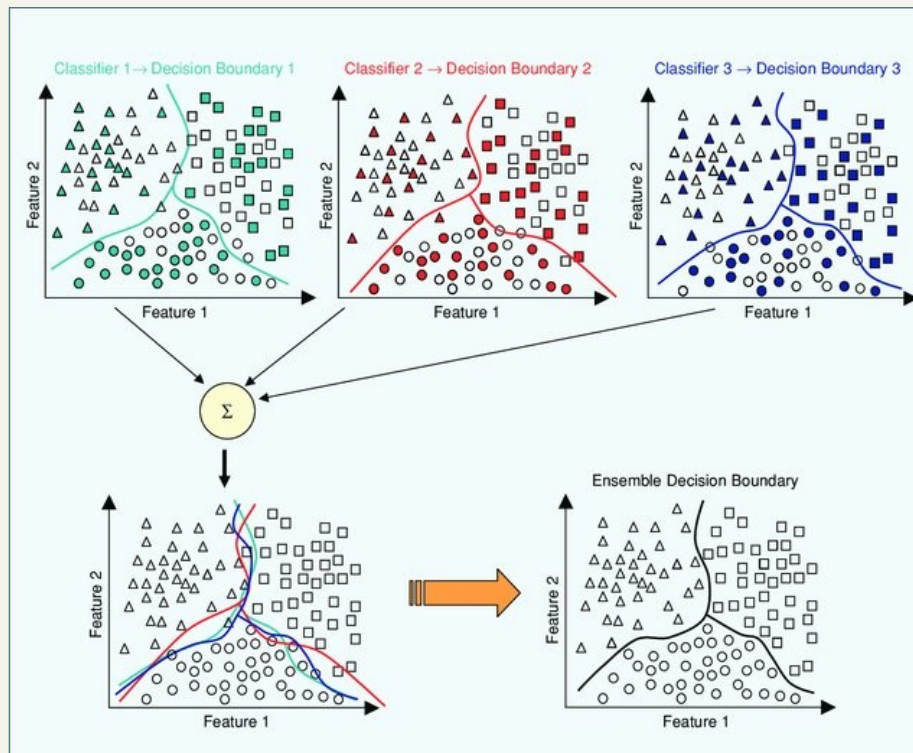
Introdução

- Visão Geral de Comitês
- Durante o treinamento de um comitê, o objetivo é construir uma coleção de preditores que, juntos, produzem uma predição melhor do que cada um separadamente.



Introdução

- Visão Geral de Comitês
- A fronteira de decisão correspondente a um modelo de comitê pode ser aproximada por meio de uma combinação apropriada dos diferentes preditores (classificadores ou regressores) componentes.



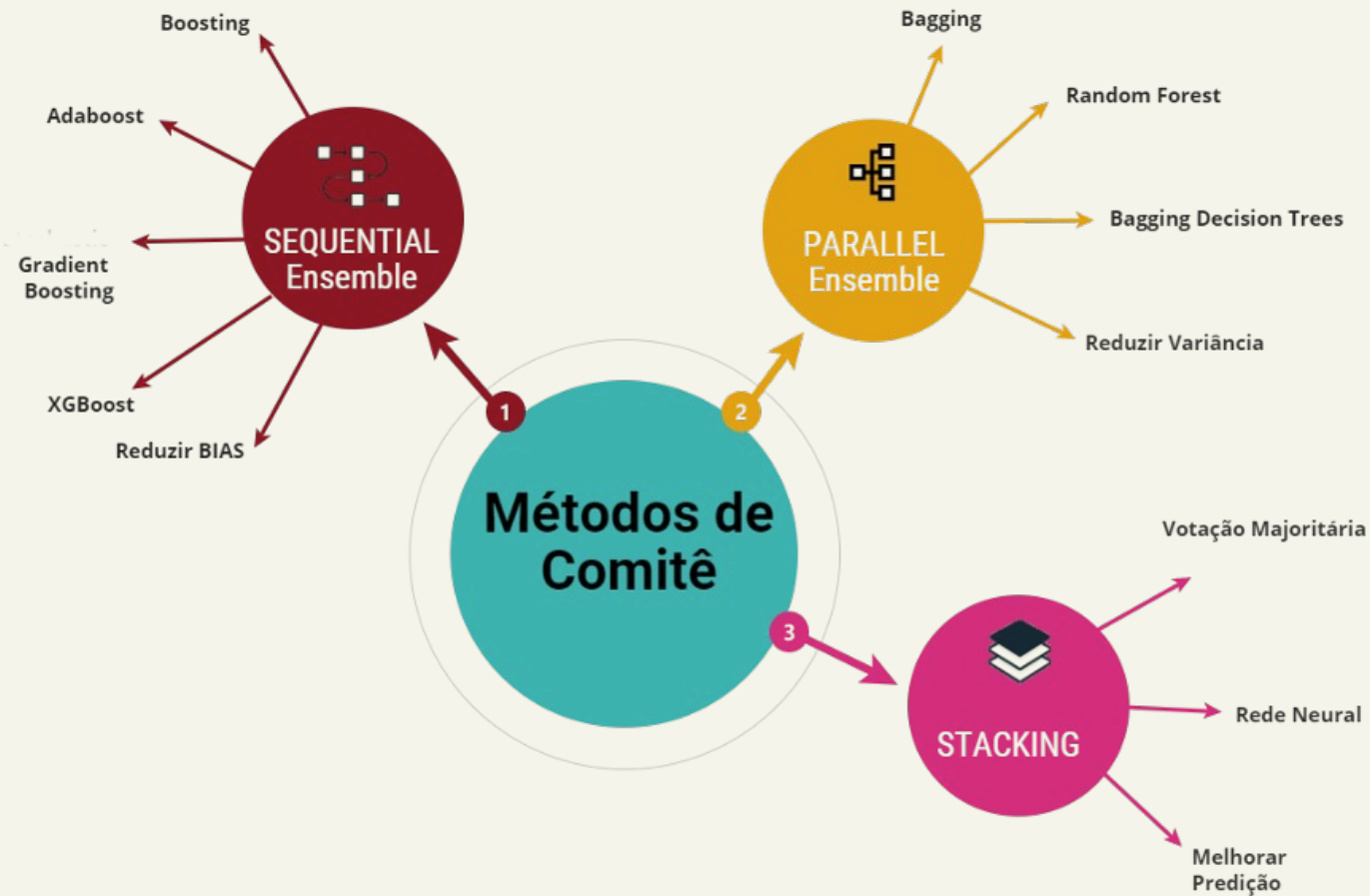
- **Comitês - Histórico**
- Tin Kam Ho [1] introduziu conceitos de Random Decision Forests em 1995 como forma de melhorar o poder preditivo das Árvores de Decisão.
 - Tarefa Classificação: dígitos manuscritos, NIST
 - Criou um conjunto de árvores de decisão no mesmo conjunto de dados e usou todas para realizar a predição.

- **Comitês - Histórico**
- Jerome Friedman[2] publicou em 2001 artigo sobre Gradient Boosting
 - Estratégia para melhorar qualquer algoritmo preditivo que ele chamou de *generalized gradient boosting*, que envolve minimizar uma função de perda.
 - No caso de árvores de regressão, a função de perda pondera (pesos) para cada folha dentro da árvore de forma diferencial, de modo que as folhas que fazem melhor predição são bem recompensadas e as que não o fazem são punidas.

- **Comitês - Histórico**
- Outra estratégia para melhorar os modelos de árvore de decisão foi desenvolvida por Leo Breiman[3], 2001.
 - Comumente conhecido como “bagging”, esta técnica cria um número adicional de conjuntos de treinamento por amostragem uniforme e com a substituição do conjunto de treinamento original.
 - **bootstrap aggregation** → (b agg) ing

Introdução

- ## Tipos de Comitês



Introdução

- *Um pouco de estatística...*

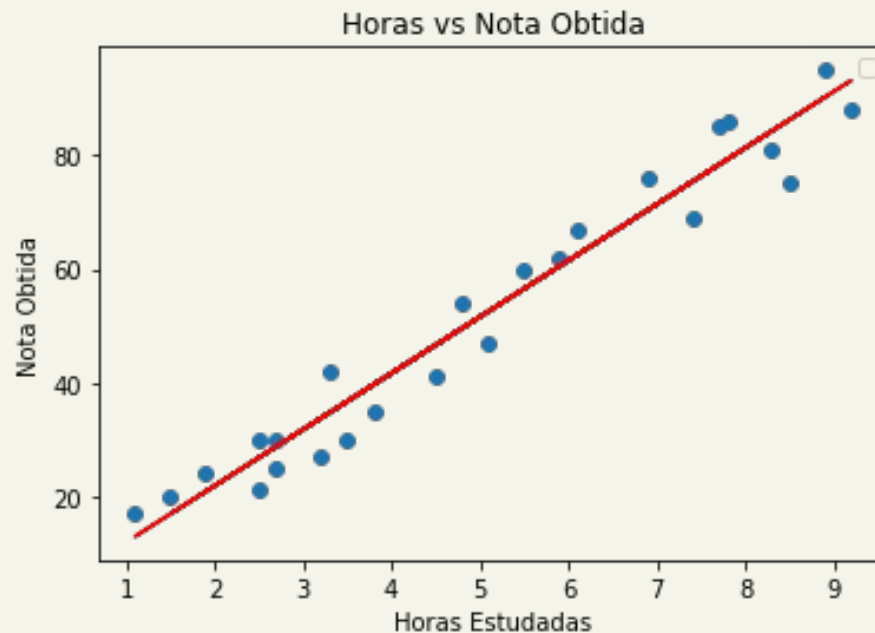


Introdução

- **Bias–Variance Tradeoff**
- Da Estatística, temos:
- $Y = f(X) + \varepsilon$ (I)
 - Y função que queremos aproximar para os pontos $x_1, x_2, x_3, \dots, x_n$ (conjunto de treinamento)
 - ε é um ruído com média 0 e variância σ .
- $\hat{Y} \rightarrow$ aproximação pelo algoritmo de aprendizado de máquina

Introdução

- **Bias–Variance Tradeoff**
- $Y=f(X) + \varepsilon \rightarrow$ pontos azuis
- $\hat{Y} \rightarrow$ aproximação por regressão linear



Introdução

- **Bias–Variance Tradeoff**
- $Y=f(X) + \varepsilon \rightarrow$ pontos azuis
- $\hat{Y} \rightarrow$ aproximação por regressão linear
- Considerando o valor esperado do erro médio quadrático, temos:
$$\text{Err}(X) = E [(Y - \hat{Y})^2] \quad (\text{II})$$

Introdução

- **Bias–Variance Tradeoff**
- Aplicando (I) em (II), temos

$$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

ou

$$\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$$

Introdução

- **Bias–Variance Tradeoff**

- $$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

ou

$$\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$$

- O viés é a diferença entre a previsão média de nosso modelo e o valor correto que estamos tentando prever.
- Podem ser erros causados por suposições incorretas no algoritmo de aprendizagem ou dados de treinamento não representativos da população de onde foram retirados.
- O modelo com alto viés presta muito pouca atenção aos dados de treinamento e simplifica demais o modelo.

Introdução

- **Bias–Variance Tradeoff**

- $$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

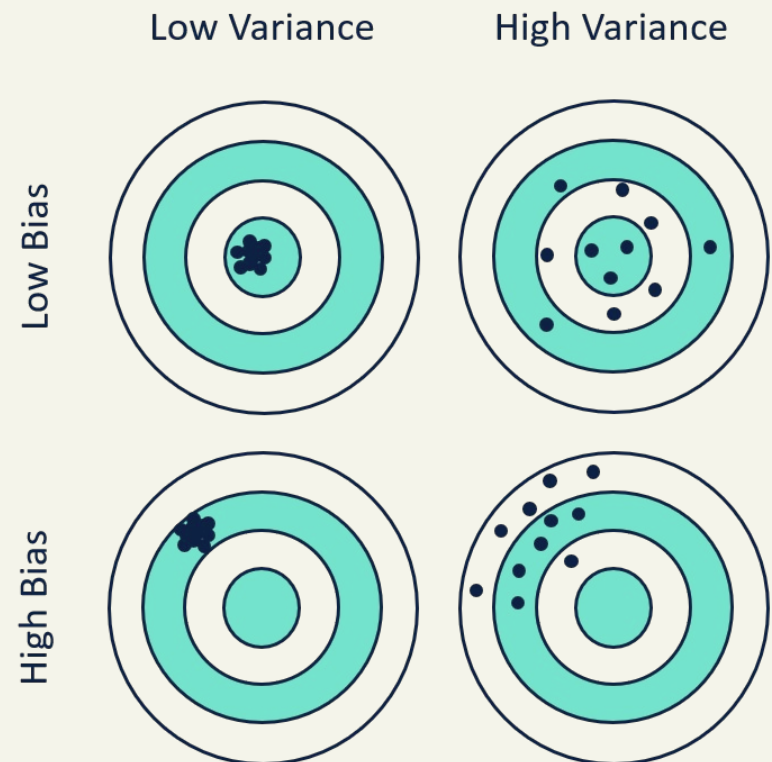
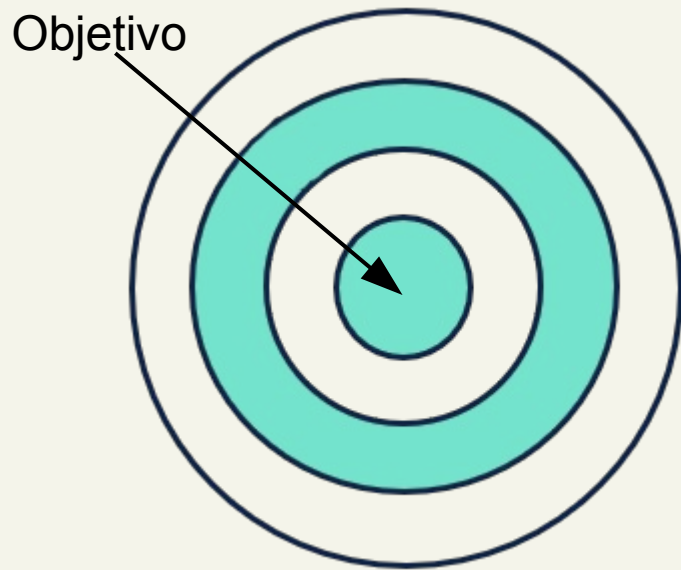
ou

$$\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irredutível}$$

- Variância é a variabilidade da previsão do modelo para um determinado ponto de dados (espalhamento das previsões).
- Sensibilidade a pequenas variações no conjunto de dados de treinamento. Pode estar dando muita atenção ao ruído.
- O modelo com alta variância dá muita atenção aos dados de treinamento e não generaliza sobre os dados que não viu antes.

Introdução

- **Bias–Variance Tradeoff**
- $\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$
- As amostras de teste → **todas concentradas no centro do alvo**



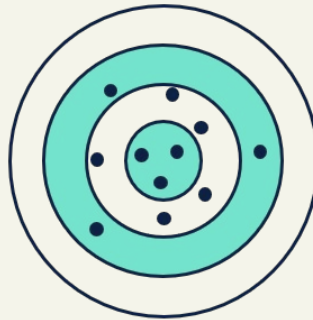
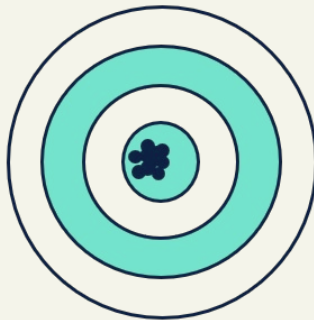
Introdução

- **Bias–Variance Tradeoff**
- $\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$
- As amostras de teste → **todas concentradas no centro do alvo**

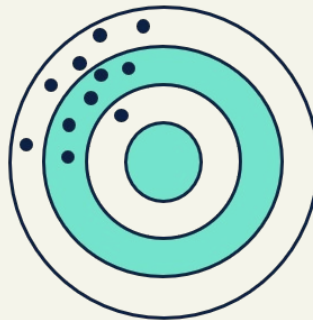
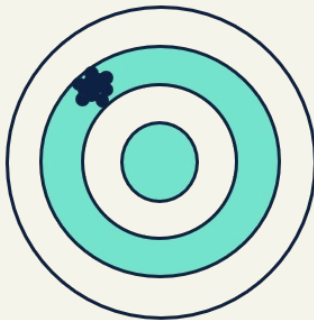
Low Variance

High Variance

Low Bias



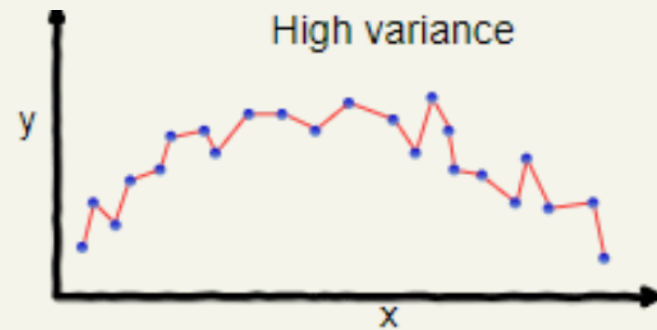
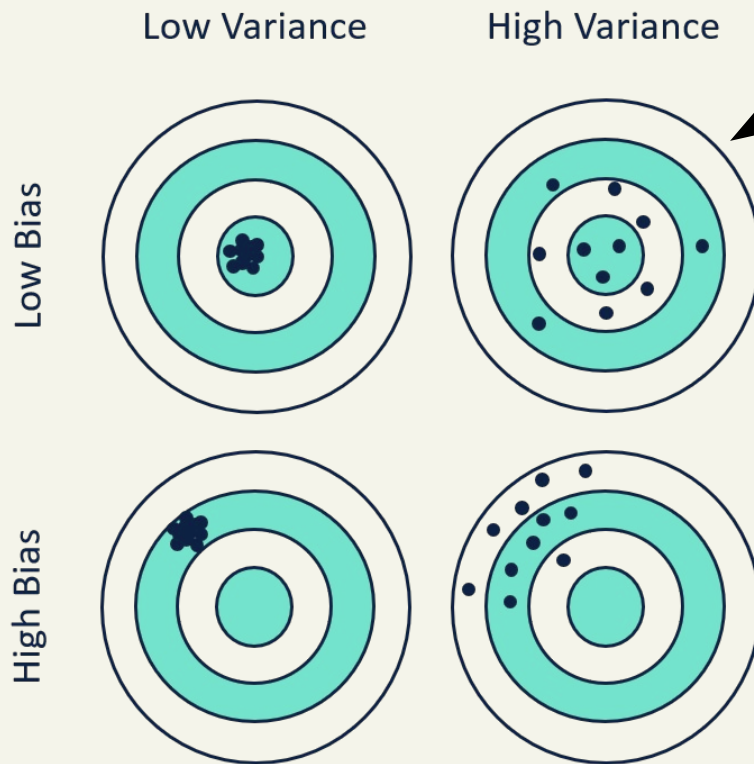
High Bias



$$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

Introdução

- **Bias–Variance Tradeoff**
- $\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$
- Overfitting ou sobreajuste

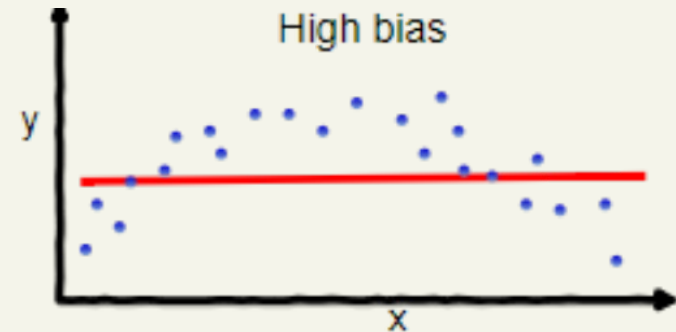
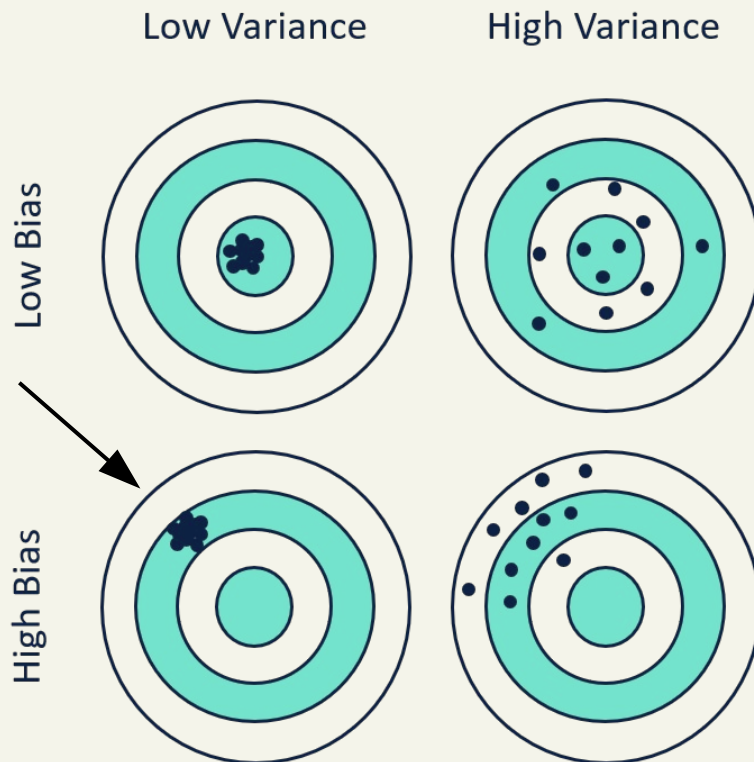


overfitting

$$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

Introdução

- **Bias–Variance Tradeoff**
- $\text{Err}(X) = \text{Viés}^2 + \text{Variância} + \text{Erro Irreduzível}$
- Underfitting ou subajuste



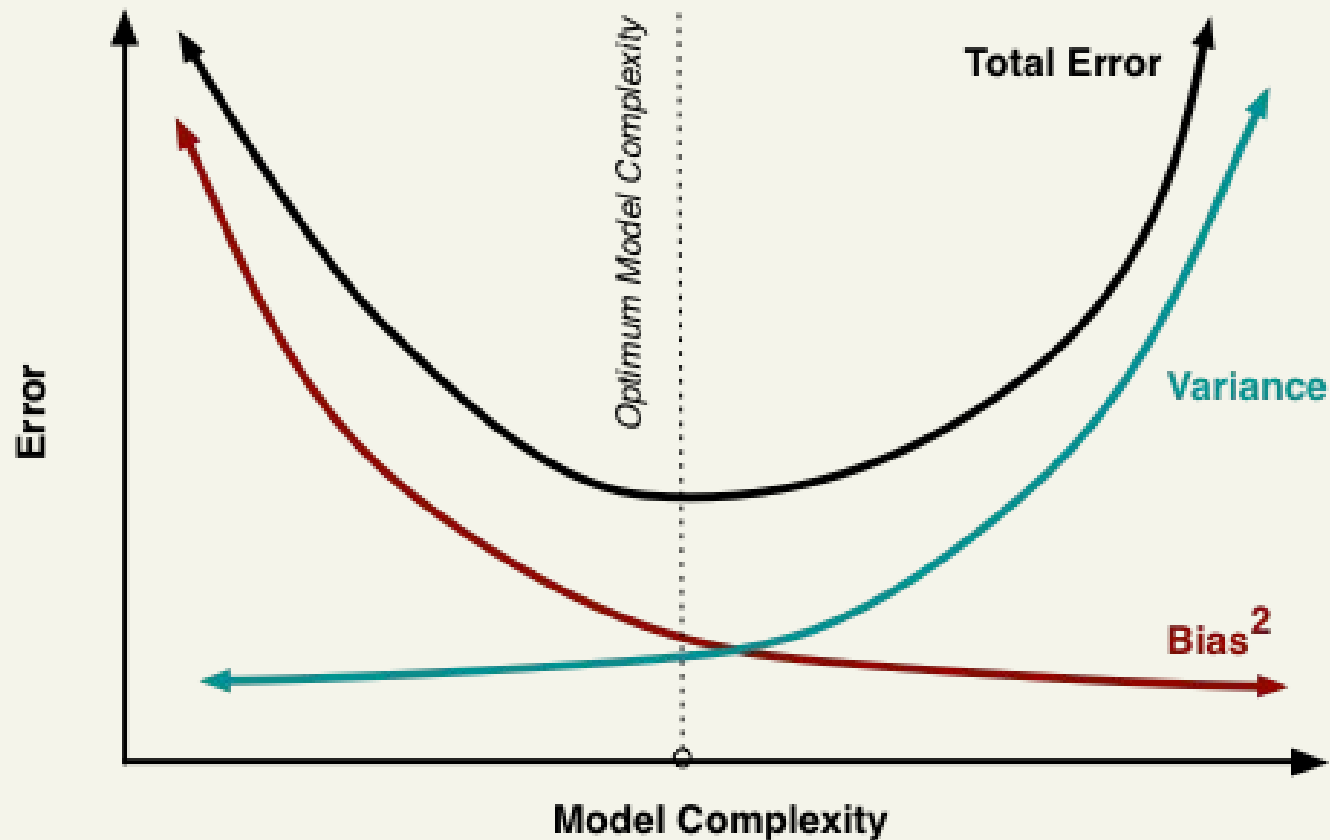
underfitting

$$\text{Err}(X) = (E[Y - \hat{Y}])^2 + E[(\hat{Y} - E[\hat{Y}])^2] + \sigma^2$$

An arrow points from the equation box to the 'underfitting' text above it.

Introdução

- Bias–Variance Tradeoff
 - Conjunto de teste

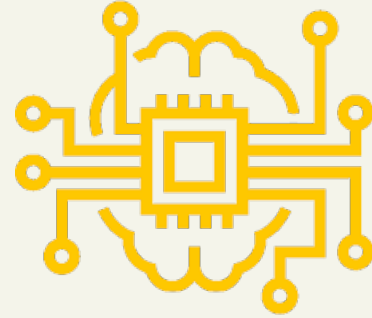


Introdução

- **Bias–Variance Tradeoff**

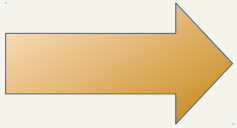


Introdução



1

Introdução



2

Classificador Bagging

3

Classificador Boosting

4

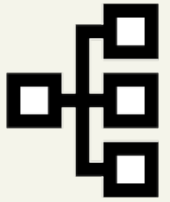
Conclusão

Introdução

- ## Tipos de Comitês

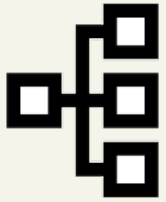


Classificador Bagging

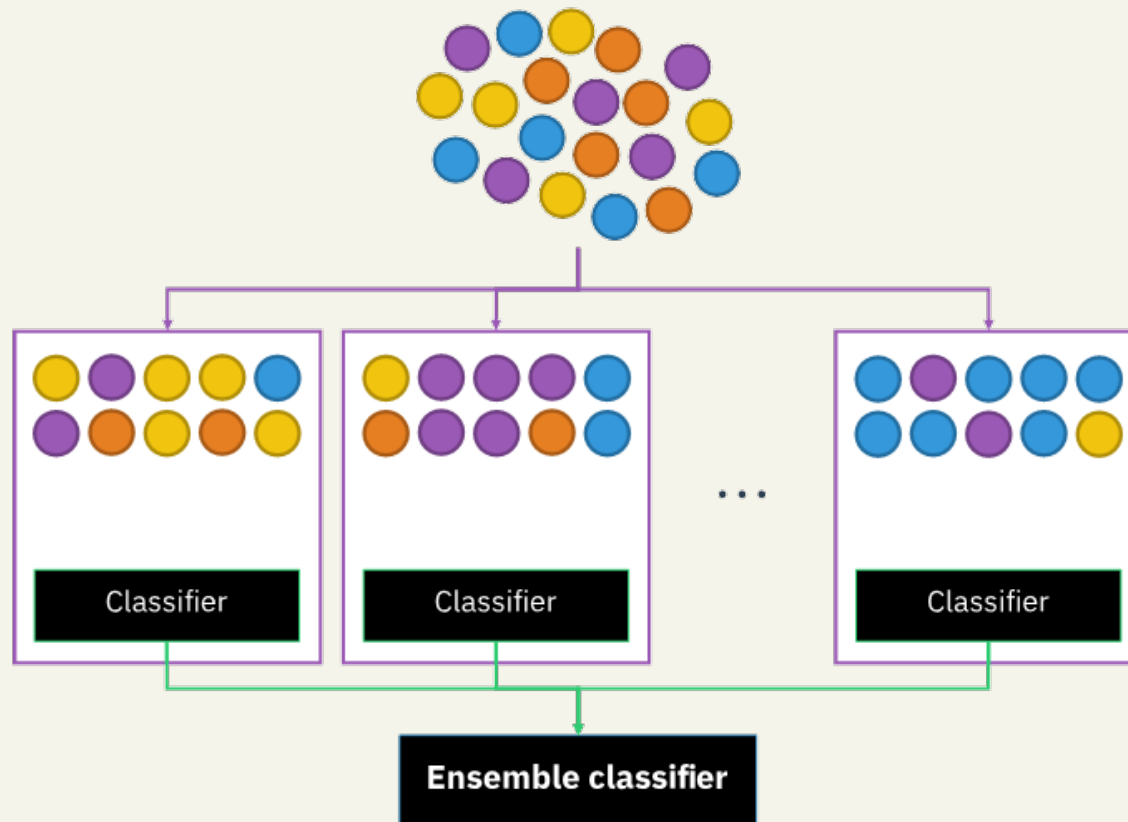


- **Classificador Bagging**
- Um classificador Bagging é um comitê que treina (fit) os classificadores base em subconjuntos aleatórios do conjunto de dados original e, em seguida, agrega suas previsões individuais por votação ou por média para formar uma previsão final.

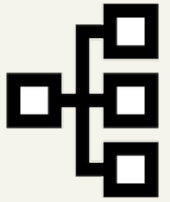
Classificador Bagging



- Classificador Bagging
 - Visão geral do Classificador Bagging



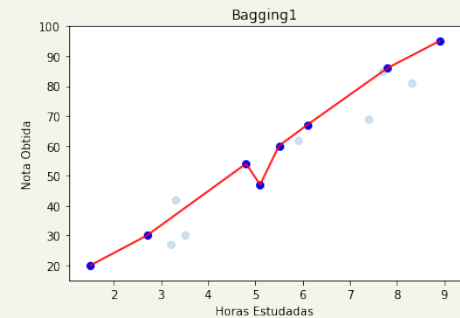
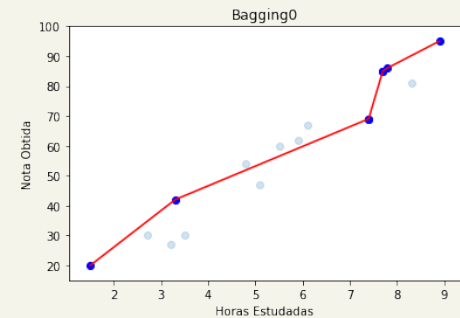
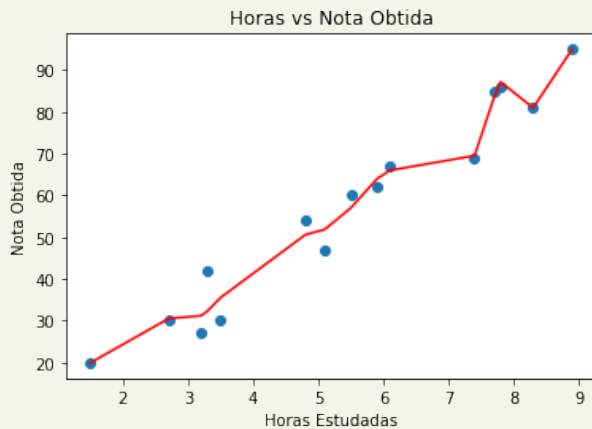
Classificador Bagging



- Classificador Bagging
- O Bagging pode normalmente ser usado como uma forma de reduzir a variância de um estimador (por exemplo, uma árvore de decisão), introduzindo aleatoriedade por meio do procedimento na construção dos *datasets*.

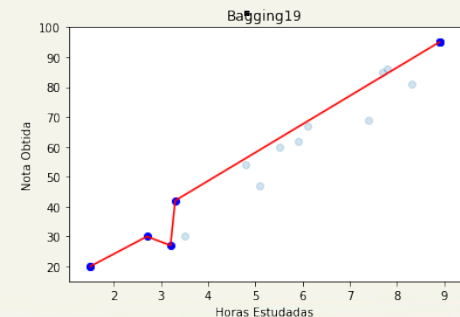
Classificador Bagging

- “Reduzir a variância alta de um estimador”

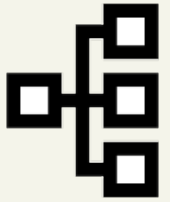


■

■



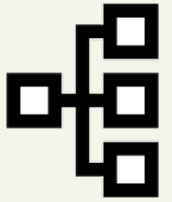
Classificador Bagging



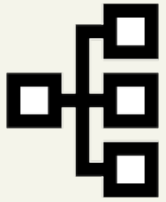
- Classificador Bagging
- Existem 4 tipos de Bagging:
 - Pasting[4]: quando os subconjuntos são subconjuntos aleatórios do conjunto de dados (não há reposição de amostras)
 - Bagging[5]: quando as amostras são retiradas com reposição.
 - Random Subspaces [6]: quando os subconjuntos de dados são gerados a partir de subconjunto de atributos.
 - Random Patches [7]: quando estimadores de base são construídos em subconjuntos de amostras e atributos.

Classificador Bagging

- Jupyter NoteBook
 - Exemplo Bagging.ipynb



Classificador Bagging



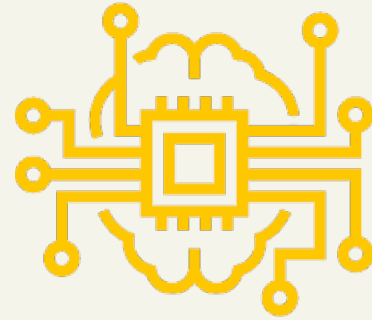
- Random Forests
- Random Forests é um classificador que é um um caso particular de bagging. Os subconjunto de dados são gerados a partir de:
 - bootstrap do conjunto de treinamento (com reposição) e
 - subconjuntos aleatórios de atributos (features).
- Devido à seleção aleatória de atributos, as árvores são mais independentes (i.e., diferentes) umas das outras quando comparado ao bagging

Classificador Bagging

- Jupyter NoteBook
 - Exemplo Random Forests.ipynb



Introdução



1

Introdução

2

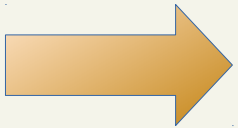
Classificador Bagging

3

Classificador Boosting

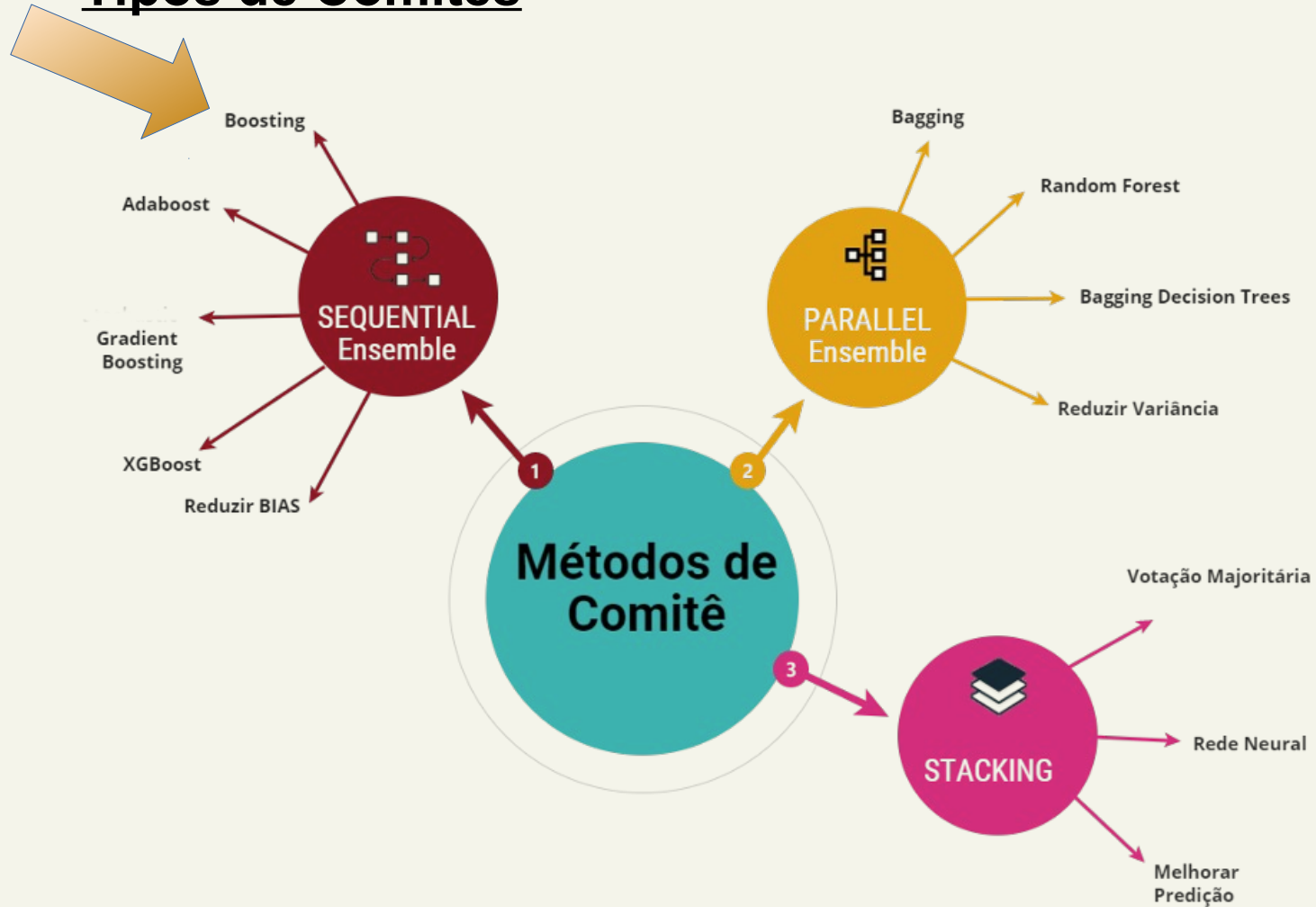
4

Conclusão



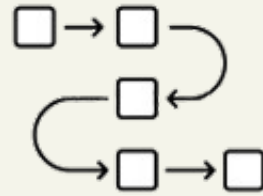
Introdução

- ## Tipos de Comitês



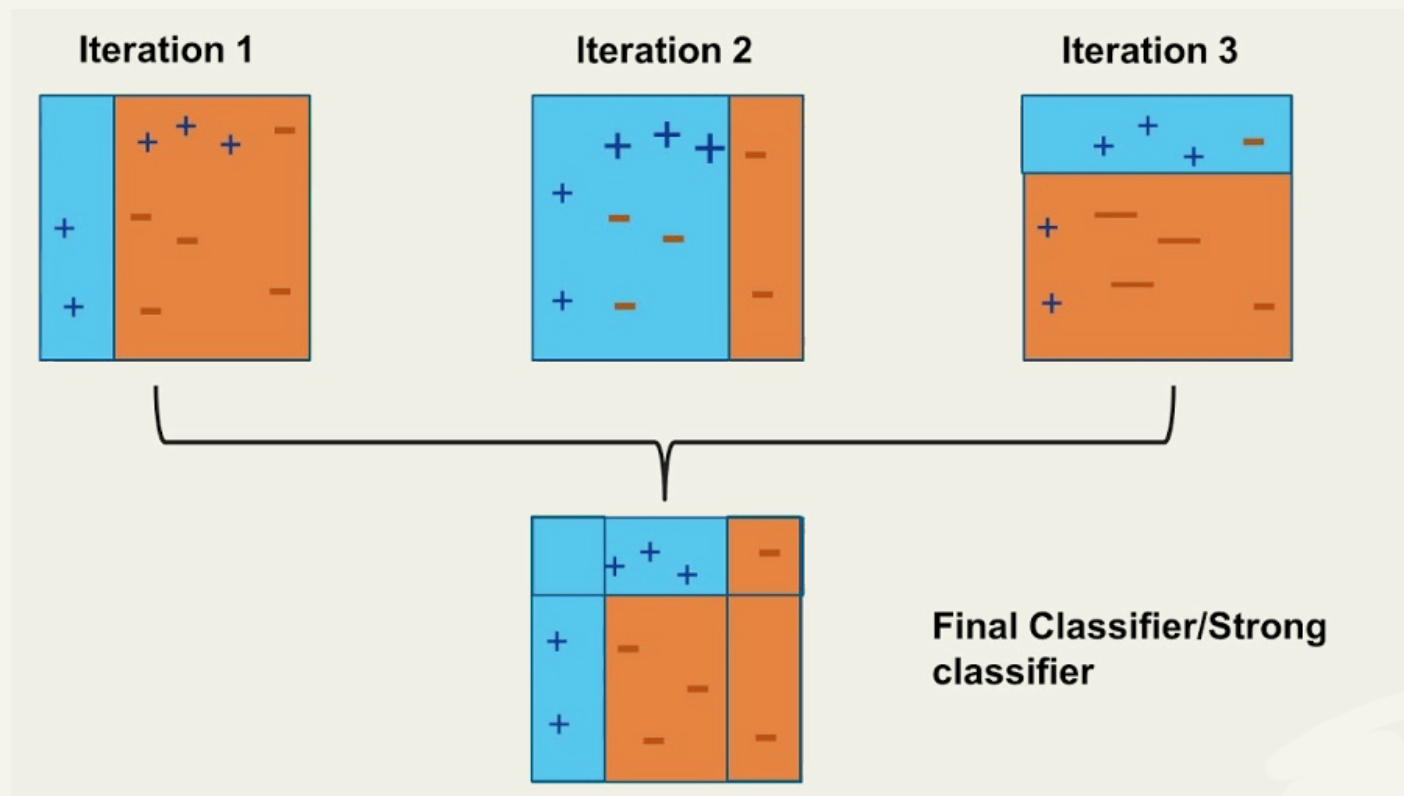
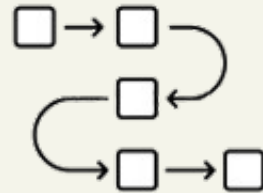
Classificador Boosting

- Classificador Boosting
- *Boosting* é uma técnica de comitê em que um classificador base aprende com os erros anteriores dos classificadores bases (comitê) para fazer melhores previsões no futuro.
- Este novo classificador base (*weak learner*) é melhor em regiões (do espaço de atributos) nas quais os modelos anteriormente adicionados ao comitê erraram.
- Sequencial → a geração de um novo modelo base depende do resultado do anterior

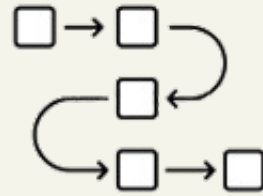


Classificador Boosting

- Classificador Boosting



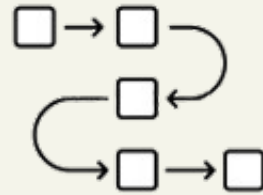
Classificador Boosting



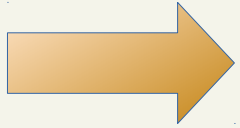
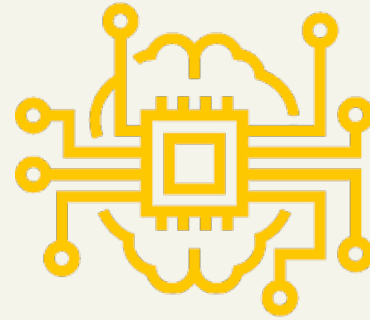
- Classificador Boosting
- Um aprendiz fraco (*weak learner*) é definido como um classificador que é ligeiramente correlacionado com a classificação verdadeira.
- Ou seja, um *weak learner* pode rotular exemplos com acurácia igual a $1/2 + \alpha$ (para classificador binário), onde α é um valor pequeno e positivo.
 - Isso significa que em um *weak learner*, o desempenho em qualquer conjunto de treinamento é ligeiramente melhor que a previsão ao acaso.
- Em contraste, um aprendiz forte (*strong learner*) é um classificador que é arbitrariamente bem correlacionado com a classificação verdadeira.

Classificador Boosting

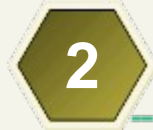
- Classificador Boosting
- Existem vários tipos de Boosting :
 - Adaptive Boosting (AdaBoost),
 - Gradient Boosting e
 - XGBoost (Extreme Gradient Boosting).



Classificador Boosting



Adaptive Boosting (AdaBoost),



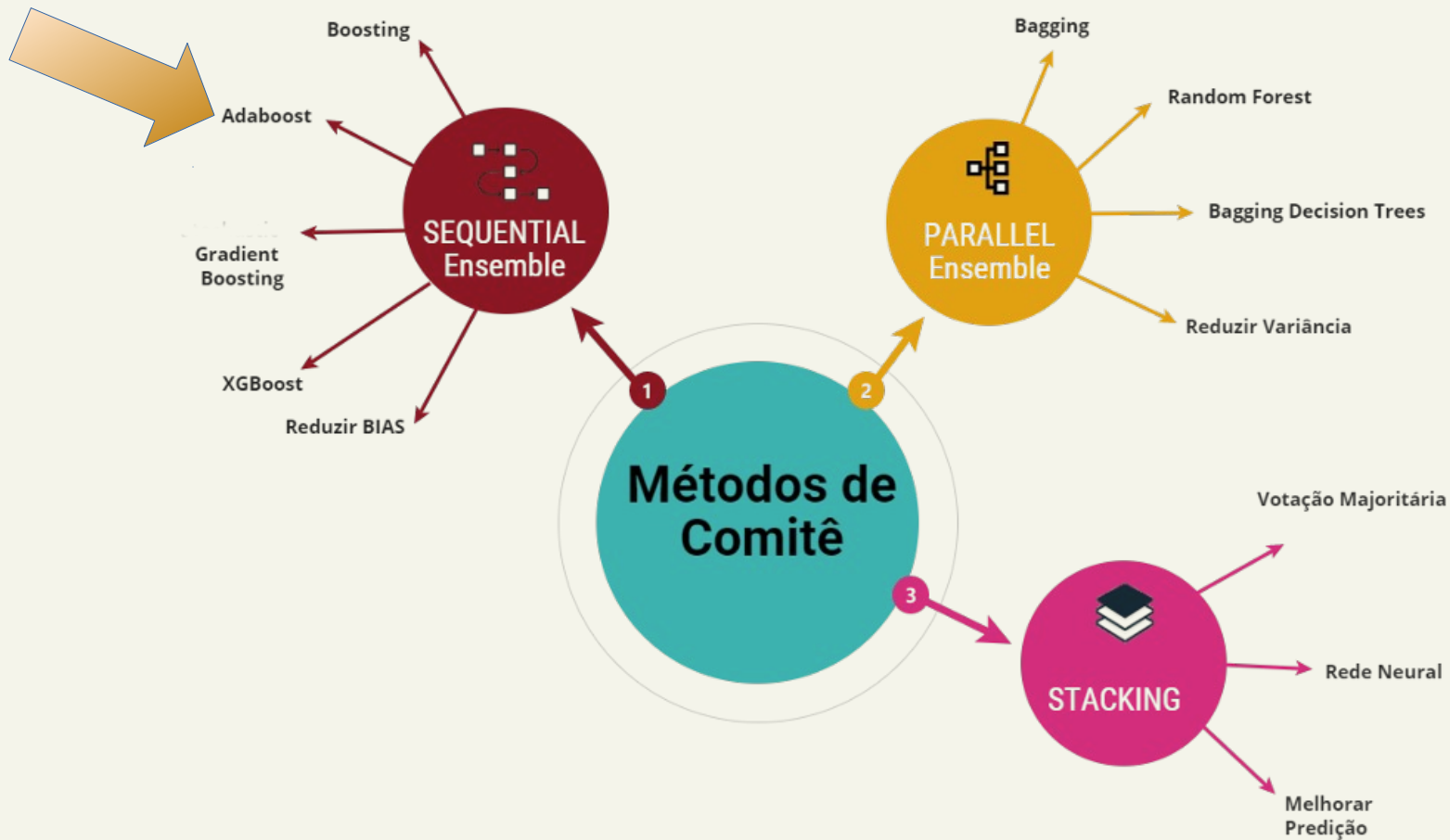
Gradient Boosting



XGBoost (Extreme Gradient Boosting)

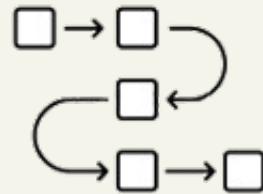
Introdução

- ## Tipos de Comitês



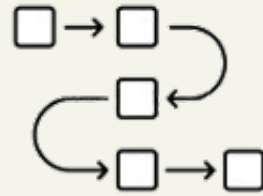
Classificador Boosting

- Adaptive Boosting (AdaBoost),
 - Visão Geral do AdaBoost.
 - As árvores de decisão usadas no boosting são chamadas de “stump” porque cada árvore de decisão tende a ser “shallow” (rasas) que não fazem sobreajuste e com alto viés.
 - Características:
 - Geralmente, as árvores tem 1 raiz e 2 folhas ! → stump!
 - Cada árvore tem seu peso na decisão final
 - Ordem importa



Classificador Boosting

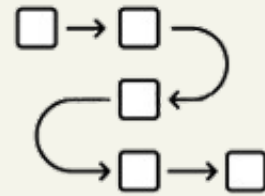
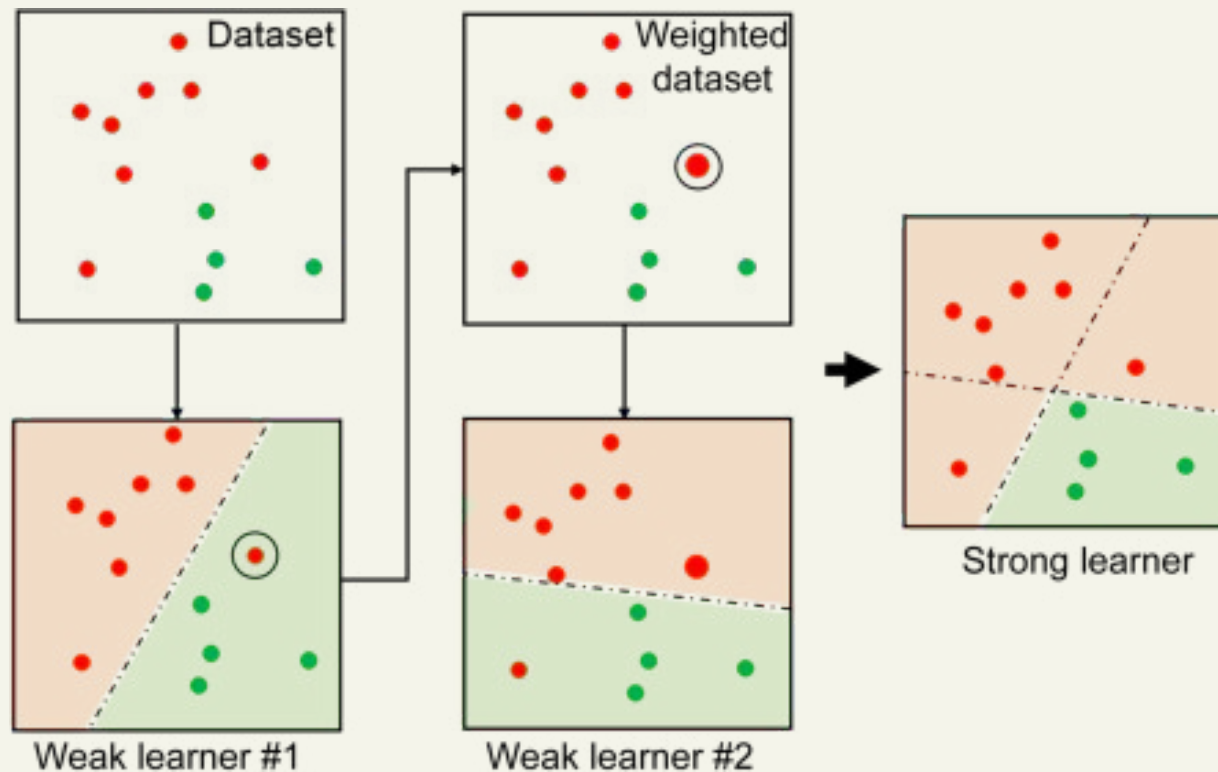
- Adaptive Boosting (AdaBoost),
- AdaBoost é um método de conjunto que treina e implanta árvores de decisão em série.
- Uma lista de pesos, com um peso para cada um dos m exemplos do conjunto de treinamento é mantida.



1. Iniciar pesos com valores uniforme ($1/m$)
2. Iterar
 - Criar uma nova instância de aprendiz L_k ;
 - Aplicar aprendiz L_k aos exemplos ponderados;
 - Aumentar pesos dos exemplos classificados incorretamente por L_k ;
 - **Atualizar pesos** e Incrementar k .

Classificador Boosting

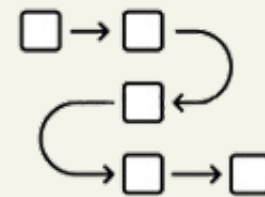
- Adaptive Boosting (AdaBoost),
 - Visão Geral do AdaBoost.



“The weight of a sample misclassified by the previous tree will **be boosted**. “

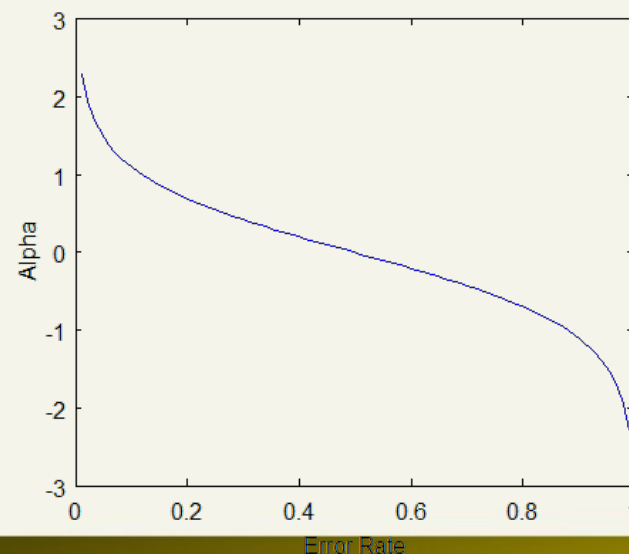
Classificador Boosting

- Adaptive Boosting (AdaBoost),
 - Cálculo do peso de cada Stump baseado na quantidade total de erros



$$\text{Alpha} = \frac{1}{2} \log\left(\frac{1 - \text{TotalError}}{\text{TotalError}}\right)$$

- TotalError → soma dos pesos das amostras erradas do Stump



Classificador Boosting

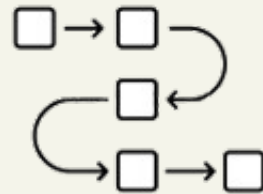
- Adaptive Boosting (AdaBoost),
- Atualização dos pesos das amostras
- 1. Aumentar o peso:

$$W_{\text{new}} = W_{\text{old}} \times e^{\alpha}$$

- 2. Diminuir o peso:

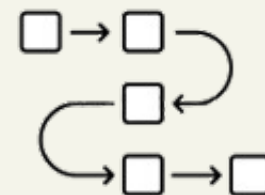
$$W_{\text{new}} = W_{\text{old}} \times e^{-\alpha}$$

- 3. Normalizar
Nova tabela de pesos

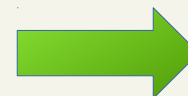


Classificador Boosting

- Adaptive Boosting (AdaBoost),
- Atualização dos pesos das amostras
 - Gerando uma nova tabela para próximo *Weak Learner*
 - Escolher número aleatório entre 0-1
 - Escolher amostra na tabela.



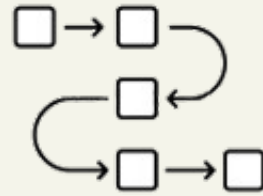
Amostra	Pesos Normalizados	Acumulado
1	0,07	0,07
2	0,07	0,14
3	0,07	0,21
4	0,07	0,28
5	0,07	0,35
6	0,37	0,72
7	0,07	0,79
8	0,07	0,86
9	0,07	0,93
10	0,07	1



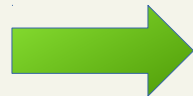
Amostra
2
7
8
6
5
6
6
4
3
2

Classificador Boosting

- Adaptive Boosting (AdaBoost),
- Estas amostras são utilizadas para treinar um novo modelo.



Amostra
2
7
8
6
5
6
6
4
3
2



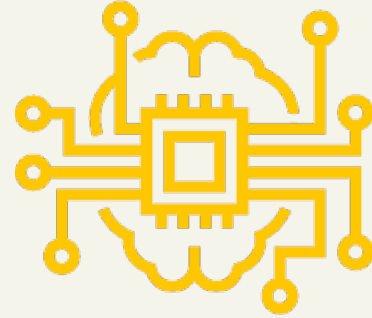
Stump

Classificador Boosting

- Jupyter Notebook
- Exemplo AdaBoostClassifier.ipynb



Classificador Boosting



1

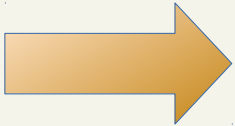
Adaptive Boosting (AdaBoost),

2

Gradient Boosting

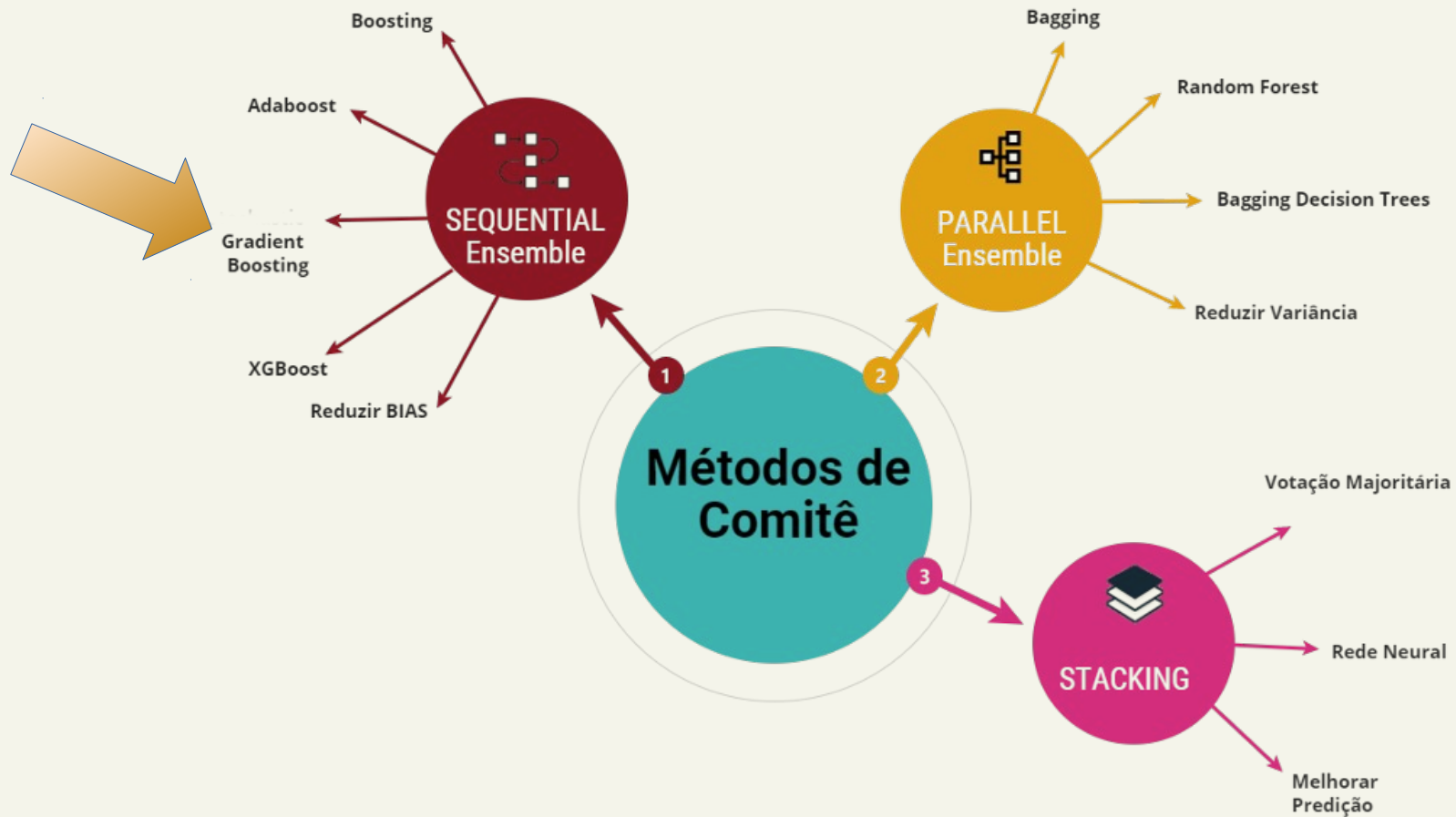
3

XGBoost (Extreme Gradient Boosting)



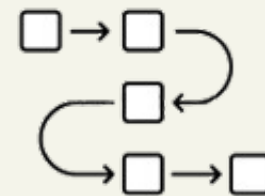
Introdução

- ## Tipos de Comitês



Classificador Boosting

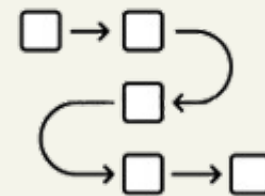
- Classificador Boosting
- Gradient Boosting
- Visão Geral → principais aspectos do funcionamento do Gboosting para Regressão!!!!
- Considere a tabela de dados abaixo



Altura	Cor favorita	Sexo	Idade	Peso
1.60	azul	fem	21	54
1.73	verde	masc	27	80
1.81	verde	masc	22	86
1.75	preto	masc	34	72
1.63	branco	fem	41	58

Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- Calcula média dos pesos: 70.0

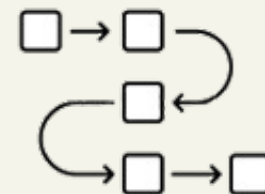


70.0

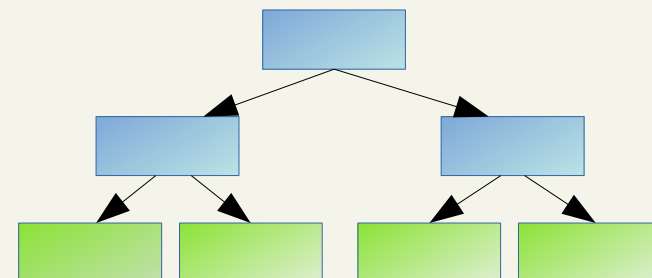
Altura	Cor favorita	Sexo	Idade	Peso
1.60	azul	fem	21	54
1.73	verde	masc	27	80
1.81	verde	masc	22	86
1.75	preto	masc	34	72
1.63	branco	fem	41	58

Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- Calcula **erro residual**: média – valor original



Altura	Cor favorita	Sexo	Idade	Peso	Res
1.60	azul	fem	21	54	-16
1.73	verde	masc	27	80	10
1.81	verde	masc	22	86	16
1.75	preto	masc	34	72	2
1.63	branco	fem	41	58	-12

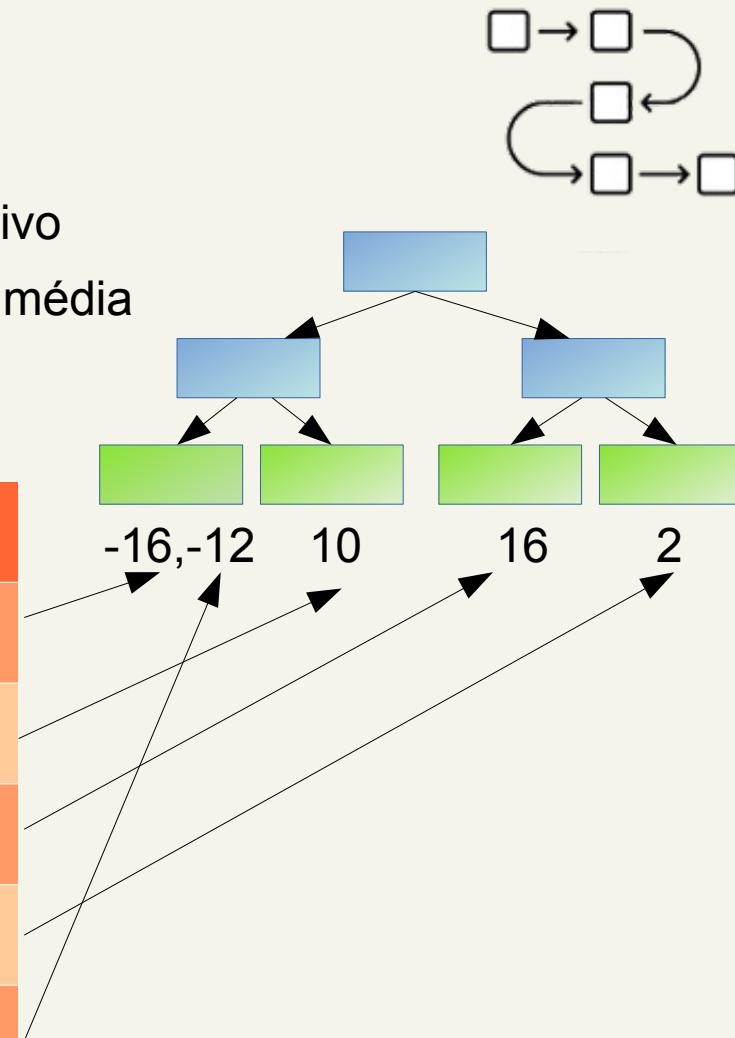


Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- Cria árvore para coluna residual como objetivo
 - Folhas com mais de uma amostra → média

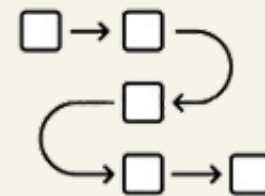
Altura	Cor favorita	Sexo	Idade	Peso
1.60	azul	fem	21	54
1.73	verde	masc	27	80
1.81	verde	masc	22	86
1.75	preto	masc	34	72
1.63	branco	fem	41	58

Res
-16
10
16
2
-12

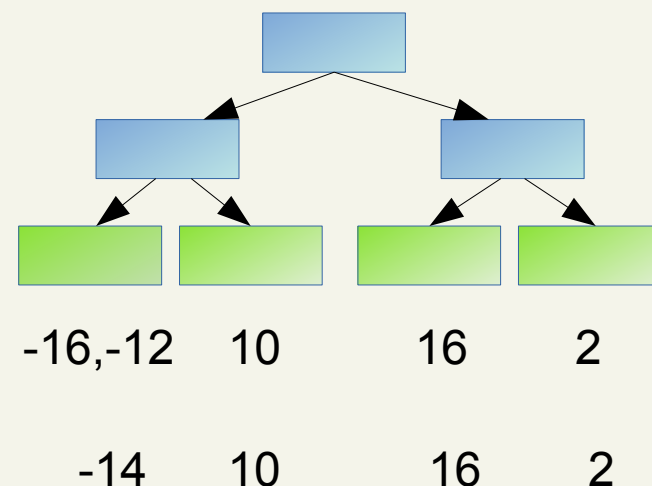


Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- Cria árvore para coluna residual como objetivo
 - Cada amostra é colocada em uma folha!

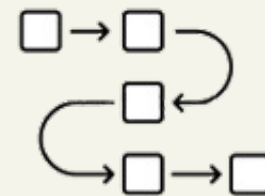


Altura	Cor favorita	Sexo	Idade	Peso	Res
1.60	azul	fem	21	54	-16
1.73	verde	masc	27	80	10
1.81	verde	masc	22	86	16
1.75	preto	masc	34	72	2
1.63	branco	fem	41	58	-12

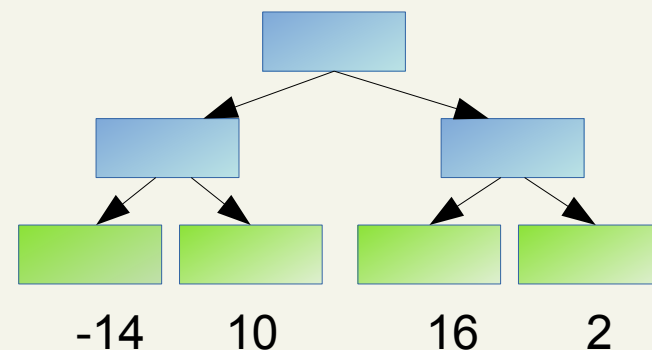


Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- Árvore pronta!

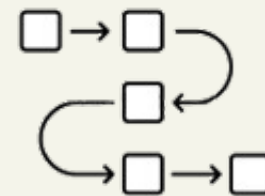


Altura	Cor favorita	Sexo	Idade	Peso	Res
1.60	azul	fem	21	54	-16
1.73	verde	masc	27	80	10
1.81	verde	masc	22	86	16
1.75	preto	masc	34	72	2
1.63	branco	fem	41	58	-12



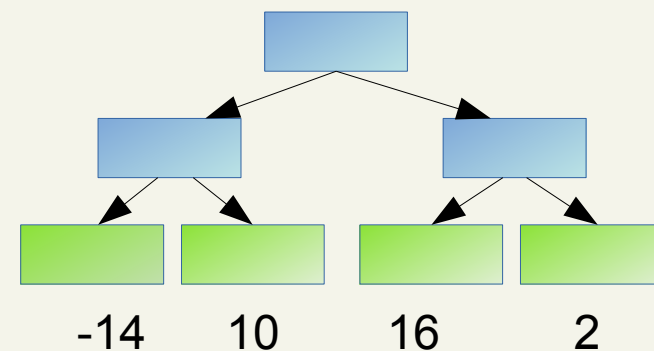
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- **SE** fizemos predição com media + árvore??



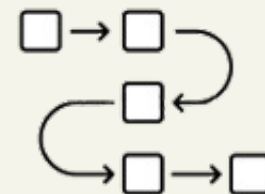
70.0

Altura	Cor favorita	Sexo	Idade	Peso	Res
1.60	azul	fem	21	54	-16
1.73	verde	masc	27	80	10
1.81	verde	masc	22	86	16
1.75	preto	masc	34	72	2
1.63	branco	fem	41	58	-12



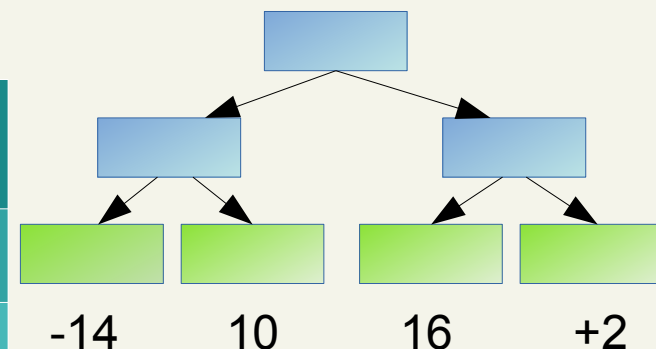
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- $\text{Pred} = 70 + \text{predição da árvore}$



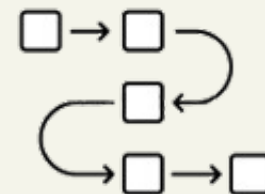
70.0

Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred
1.60	azul	fem	21	54	-16	70-14
1.73	verde	masc	27	80	10	70+10
1.81	verde	masc	22	86	+16	70+16
1.75	preto	masc	34	72	2	70+2
1.63	branco	fem	41	58	-12	70-14



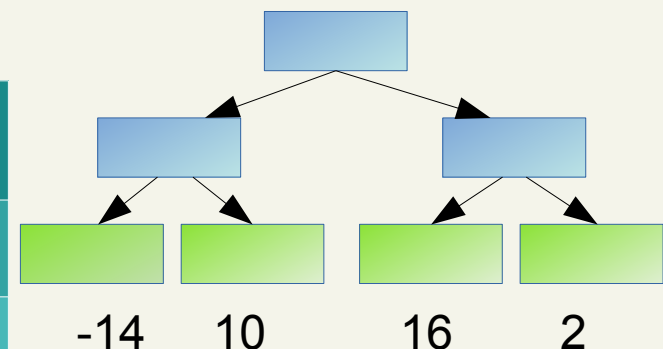
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- OVERFITTING!!!



70.0

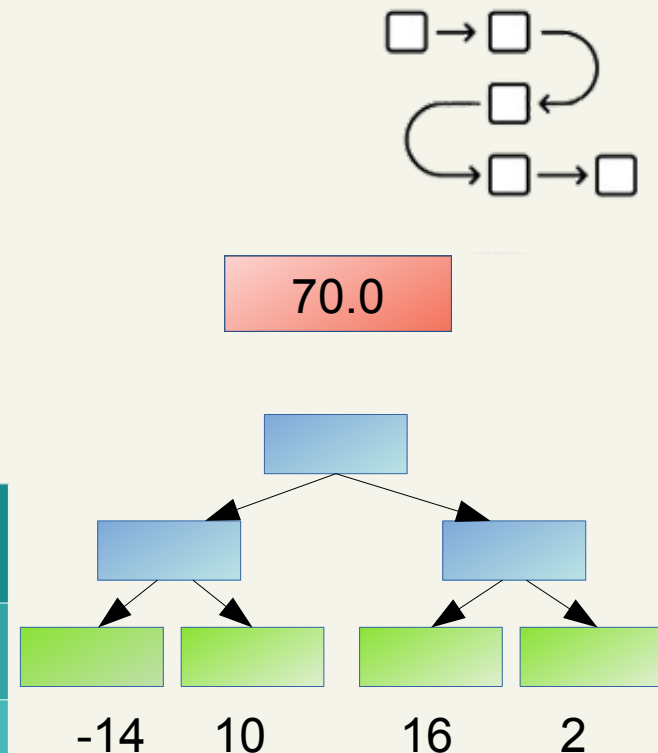
Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred
1.60	azul	fem	21	54	-16	56
1.73	verde	masc	27	80	+10	80
1.81	verde	masc	22	86	+16	86
1.75	preto	masc	34	72	+2	72
1.63	branco	fem	41	58	-12	56



Classificador Boosting

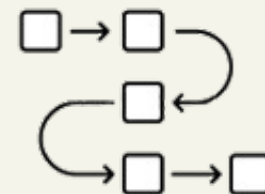
- Classificador Boosting
- Gradient Boosting
- O que fazer ???
 - Learning Rate

Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred
1.60	azul	fem	21	54	-16	56
1.73	verde	masc	27	80	+10	80
1.81	verde	masc	22	86	+16	86
1.75	preto	masc	34	72	+2	72
1.63	branco	fem	41	58	-12	56



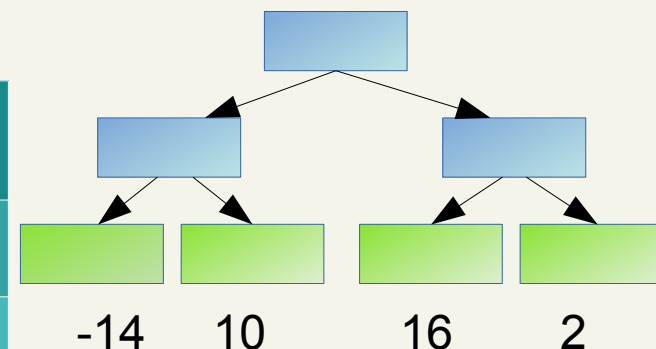
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- $\text{Pred} = \text{Residual} + (\text{Lr} \times \text{Predição Árvore})$
- $\text{LR} = 0.1$



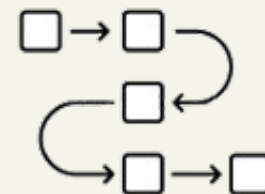
70.0

Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred
1.60	azul	fem	21	54	-16	56
1.73	verde	masc	27	80	+10	80
1.81	verde	masc	22	86	+16	86
1.75	preto	masc	34	72	+2	72
1.63	branco	fem	41	58	-12	56



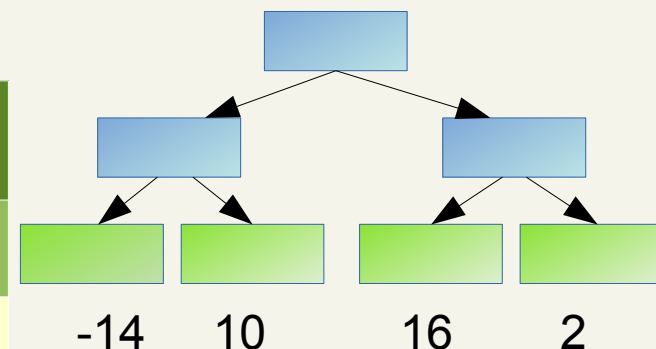
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- $\text{Pred} = \text{Residual} + (\text{Lr} \times \text{Predição Árvore})$
- $\text{LR} = 0.1$



70.0

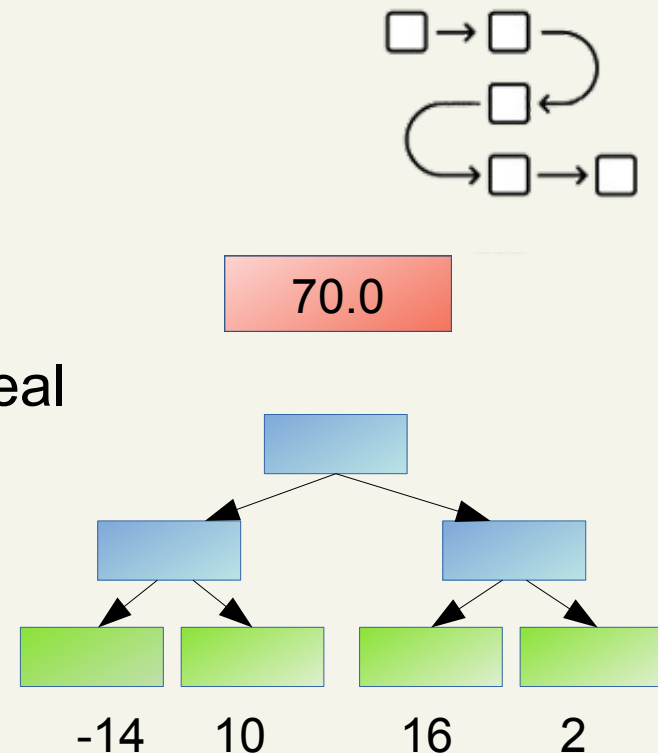
Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred
1.60	azul	fem	21	54	-16	68.6
1.73	verde	masc	27	80	+10	71
1.81	verde	masc	22	86	+16	71.6
1.75	preto	masc	34	72	+2	70.2
1.63	branco	fem	41	58	-12	68.6



Classificador Boosting

- Classificador Boosting
- Gradient Boosting
- A primeira árvore está pronta.
- Calculamos novo **Erro residual**: Pred-Real

Altura	Cor favorita	Sexo	Idade	Peso	Res	Pred	Res
1.60	azul	fem	21	54	-16	68.6	-14.6
1.73	verde	masc	27	80	+10	71	9
1.81	verde	masc	22	86	+16	71.6	14.4
1.75	preto	masc	34	72	+2	70.2	1.8
1.63	branco	fem	41	58	-12	68.6	-10.6

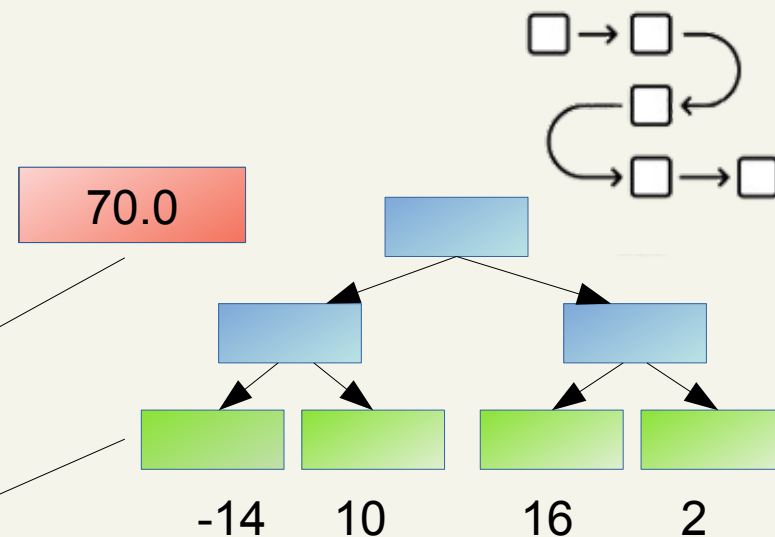


Classificador Boosting

- Classificador Boosting
- Gradient Boosting
Resíduos diminuíram

Altura	Cor favorita	Sexo	Idade	Peso
1.60	azul	fem	21	54
1.73	verde	masc	27	80
1.81	verde	masc	22	86
1.75	preto	masc	34	72
1.63	branco	fem	41	58

Res	Res
-16	-14.6
+10	9
+16	14.4
+2	1.8
-12	-10.6



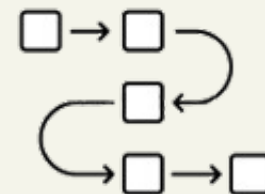
Classificador Boosting

- Classificador Boosting
- Gradient Boosting

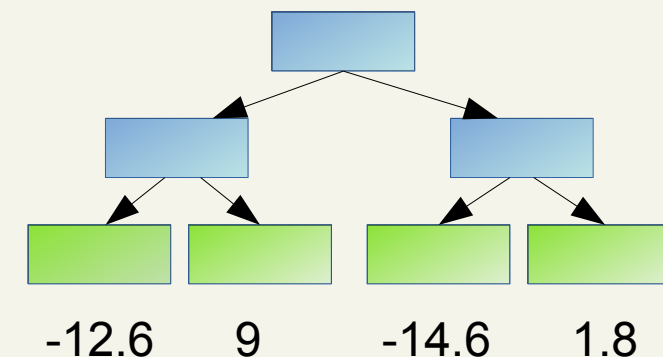
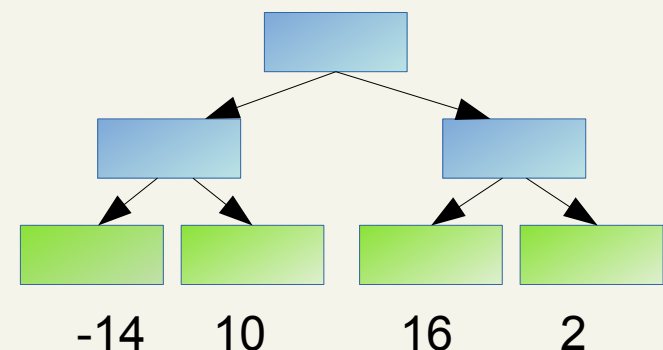
Construir uma nova árvore com Resíduos

– Mais um passo na direção certa!

Altura	Cor favorita	Sexo	Idade	Peso	Res	Res
1.60	azul	fem	21	54	-16	-14.6
1.73	verde	masc	27	80	+10	9
1.81	verde	masc	22	86	+16	14.4
1.75	preto	masc	34	72	+2	1.8
1.63	branco	fem	41	58	12	-10.6



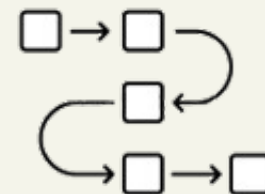
70.0



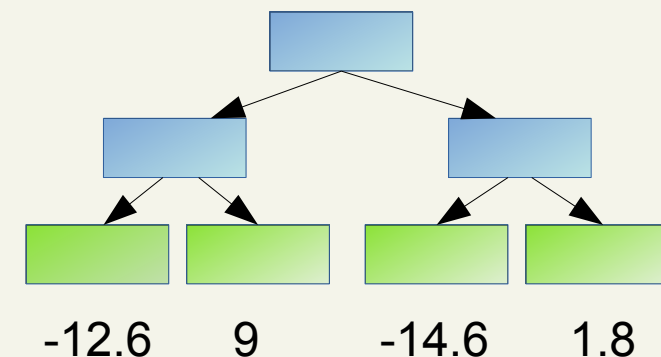
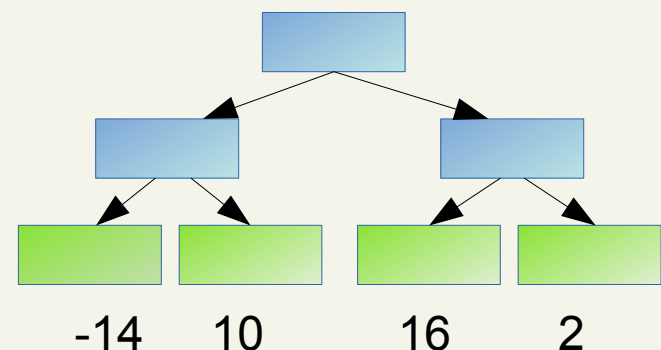
Classificador Boosting

- Classificador Boosting
- Gradient Boosting
 - Repetir etapas anteriores
- Segredo:
- A cada nova árvore, as amostras se encaixam em folhas diferentes
- LR → define o quanto ir na direção certa!

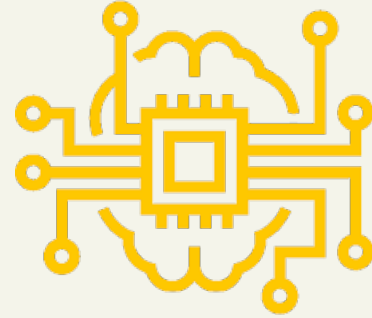
Predição = Média + 0,1 x Árvore1 + 0,1x Árvore2 + ...



70.0



Classificador Boosting



1

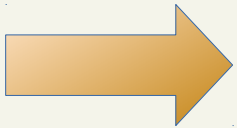
Adaptive Boosting (AdaBoost),

2

Gradient Boosting

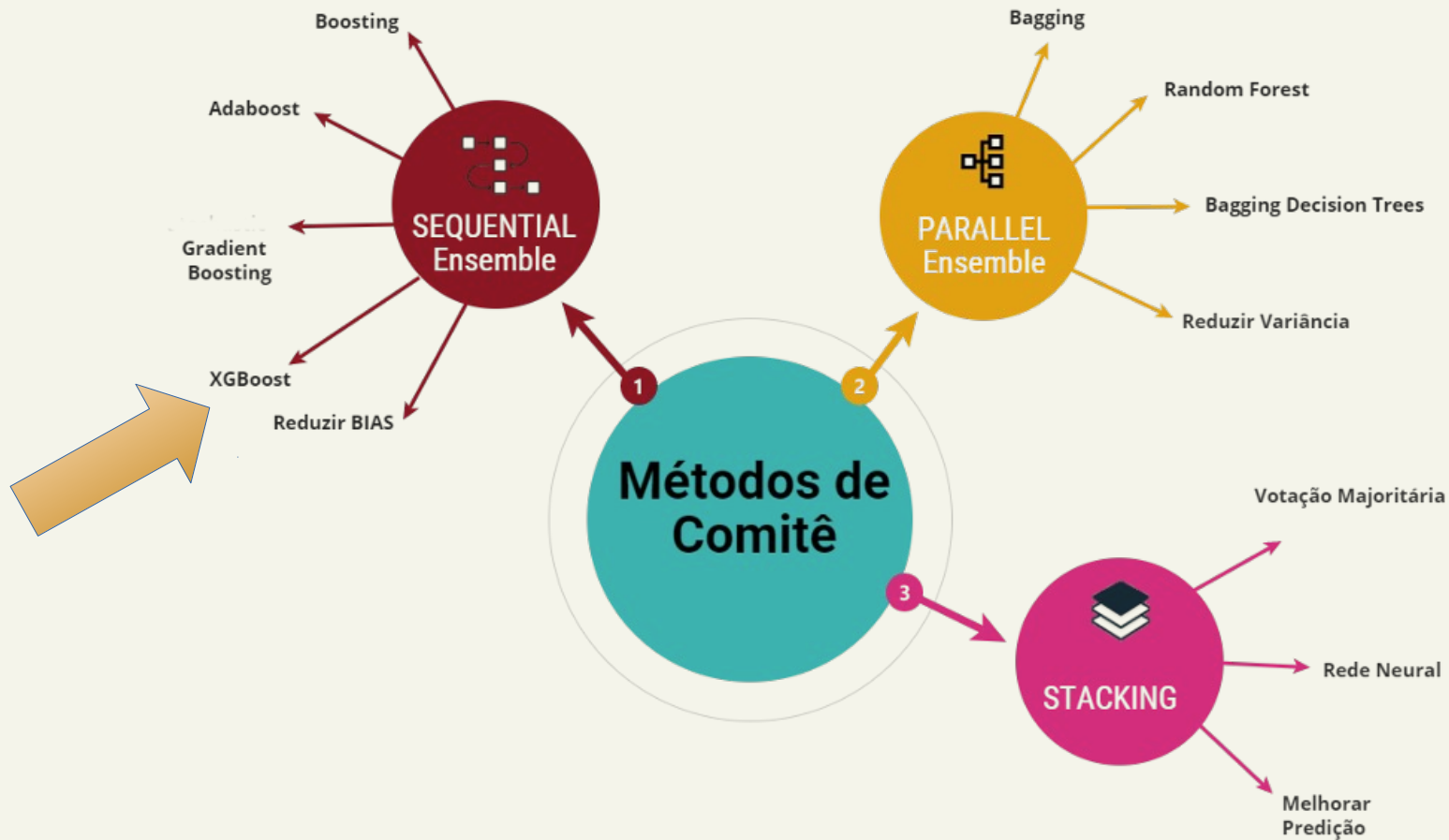
3

XGBoost (Extreme Gradient Boosting)



Introdução

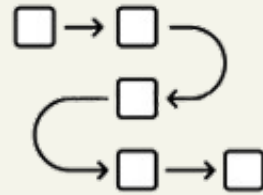
- ## Tipos de Comitês



Classificador Boosting

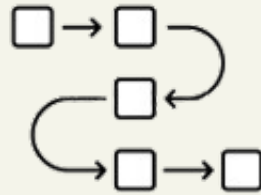
- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Como Anthony Goldbloom CEO da Kaggle disse em 2016, quando o XGBoost estava se tornando “grande” em aprendizado de máquina competitivo:

“It has almost always been ensembles of decision trees that have won competitions. It used to be random forest that was the big winner, but over the last six months a new algorithm called XGboost has cropped up, and it’s winning practically every competition in the structured data category.”



Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Xgboost Tree: qualidade da Árvore
 - Valor inicial (média por exemplo)
 - Cálculo de resíduos (similar ao Gboost)
 - Qualidade da Árvore

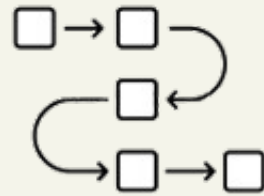
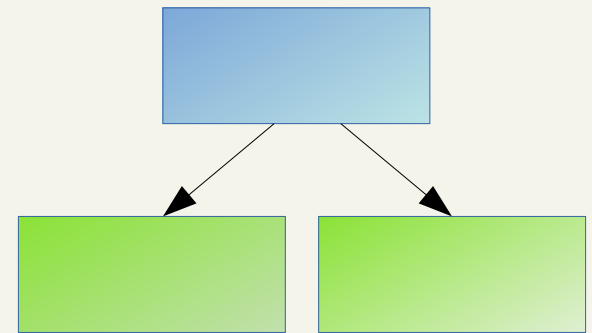


$$\text{Score de Similaridade} = \frac{(\text{Soma dos Resíduos})^2}{\text{Número de Resíduos} + \lambda}$$

Classificador Boosting

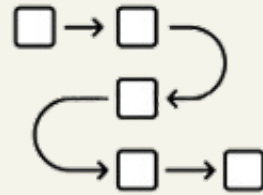
- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Consideremos a tabela abaixo de efetividade de um medicamento.

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Preditor 0,5 (*)



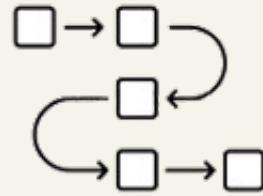
0,5

-4, -3, 5, 6, 6, -2

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Calcular resíduos



0,5

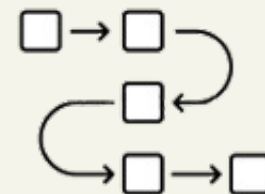
-4, -3, 5, 6, 6, -2

4.5, 3.5, -4.5, -5.5, -5.5, 2.5

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Calculamos SS para **resíduos**



0,5

-4, -3, 5, 6, 6, -2

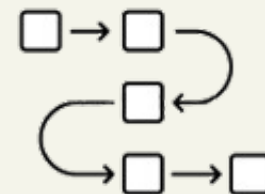
4.5, 3.5, -4.5, -5.5, -5.5, 2.5

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

$$SS = (4.5 + 3.5 - 4.5 - 5.5 - 5.5 + 2.5)^2 / (6 + 0)$$
$$SS = 4.166$$

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Deixamos anotado o SS



0,5

SS=4.16

4.5, 3.5, -4.5, -5.5, -5.5, 2.5

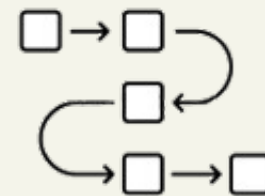
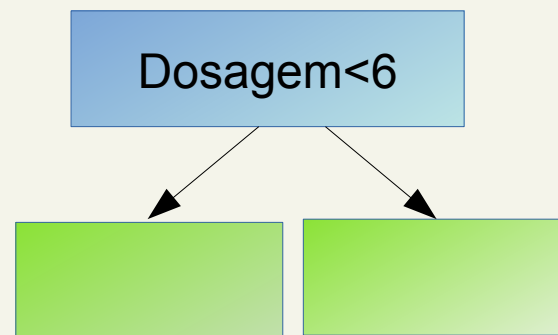
Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Escolhemos um critério

dosagem < média entre 2 amostras consecutivas

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

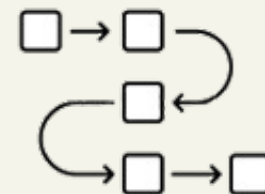


0,5

SS=4.16

Classificador Boosting

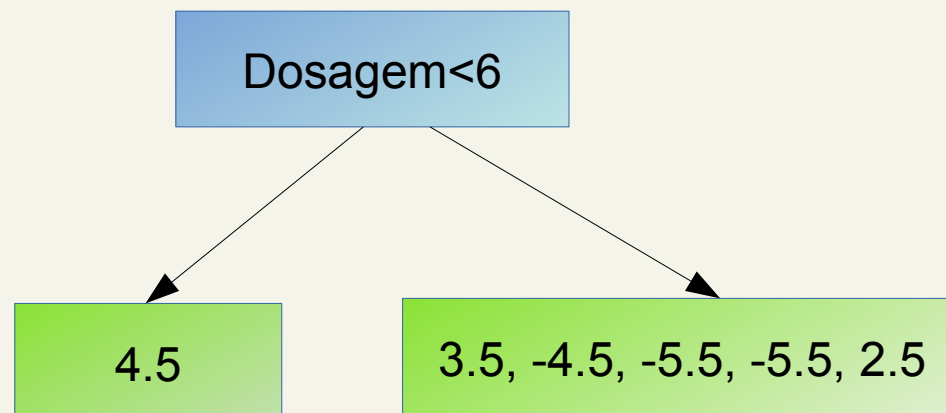
- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Dividimos efetividade RESIDUAL folhas



0,5

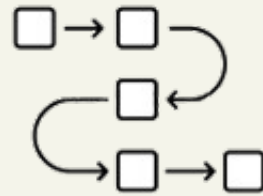
SS=4.16

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



Classificador Boosting

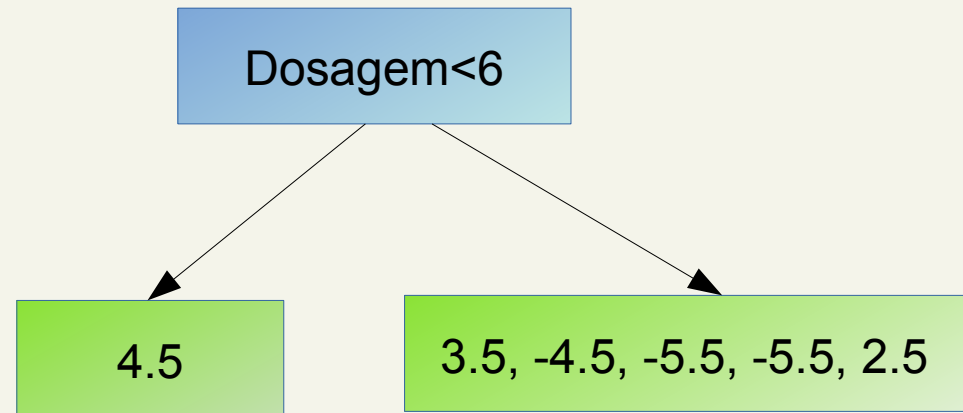
- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Calculamos SS para cada folha



0,5

SS=4.16

Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

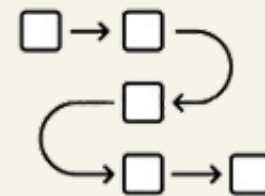


$$SS = (4.5)^2 / (1 + 0)$$
$$SS = 20.25$$

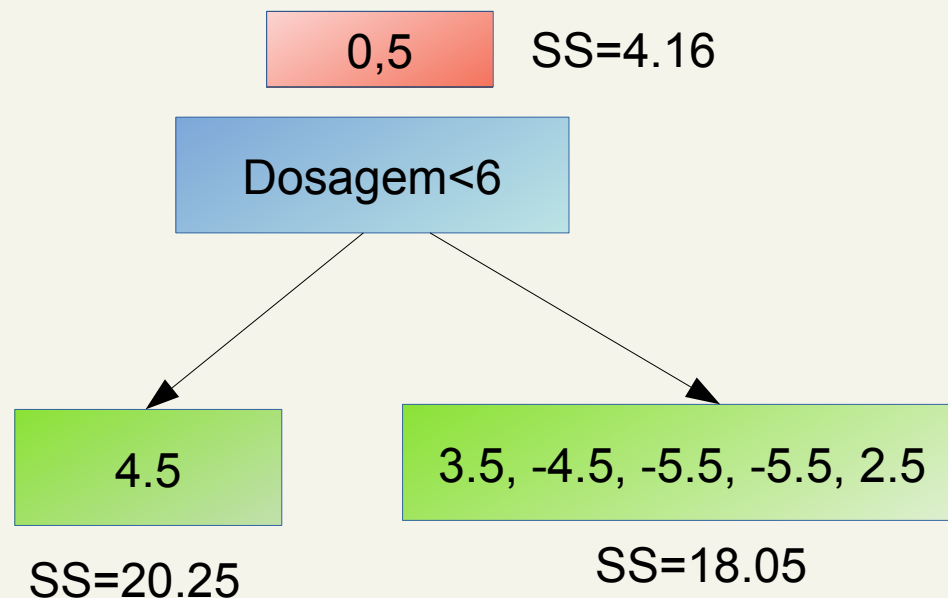
$$SS = (3.5 - 4.5 - 5.5 - 5.5 + 2.5)^2 / (5 + 0)$$
$$SS = 18.05$$

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Calculamos SS para cada folha

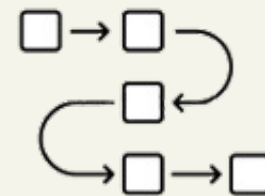


Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

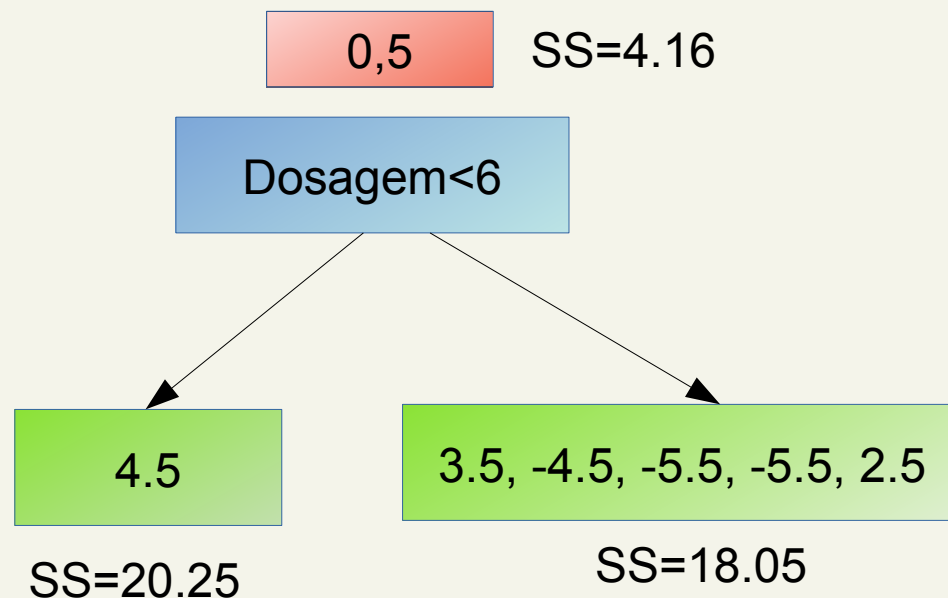


Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Calculamos ganho de



Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

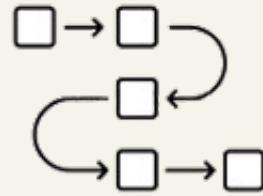


$$\text{Ganho} = 20.25 + 18.05 - 4.16$$

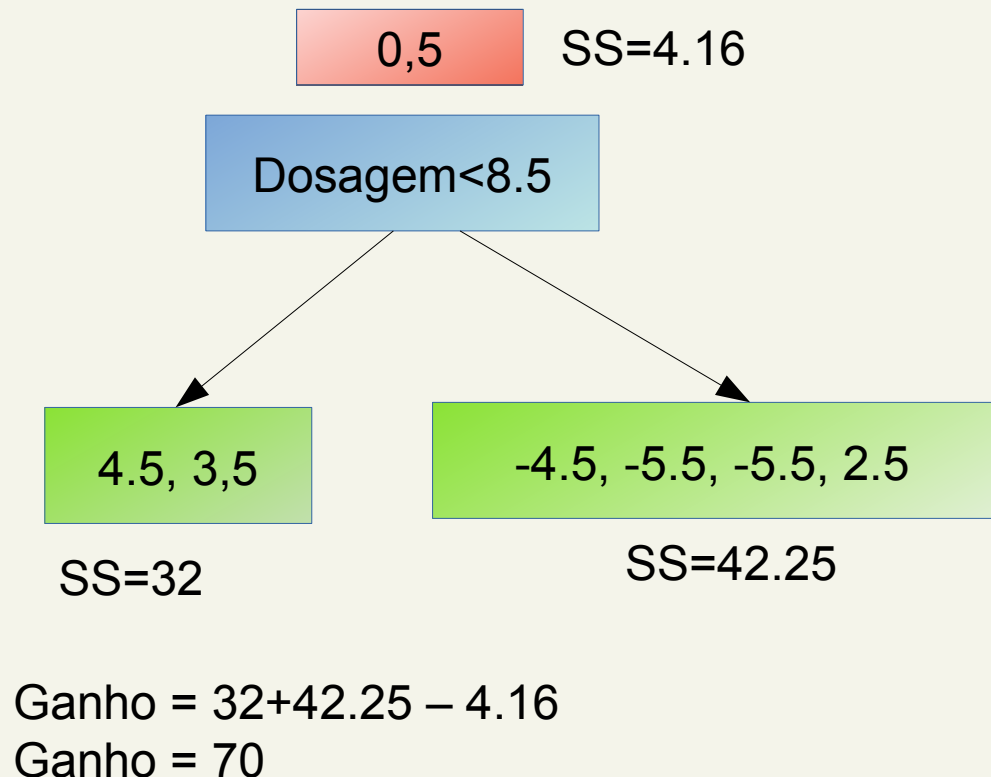
$$\text{Ganho} = 34.14$$

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Repete Processo para outros thresholds

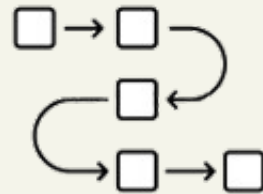


Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

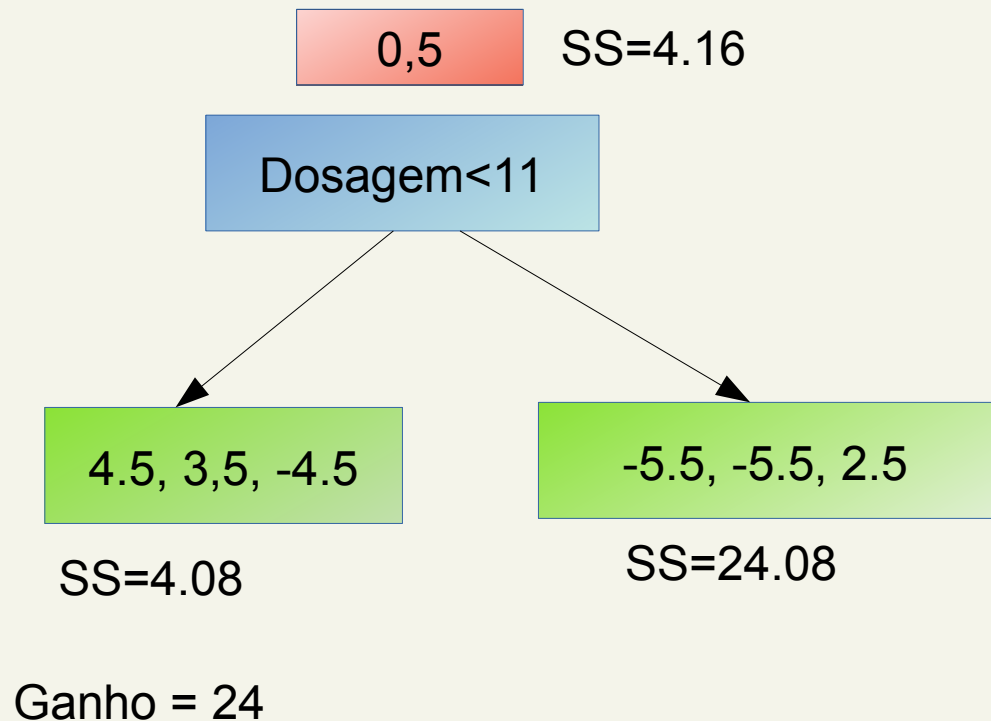


Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Repete Processo para outros thresholds

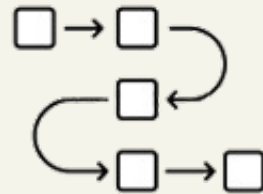


Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

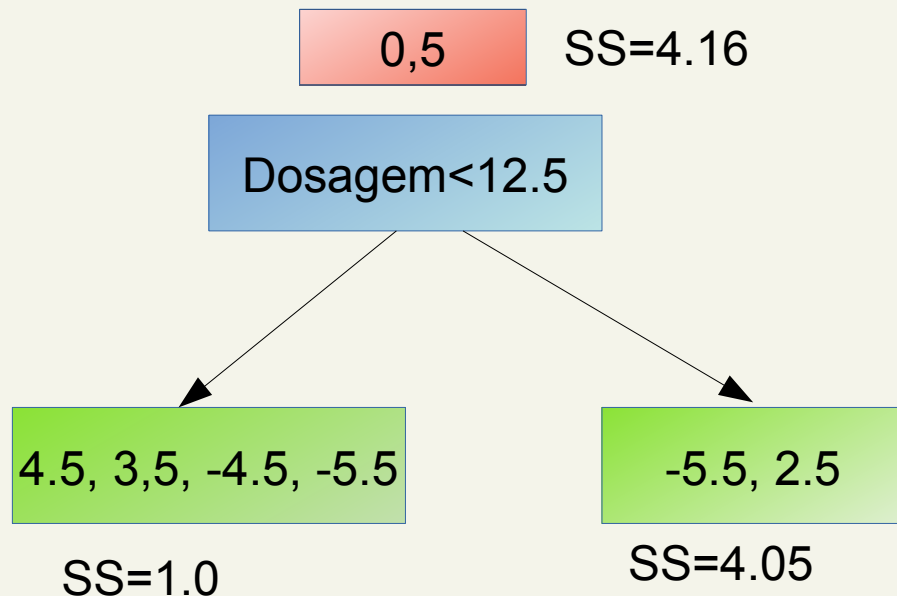


Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Repete Processo para outros thresholds



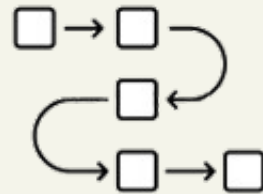
Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



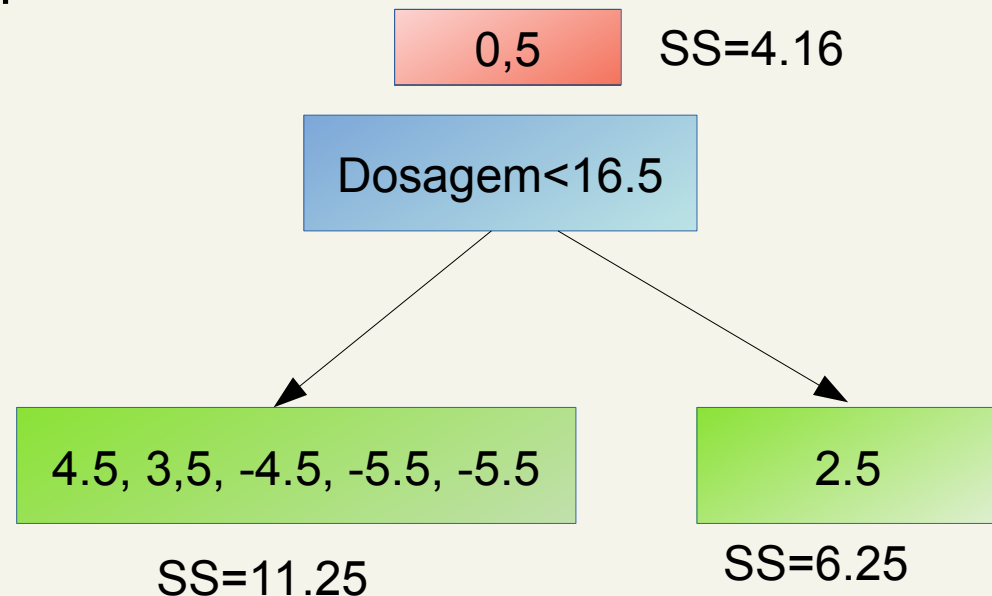
Ganho = 1.33

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Repete Processo para outros thresholds



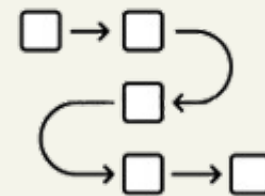
Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



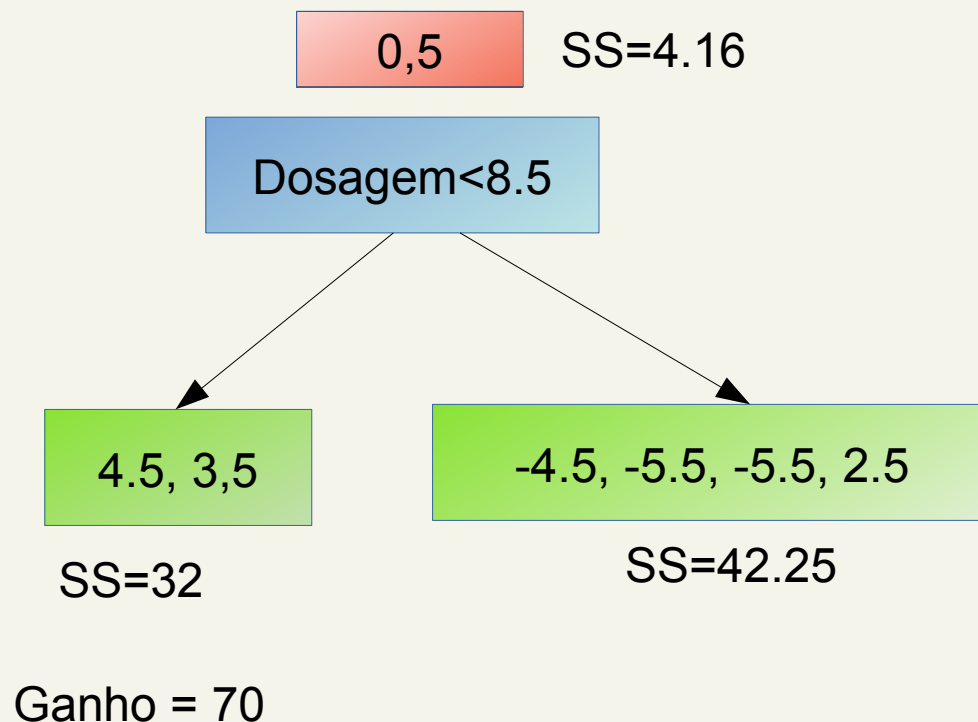
Ganho = 13.33

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- **Escolher o maior GANHO**

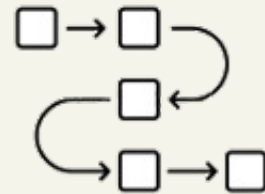


Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- O que é maior Ganho ??
 - Todos negativos em uma folha e positivos em outra
 - E o λ ??

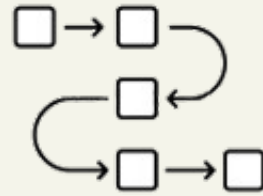


Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2

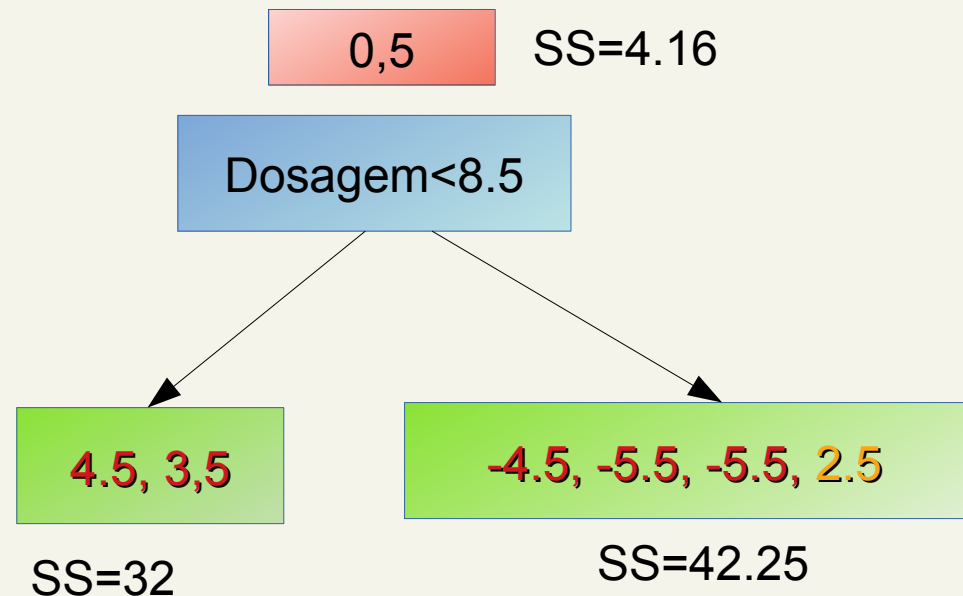
$$\text{Score de Similaridade} = \frac{(\text{Soma dos Resíduos})^2}{\text{Número de Resíduos} + \lambda}$$

Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- O que é maior Ganho ??
 - Todos negativos em uma folha e positivos em outra



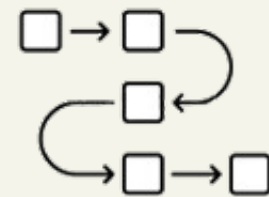
Dosagem	Efetividade
5	-4
7	-3
10	5
12	6
13	6
20	-2



Ganho = 70

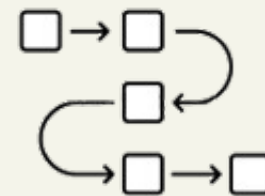
Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Objetivo: foi o de apresentar uma intuição sobre como gerar XGBoost Trees
 - Pelo menos a forma mais comum implementada
- E o que tem mais no XGBoost ???



Classificador Boosting

- Classificador Boosting
- XGBoost (Extreme Gradient Boosting)
- Diversos algoritmos:
 - Gradient Boosting
 - Regularization
 - **Unique Regression Tree**
 - Approximate Greedy Algorithm
 - Weighted Quantile Sketch
 - Sparsity-Aware Split Finding
 - Cache-Aware Access
 - Blocks for out-of-core Computation

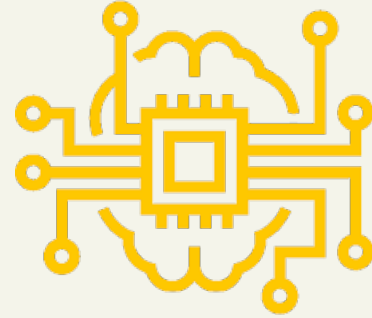


Classificador Boosting

- Jupyter NoteBook
- Exemplo XGBoostRegressor.ipynb



Introdução



1

Introdução

2

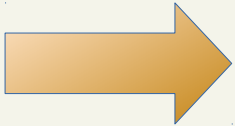
Classificador Bagging

3

Classificador Boosting

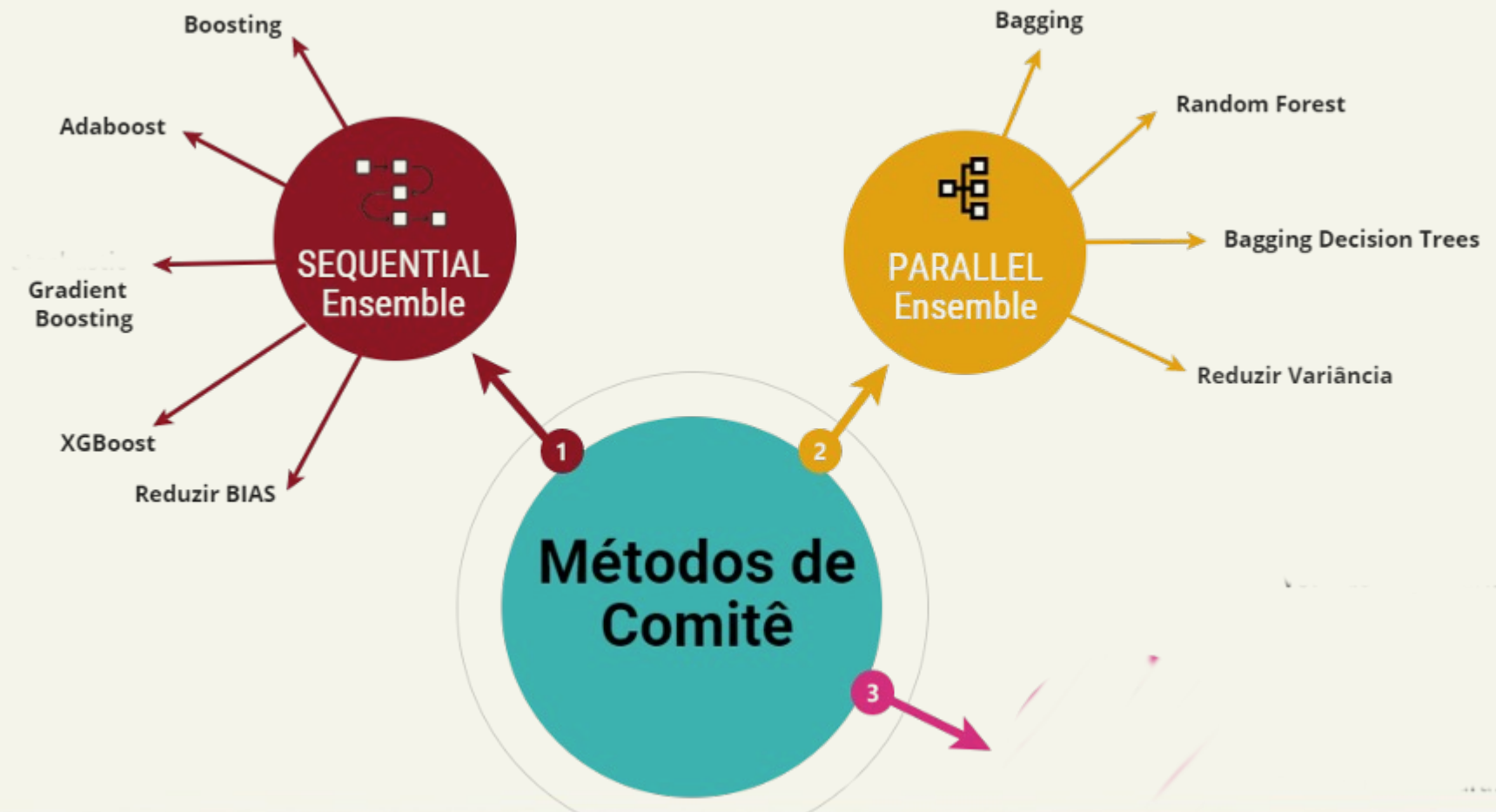
4

Conclusão



Conclusão

- Take Away



Referência

[1] Ho, Tin Kam. "Random decision forests." Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE, 1995.

[2] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.

[3] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[4] Breiman, Leo. "Pasting small votes for classification in large databases and on-line." Machine learning 36.1-2 (1999): 85-103.

[5] Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

[6] Ho, Tin Kam. "The random subspace method for constructing decision forests." IEEE transactions on pattern analysis and machine intelligence 20.8 (1998): 832-844.

[7] Louppe, Gilles, and Pierre Geurts. "Ensembles on random patches." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2012.

[8] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

Obrigado!!!

The background is a dark, textured surface with a grid pattern. It features several elements: binary code (0s and 1s) scattered throughout; wavy, glowing yellow lines that resemble signal waves or data paths; a central target symbol with concentric circles and a crosshair; and various mathematical symbols like plus (+) and minus (-) signs. The overall color palette is dominated by dark greens, yellows, and greys, giving it a high-tech, digital feel.