

Engenharia de Computação
Inteligência Artificial
15 - Algoritmos Ensemble
Fundamentos e métodos

Organização da aula

- Retomada: DT, KNN, SVM, viés, variância e diversidade.
- Voting: votação simples e ponderada.
- Bagging e Random Forest: amostragem e diversidade.
- Boosting: AdaBoost e Gradient Boosting.
- Stacking, comparação experimental e preparação para a prática.

Fio condutor

Como combinar modelos de aprendizagem para melhorar a generalização?

Retomada: modelos individuais

Modelo	Decisão	Sensibilidade
Árvore	regras e partições	variância
KNN	vizinhança	escala e k
SVM	margem	escala e kernel

Para discutir

Em quais exemplos esses modelos podem discordar?

O que é um ensemble?

Conjunto de modelos

- Cada estimador aprende uma função a partir dos dados.
- Uma regra ou outro modelo combina as saídas.
- O objetivo é melhorar desempenho ou estabilidade, sem garantia de superioridade.

Observação

Uma população de candidatos em AG, ACO, PSO ou CLONALG não é, por si só, um ensemble de preditores.

Fio condutor: classificação

Problema comum às aulas anteriores

- Breast Cancer Wisconsin: atributos numéricos, duas classes.
- Classe positiva: maligno (1); negativa: benigno (0).
- Comparar DT, KNN e SVM com conjuntos de modelos.

Para discutir

Um classificador que acerta casos difíceis diferentes pode ajudar o conjunto?

Viés e variância

Viés

- Erro sistemático associado às hipóteses do modelo.
- Exemplo: árvore muito rasa para fronteira complexa.

Variância

- Sensibilidade a mudanças na amostra de treinamento.
- Exemplo: árvore profunda pode mudar bastante com poucos dados.

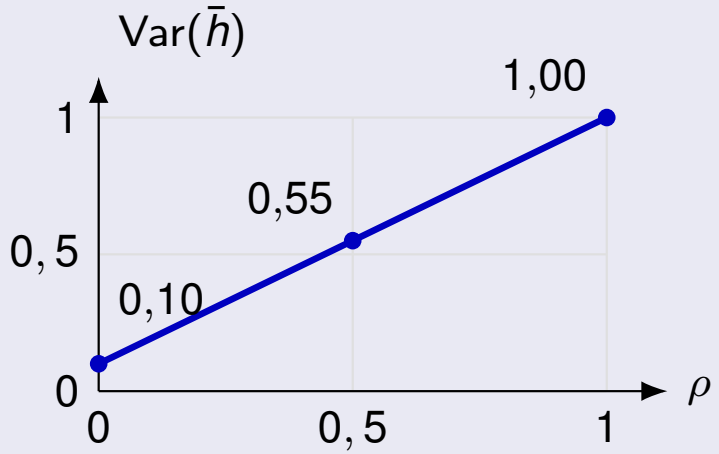
Por que a diversidade importa?

$$\text{Var}(\bar{h}) = \sigma^2 \left(\rho + \frac{1 - \rho}{M} \right)$$

Significado dos símbolos

- ρ (rho): correlação entre preditores.
- σ^2 (sigma ao quadrado): variância individual.
- M: número de preditores.
- $\bar{h}$: média das predições.

Exemplo: $M = 10$, $\sigma^2 = 1$



Interpretação

$\rho = 0$: variância σ^2/M; $\rho = 1$: sem redução. Supõe variâncias iguais e correlação comum. Não mede acurácia.

Atividade: diversidade dos erros

Modelo	A	B	C	D
Real	1	0	1	0
DT	1	1	1	0
KNN	1	0	0	0
SVM	0	0	1	0

Atividade

Determine a acurácia de cada modelo e da votação majoritária.

Discussão da atividade

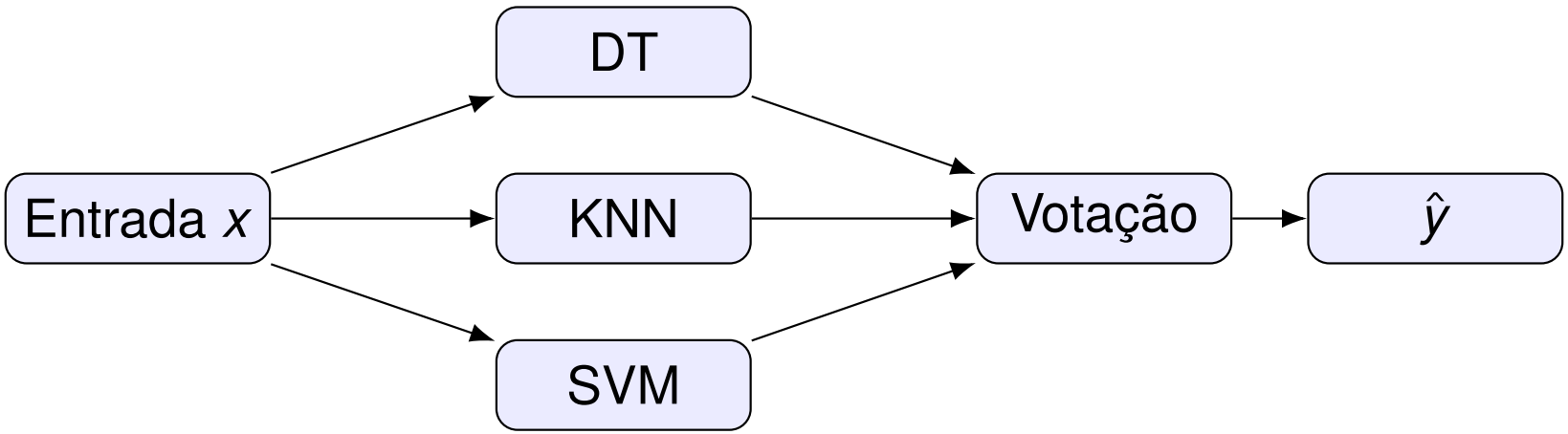
Resultado

- Cada modelo acerta três entre quatro casos (75%).
- Votação: 1, 0, 1, 0; quatro acertos (100%).
- Neste exemplo, os três erros ocorrem em casos distintos.

Observação

Não é uma garantia: modelos podem concordar justamente nos erros.

Votação: arquitetura



Treinamento

- Os modelos são ajustados sobre os dados de treinamento.
- A combinação pode ser fixa ou ponderada.

Hard voting

$$\hat{y} = \arg\max_{c} \sum_{m=1}^{M} w_m \mathbf{1}[h_m(x) = c]$$

Significado dos símbolos

- x: entrada; c: classe candidata; $\hat{y}$: classe escolhida.
- M: total de modelos; $h_m(x)$: classe prevista pelo modelo m; w_m: peso de seu voto.
- $\mathbf{1}[\cdot]$: vale 1 se a condição é verdadeira, 0 caso contrário.
- $\arg\max_c$: escolhe a classe de maior soma de votos.

Para discutir

Com pesos iguais, DT=1, KNN=0 e SVM=1: quem vence? E se houver empate?

Soft voting

$$\hat{y} = \arg\max_{c} \frac{\sum_{m=1}^{M} w_m p_m(c \mid x)}{\sum_{m=1}^{M} w_m}$$

Significado dos símbolos

- $p_m(c \mid x)$: probabilidade estimada pelo modelo m para a classe c, dada a entrada x.
- $w_m \geq 0$: peso do modelo; a soma dos pesos deve ser positiva.
- O denominador normaliza os pesos: média ponderada.
- $\arg\max_c$: escolhe a classe de maior média.

Observação

Alinhe as classes. Probabilidades mal calibradas podem prejudicar a combinação.

Atividade: duas formas de votar

Modelo	$P(0)$	$P(1)$
DT	0,49	0,51
KNN	0,48	0,52
SVM	0,95	0,05

Atividade

Calcule hard voting e soft voting com pesos iguais.

Discussão: hard e soft

$$P(1) = (0{,}51 + 0{,}52 + 0{,}05)/3 = 0{,}36$$

Resultados

- Hard: dois votos para classe 1.
- Soft: probabilidade média 0,36 para classe 1; prediz 0.
- As regras podem divergir; nenhuma é universalmente superior.

Votação ponderada e Trabalho 02

$$w = (0{,}5, 0{,}3, 0{,}2), \quad \sum_{m=1}^{3} w_m = 1$$

Exemplo de representação

DT, KNN e SVM contribuem com 50%, 30% e 20%. Para a classe 1:

$$P(1) = 0{,}5(0{,}51) + 0{,}3(0{,}52) + 0{,}2(0{,}05) = 0{,}421$$

Possível problema de otimização

Encontrar pesos não negativos que maximizem o F1 de validação, por exemplo, usando AG, PSO ou CLONALG, conforme o tema.

Observação

Nunca utilize o teste para encontrar os pesos.

Bagging: ideia geral

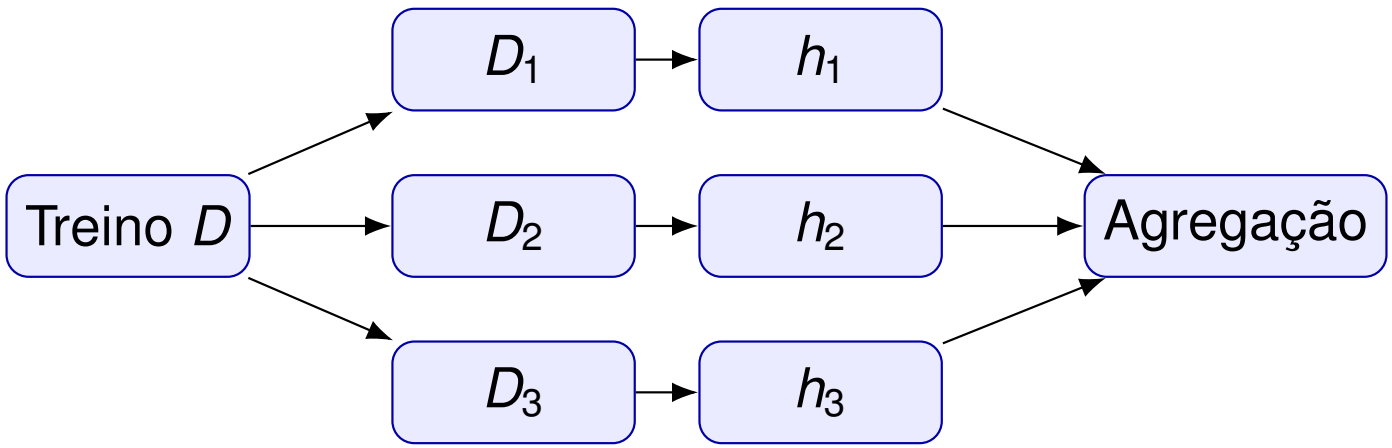
Bootstrap Aggregating

- Criar M amostras com reposição do treino.
- Ajustar um estimador independente por amostra.
- Agregar as predições por voto ou média.

Para discutir

Por que árvores diferentes podem aprender regras distintas?

Bagging: fluxo do treinamento



Duas fontes de decisão

O sorteio das amostras cria diversidade; a regra de agregação combina as predições.

Bootstrap: exemplo

Conjunto	Índices
Treino	1, 2, 3, 4, 5
Amostra 1	1, 1, 3, 5, 3
Amostra 2	2, 4, 4, 1, 5
Amostra 3	3, 2, 5, 2, 1

Atividade

Quais elementos estão fora da amostra 1? E por que a amostra tem tamanho 5?

Discussão: out-of-bag

$$P(i \text{ ausente}) = (1 - 1/n)^n \xrightarrow{n \to \infty} e^{-1} \approx 0{,}368$$

De onde vem essa probabilidade?

- n: tamanho do treino e número de sorteios; i: índice de um exemplo.
- $1/n$: chance de sortear i uma vez; $1 - 1/n$: chance de não sortear.
- Elevamos a n porque os sorteios com reposição são independentes; e é a base do logaritmo natural.

Observação

Faltam 2 e 4 na amostra 1. OOB avalia exemplos em modelos que não os usaram; não substitui automaticamente o teste externo.

Bagging: pseudocódigo

Algoritmo

Entrada: Treino D, número de modelos M
Para $m \leftarrow 1$ **até** M **faça**
$D_m \leftarrow \text{bootstrap}(D)$ $h_m \leftarrow \text{treinar}(D_m)$
Saída: Combinação de $h_1, \ldots, h_M$

Para discutir

Em que etapa é possível paralelizar o treinamento?

Random Forest

Além do bootstrap

- Cada árvore é ajustada sobre uma amostra do treino.
- Em cada divisão, considera-se um subconjunto aleatório de atributos candidatos.
- As árvores tornam-se menos correlacionadas.

Observação

A seleção de atributos ocorre a cada nó, não apenas uma vez por árvore.

Random Forest: divisão de um nó

Árvore	Atributos candidatos no nó
1	x_1, x_4, x_7
2	x_2, x_3, x_7
3	x_1, x_5, x_6

Escolha

- Em cada nó, selecionar a melhor divisão entre os candidatos.
- Exemplo: redução de impureza Gini.

Para discutir

Qual a diferença para bagging de árvores com todos os atributos candidatos?

Hiperparâmetros e efeitos

Parâmetro	Efeito principal
n_estimators	quantidade de árvores
max_depth	complexidade de cada árvore
max_features	diversidade nas divisões
min_samples_leaf	suavização das folhas

Observação

Mais árvores aumentam custo; o ganho pode estabilizar.

Boosting: aprendizagem sequencial

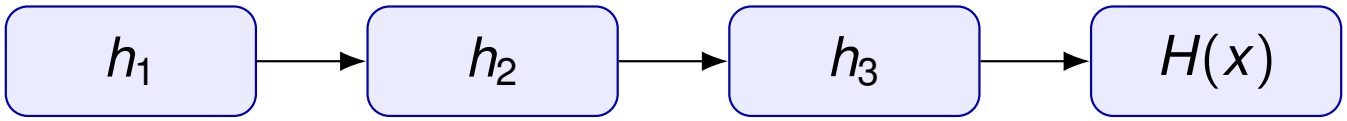
Ideia geral

- Construir estimadores de forma dependente.
- Cada novo estimador busca corrigir aspectos não explicados pelos anteriores.
- A combinação final é aditiva ou ponderada.

Para discutir

Por que essa dependência dificulta paralelizar o treinamento entre etapas?

Boosting: fluxo do treinamento



Cada etapa depende da anterior.

Contraste com bagging

Em bagging, os estimadores podem ser ajustados de forma independente; em boosting, a construção é sequencial.

AdaBoost: pesos das amostras

$$\epsilon_t = \sum_i w_i \mathbf{1}[y_i \neq h_t(x_i)], \quad \sum_i w_i = 1$$

$$\alpha_t = \tfrac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

Erro e peso do estimador

- t: etapa; i: exemplo; w_i: peso atual do exemplo.
- y_i: rótulo real; $h_t(x_i)$: rótulo previsto.
- ϵ_t: soma dos pesos dos exemplos incorretos.
- α_t: peso do estimador; ln: logaritmo natural.

Observação

Menor erro, maior peso. Caso binário clássico: rótulos ± 1 e $0 < \epsilon_t < 0{,}5$. Bibliotecas podem usar variantes.

AdaBoost: atualização e decisão

$$w_i \leftarrow \frac{w_i \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$H(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$$

Interpretação

- Acerto: $y_i h_t(x_i) = +1$; fator $e^{-\alpha_t}$ reduz o peso relativo.
- Erro: $y_i h_t(x_i) = -1$; fator e^{α_t} aumenta o peso relativo.
- Z_t: soma dos pesos após a multiplicação pelos fatores, para normalizar a soma em 1.
- $H(x)$: classe final; sign escolhe o sinal da soma dos votos ponderados. Empates exigem regra.

Atividade: AdaBoost

Atividade

Quatro exemplos começam com peso 0,25. Um estimador erra somente o quarto. Calcule ϵ e α e determine os pesos normalizados após a atualização.

Discussão: AdaBoost

$$\epsilon = 0{,}25, \quad \alpha = \tfrac{1}{2}\ln(0{,}75/0{,}25) \approx 0{,}5493$$

Atualização passo a passo

- Cada acerto: $0{,}25e^{-\alpha} \approx 0{,}1443$.
- O erro: $0{,}25e^{\alpha} \approx 0{,}4330$.
- Normalização: $Z \approx 3(0{,}1443) + 0{,}4330 = 0{,}8660$.

$$w \approx (0{,}1667, 0{,}1667, 0{,}1667, 0{,}5)$$

Observação

O exemplo incorreto recebe metade do peso total; ruídos podem receber atenção excessiva.

Gradient Boosting

$$F_t(x) = F_{t-1}(x) + \eta h_t(x)$$

Significado dos símbolos

- x: entrada; t: etapa de boosting.
- $F_{t-1}(x)$: predição acumulada anterior; $F_t(x)$: nova predição.
- $h_t(x)$: contribuição do novo estimador, ajustado para reduzir a perda atual.
- η: taxa de aprendizagem; controla o tamanho da correção.

Observação

Para $L = \frac{1}{2}(y - F)^2$, o gradiente negativo é $y - F$ (resíduo). Em classificação, depende da perda.

Exemplo: boosting para regressão

Exemplo	Real	F_0	Resíduo
A	2	4	-2
B	4	4	0
C	6	4	2

Atividade

Se h_1 reproduz exatamente os resíduos e $\eta = 0,5$, quais as novas predições?

Discussão: correção gradual

$$F_1 = (4, 4, 4) + 0{,}5(-2, 0, 2) = (3, 4, 5)$$

Resultado

- $F_0 = (4, 4, 4)$: predições iniciais.
- $h_1 = (-2, 0, 2)$: resíduos $y - F_0$; $\eta = 0{,}5$ aplica metade da correção.
- Erro quadrático médio: $\frac{1}{3} \sum_i (y_i - F(x_i))^2$.
- Antes: $(4 + 0 + 4)/3 = 8/3$; depois: $(1 + 0 + 1)/3 = 2/3$.

Observação

Para classificação, utiliza-se uma perda apropriada, como log-loss.

Boosting: parâmetros

Parâmetro	Interpretação
n_estimators	número de etapas
learning_rate	tamanho da contribuição
max_depth	complexidade do estimador (GB)

Observação

Valores de taxa e número de etapas interagem; ajuste usando validação.

Stacking: combinação aprendida

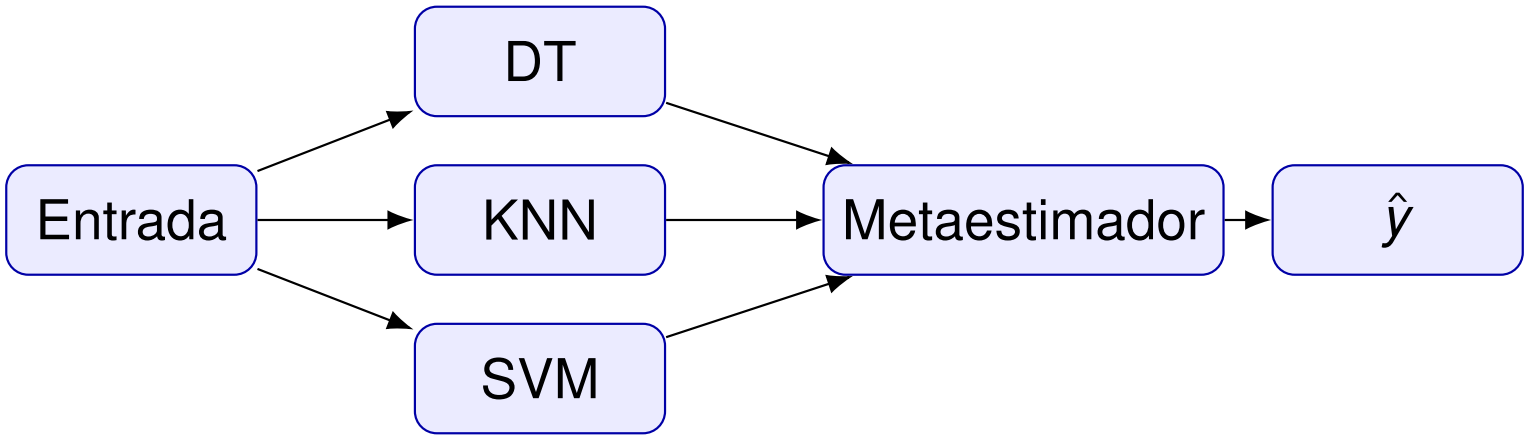
Dois níveis

- Nível 0: modelos base, como DT, KNN e SVM.
- Nível 1: metaestimador aprende a combinar suas predições.
- Exemplo: regressão logística como metaestimador.

Para discutir

Se treinarmos o metaestimador nas predições dos próprios dados de ajuste, qual o risco?

Stacking: estrutura



Observação

No treinamento, as entradas do metaestimador são predições geradas fora do fold. Na inferência, são predições dos modelos base reajustados.

Stacking: predições fora do fold

Fold de validação	Base treinada em	Predição para
1	2, 3	1
2	1, 3	2
3	1, 2	3

Construção

- Concatenar as predições fora do fold de todos os estimadores.
- Ajustar o metaestimador com essas características e rótulos.

Stacking: inferência

Após construir as características OOF

- Reajustar estimadores base em todo o treinamento.
- Para uma nova entrada, obter predições dos modelos base.
- O metaestimador transforma essas predições na saída final.

Observação

O teste permanece isolado durante toda a construção.

Atividade: vazamento de dados

Atividade

Uma árvore profunda é ajustada em toda a base de treino e gera probabilidades para os mesmos exemplos. Essas probabilidades são usadas para treinar a regressão logística. Explique por que esse procedimento é problemático e como corrigi-lo.

Discussão: treinamento correto

Problema

- Predições no próprio treino representam desempenho excessivamente otimista.
- O metaestimador aprende em condições diferentes daquelas de inferência.

Correção

- Gerar características out-of-fold; reajustar bases no treino completo.

Comparação entre famílias

Método	Diversidade	Combinação
Voting	modelos diferentes	regra fixa
Bagging	bootstrap	voto/média
Random Forest	bootstrap + atributos	média de probabilidades
Boosting	ajuste sequencial	soma ponderada
Stacking	modelos base	metaestimador

Observação

As categorias não são mutuamente exclusivas: ensembles podem combinar outros ensembles.

Protocolo de avaliação

Separação dos dados

- Reservar teste estratificado antes de ajustar modelos.
- Na parte de treinamento, usar validação cruzada com folds comuns.
- Escalonamento, imputação e seleção de atributos devem estar dentro do pipeline.

Para discutir

Por que ajustar PCA antes da divisão pode provocar vazamento?

Métricas e custo

$$F_1 = 2\frac{P\,R}{P+R}$$

Não olhar apenas a acurácia

- $P = VP/(VP + FP)$: precisão, qualidade dos positivos preditos.
- $R = VP/(VP + FN)$: recall, cobertura dos positivos reais.
- VP: verdadeiros positivos; FP: falsos positivos; FN: falsos negativos.
- F_1: média harmônica entre precisão e recall; penaliza valores baixos de uma das duas métricas.

Observação

Compare média, dispersão, matriz de confusão e tempo. Escolha métricas conforme a aplicação.

Conexão com Trabalho 02

Aprendizagem de máquina

- Utilizar a mesma base, definição do alvo, divisão e métricas para os modelos.
- Comparar estimadores individuais com ensembles.
- Justificar pré-processamento, parâmetros e limitações.

Metaheurísticas

- Podem otimizar pesos ou hiperparâmetros quando isso condiz com o tema; manter a avaliação objetiva e isolada do teste.

Síntese

O que aprendemos?

- Voting combina decisões; bagging busca estabilidade.
- Random Forest introduz diversidade por atributos.
- Boosting corrige a perda sequencialmente.
- Stacking aprende como combinar predições.

Para discutir

Qual abordagem você experimentaria primeiro e por quê?

Referências

- Russell e Norvig. Artificial Intelligence: A Modern Approach, 4ª ed.
- Breiman. Bagging Predictors, 1996; Random Forests, 2001.
- Freund e Schapire. A Decision-Theoretic Generalization of On-Line Learning, 1997.
- Friedman. Greedy Function Approximation: A Gradient Boosting Machine, 2001.
- Wolpert. Stacked Generalization, 1992.

Próxima aula: prática

Objetivos

- Executar a comparação no mesmo dataset da Aula 11.
- Investigar parâmetros, erros e estabilidade.
- Adaptar o protocolo à base do Trabalho 02.

Dúvidas



Verificação de aprendizagem

Atividade

Explique a diferença entre bagging e boosting; calcule uma votação soft e explique por que stacking precisa de predições fora do fold. Indique uma estratégia de comparação justa.