

\_\_\_\_\_

**Professora Stella Sposito**  
stella.sposito@facens.br

# Ensemble

---

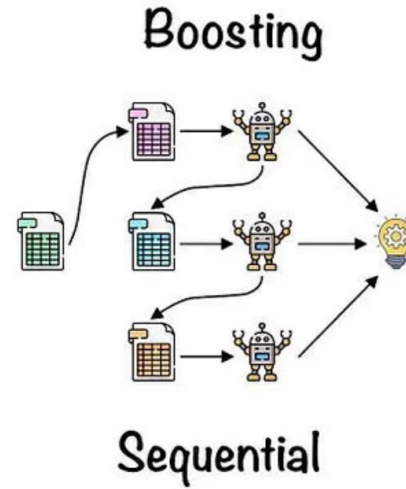
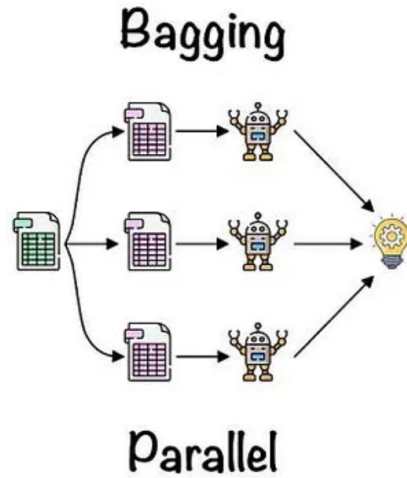
- Se temos que responder uma pergunta complexa, é melhor:
  - perguntar para apenas uma pessoa especialista a resposta ou
  - perguntar para várias pessoas (que podem ser especialistas também) e agrupar as respostas?



## Aprendizado Ensemble

Predições agregadas de um grupo de preditores (como classificadores ou regressores)

# Métodos de Ensemble



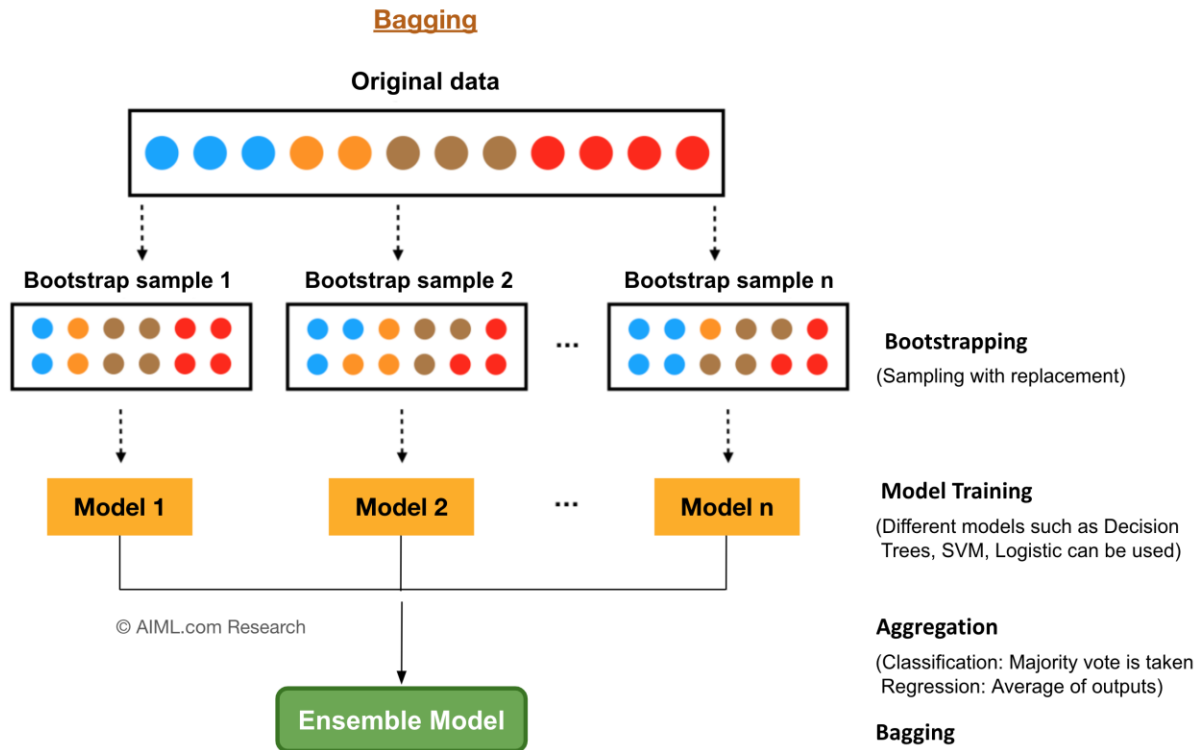


---

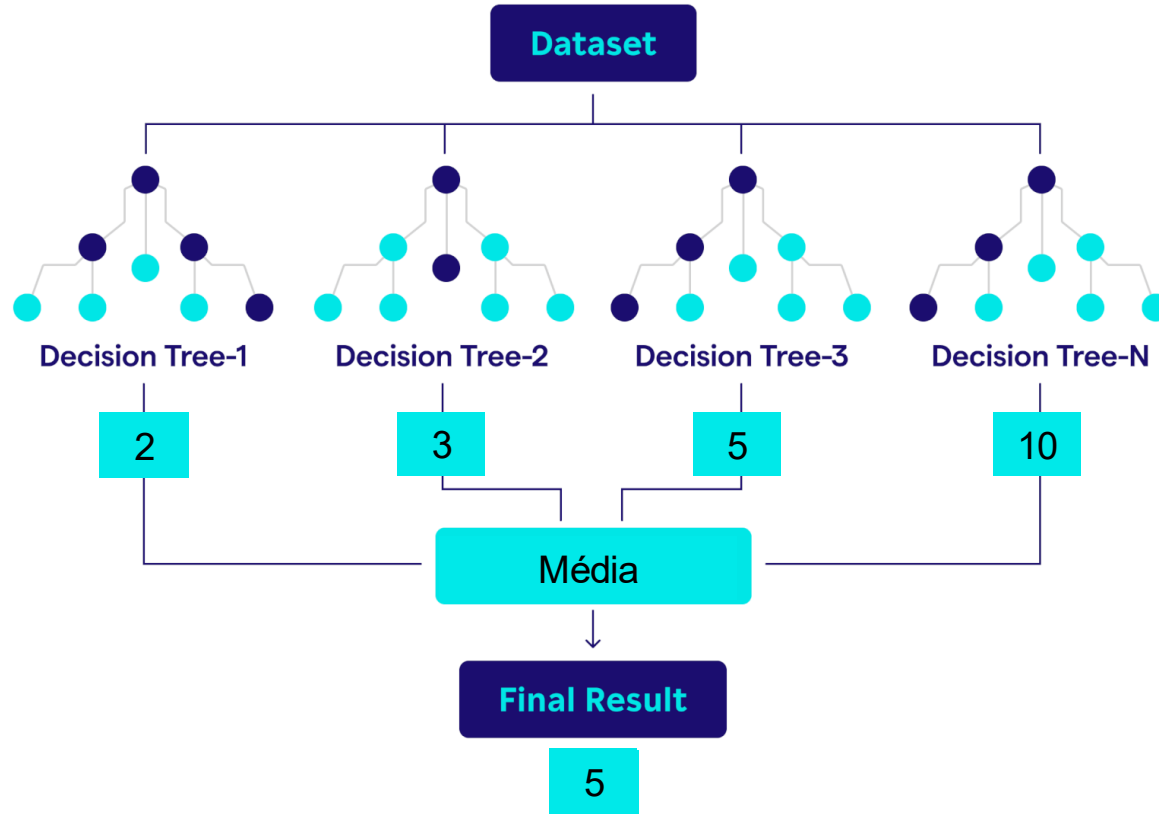
# Bagging

# Bagging

- Bagging é uma técnica de criar conjuntos de dados de treinamento por meio de amostragem aleatória com reposição.
- Vem da abreviação de bootstrap aggregation.
- Após o treinamento dos modelos individuais, as previsões de cada modelo são combinadas para obter uma previsão final por **votação** ou **média**.



# Bagging – Random Forest



# Bagging – Random Forest

- É um conjunto de árvores de decisão que funcionam com o método bagging para escolher o melhor resultado. Ela possui, basicamente, 4 passos:
  1. Seleção aleatória de algumas features;
  2. Seleção da feature mais adequada para a posição de nó raiz;
  3. Geração dos nós filhos;
  4. Repetição dos passos até que se atinja a quantidade de árvores necessárias.

## Treinamento

Várias árvores são criadas;

Cada árvore aprende regras próprias usando bootstrap (cada uma tenta reduzir impureza – gini e entropia – reduzir o erro e maximizar o ganho)

Combina todas as árvores ao final.

## Teste

Cada árvore treinada faz sua previsão;

Random forest combina todas (votação ou média) e roda para cada dado de teste.



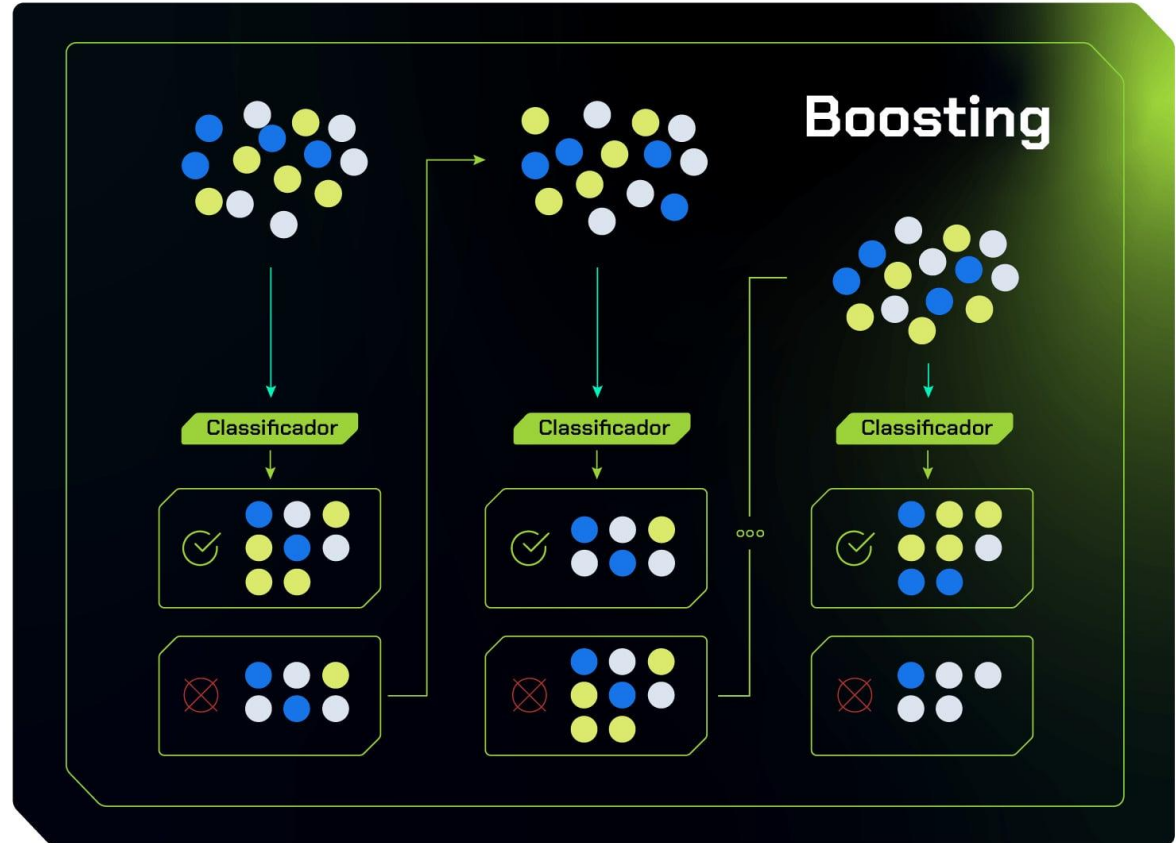
---

# Boosting



# Boosting

- O boosting concentra-se no treinamento sucessivo de modelos básicos focando em corrigir os erros de iterações anteriores.
- O modelo aprende com seus erros e melhora seu desempenho iterativamente.



# Boosting

---

1. **Inicialização de pesos:** cada exemplo de treinamento recebe o mesmo peso;
2. **Treinamento de um classificador fraco:** um modelo simples;
3. **Cálculo do erro:** o erro do classificador fraco é calculado;
4. **Atualização de pesos:** os exemplos classificados incorretamente recebem pesos maiores;
5. **Repetição:** os passos 2 a 4 são repetidos várias vezes;
6. **Combinação de aprendizes fracos:** o modelo final é composto por todos os aprendizes fracos treinados nas etapas anteriores;
7. **Previsão:** o modelo finalizado é usado para prever os rótulos de classes de novas instâncias.

# Boosting - Algoritmos

## AdaBoost

### Adaptive Boosting

- Foca em **ajustar pesos** das observações.
- A cada iteração, aumenta o peso dos exemplos difíceis (mal classificados).
- O próximo modelo tenta acertar esses pesos maiores.
- Geralmente usa **árvores muito rasas**.
- A adaptação está nos **pesos das amostras**.

## Gradient Boosting

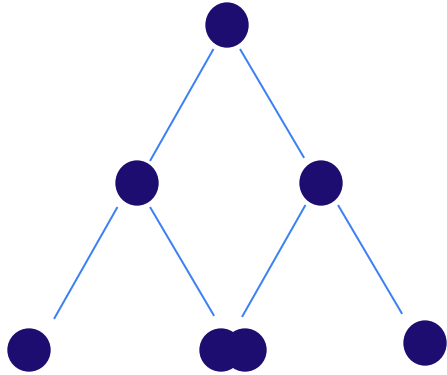
- Foca em ajustar o modelo via **gradiente da função de perda**.
- Cada nova árvore “aprende” os **resíduos** (erros) do modelo anterior.
- Usa uma função de perda diferenciável (log-loss, MSE, MAE, etc.).
- As árvores podem ser **mais profundas**.
- A adaptação está nos **resíduos (gradiente da perda)**.



LightGBM, XGBoost e CatBoost

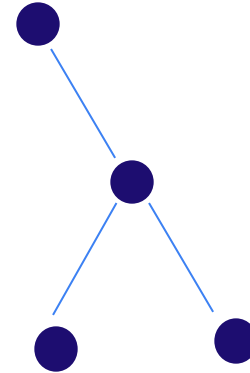
# XGBoost X LightGBM

## XGBoost – Level-wise



- Expande todos os nós de um nível antes de descer.
- Mais equilibrado.
- Mais lento, porém mais estável (menor risco de overfitting).

## LightGBM – Leaf-wise



- Cresce verticalmente em profundidade.
- Muito mais rápido.
- Mais preciso, mas pode **overfittar** se não limitar max\_depth.

Ambos suportam variáveis categóricas, mas quando temos muitas variáveis categóricas, o indicado é usar CatBoost.



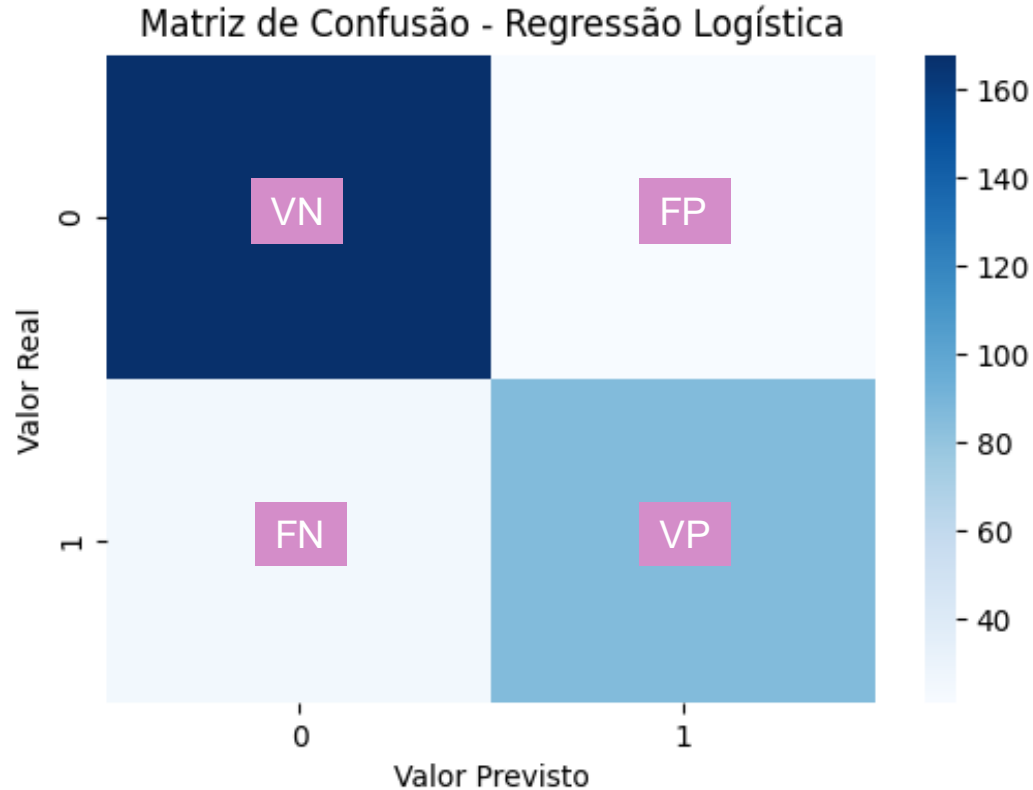
---

## Métodos de Avaliação

# Matriz de Confusão

		Valor Predito	
		SIM	NÃO
Valor Real	SIM	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	NÃO	Falso Positivo (FP)	Verdadeiro Negativo (VN)

# Matriz de Confusão



# Matriz de Confusão

---

```
Matriz de Confusão  
[[164  25]  
 [ 16  95]]
```

	Predito	
	0	1
Real 0	TN	FP
Real 1	FN	TP



# Acurácia

---

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP}$$

- De tudo que o modelo classificou, quantos ele acertou?
- Útil para datasets balanceados, extremamente preocupante quando há desbalanceamento de classes.

# Precisão

---

$$\textit{precisão} = \frac{VP}{VP + FP}$$

- Das classes que o modelo disse que eram positivas, quantas realmente eram?
- Útil quando o custo de um falso positivo é alto, por exemplo, no caso de fraude, você não quer bloquear cartões de clientes honestos (FP), porque isso causa reclamações e perda de confiança.

# Recall

---

$$\text{Recall} = \frac{VP}{VP + FN}$$

- Entre todas as classes **realmente** positivas, quantas o modelo encontrou?
- Útil quando o custo de um FN é alto, por exemplo, para o caso de diagnóstico de câncer, um falso negativo indica que uma pessoa com câncer foi classificada como saudável.

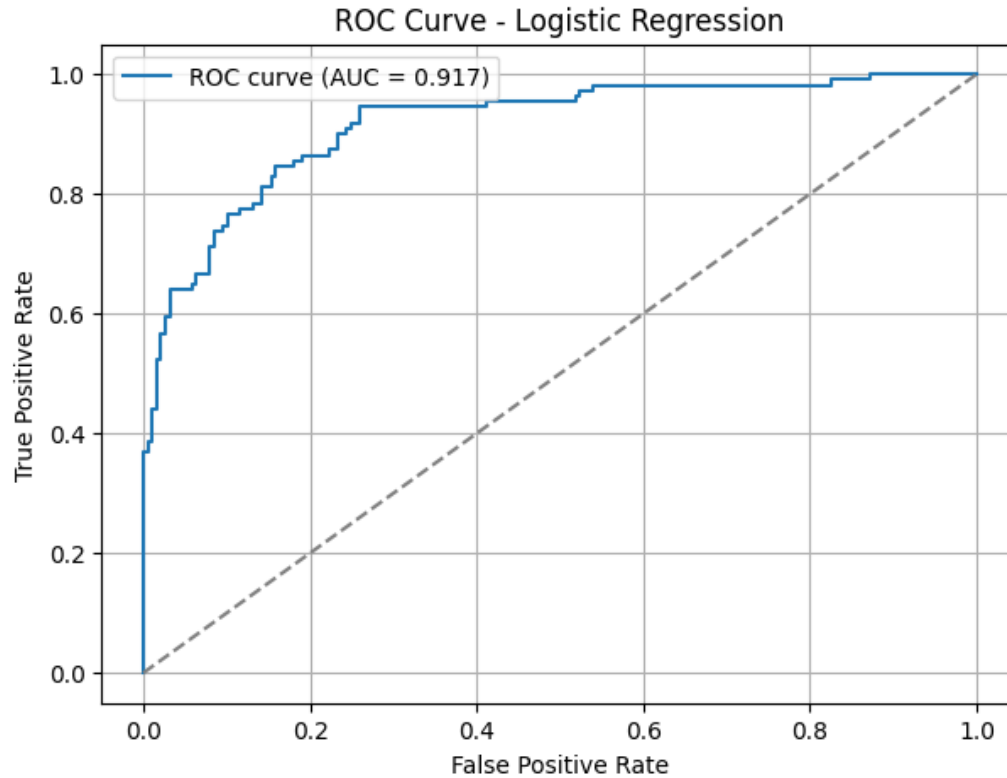
# F1 Score

---

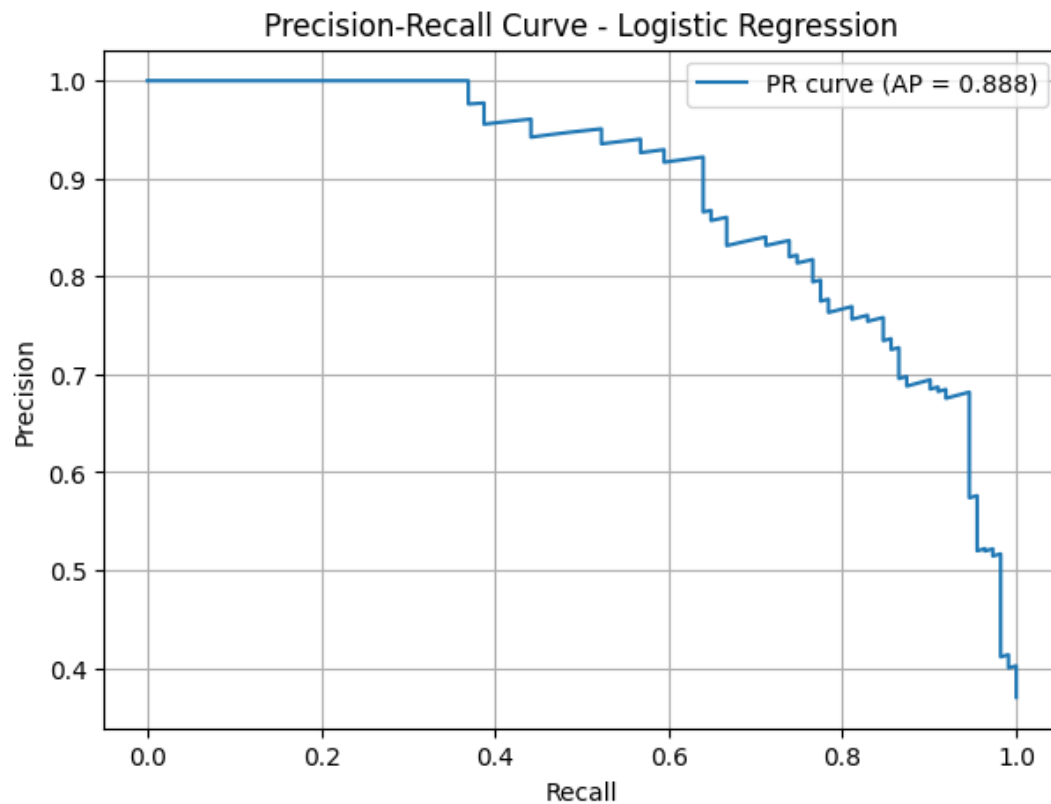
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Mede o equilíbrio entre precisão e recall
- Útil quando as classes são desbalanceadas e você precisa equilibrar entre as duas métricas, quando as duas são importantes para analisar.

# Curva ROC AUC



# Curva AUC-PR





---

## Tuning de Hiperparâmetros

# O que é e para que serve?

---

Tuning de hiperparâmetros é o processo de encontrar a melhor combinação de configurações que controlam **como** um modelo de machine learning aprende.

Exemplos: *learning\_rate*, *max\_depth*, *n\_estimators*, *regularização*, etc.

Esses hiperparâmetros **não** são aprendidos pelo modelo, precisamos escolhê-los.

Aumenta a performance do modelo;

Evita overfitting ou underfitting;

Melhora a generalização para novos dados;

Toma o modelo mais eficiente;

Pode ser determinante para modelos complexos como de bagging e boosting.



# Métodos para otimização

---

## Grid Search

Testa todas as combinações possíveis de hiperparâmetros;  
Lento e caro computacionalmente;  
Bom quando o espaço de busca é pequeno.

## Random Search

Testa amostras aleatórias do espaço de hiperparâmetros;  
Mais rápido que Grid Search;  
Pode encontrar boas soluções sem explorar todas as combinações.

## Optuna

Otimização inteligente e automática;  
Modela estatisticamente a relação entre hiperparâmetros e desempenho;  
Muito mais eficiente que Grid e Random Search e otimiza qualquer modelo.

## Ray Tune

Ideal para paralelizar tuning em múltiplas CPUs/GPUs;  
Suporta vários algoritmos de busca (Random, Bayes, Optuna);  
Escalável para grandes datasets e deep learning.



---

## Interpretabilidade de Modelos

# SHAP Values

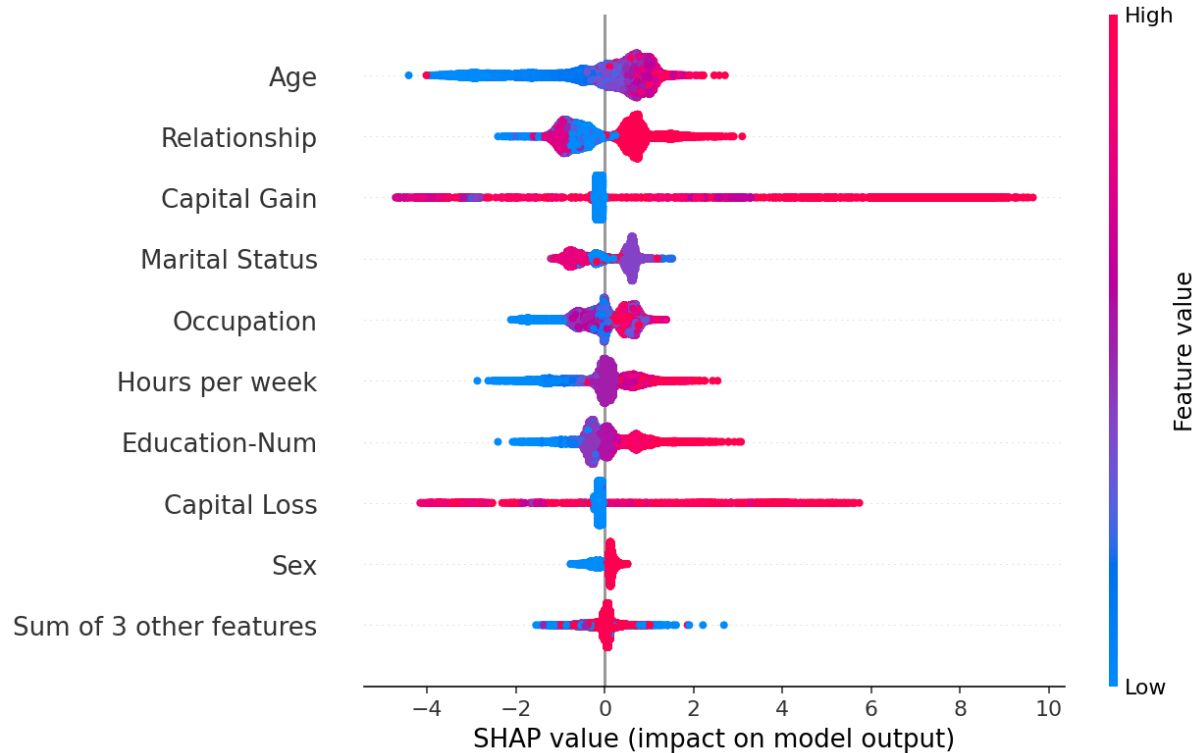
- **SHAP (SHapley Additive exPlanations)** é uma técnica de interpretação baseada na teoria dos valores de Shapley, da Teoria dos Jogos.
- Ele calcula **quanto cada feature contribui para aumentar ou diminuir a predição do modelo**.
- Cada previsão é decomposta em:
  - o **Valor base (baseline)**: predição média do modelo.
  - o **Contribuições das features**: impacto individual de cada variável.

Explicar **como** e **por que** o modelo tomou determinada decisão.

Identificar **principais variáveis** que influenciam predições.

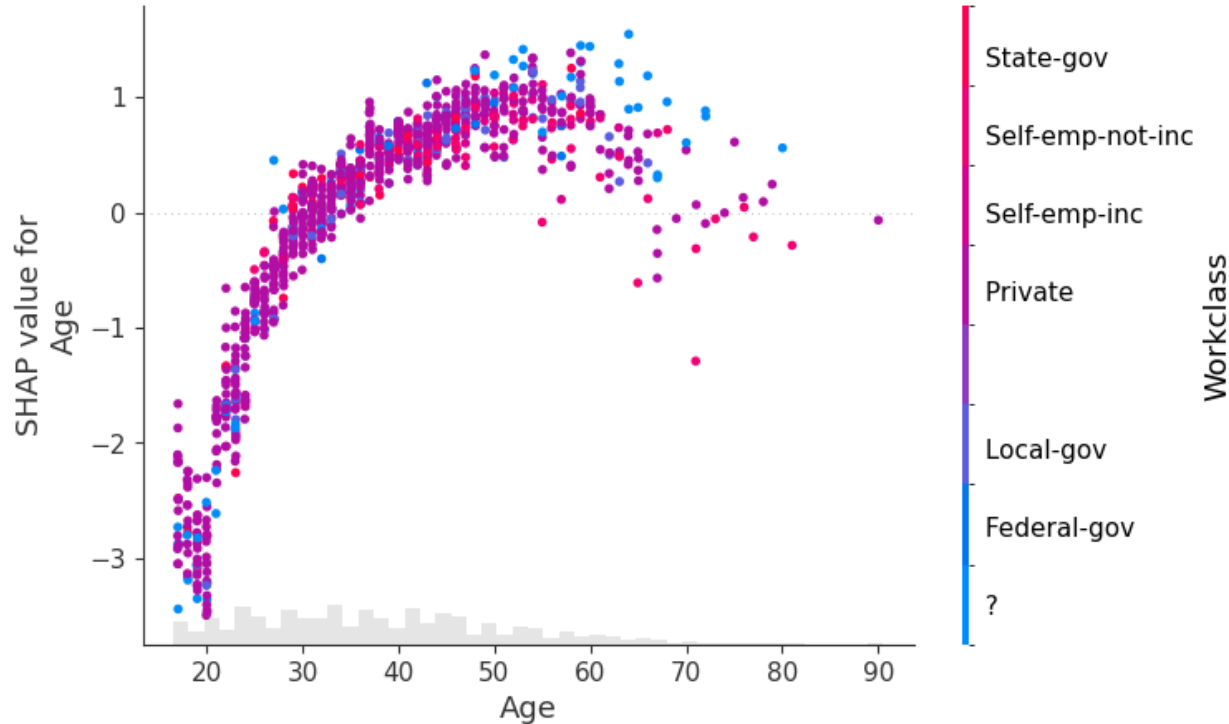
Aumentar **transparência** e **confiabilidade** em modelos complexos (tree boosting, deep learning etc.).

# Tipos de SHAP



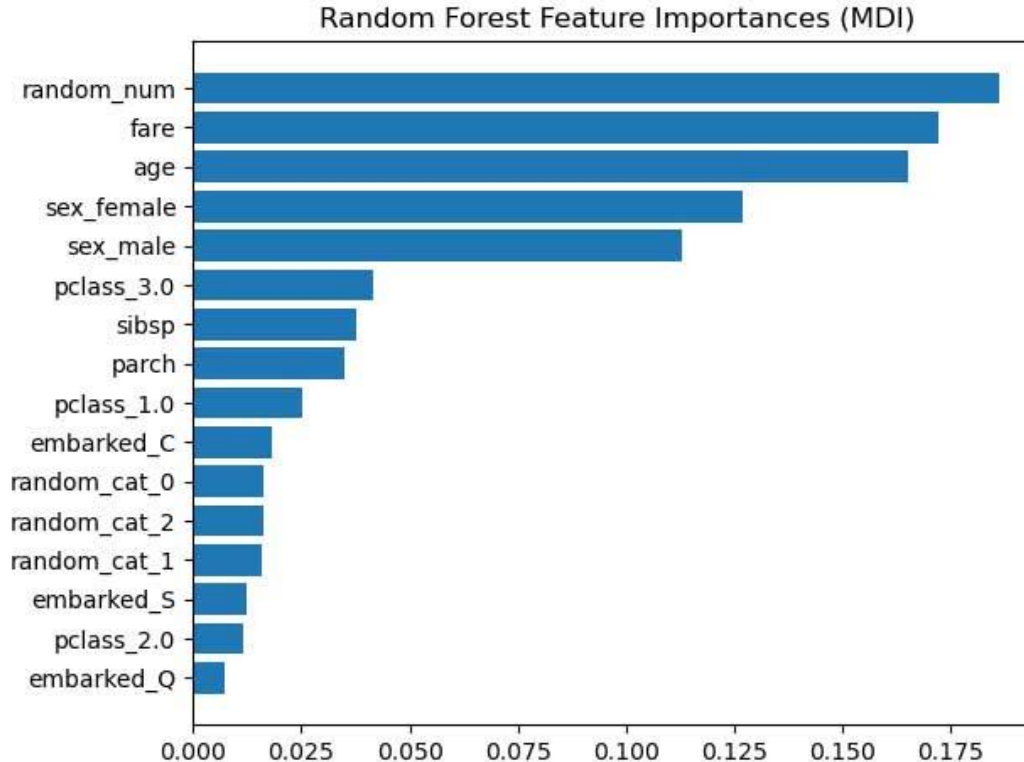
- A cor mostra o valor da feature (alto/baixo).
- A dispersão mostra **quanto a variável impacta** a predição.

# Tipos de SHAP



- Esse gráfico mostra a relação entre **uma feature** e seu impacto no modelo.
- Revela **não linearidades** e **interações** entre variáveis.
- Ajuda a entender como mudanças na feature alteram a predição.

# Feature Importante – Random Forest





---

## Mão na Massa

## Tarefa de Aula

---

- Entregar notebook feito em aula juntamente com a professora.

## Tarefa de Casa

---

- Escolher um dos métodos de tuning de hiperparâmetros e rodar para otimizar um dos modelos feitos em aula (fica a seu critério) e responder as perguntas a seguir:
  - Quanto tempo demorou para rodar?
  - Você conseguiu otimizar algum resultado? Se sim, qual? Se não, o que você acha que aconteceu?





Facens

AQUI TEM ENGENHARIA