

# Curso Fundamentos de Machine Learning:

Inteligência Artificial: Ideia, conceito, construindo inteligência, máquinas, machine learning

Machine Learning, capacidade do algoritmo aprender sem explicitamente serem programados. Através dos dados

Modelo: Simular um comportamento, permitindo o estudo da mudança do comportamento (conceito, uma ideia)

Algoritmo: Visa obter uma solução para um determinado tipo de problema.

Raciocínio Indutivo: Extrai regras e padrões de grandes conjuntos de dados!

Aprendizado supervisionado, Classificação e Regressão  
não supervisionado,  
semi supervisionado e  
reforço.

## Aprendizado supervisionado em classificação:

### KNN

**Classificação:** Prever categorias ou classes distintas. Exemplo: identificar se um e-mail é "spam" ou "não spam". A saída é qualitativa (discreta).

**Regressão:** Prever valores numéricos contínuos. Exemplo: estimar o preço de uma casa com base em características. A saída é quantitativa (contínua).

**K-Nearest Neighbors (KNN) na classificação:** O algoritmo identifica os  $k$  vizinhos mais próximos de um ponto no espaço de características e atribui a classe mais comum entre eles. Exemplo: Classificar uma fruta como "maçã" ou "laranja" com base no tamanho e na cor.

**K-Nearest Neighbors (KNN) na regressão:** O algoritmo calcula a média (ou outro valor agregado) dos valores numéricos associados aos  $k$  vizinhos mais próximos para prever um

valor contínuo. Exemplo: Prever o preço de uma casa com base nos preços das casas vizinhas.

## Ex Precision x Recall

Precisão - Toda vez que o algoritmo falar que não vai renovar, realmente não renova  
Recall - Todo mundo que é possível não renovar, acertar a maioria deles.

A **precisão** é útil quando queremos garantir que as previsões positivas sejam confiáveis, enquanto o **recall** é crucial quando queremos garantir que o modelo identifique a maior quantidade possível de instâncias positivas (por exemplo, detectar todos os doentes). O equilíbrio entre essas métricas depende do objetivo do modelo e do problema que está sendo resolvido.

Na Matriz de Confusão

Recall -  $tp / (tp + fn)$  Horizontal

Precisão -  $tp / (tp + fp)$  Vertical

Exercícios:

- ▶ 1. Rafaça o código de treinamento da aula 12: "K-Nearest Neighbors - Prática" no seu computador usando o Jupyter Notebook ou o Google Colabs.
- ▶ 2. Retreino o algoritmo com os seguintes valores para K: [3, 5, 7, 9, 11, 13, 15, 17, 19 e 21] e anote a acurácia.
  
- ▶ 5. Escreve um trecho de código que automatize o treinamento do algoritmo K-NN, a fim de encontrar o melhor valor para K, do exercício 2.

```

dicionario = {}
#Definição de parametros do treinamento
for x in range(3, 22, 2):
    k = x
    knn_classifier = KNeighborsClassifier(n_neighbors=k)
    #Treinando algoritmo
    knn_classifier.fit(x_train, y_train)
    y_pred = knn_classifier.predict(x_train)
    acuracia = mt.accuracy_score(y_train, y_pred)
    dicionario[k] = acuracia

9] ✓ 3.8s

dicionario
✓ 0.0s
{3: 0.8854736842105263,
 5: 0.8667368421052631,
 7: 0.8590526315789474,
 9: 0.8528421052631578,
 11: 0.85,
 13: 0.8492631578947368,
 15: 0.848,
 17: 0.8478947368421053,
 19: 0.8478947368421053,
 21: 0.8462105263157895}

```

► 3) Qual o problema principal de usar a métrica acurácia? Escreve um exemplo hipotético, no qual o problema acontece.

O problema de usar a acurácia é que nem sempre ela te trará uma informação precisa. Por exemplo, em um banco, você deseja pegar as transferências malignas (golpes, para poder bloquear ou solicitar alguma senha do cartão) a acurácia do seu modelo pode estar em 95%, porém ela pode ter acertado quase todas as transferências que não teve golpe e errado todas que teve golpe.

Ex:

95 - Nao golpe

5 - Golpe

Seu modelo acertou 95 do não golpe e errou as 5 do golpe, que é o objetivo do modelo, detectar golpes.

Por isso é bem importante não ter um dado tão discrepante em quantidade quanto outro, pois pode acabar te dando uma acurácia imprecisa.

► 4. Explique um pequeno texto ilustrando a diferença entre a métrica de Precision e Recall e mostrando quando usa deve ser escolhida em relação a outra.

Em uma loja de TV, aconteceu um lote em que as TVs foram enviadas com um problema, onde depois de um tempo ela congelou a imagem.

Recall - Você encontraria a maior quantidade de pessoas que compraram a TV para enviar a mensagem solicitando que entre em contato com o time para assistência, porém com menor precisão, podendo chegar em pessoas que não compraram a tv.

Precision - Você encontraria uma quantidade menor de pessoas, mas com uma precisão bem maior.

- 6. Escreva um pequeno texto, explicando as 6 denominações da matriz de confusão: P, N, TP, FN, FP e TN

P positive, N negative

Verdadeiro Positivo (VP): O modelo acertou e previu corretamente algo como positivo.

Falso Positivo (FP): O modelo disse que era positivo, mas estava errado.

Falso Negativo (FN): O modelo disse que era negativo, mas estava errado.

Verdadeiro Negativo (VN): O modelo acertou e previu corretamente algo como negativo.

## Regressão

Estudar o fenômeno ou criar um modelo para fazer a previsão.

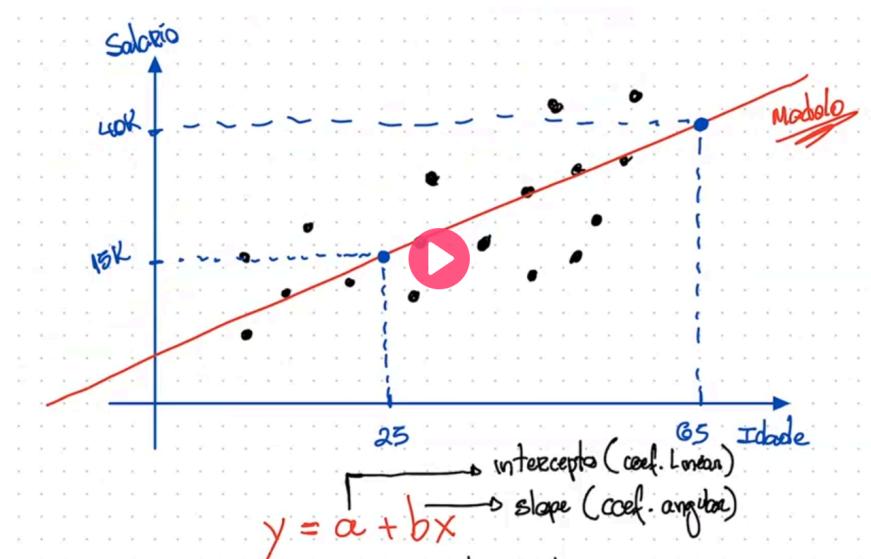
### 3.1 Regressão

1. Linear Regression
2. Regularized Linear Regression
  - a. Ridge Regression - L2 Norm
  - b. Lasso Regression - L1 Norm
3. Polynomial Regression
4. Neural Network Regression
5. Decision Tree Regression
6. Random Forest Regression
7. KNN Regression
8. Gaussian Regression

Regressão Linear =  $y = a + bx$

O modelo encontra a melhor reta para previsão dos dados

Técnica do mínimos quadrados, encontrar os melhores parâmetros da reta (a e b)



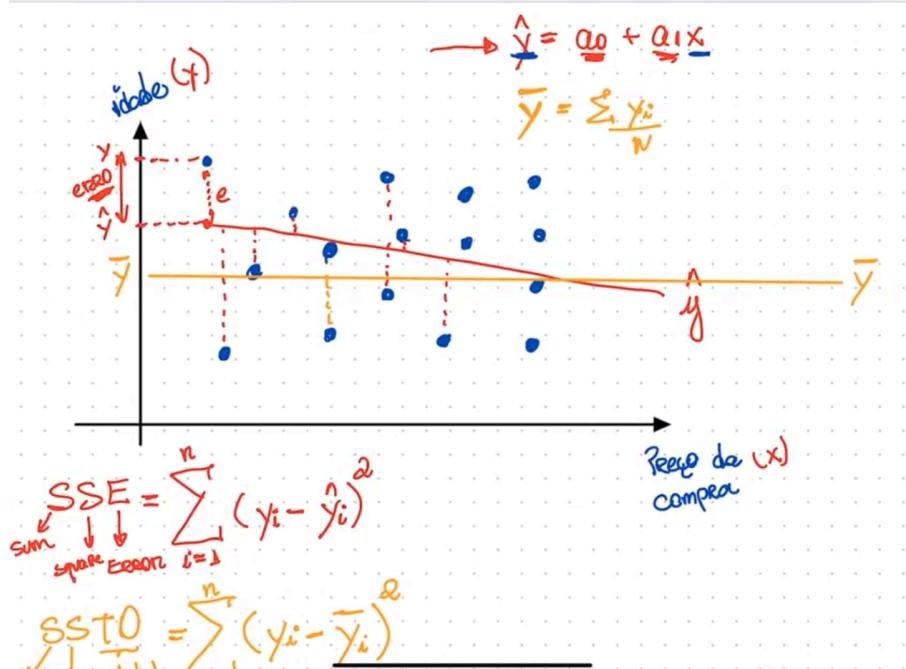
Fórmula do mínimos quadrados

$$a = \bar{y} - b\bar{x}$$

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$\hat{y}$   
Estimando

$\bar{y}$   
Média



Relação != causalidade

Relação, se uma variável subiu e outra subiu também quer dizer que possui uma relação, mas pode ter outros fatores então é impossível dizer que é a causa.

Para dizermos que é a causa temos que ter um ambiente com dados similares, só com um ponto diferente, exemplo, temos que pegar uma população de pessoas que moram no mesmo lugar, tem a faixa de salário bem parecidos, enviar o cupom no mesmo horário etc. Com todos dados bem parecidos, se manter a relação aí sim podemos dizer que é a causa.

Relação (correlação) entre variáveis indica que elas estão associadas, mas não que uma causa a outra. Para provar a causalidade, é necessário controlar outros fatores, como em um experimento onde tudo é mantido igual, exceto a variável testada. Sem esses controles, a relação pode ser apenas coincidência ou resultado de outra causa comum.

## ▼ Exercícios

- ▶ 1. Rafaça o código de treinamento da aula 19: "Linear Regression - Prática" no seu computador usando o Jupyter Notebook ou o Google Colabs e compute o R2, o MSE e o RMSE
- ▶ 2. Qual o problema principal de usar a métrica MSE? Escreve um exemplo hipotético, no qual o problema acontece.
- ▶ 3. Explique com um pequeno texto ilustrando o benefício de usar a métrica RMSE.

2 O problema principal de usar a métrica mse é que ela não está na mesma escala do seu target, não sendo possível comparar e saber se o modelo está bom ou ruim, também a métrica mse dá muita importância aos outliers, dando resultados errôneos

3 A métrica rmse coloca os valores na mesma escala da target, podendo assim comparar e realmente entender o erro do seu modelo

## Treino, Teste, Validação

Dividir o df em teste e treino para que você teste o modelo com dados que não foram vistos, dados novos.

Para classificação usar +- 80% treino 20% teste e ter a proporção dos targets igualmente divididos em treino e teste.

Com a validação - 70 treino, 15 teste, 15 validação

### Tipos de validação:

#### **Validação Holdout:**

Separa um terceiro conjunto de dados, que será usado para selecionar os parâmetros que trazem a melhor performance ao algoritmo.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

values = [i for i in range( 1, 60 )]

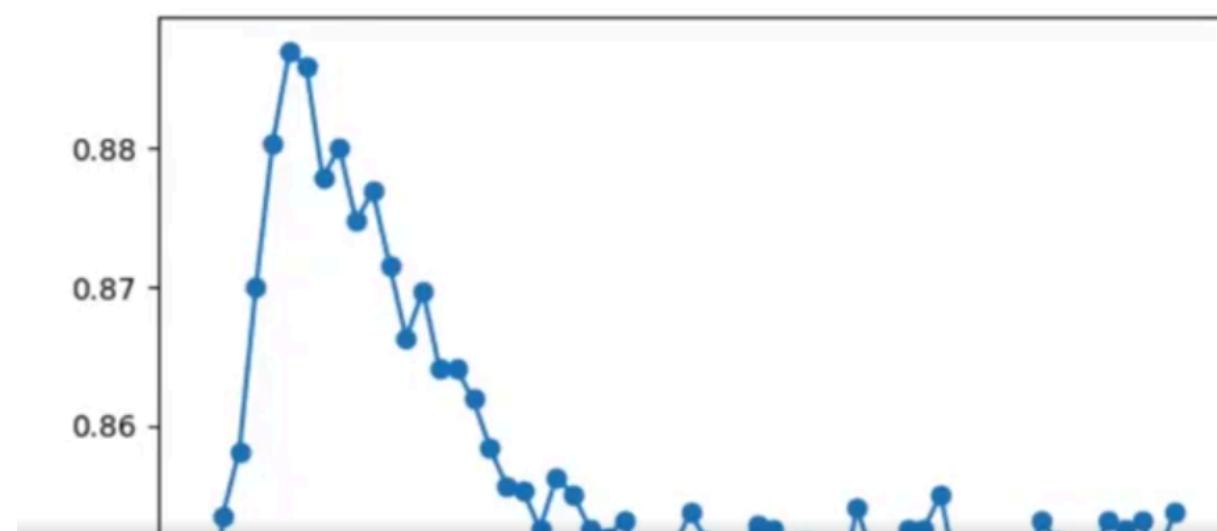
test_score = list()
for i in values:
    model = tr.DecisionTreeClassifier( max_depth=i )
    model.fit( X_train, y_train )

    # classificacao sobre o teste
    yhat_test = model.predict( X_test )
    acc_test = mt.accuracy_score( y_test, yhat_test )

    test_score.append( acc_test )

plt.plot( values, test_score, '-o', label='Test' )

[<matplotlib.lines.Line2D at 0x2988d6b20>]
```



Pode ter vazamento, mas quando não é possível ter validacao:

## Publicacao do algoritmo em Producao

```
# modelo treinado
model_last = tr.DecisionTreeClassifier( max_depth=6 )
model_last.fit( np.concatenate( (X_train, X_test) ),
                np.concatenate( (y_train, y_test) ) )

# classificacao sobre os dados de producao
yhat_prod = model_last.predict( X_prod )
acc_prod = mt.accuracy_score( y_prod, yhat_prod )

print( 'Accuracy Over Production: {}'.format( acc_prod ) )
```

Accuracy Over Production: 0.88175



Estratégia treino validação teste:

```
# Import libraries
import numpy as np
from sklearn import datasets as ds
from sklearn import tree as tr
from sklearn import metrics as mt
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
# 1.0 Treinamento como o Joaquim: O DS Novato
## Dados sintéticos para produção
n_samples = 20000
n_features = 2
n_informative = 2
n_redundant = 0
random_state = 0
# Dados para treinamento
X, y = ds.make_classification( n_samples=n_samples, n_features=n_features,
n_informative=n_informative, n_redundant=n_redundant,
random_state=random_state )

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=0)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=0)
X_train.size

values = [i for i in range(1,61)]
```

```

val_score = list()
for i in values:
    model = tr.DecisionTreeClassifier( max_depth=i )
    model.fit(X_train, y_train)
    model_pred = model.predict(X_val)
    acc_val = mt.accuracy_score( y_val, model_pred )
    val_score.append(acc_val)

model = tr.DecisionTreeClassifier( max_depth=10 )
model.fit(X_train, y_train)
model_pred = model.predict(X_val)
acc_val = mt.accuracy_score( y_val, model_pred )
acc_val

model = tr.DecisionTreeClassifier( max_depth=10 )
model.fit( np.concatenate( (X_train, X_val) ), np.concatenate((y_train, y_val)) )
model_pred = model.predict(X_test)
acc_val = mt.accuracy_score( model_pred, y_test )
acc_val

```

=====

## Overfitting

Generalização, exemplo do taxista, uma pessoa pegou um taxista muito caro e disse que todos os taxistas da cidade são ladrões. Os algoritmos também podem cair nessa armadilha, generalizando o que aprendeu nas amostras. Ou seja, o modelo comete muitos erros em dados não vistos. Máximo 0.5 de diferença de acurácia do validacao para o treino

O que causa?

Complexidade do Modelo, pode se ajustar demais aos dados de treinamento e ter mau desempenho em novos dados.

Conjunto de dados pequenos, pode aprender padrões aleatórios.

Treinamento Excessivo, em rede neural, se o modelo é treinado por muitas épocas, pode acabar decorando os dados e não generalizar bem.

Vazamento, informações dos dados de teste vazam para o modelo durante o treinamento.

### Soluções

Validação Cruzada - Dividir o conjunto em treinamento e validacao

Regularização

Early Stopping

Aumentar o tamanho do conjunto de dados

## **Underfitting**

O modelo está uma complexibilidade abaixo do que a complexibilidade do problema

Exemplos de causas de underfitting:

O modelo escolhido é muito básico (ex.: um modelo linear tentando ajustar dados claramente não lineares).

Falta de treino suficiente (ex.: poucas épocas em um modelo de redes neurais).

### **Soluções**

Aumentar a complexibilidade do modelo

Adicionar mais features

Aumentar o tempo de treinamento

Alterar os parâmetros do modelo

## **Aprendizado não supervisionado**

Não possui rótulos

Usamos para descobrir padrões, agrupar indivíduos com características ou comportamentos semelhantes.

Exemplo: Clusterização

Usos:

Campanhas de marketing, grupos de clientes com comportamentos semelhantes, etc.

Identificação de fraudes, detectar atividades suspeitas.

Segmentação de imagens, exemplo agrupar cores semelhantes em uma imagem

K-Means

Hierarchical Clustering Analysis

DBSCAN

Meanshift

GMM

Não tem aprendizagem, não tem rótulos, não tem feedback.

# KMeans:

Passo 1: Carregar os dados.

Passo 2: Inicializar aleatoriamente os K centroides no espaço dos dados.

Passo 3: Calcular a distância entre cada ponto e os K centroides.

Passo 4: Atribuir cada ponto ao centroide mais próximo.

Passo 5: Determinar o ponto médio de cada cluster K e reposicionar o centroide nesse ponto.

Passo 6: Repetir os passos 3 a 5 até que não ocorram mais alterações na atribuição de pontos entre os clusters.

Média pela soma do erro das distâncias quadráticas entre cada ponto do cluster e seu centroide dentro do próprio cluster - Within cluster sum of squares

$$WCSS = \sum_{i=1}^n (x_i - c_j)^2$$

n - total de pontos

x - ponto

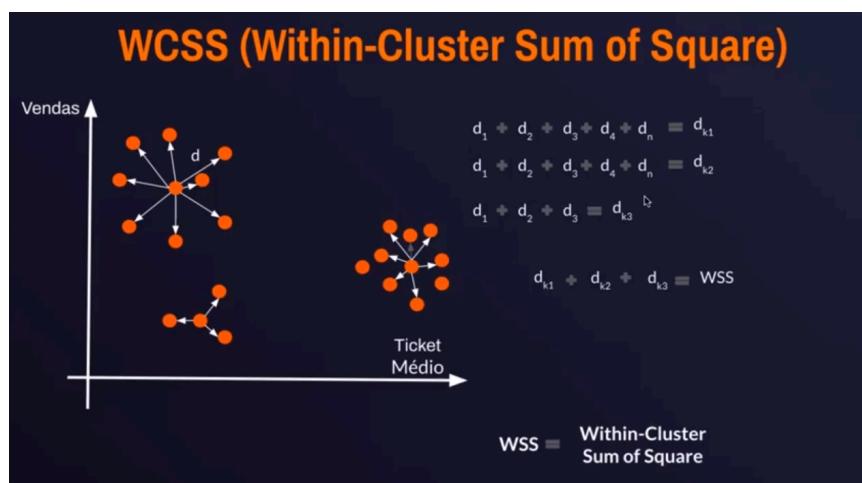
i - qual ponto, 1, 2, 3

c - centroide

j - qual centroide 1, 2, 3

Coesão

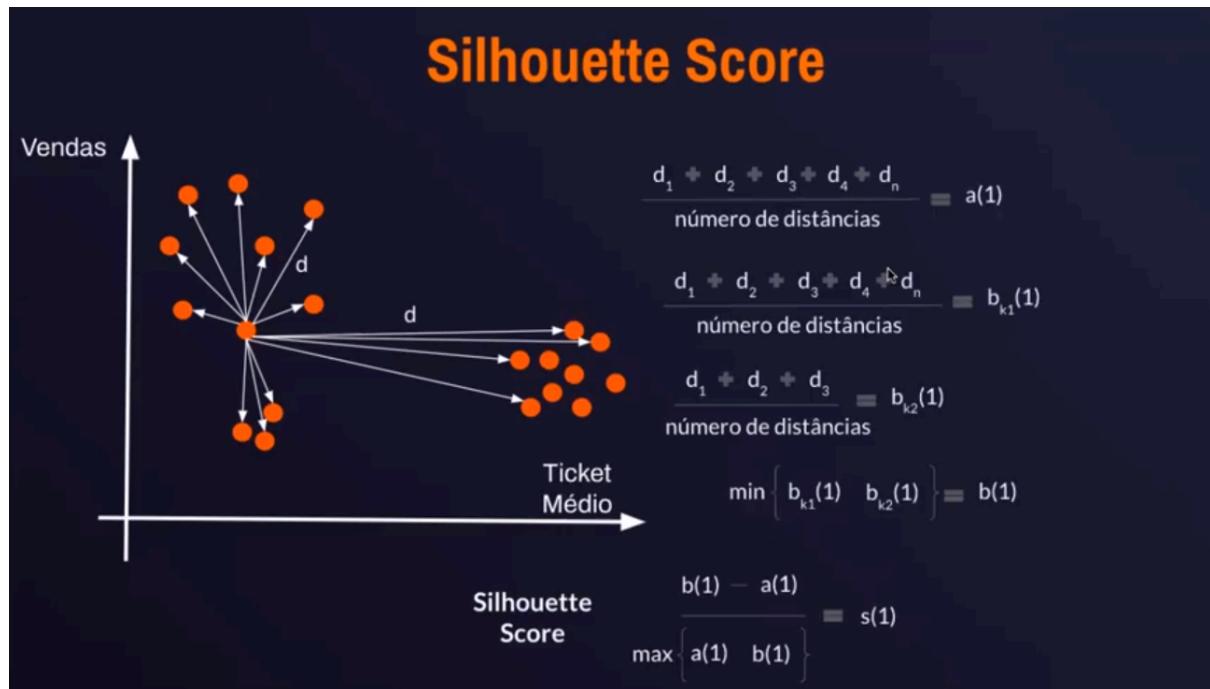
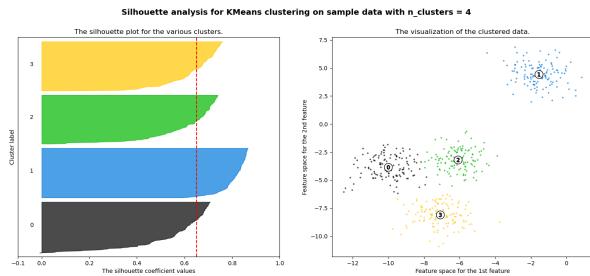
WCSS



o d já é a distância ao quadrado

## Silhouette score

É uma medida de avaliação para medir a qualidade de um agrupamento. Calculado para cada observação e mede o quanto bem ela se encaixa no cluster atual, sendo 1 está bem ajustada ao seu cluster e -1 está mal ajustada ao seu cluster



O quanto distante os pontos estão da borda

Se  $s(i)$  for próximo de 1: O ponto está bem inserido no seu cluster e distante de outros clusters.

Se  $s(i)$  for próximo de 0: O ponto está próximo da fronteira entre dois clusters.

Se  $s(i)$  for negativo: O ponto pode estar em um cluster "errado", pois está mais perto do cluster vizinho.

## Elbow Method

Cotovelo

### Identificar o melhor k

Quantos Clusters tem no meu conjunto de dados.

Variar o valor de K e observar as métricas de avaliação e performance

Exemplo de uso da clusterização

programas de fidelidade

identificação de fraudes - recomendado apenas para análise primária, criação de rótulos

detecção de objetos

compactação de áudio

## Árvores - Classificação

### Decision Tree

Usado tanto para regressão quanto para classificação

Funcionamento - Dividir o conjunto de dados em subconjuntos menores, de forma recursiva, até chegar a um conjunto de dados homogêneo ou até o limite máximo de crescimento ser atingido.

Recortes no espaço - Paralelos aos eixos, as features

Pureza e impureza

Suponhamos que em um grupo de 18 elementos foram distribuídos entre folhas

Uma dessas folhas possui 10 elementos da classe A e 0 elementos da classe B. Essa folha possui um grau de impureza zero, pois todos os seus elementos pertencem a uma única classe.

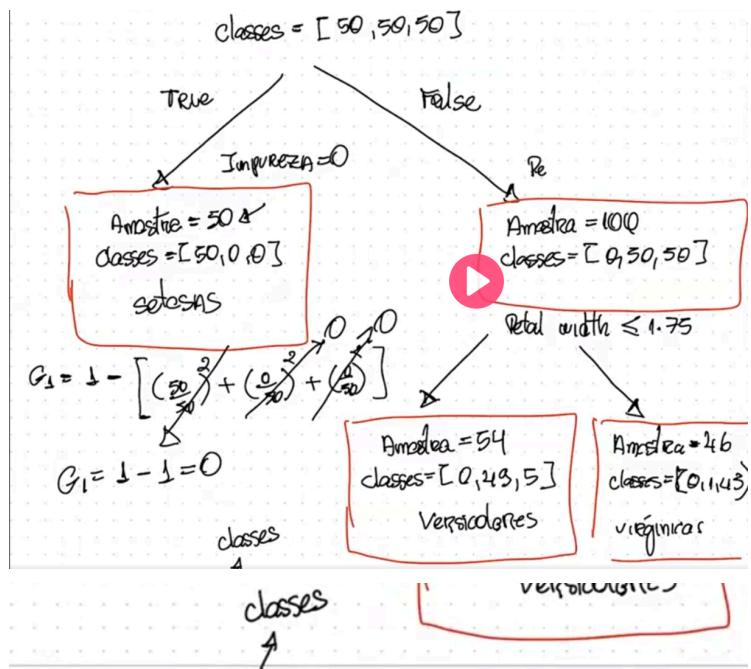
Por outro lado, uma dessas folhas possui 3 elementos da Classe A e 5 elementos da classe B. Essa folha tem um certo grau de impureza, pois 37.5% dos seus elementos pertencem à classe A e 62.5% à classe B.

% não significa a impureza.

Para determinar o grau de impureza de uma folha são utilizados algumas medidas

- Gini
- Entropia
- Ganho de informação

## Gini



$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2 = \frac{\text{Número elementos}}{\text{Total}}$$

## Entropia

### Entropia

$$E_i = \sum_{k=1}^{N_i} p(x_k) \log_2 p(x_k)$$

$$\begin{aligned} E_3 &= \left( \frac{0}{54} \right) \log_2 \left( \frac{0}{54} \right) + \left( \frac{43}{54} \right) \log_2 \frac{43}{54} + \\ &\quad + \left( \frac{5}{54} \right) \log_2 \left( \frac{5}{54} \right) = \end{aligned}$$

## Information Gain

$$IG(D_p, a) = I(D_p) - \sum_{j=1}^v \frac{N_j}{N_p} I(D_j)$$

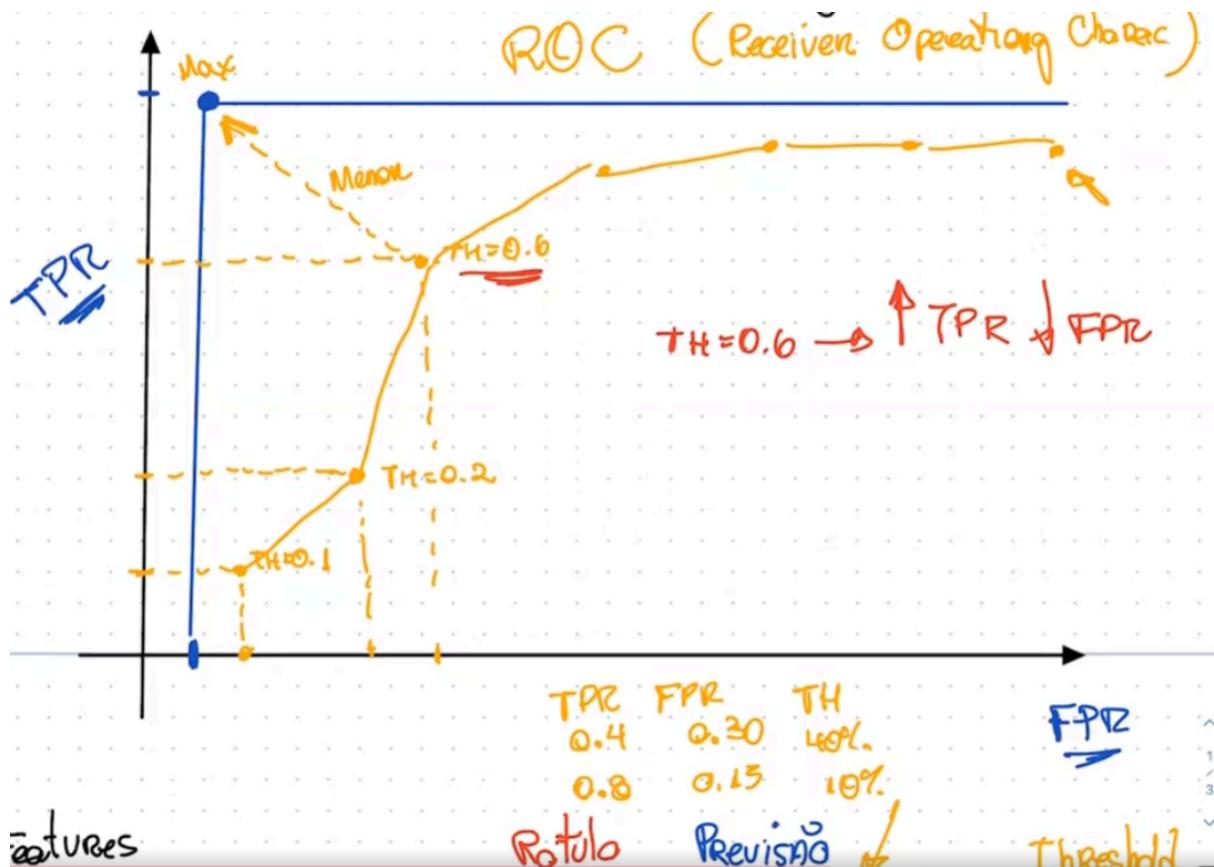
## ▼ 4. Os 6 passos do treinamento da Decision Tree

1. Escolha uma atributo ( coluna ) do conjunto de dados.
2. Para cada possível valor do atributo selecionado, use a função de custo para encontrar o valor da impureza da separação.
3. Repita os passos 1 e 2 para todas as combinações de atributo e valores, a fim de encontrar a combinação atributo-valor que retorne o menor valor da função custo da Decision Tree.
4. Uma vez definido o para atributo-valor, faça a separação do conjunto de dados em dois nós filhos.
5. Repita os passos de 1 a 3, para encontrar a segunda combinação atributo-valor para causar uma nova separação dos dados.
6. Repita o processo 5 até os valores dos parâmetros serem atendidos.

É importante limitar o crescimento da árvore para evitar o overfitting.

```
#model_tree = tr.DecisionTreeClassifier(max_depth=3) -> Controle do tamanho máximo de
quebras da árvore. O tamanho máximo de crescimento da árvore ou a quantidade de
recortes do espaço.
#min_samples_leaf = o número mínimo de amostras que a folha seguinte deve ter após a
divisão
#min_samples_split = o número mínimo de amostras que a folha deve ter para gerar um
novo split
#max_features = o número máximo de atributos avaliados para a divisão
```

## Curva ROC



## Roc x recall precision curve

As curvas ROC são apropriadas quando as observações são balanceadas entre cada classe, enquanto as curvas de precisão-recall são apropriadas para conjuntos de dados desbalanceados.

Exemplo de um conjunto de dados desbalanceado:

Classe positiva (interesse principal): 10% dos dados

Classe negativa: 90% dos dados

principais diferenças entre Precision-Recall (PR) e Curva ROC de forma objetiva:

Foco da métrica:

PR Curve: Foca em como o modelo lida com as classes positivas (verdadeiros positivos e falsos positivos).

ROC Curve: Foca em como o modelo lida com as classes negativas (verdadeiros negativos e falsos positivos).

Composição das métricas:

PR Curve: Calcula Precision (precisão) e Recall (sensibilidade).

ROC Curve: Calcula True Positive Rate (TPR) e False Positive Rate (FPR).

Sensibilidade a desbalanceamento de classes:

PR Curve: Melhor para dados desbalanceados, pois foca na classe positiva.

ROC Curve: Pode ser enganosa em dados desbalanceados, pois o FPR considera a classe negativa em relação ao total de negativos.

Interpretação:

PR Curve: Boa para avaliar modelos quando o interesse principal é minimizar falsos positivos e maximizar a detecção de positivos.

ROC Curve: Boa para avaliar a discriminação geral entre as classes positivas e negativas.

Em resumo:

PR Curve: Mais útil para desbalanceamento de classes, foca nas classes positivas.

ROC Curve: Boa para avaliar a performance geral do modelo, mas pode ser menos informativa em dados desbalanceados.

# Árvores - Regressão

## Decision Tree

Árvores de decisão para regressão não conseguem extrapolar previsões fora do intervalo dos dados de treinamento.

Ao invés de usar critérios de impureza (como Gini ou Entropia), utilizam métricas baseadas na redução de erro, como o MSE.

O valor previsto para um exemplo é a média dos valores das amostras na folha onde o exemplo foi classificado.

MSE e MAE - O MSE (Mean Squared Error) e o MAE (Mean Absolute Error) são métricas de avaliação utilizadas em modelos de regressão. Ambas medem o erro entre os valores previstos pelo modelo e os valores reais.

### 1. MSE (Erro Quadrático Médio):

- **Cálculo:** Média dos quadrados das diferenças entre os valores reais e previstos.
- **Objetivo:** Penalizar erros maiores de forma mais intensa.
- **Quando usar:** Se erros grandes são mais críticos para o problema, pois o MSE aumenta exponencialmente com erros maiores.
- **Desvantagem:** Sensível a outliers.

Fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2$$

---

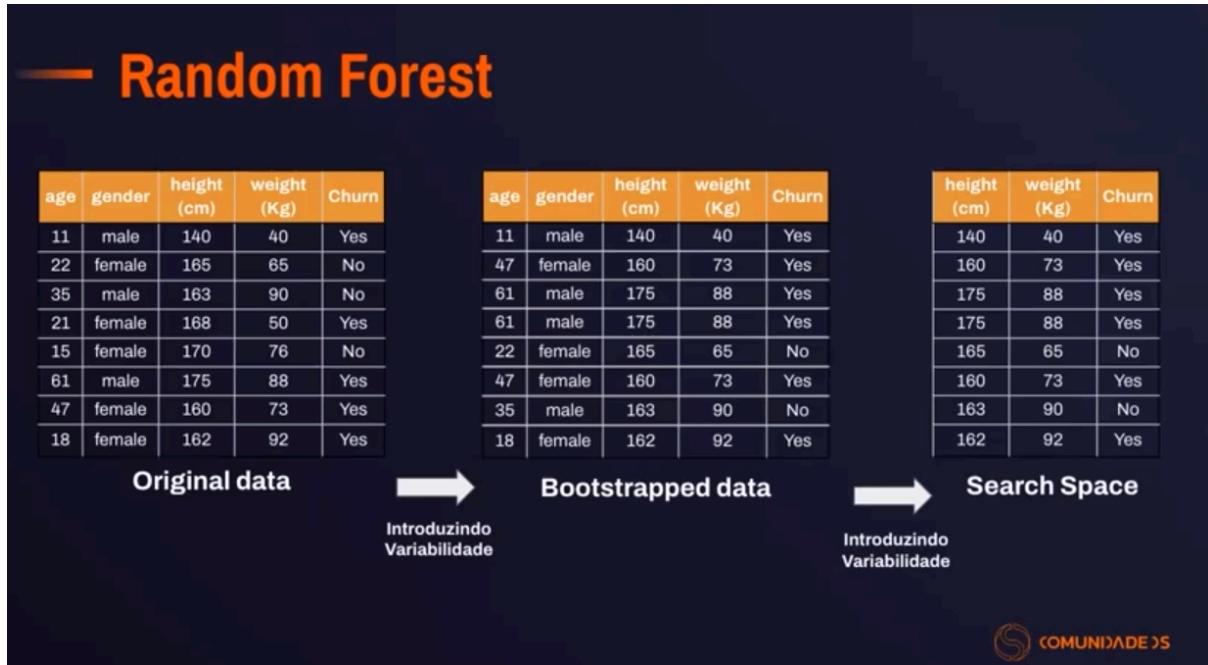
### 2. MAE (Erro Absoluto Médio):

- **Cálculo:** Média das diferenças absolutas entre os valores reais e previstos.
- **Objetivo:** Medir o erro médio sem penalizar excessivamente outliers.
- **Quando usar:** Se todos os erros têm peso igual no problema.
- **Desvantagem:** Não é tão sensível a grandes erros como o MSE.

Fórmula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

# Random Forest



Sorteia linhas do df original e cria o bootstrapped data (Pode conter dados repetidos) - cada bootstrap é uma árvore da floresta.

Sorteia colunas do df

Quando passa um novo dado, eles passam por todas as árvores, cada árvore terá seu resultado, o resultado que mais for obtido é o resultado final.



1. A Random Forest é a composição de resultados das Decision Tree.
2. Em problemas de classificação, usamos o sistema de votação.
3. Em problemas de regressão, usamos a média dos resultados das Decision Tree Regression.
4. Random Forest são Robustas contra Outliers.
5. Para o treinamento, não é necessário normalização ou rescala dos atributos.
6. Tanto a Random Forest quanto as Decision Tree não são capazes de extrapolar um valor.

## Feature Importance

```
feature_names = [f'Feature {i}' for i in range(X.shape[1])]
importances = model.feature_importances_
forest_importances = pd.Series(importances, index=feature_names)
forest_importances
```

0.0s

| Feature   | Importance |
|-----------|------------|
| Feature 0 | 0.000000   |
| Feature 1 | 0.022721   |
| Feature 2 | 0.061260   |
| Feature 3 | 0.354245   |
| Feature 4 | 0.037451   |
| Feature 5 | 0.181679   |
| Feature 6 | 0.342643   |

dtype: float64

quantifica a influência de cada variável preditora no modelo, indicando quais características contribuem mais para as previsões. Ela é calculada principalmente por dois métodos: Mean Decrease Impurity (MDI), que mede a redução da impureza dos nós causada por uma característica (computacionalmente eficiente, mas com possível viés para alta cardinalidade), e Mean Decrease Accuracy (MDA), que avalia o impacto da permutação aleatória de uma característica na precisão do modelo (mais robusto, porém mais custoso). Valores mais altos de importância indicam maior relevância da característica para o modelo, auxiliando na seleção de características, interpretação do modelo e engenharia de características.

## F1-Score

É o equilíbrio do recall e precision.

Como é calculado o f1 score -

$$F1 = \frac{2}{\left( \frac{1}{precision} + \frac{1}{recall} \right)}$$

ou

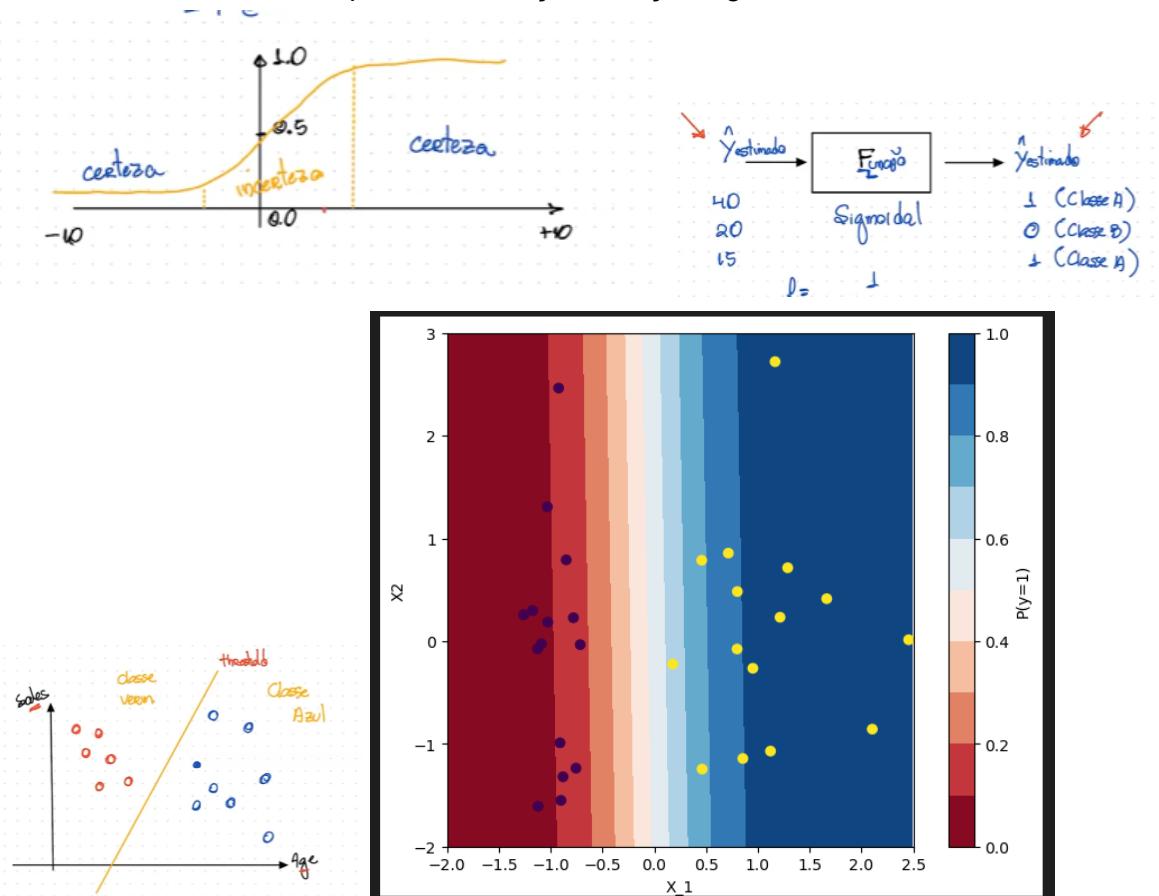
$$2 \times \frac{precision \times recall}{precision + recall}$$

| precision | recall | 1/precision | 1/recall |  |
|-----------|--------|-------------|----------|--|
| 0,8       | 0,02   |             | 1,25     |  |
|           |        |             | 50       |  |

**F1-Score** 0,03902439024

## Logistic Regression

Pode também ser utilizado para classificação. Função sigmoidal.



## Diferença entre regressão linear e logística

A regressão linear é usada para prever valores contínuos, como preços ou quantidades, enquanto a regressão logística é usada para prever probabilidades ou classificar categorias, como 'sim/não'. A saída da regressão linear é um número contínuo, enquanto na logística é uma probabilidade entre 0 e 1, que geralmente é convertida em uma classificação."

## Regularização

Lambda é o parâmetro de regularização que controla a força da penalidade

### L1(Lasso)

Incentiva o modelo a utilizar apenas os parâmetros mais importantes. Zerando os parâmetros menos relevantes.

Ajuda a selecionar as features.

### L2 (Ridge)

Suavizar os parâmetros da curva, mas não vai zerar. Função de custo é minimizada quando os valores dos parâmetros são pequenos

Lambda é o parâmetro de regularização que controla a força da penalidade.

Suaviza as diferenças entre os valores dos parâmetros, reduzindo a complexidade do modelo e evitando o overfitting.

### L1 e L2 (Elastic Net)

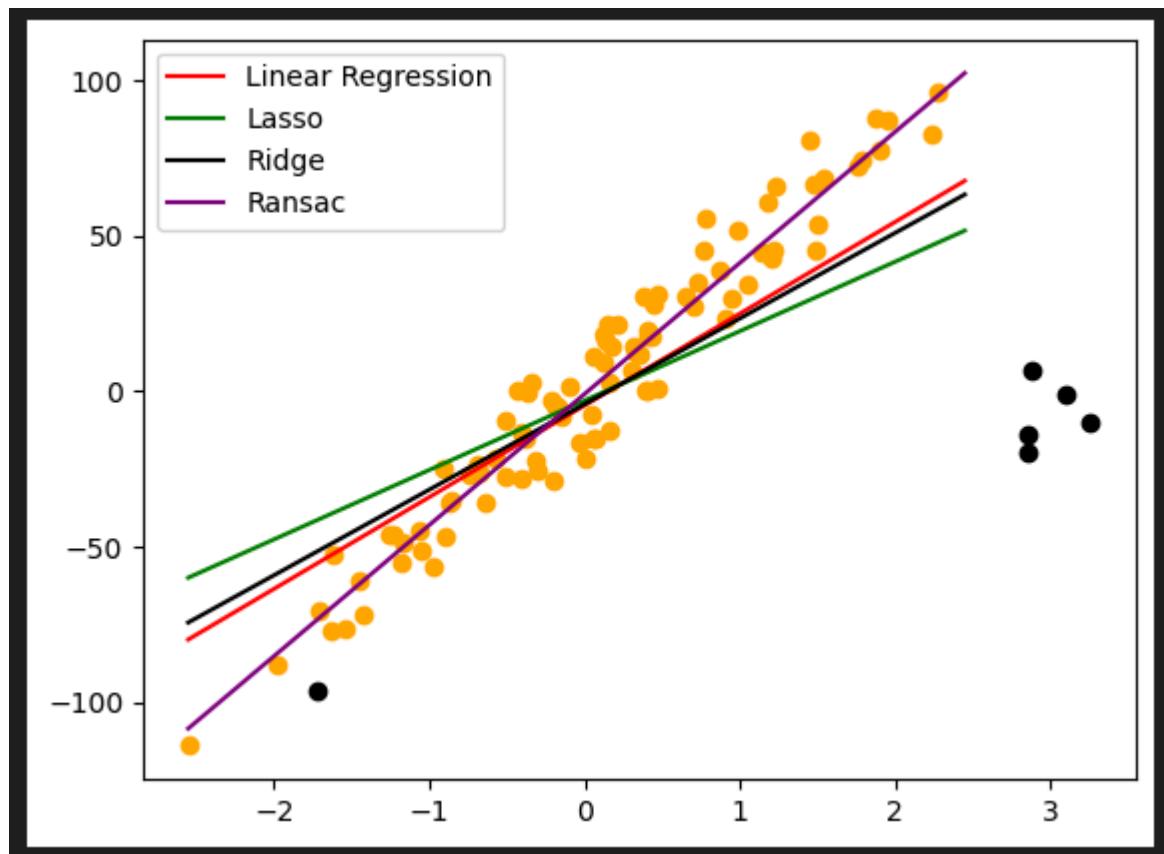
Combinação dos dois.

O Elastic Net combina L1 e L2, penalizando parâmetros irrelevantes (como L1) e suavizando os demais (como L2). É útil quando há muitas features correlacionadas ou em datasets esparsos.

Exemplo: Seleção de genes em bioinformática, onde muitas variáveis são correlacionadas e algumas podem ser irrelevantes.

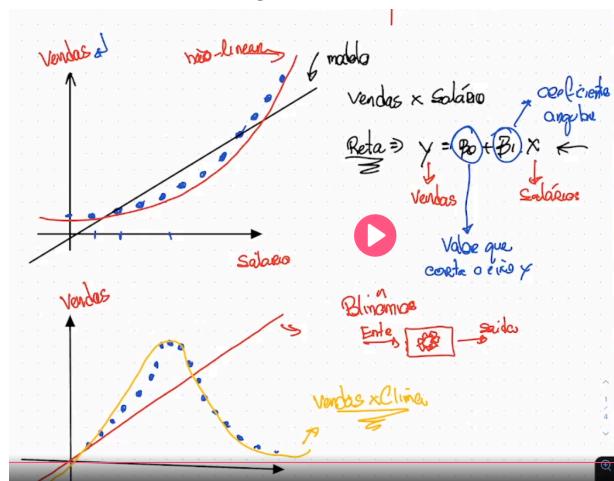
## RAMSAC

Ignora os outliers



## Regressão Polinomial

É usada para modelar relações não lineares entre variáveis em um conjunto de dados. Conforme sobe os graus, aumenta as curvas.



## Métrica MAE

| Idade | Clima | Vendas | Previsões | Eroo | Eroo Quadratric | Absolute |
|-------|-------|--------|-----------|------|-----------------|----------|
| 18    | 23    | 10     | 8         | 2    | +4              | 2        |
| 25    | 35    | 18     | 25        | -7   | +49             | +7       |
| 30    | 5     | 2      | 0         | 2    | +4              | 2        |
| 13    | 10    | 70     | 70        | 0    | 0               | 0        |
|       |       |        |           |      | 57              | 11       |

valor total do erro

Média Quadrada

MAE =  $\frac{11}{4} = 2,75$  é o erro absoluto médio

$$MAE = \frac{1}{N} \sum_{i=1}^n |y_i - \hat{y}_i|$$

MAE é mais robusto a outliers.

Altera o sinal do erro, coloca os valores negativos para negativos, soma e divide pela quantidade de valores.

Caso precisar dar mais ênfase ao outliers, podemos usar o rmse

# MAPE

Mais fácil para comunicar a equipe de negócio.  
Resposta dada em %

| Idade | Clima | Vendas | Previsto | ERRO | Abs | Percentual Abs |
|-------|-------|--------|----------|------|-----|----------------|
| 18    | 25    | 50     | 8        | 2    | 2   | 0,2            |
| 15    | 13    | 40     | 32       | 8    | 8   | 0,2            |
| 25    | 18    | 8      | 16       | -8   | 8   | +1             |
| 50    | 30    | 70     | 70       | 0    | 0   | 0              |
|       |       |        |          |      | 18  | 3,4            |

$MAE = \frac{18}{4} = 4,5$  é erro médio absoluto  
no valor das vendas

$MAPE = \frac{1,4}{4} = 0,35 = 35\%$

Median  
Absolute  
Percent  
Error

Erro percentual médio absoluto.

$(Real - Previsto) / real$

Real deve ser diferente de 0.

Caso tenha o valor 0, não conseguimos usar o mape

# Affinity Propagation

Clusterização

Não precisa dizer quantos clusters.

Agrupar gostos semelhantes a outros grupos. - Facebook, instagram, spotify etc.

| Participantes | Alice | Bob | Cary | Doug | Edna |
|---------------|-------|-----|------|------|------|
| Alice         | 5     | -16 | -15  | -11  | -21  |
| Bob           | 5     | -15 | -25  | -15  | -25  |
| Cary          | 5     | -26 | -15  | -17  | -25  |
| Doug          | -9    | -29 | -30  | 5    | -10  |
| Edna          | -14   | -34 | -33  | -5   | -10  |

Matriz de Similaridade: Representa quanto semelhante cada par de pontos está, usada para definir afinidades.

Matriz de Responsabilidade: Mede como um ponto sugere que outro seja representante (exemplar).

Matriz de Disponibilidade: Indica a adequação de um ponto para ser representante, considerando suporte global.

Matriz de Critério: Combina responsabilidade e disponibilidade para decidir os representantes finais.

=====  
Matriz Similaridade:  $D = -(x_1 - y_1)^2 \dots$

Troca os valores da diagonal pelo menor valor da tabela

Matriz Similaridade (S)

|       | Alice | Bob | Cary | Doug | Edna |
|-------|-------|-----|------|------|------|
| Alice | -22   | -7  | -6   | -12  | -17  |
| Bob   | -7    | -22 | -17  | -17  | -22  |
| Cary  | -6    | -17 | -22  | -18  | -21  |
| Doug  | -12   | -17 | -18  | -22  | -3   |
| Edna  | -17   | -22 | -21  | -3   | -22  |

$$d = \{ (x_1 - y_1) + (x_2 - y_2) + \dots \}$$

↑ Film 1      ↑ Film 2  
↓            ↓  
Alice        Alice     Alice

## Matriz Responsabilidade

$$R(i, k) = S(i, k) - \max_{k' \neq k} \{ D(i, k') + S(i, k') \}$$

Linha  $i$  coluna  $k$   
 Matriz Similitude  
 Matriz Disponibilidade

Alice Bob Alice Bob Bob Alice valor Alice valor Alice valor  
 Alice Bob Alice Bob Bob Alice valor Alice valor Alice valor

$$R(Alice, Bob) = S(Alice, Bob) - \max_{k' \neq k} \{ D(Alice, k') + S(Alice, k') \}$$

$D(Alice, Cary) + S(Alice, Cary)$   
 $D(Alice, Doug) + S(Alice, Doug)$   
 $D(Alice, Edna) + S(Alice, Edna)$   
 $D(Alice, Alice) + S(Alice, Alice)$

$$R(Alice, Bob) = S(Alice, Bob) - \max_{k' \neq k} \{ 0 + (-6), 0 + (-12), 0 + (-17), 0 + (-22) \}$$

$$R(Alice, Bob) = S(Alice, Bob) - \max \{ -6, -12, -17, -22 \}$$

$$R(Alice, Bob) = -7 - (-6) = -7 + 6 = -1$$

## Matriz de Responsabilidade

|       | Alice | Bob | Cary | Doug | Edna |
|-------|-------|-----|------|------|------|
| Alice | -16   | -1  | 1    | -6   | -11  |
| Bob   | 10    | -15 | -10  | -10  | -15  |
| Cary  | 11    | -11 | -16  | -12  | -15  |
| Doug  | -9    | -14 | -15  | -18  | 9    |
| Edna  | -14   | -13 | -18  | 14   | -13  |

## Matriz de Disponibilidade:

Fazer a diagonal primeiro

## Matriz de Disponibilidade

|       | Alice | Bob | Cary | Doug | Edna |
|-------|-------|-----|------|------|------|
| Alice | 0     | 0   | 0    | 0    | 0    |
| Bob   | 0     | 0   | 0    | 0    | 0    |
| Cary  | 0     | 0   | 0    | 0    | 0    |
| Doug  | 0     | 0   | 0    | 0    | 0    |
| Edna  | 0     | 0   | 0    | 0    | 0    |

$$D(k, k) = \max \{ 0, R(i, k) \}$$

Alice Alice  
 Bob Bob  
 Cary Cary  
 Doug Doug  
 Edna Edna

$$D(Alice, Alice) = \sum_{i \neq k} \max \{ 0, R(i, k) \}$$

$0, R(Bob, Alice)$   
 $0, R(Cary, Alice)$   
 $0, R(Doug, Alice)$   
 $0, R(Edna, Alice) \}$

$$D(Alice, Alice) = \sum_{i \neq k} \max \{ 0, 10, 0, 11, 0, -9, 0, -14 \}$$

$$D(Alice, Alice) = \sum_{i \neq k} 10, 11, 0, 0 = 21$$

## Matriz de Disponibilidade

|       | Alice | Bob | Cary | Doug | Edna |
|-------|-------|-----|------|------|------|
| Alice | 21    | 0   | 1    | 4    | 9    |
| Bob   | 0     |     |      |      |      |
| Cary  |       |     |      |      |      |
| Doug  |       |     |      |      |      |
| Edna  |       |     |      |      |      |

Restante fora da diagonal

$$D(i, k) = \min \{0, R(i, k)\} + \sum_{j \neq k} \max \{0, R(i, j)\}$$

↓      ↓  
 Alice   Bob  
 ↓      ↓  
 Bob, Bob  
 ↓  
 i ≠ k  
 ↓      ↓  
 outra   Bob  
 valor

$$D(Alice, Bob) = \min \{0, R(Bob, Bob)\} + \sum_{i \neq k} \max \{0, R(i, Bob)\}$$

$$D(Alice, Bob) = \min \{0, -15\} + \sum \max \{0, R(Alice, Bob), 0, R(Carey, Bob), 0, R(Doug, Bob), 0, R(Edna, Bob)\}$$

$$D(Alice, Bob) = -15 + \sum \max \{0, -1, 0, -11, 0, -14, 0, -13\}$$

$$D(Alice, Bob) = -15 + \sum 0, 0, 0, 0$$

$$D(Alice, Bob) = -15 + 0 = -15$$

## Matriz de Disponibilidade

|       | Alice | Bob | Carey | Doug | Edna |
|-------|-------|-----|-------|------|------|
| Alice | 21    | -15 | -14   | -5   | -10  |
| Bob   | -5    | 0   | -15   | -5   | -10  |
| Carey | -6    | -15 | 1     | -5   | -10  |
| Doug  | 0     | -15 | -15   | 14   | -19  |
| Edna  | 0     | -15 | -15   | -19  | 9    |

## Matriz de Critérios

Soma da Matriz de Responsabilidade + Disponibilidade

$$C(i, k) = R(i, k) + D(i, k)$$

Alice   Bob   Alice   Bob   Alice

## Matriz de Critérios

|       | Alice | Bob | Carey | Doug | Edna |
|-------|-------|-----|-------|------|------|
| Alice | 5     | -16 | -15   | -11  | -21  |
| Bob   | 5     | -15 | -25   | -15  | -25  |
| Carey | 5     | -26 | -15   | -17  | -23  |
| Doug  | -9    | -29 | -30   | 5    | -10  |
| Edna  | -14   | -34 | -33   | 5    | -10  |

Alice = 5  
 Bob = 5  
 Carey = 5  
 Doug = -5  
 Edna = -5

Alice  
 Bob  
 Carey  
 Doug  
 Edna  
 cluster 1

Doug  
 Edna  
 cluster 2

Pegar os maiores valores, unir os semelhantes e formar os clusters.