



AULA 9 - Ensemble

#BootcampMIA2022 #SomosMIA



Quem somos?



Jéssica dos Santos

Head de Dados na NeuralMed

- Mestra em Sistemas de Informação pela USP com pesquisa em inteligência artificial e área de aplicação em saúde
- Co-fundadora da MIA



jessica-santos



jessica-santos-oliveira



Quem somos?



Carol Silva

Data Scientist na Neuralmed

- Mestra em processamento e análise de sinais e graduada em Engenharia elétrica pela PUC-MG



carolinasilvadev



carolina-fernanda-silva



Uma inspiração...

"A pergunta não é quem vai me deixar; a pergunta é quem vai me impedir."

(Ayn Rand)

**Comentamos sobre
clustering na aula passada...**



Alguém lembra o que é?





Aprendizado supervisionado e não supervisionado

Aprendizado supervisionado

- ▷ Regressão 
- ▷ Classificação 

Classificação



Regressão

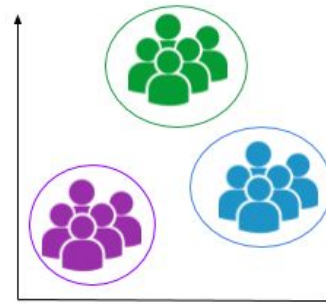


Aprendizado
supervisionado

Aprendizado não supervisionado

- ▷ Clustering

Agrupamento

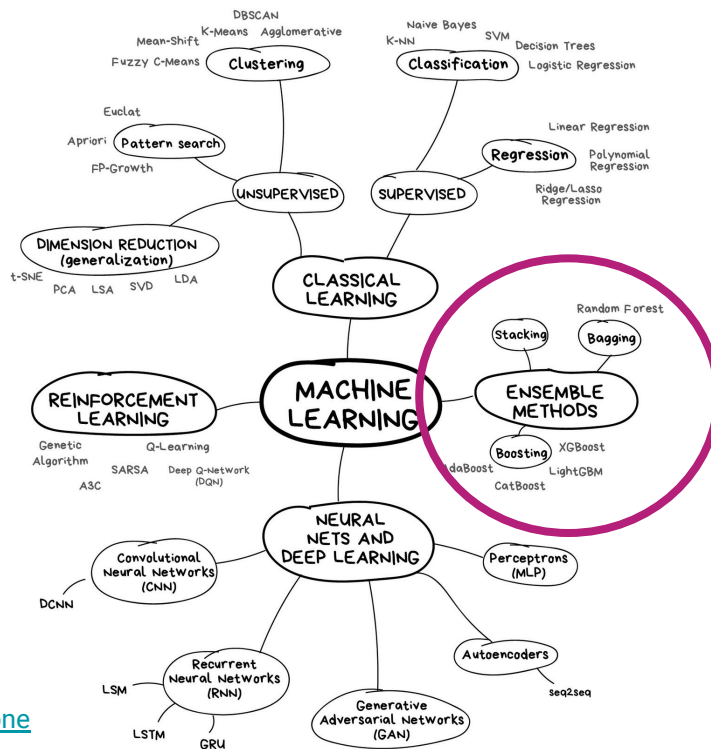


Aprendizado não
supervisionado

**O que
veremos na
aula de hoje?**



Aprendizado supervisionado e não supervisionado



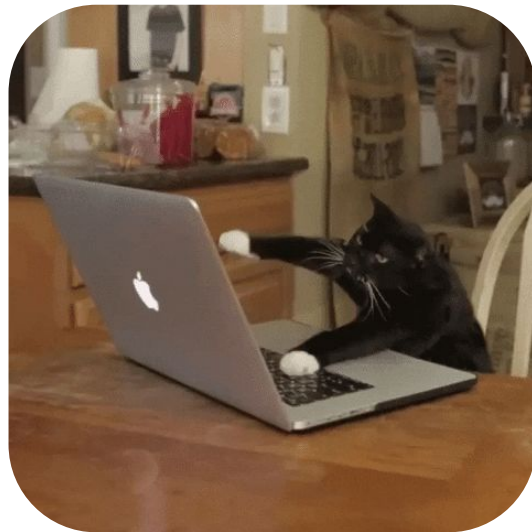
Ensemble

Junção de vários modelos, geralmente mais fracos, que juntos geram um melhor preditor. Basicamente segue a ideia de que várias "cabeças" pensam melhor do que uma.



Ensemble

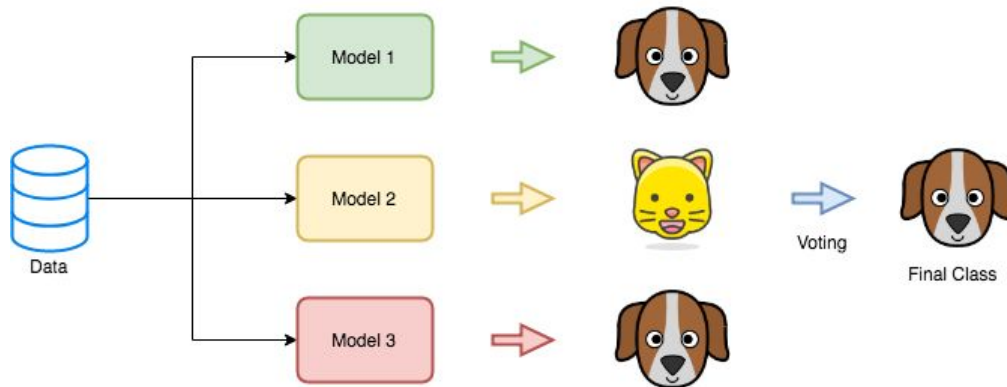
Vamos ver na prática:
<Notebook>



Tipos de Ensemble:

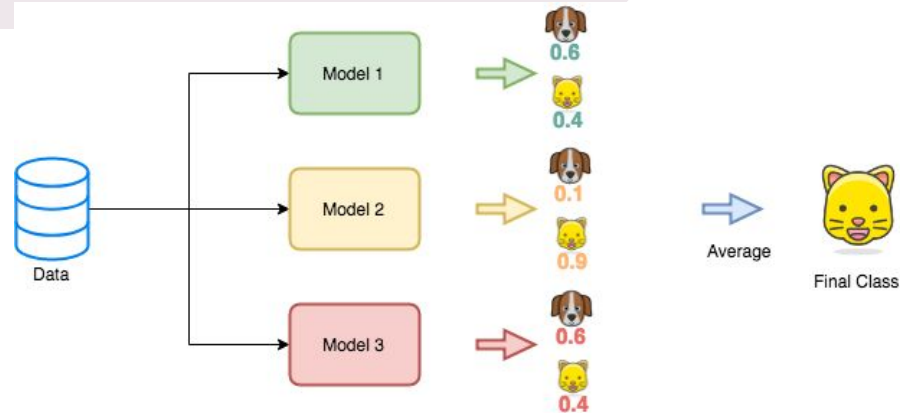
- ▷ Voting Based Classifier (o que acabamos de ver):
 - Majority Vote
 - Average Classifier
- ▷ Stacking
- ▷ Boosting
- ▷ Bagging

Majority Vote



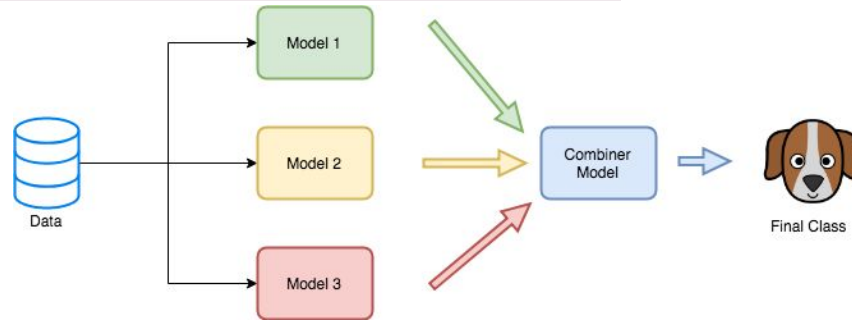
A ideia é fazer uma votação entre as predições dos modelos. A classe que tiver mais votos vence. Também podemos ter uma variação desse algoritmo, o **Weighted Voting Classifier**, em que na votação alguns modelos têm mais peso que outros.

Average Classifier



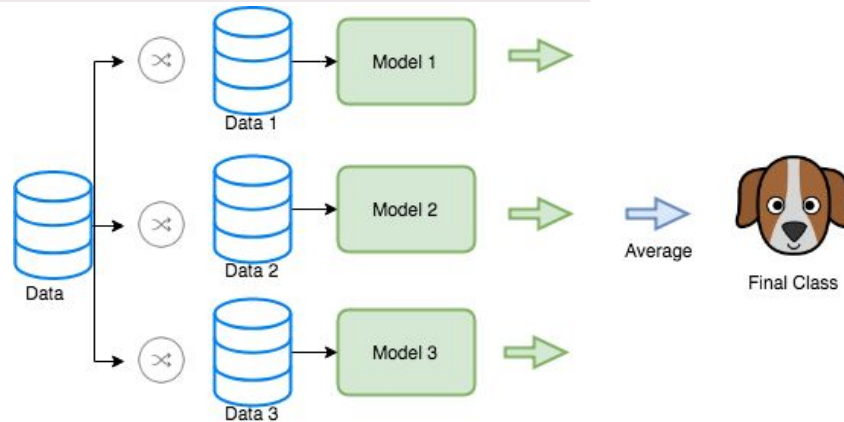
A ideia é similar ao anterior, porém ao invés de uma votação é calculada a média das previsões. Da mesma forma podemos ter alguns modelos com mais peso que outros tendo um **Weighted Average Classifier**

Stacking



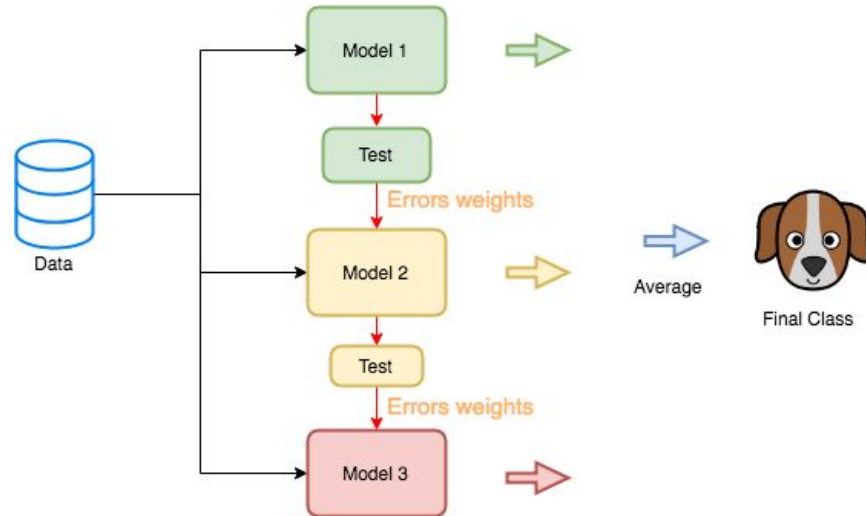
Nesse modelo as previsões dos modelos anteriores são combinadas por um outro modelo para obter a saída final. Podem ser criadas **várias camadas** com modelos diferentes.

Bagging



Todos os modelos deste tipo de ensemble são **do mesmo algoritmo**, porém os dados de entrada de cada um são **amostras do dado original**, com a mesma quantidade de dados do dataset original, selecionadas usando o método **bootstrap** (aleatória com repetição). Ex.: **RANDOM FOREST**

Boosting



Os modelos são treinados com os mesmos datasets, porém os pesos das instâncias são ajustados de acordo com o erro das predições anteriores. Ex. XGBoost, Light GBM

Random Forest

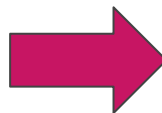
Etapas do algoritmo:

1. Criar dataset **com bootstrap** (seleção aleatória com repetição)

Random Forest

DATASET ORIGINAL

Cor	Estampa	Categoria	Bem avaliado
Verde	Flores	Casaco	Não
Azul	Liso	Casaco	Sim
Amarelo	Flores	Saia	Não
Azul	Flores	Saia	Sim



NOVO DATASET

Cor	Estampa	Categoria	Bem avaliado
Verde	Flores	Casaco	Não
Amarelo	Flores	Saia	Não
Azul	Liso	Casaco	Sim
Verde	Flores	Casaco	Não

* Perceba que a linha 1 foi selecionada duas vezes, enquanto a linha 4 não foi selecionada nenhuma vez nesse exemplo

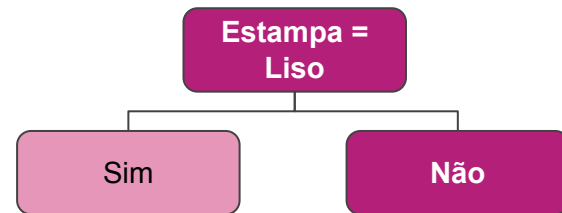
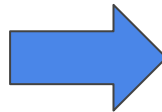
Random Forest

Etapas do algoritmo:

1. Criar dataset **com bootstrap** (seleção aleatória com repetição)
2. Criar uma árvore de decisão para o novo dataset utilizando um subconjunto randômico de variáveis

Random Forest

Cor	Estampa	Categoria	Bem avaliado
Verde	Flores	Casaco	Não
Amarelo	Flores	Saia	Não
Azul	Liso	Casaco	Sim
Verde	Flores	Casaco	Não



Random Forest

Etapas do algoritmo:

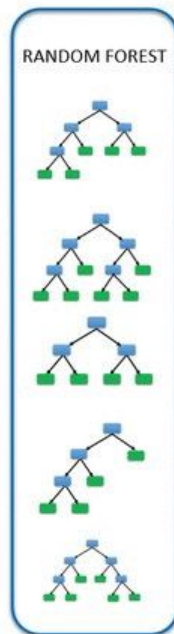
1. Criar dataset **com bootstrap** (seleção aleatória com repetição)
2. Criar uma árvore de decisão para o novo dataset utilizando um subconjunto randômico de variáveis
3. Repita esse processo várias e várias vezes, criando diferentes árvores (em média 100 árvores)
4. O classificador final é a média (ou voto) de todas as árvores

Random Forest

OOB Score:

Cada dado que não foi utilizado em uma árvore é usado para calcular o desempenho da mesma.

NEW DATA ARRIVES FOR TESTING



MISTAKES	CORRECT PREDICTIONS
0	0

Random Forest

Implementando Random Forest:
<Notebook>



Implementações do Boosting

1996

2002

2015

2017

2018

AdaBoost

Implementa a ideia de boosting geralmente usando como classificador fraco árvores de nível 1 (stump).

Gradient Boosting

Define um classificador inicial, calcula os resíduos e cria um novo classificador a partir dele

XGBoost

Primeira implementação do GBM, muito utilizado em competições do Kaggle

Light GBM

Versão com algumas alterações na implementação para melhorar performance. Trata variáveis categóricas

CatBoost

Versão com algumas alterações na implementação para melhorar performance. Trata variáveis categóricas de outra forma.

Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.

Gradient Boosting

Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade
Sim	Azul	Sim	12
Sim	Verde	Não	87
Não	Azul	Não	44
Sim	Vermelho	Não	19
Não	Verde	Sim	32
Não	Azul	Sim	14

Vamos tirar a média

34,7

Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)

Gradient Boosting

35

Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade	Resíduos
Sim	Azul	Sim	12	-23
Sim	Verde	Não	87	52
Não	Azul	Não	44	9
Sim	Vermelho	Não	19	-16
Não	Verde	Sim	32	-3
Não	Azul	Sim	14	-21

* Esse resíduo na verdade é a derivada da função de perda (Loss Function) e o chamamos de **Gradiente**

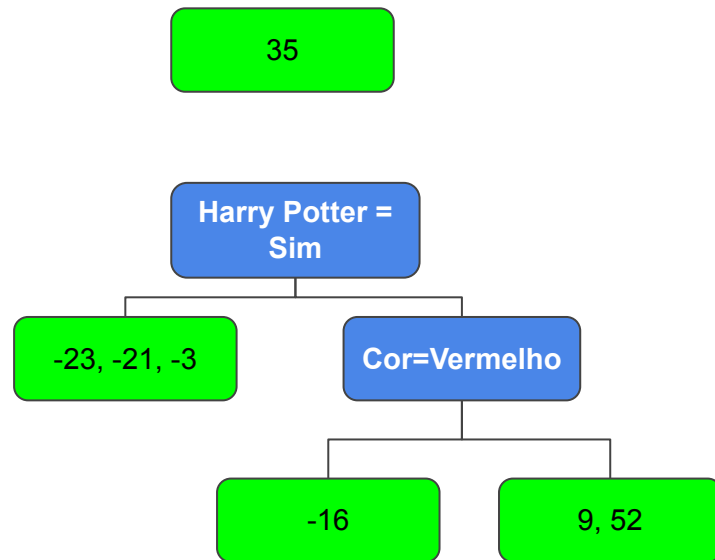
Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**

Gradient Boosting

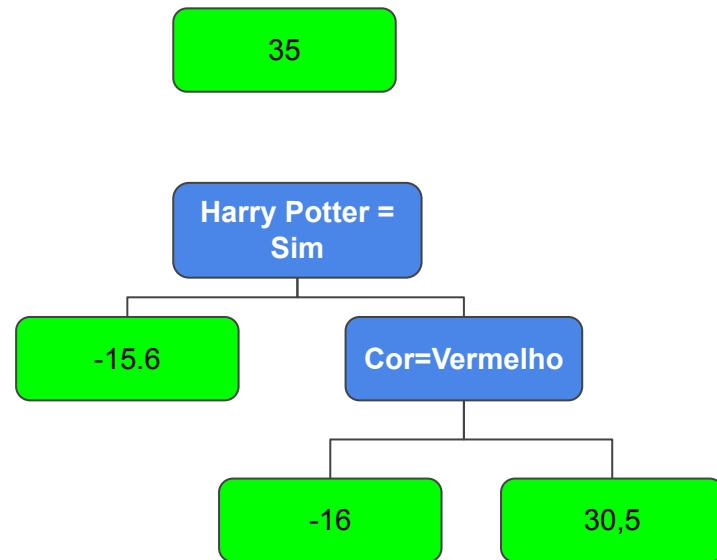
Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade	Resíduos
Sim	Azul	Sim	12	-23
Sim	Verde	Não	87	52
Não	Azul	Não	44	9
Sim	Vermelho	Não	19	-16
Não	Verde	Sim	32	-3
Não	Azul	Sim	14	-21



*no GB há um limite de folhas, geralmente entre 8 e 32.
Aqui limitamos em 3

Gradient Boosting

Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade	Resíduos
Sim	Azul	Sim	12	-23
Sim	Verde	Não	87	52
Não	Azul	Não	44	9
Sim	Vermelho	Não	19	-16
Não	Verde	Sim	32	-3
Não	Azul	Sim	14	-21



*no GB há um limite de folhas, geralmente entre 8 e 32.
Aqui limitamos em 3

Gradient Boosting

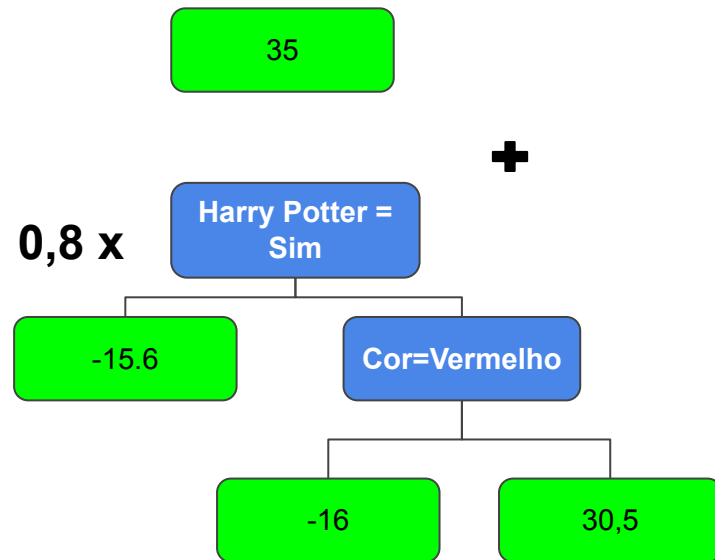
Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**
4. Usaremos a **“somatória” dos classificadores** para realizar a nova predição

Gradient Boosting

Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade	Resíduos	Pred
Sim	Azul	Sim	12	-23	22.5
Sim	Verde	Não	87	52	59.5
Não	Azul	Não	44	9	59.5
Sim	Vermelho	Não	19	-16	22.2
Não	Verde	Sim	32	-3	22.5
Não	Azul	Sim	14	-21	22.5

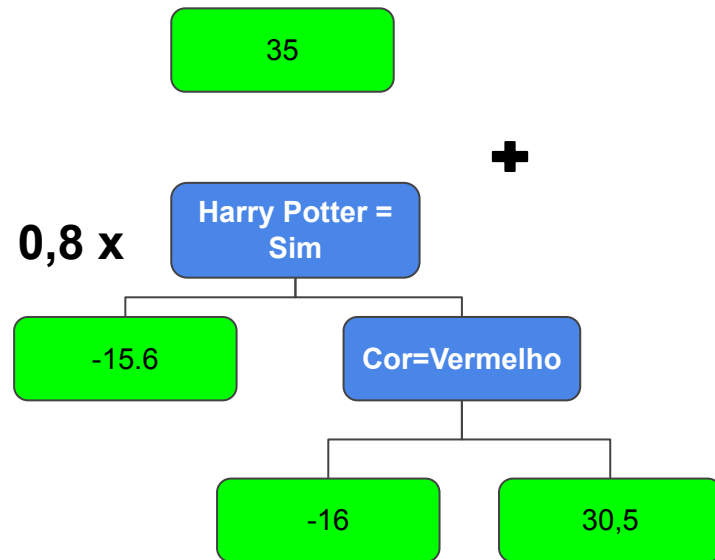
*Learning Rate geralmente é pequeno, em torno de 0,1 ou menor.



$$\text{pred: } 35 + (0,8 * -15,6) = 22.5$$

Gradient Boosting

Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Idade	Resíduos	Pred	Resíduos 2
Sim	Azul	Sim	12	-23	22.5	-10,5
Sim	Verde	Não	87	52	59.5	27,5
Não	Azul	Não	44	9	59.5	-15,5
Sim	Vermelho	Não	19	-16	22.2	-3,2
Não	Verde	Sim	32	-3	22.5	9,5
Não	Azul	Sim	14	-21	22.5	-8,5



$$\text{pred: } 35 + (0,8 * -15,6) = 22.5$$

*Learning Rate geralmente é pequeno, em torno de 0,1 ou menor.

Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**
4. Usaremos a **“somatória” dos classificadores** para realizar a nova predição
5. Crie um novo classificador para predizer os novos resíduos
... e assim por diante até atingir o número máximo de árvores (geralmente 100) ou um valor mínimo de resíduo

Gradient Boosting

E para **CLASSIFICAÇÃO??**

Para o cálculo das médias e respostas finais são usadas as funções de **probabilidade de logit** que vocês viram na aula de regressão logística.

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter
12	Sim	Azul	Sim
87	Sim	Verde	Sim
44	Não	Azul	Não
19	Sim	Vermelho	Não
32	Não	Verde	Sim
14	Não	Azul	Sim

Vamos usar a função que aprendemos em logit

??

Gradient Boosting

Idade	Gostou do Pipoca
12	Sim
87	Sim
44	Sim
19	Sim
32	Sim
14	Não

Precisamos transformar nosso resultado em probabilidade para utilizar o classificador. Vamos usar a Função Logística:

$$\log\left(\frac{p}{1-p}\right)$$

$$\log_e(4 / 2)$$

(temos 4 sims e 2 não)

0,7

Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)

Gradient Boosting

0,7

Idade	Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Resíduos
12	Sim	Azul	Sim	0,3
87	Sim	Verde	Sim	0,3
44	Não	Azul	Não	-0,7
19	Sim	Vermelho	Não	-0,7
32	Não	Verde	Sim	0,3
14	Não	Azul	Sim	0,3

Assumimos: Sim=1 e Não=0

Resíduo = (Valor Observado - Valor Predito)

* Esse resíduo na verdade é a derivada da função de perda (Loss Function) e o chamamos de **Gradiente**

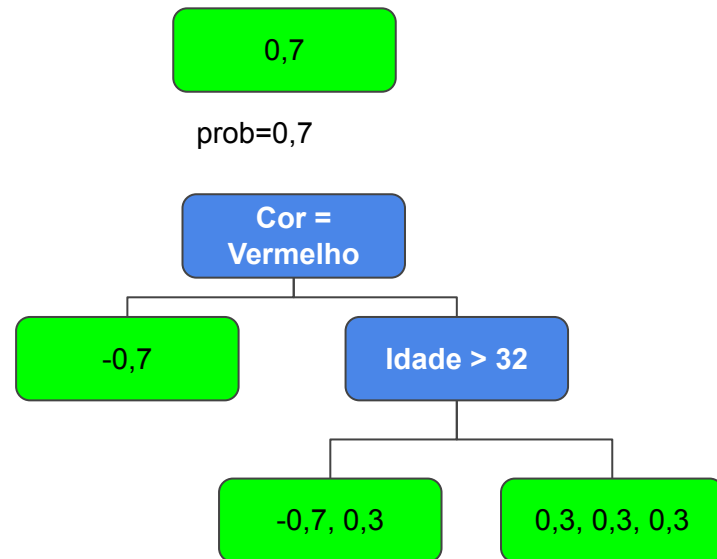
Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Resíduos
12	Sim	Azul	Sim	0,3
87	Sim	Verde	Sim	0,3
44	Não	Azul	Não	-0,7
19	Sim	Vermelho	Não	-0,7
32	Não	Verde	Sim	0,3
14	Não	Azul	Sim	0,3



*no GB há um limite de folhas, geralmente entre 8 e 32.
Aqui limitamos em 3

Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**
4. Usaremos a “**somatória**” dos **classificadores** para realizar a nova predição

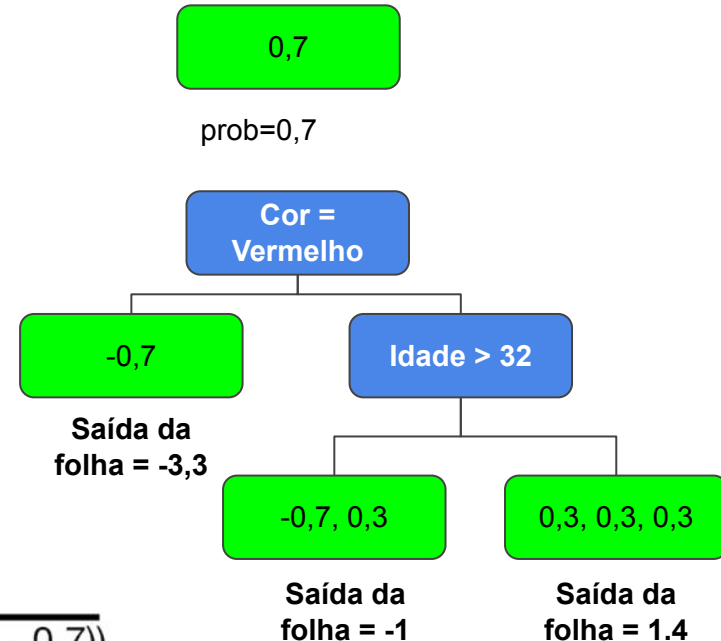
Gradient Boosting

$$\frac{\sum \text{Residual}_i}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)]}$$

$$\frac{-0.7}{0.7 \times (1 - 0.7)}$$

$$\frac{0.3 + -0.7}{(0.7 \times (1 - 0.7)) + (0.7 \times (1 - 0.7))}$$

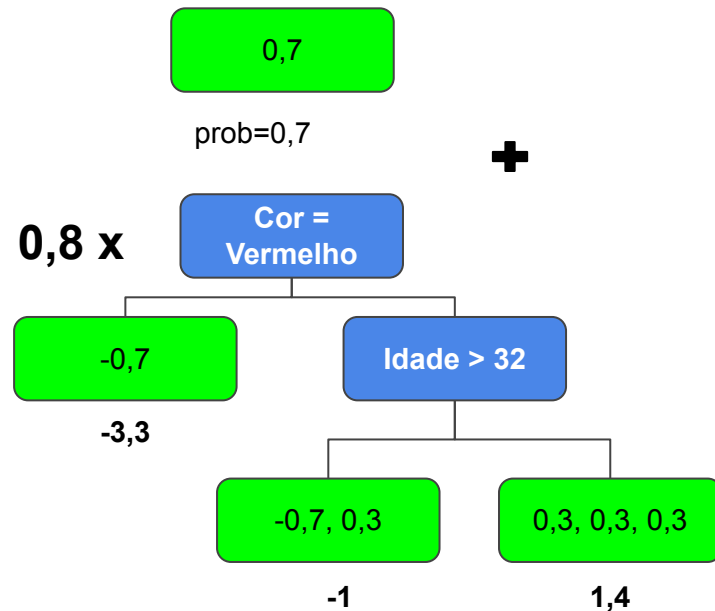
* Se fosse uma regressão apenas somaríamos os resultados, como é uma classificação precisamos antes fazer transformações probabilísticas



Imagens do canal [StatQuest](#)

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de Harry Potter	Rs	Logs
12	Sim	Azul	Sim	0,3	1,8
87	Sim	Verde	Sim	0,3	-0,1
44	Não	Azul	Não	-0,7	-0,1
19	Sim	Vermelho	Não	-0,7	-1,94
32	Não	Verde	Sim	0,3	1,8
14	Não	Azul	Sim	0,3	1,8

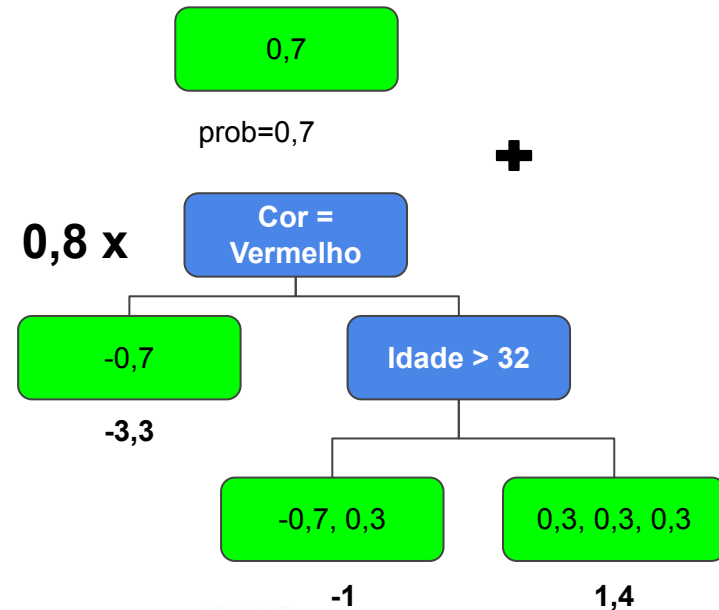


$$\text{logs: } 0,7 + (0,8 * 1,4) = 1,8$$

*Learning Rate geralmente é pequeno, em torno de 0,1 ou menor.

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de HP	Rs	logs	probs
12	Sim	Azul	Sim	0,3	1,8	0,9
87	Sim	Verde	Sim	0,3	-0,1	0,5
44	Não	Azul	Não	-0,7	-0,1	0,5
19	Sim	Vermelho	Não	-0,7	-1,94	0,1
32	Não	Verde	Sim	0,3	1,8	0,9
14	Não	Azul	Sim	0,3	1,8	0,9

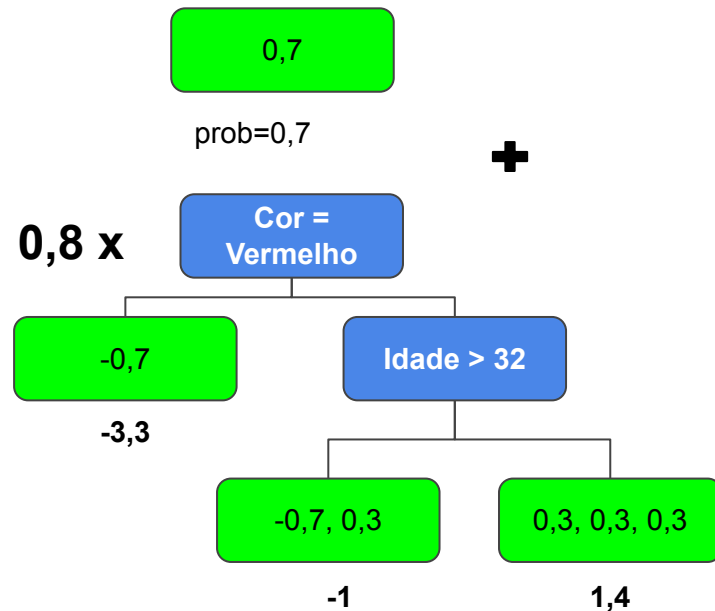


$$\text{Probability} = \frac{e^{1.8}}{1 + e^{1.8}} = 0.9$$

*Learning Rate geralmente é pequeno, em torno de 0,1 ou menor.

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de HP	Rs	pb	Rs 2
12	Sim	Azul	Sim	0,3	0,9	0,1
87	Sim	Verde	Sim	0,3	0,5	0,5
44	Não	Azul	Não	-0,7	0,5	-0,5
19	Sim	Vermelho	Não	-0,7	0,1	-0,1
32	Não	Verde	Sim	0,3	0,9	0,1
14	Não	Azul	Sim	0,3	0,9	0,1



*Learning Rate geralmente é pequeno, em torno de 0,1 ou menor.

prever os novos resíduos:
(Valor Observado - Novo Valor Previsto)

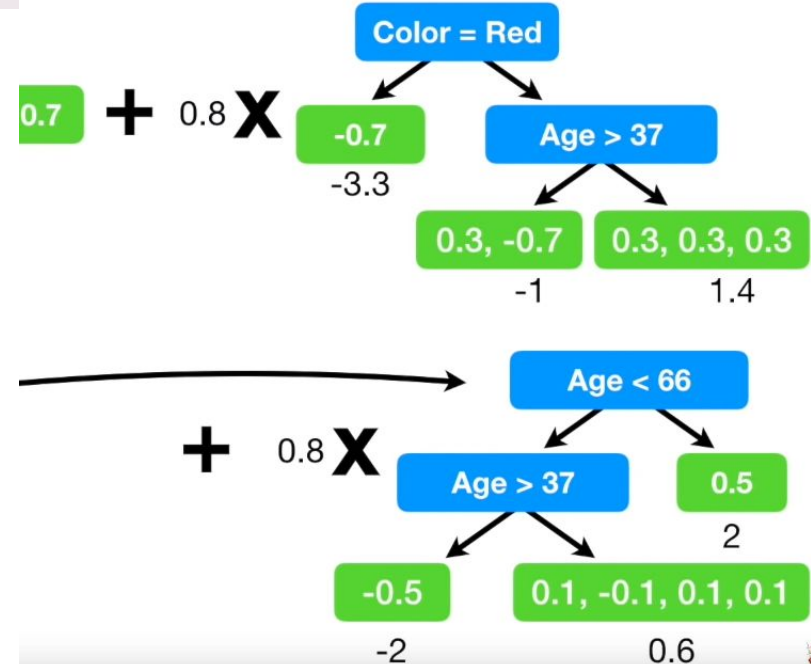
Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**
4. Usaremos a **“somatória” dos classificadores** para realizar a nova predição
5. Crie um novo classificador para predizer os novos resíduos

Gradient Boosting

Idade	Gosta de Pipoca	Cor Favorita	Gosta de HP	Rs	pb	Rs 2
12	Sim	Azul	Sim	0,3	0,9	0,1
87	Sim	Verde	Sim	0,3	0,5	0,5
44	Não	Azul	Não	-0,7	0,5	-0,5
19	Sim	Vermelho	Não	-0,7	0,1	-0,1
32	Não	Verde	Sim	0,3	0,9	0,1
14	Não	Azul	Sim	0,3	0,9	0,1



*Para fazer uma nova predição eu percorro a árvore calculando o log e posteriormente a probabilidade

dataset baseado no do [StatQuest](#)

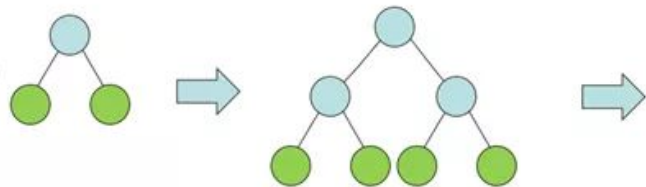
Gradient Boosting

Etapas do algoritmo:

1. Criamos uma **folha inicial**, que será nosso primeiro classificador. Para simplificar vamos imaginar que ela sempre será a média dos valores resposta.
2. Usaremos o nosso classificador para **calcular os resíduos** (erros)
3. Criamos um novo classificador para **predizer os resíduos**
4. Usaremos a **“somatória” dos classificadores** para realizar a nova predição
5. Crie um novo classificador para predizer os novos resíduos
... e assim por diante até atingir o número máximo de árvores (geralmente 100) ou um valor mínimo de resíduo

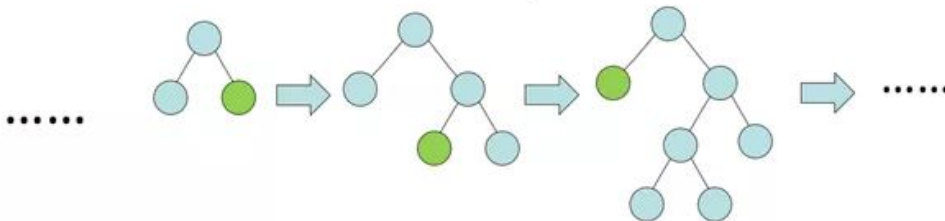
XG Boosting vs Light GBM

Level-wise growth



- Cresce as árvores em nível**
- Separação por histograma: cria bins (categorias) para as features contínuas.

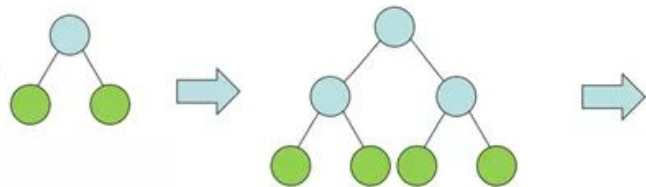
Leaf-wise growth



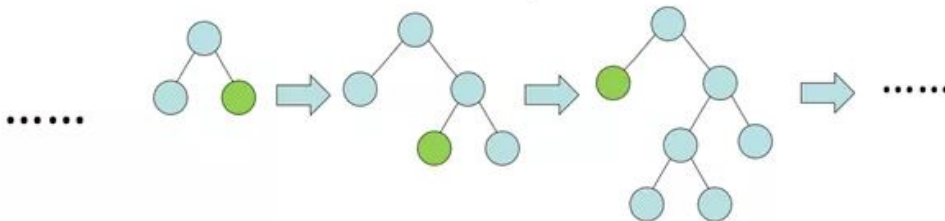
- Cresce as árvores por folha
- Separação por histograma, porém faz uma vez para todo o treinamento
- usa GOSS (Gradient Based One Side Sampling): faz downsampling do dataset:
 1. calcula o gradiente de cada linha
 2. seleciona todas as que tem um alto valor de gradiente
 3. faz seleção aleatória das que tem baixo valor

XG Boosting vs Light GBM

Level-wise growth



Leaf-wise growth



Gradient-based One-Side Sampling

Row Id	gradients
4	-5
3	3
2	0.5
6	0.2
5	0.1
1	0

select top 2
and randomly
sample 2
from the rest

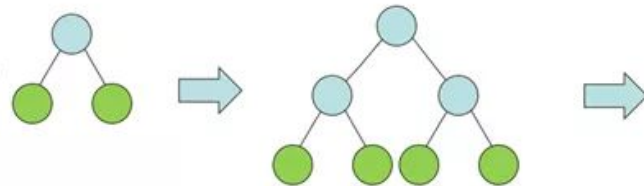
Sampling data

Row Id	gradients	weights
4	-5	1
3	3	1
6	0.2	2
5	0.1	2

- Cresce as árvores por folha
- Separação por histograma, porém faz uma vez para todo o treinamento
- usa GOSS (Gradient Based One Side Sampling): faz downsampling do dataset:
 1. calcula o gradiente de cada linha
 2. seleciona todas as que tem um alto valor de gradiente
 3. faz seleção aleatória das que tem baixo valor

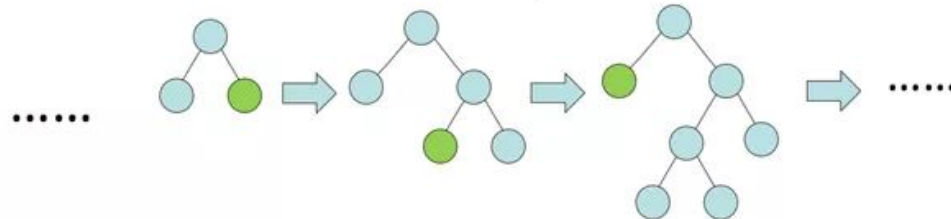
XG Boosting vs Light GBM

Level-wise growth



- Cresce as árvores em nível**
- Separação por histograma: cria bins (categorias) para as features contínuas.

Leaf-wise growth



- Cresce as árvores por folha
- Separação por histograma, porém faz uma vez para todo o treinamento
- usa GOSS (Gradient Based One Side Sampling)
- usa EFB (Exclusive Feature Bundling): diminui a quantidade de features juntando as que são esparsas de forma exclusiva, ou seja, onde está 0 em uma feature tem valor na outra.

LGBM

Implementando Light GBM:
<Notebook>



Material Complementar

- Canal do Youtube (inglês): [StatQuest](#)
- Canal do Youtube (em pt): [Bee Data \(USP\)](#)
- [Métodos de Ensemble - árvores \(UFPR\)](#)
- [Comitê de Máquinas](#)
- [Gradient Boosting \(inglês\)](#)
- Materiais do Kaggle:
 - <https://www.kaggle.com/code/pavansanagapati/ensemble-learning-techniques-tutorial/notebook>
 - <https://www.kaggle.com/code/satishgunjal/ensemble-learning-bagging-boosting-stacking/notebook>



**Quick view: O que
faltou de
classificação?**

Tópicos - Feature Engineering:

1. Algoritmos de redução da dimensionalidade:
 - a. ReliefF (**Feature selection using Relief algorithms with python example**)
 - b. PCA (**Principal Component Analysis from Statistical and Machine Learning Perspectives**)

São algoritmos que tentam diminuir a quantidade de features fazendo seleção das melhores ou combinação entre features. Como vimos os algoritmos recentes já fazem isso internamente.

Tópicos - Modelagem:

2. Hiperparametrização de algoritmos:
 - a. GridSearch (**GridSearch na tua cara**)
 - b. AutoML: TPot (**TPOT**)

Formas de encontrar os parâmetros ideais para um algoritmo. Os algoritmos atuais costumam funcionar bem com os parâmetros default.

Tópicos - Avaliação:

3. Validação:

a. Validação cruzada (k-fold cross validation):

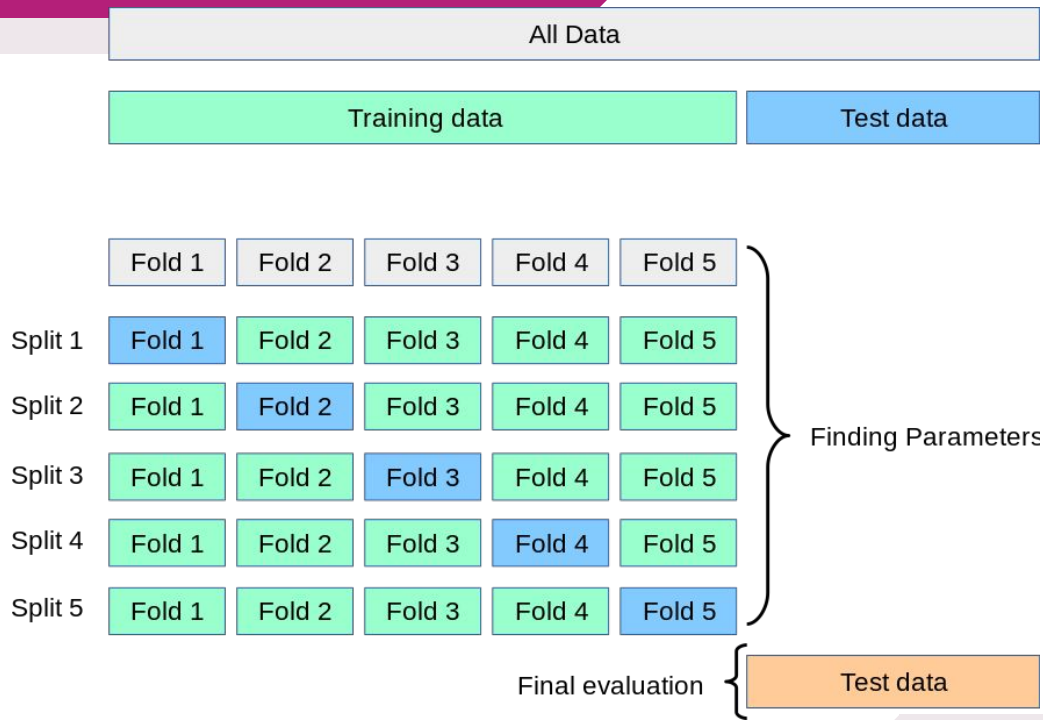
Geralmente utilizado para garantir que os parâmetros escolhidos para o algoritmo não estão causando Overfitting. Divide-se os dados em uma determinada quantidade de blocos (folds), e em cada rodada um desses blocos é usado para teste.

Tópicos - Avaliação:

Documentação de validação cruzada do scikit-learn.

```
import numpy as np
from sklearn.model_selection import KFold
```

```
X = ["a", "b", "c", "d"]
kf = KFold(n_splits=2)
for train, test in kf.split(X):
    print("%s %s" % (train, test))
```

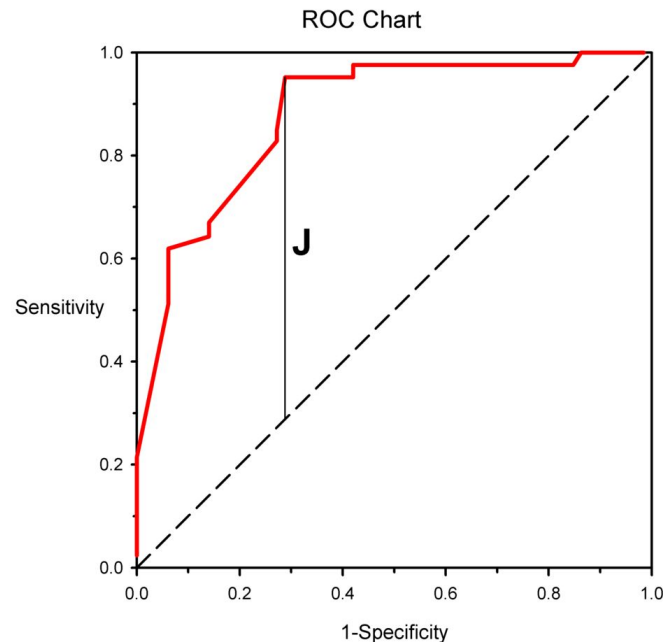


Tópicos - Avaliação:

3. Validação:

b. Cálculo de Threshold:

Geralmente assumimos que se a probabilidade for maior do que 0.5, o dado pertence a classe. Mas isso nem sempre é uma boa prática, há fórmulas para definir o melhor threshold baseado na curva ROC. Uma delas é o **Índice de Youden**.



Exercícios para casa: Pratiquem!!

1. Tenho um dataset em que o minha tarefa é dizer o preço de um produto:
 - a. Que tipo de algoritmo devo usar? Supervisionado ou não-supervisionado? De classificação ou de regressão?
 - b. Posso usar uma Random Forest para essa Tarefa?
 - c. Se no meu dataset tiver a variável “valor por kg” eu posso usar como entrada para o meu algoritmo? Por quê?

Lembram do desafio do titanic?

2. Tentem resolvê-lo com cada um dos algoritmos vistos hoje.

No notebook tentem ir explicando por passos o que estão fazendo, tanto a parte de Feature Engineering quando a Modelagem em si.

Aproveitem para colocar no github e usar como início de portfólio :D



E essa foi a nossa última aula! <3

Mas acabou?? =(

E essa foi a nossa
última aula! <3

NÃO

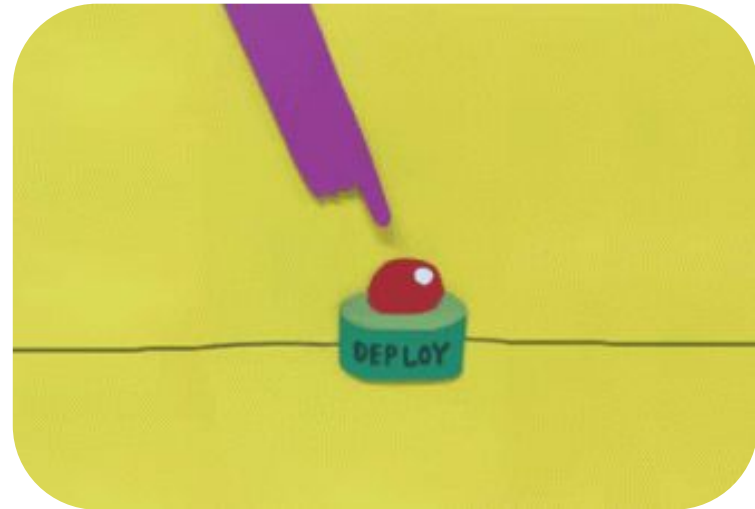
Mas acabou?? =(

Workshop de deploy de Modelos

Parte 1 - desenvolvendo o modelo e colocando em script

Parte 2 - Fazendo a API para deploy do modelo

Link do Github



Projeto final do curso: Mentoria




Karol:  [anakarolinafernandes](#)



Marielen:  [ferreiramarielen/](#)



Nathália:  [nath%C3%A1liatito/](#)

Projeto final do curso: Os projetos

Cada grupo receberá um tema, com os dados e a descrição! São 4 temas no total, portanto alguns grupos receberão o mesmo. (até terça-feira será enviado o e-mail com os detalhes)

Alguns são mais puramente analíticos e outros com modelos, podem criar uma apresentação mostrando os dados, gráficos e insights.

Vamos ver o exemplo de um: **Desafio Microsoft**



Não é nenhum bicho de sete cabeças!!

Aula de revisão e dúvidas

Por enquanto não teremos, nem $\frac{1}{3}$ respondeu pedindo.

Se quiserem a aula respondam até a próxima quarta para nos planejarmos!!



Formatura!

Quando: Dias 05 e 06 de outubro, às 19h

Onde: Online, com transmissão para o Youtube, podem chamar os amigos para assistir!

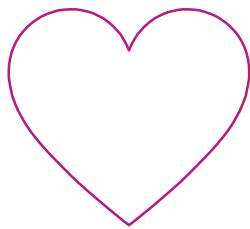
O que teremos:

- Cada grupo apresentará o desafio e receberão os feedbacks nossos e de alguns representantes de empresas
- Poderão escolher um representante por grupo para falar sobre a experiência no bootcamp.
- Painel de carreira
- Entrega de prêmios



Vamos preencher o formulário de feedback???





Muito obrigada!

Dúvidas? Podem nos procurar! 🙄



Nossos contatos



mulheres.em.ia@gmail.com



mulheres-em-ia



@mulheres.em.ia



@mulheres.em.ia



@MulheresemInteligenciaArtificial



Canal: Mulheres em IA

Linktree

<https://linktr.ee/mulheres.em.ia>

Grupo Telegram para Mulheres

https://t.me/mulheres_em_ia